

Reti di Telecomunicazioni



© 2002–2026 Gianluca Mazzini

Licenza Creative Commons CC BY-NC-ND 4.0

[Release 2.2](#)

... In cambio, aveva il dono quasi soprannaturale di vedere subito l'applicazione di una teoria, in modo di servirsene. Nel gelido blocco di marmo della struttura astratta, riusciva a scorgere, senza difficoltà evidente, il disegno complicato di un congegno meraviglioso: il blocco si apriva come per un tocco magico e al suo posto restava il congegno...

Isaac Asimov, *Billiard Ball*

Indice

Premessa	xi
1 Introduzione	1
1.1 Classificazione delle reti	1
1.2 Localizzazione del traffico	2
1.3 Servizi	2
1.4 Tipi di comunicazione	2
1.4.1 Commutazione di circuito	2
1.4.2 Commutazione di pacchetto	2
1.4.3 Datagram e circuito virtuale	3
1.5 Tecniche di accesso	3
1.6 Tipologie di canali	3
1.6.1 Simplex, half duplex e full duplex	3
1.6.2 Canali punto-punto	3
1.6.3 Canali condivisi	4
1.7 Architettura di rete e organizzazione a livelli	4
1.7.1 Piano dati, piano di controllo e gestione	5
1.7.2 Livello fisico	5
1.7.3 Livello di collegamento dati	6
1.7.4 Livello di rete	6
1.7.5 Livello di trasporto	6
1.7.6 Livelli applicativi	6
1.7.7 Colloquio tra livelli	6
2 Elementi di teoria delle code	9
2.1 Richiami utili: momenti e trasformate	9
2.1.1 Variabili discrete non negative	9
2.1.2 Variabili continue non negative	9
2.2 Generalità sui processi stocastici	11
2.3 Catene di Markov a tempo continuo	11
2.3.1 Sulla distribuzione dei tempi di permanenza	11
2.3.2 Evoluzione dello stato nei sistemi a tempo continuo	12
2.3.3 Equazioni in avanti di Chapman Kolmogorov	13
2.3.4 Osservazioni sui tassi di transizione	14
2.3.5 Equazioni in indietro di Chapman Kolmogorov	14
2.3.6 Soluzione generale per sistemi a tempo continuo	15
2.3.7 Probabilità di stato	15
2.4 Catene di Markov a tempo continuo omogenee	15
2.5 Processi di nascita e morte	16
2.5.1 Condizione di adiacenza	16
2.5.2 Analisi processi nascita e morte	17
2.5.3 Processi di Poisson	18
2.5.4 Momenti della distribuzione di Poisson	19

2.5.5	Tempi interarrivo, distribuzione esponenziale	20
2.5.6	Momenti della distribuzione esponenziale	21
2.5.7	Tempo per collezionare k arrivi, distribuzione di Erlang	21
2.6	Processi di nascita e morte: soluzioni in equilibrio	21
2.6.1	Soluzione generale	22
2.6.2	Risultato di Little	23
2.6.3	Notazione di Kendall	24
2.6.4	Parametri caratteristici dei sistemi a coda	24
2.6.5	Sistema $M/M/1$	25
2.6.6	Sistema $M/M/1$ con arrivi rallentati	27
2.6.7	Sistema $M/M/\infty$	28
2.6.8	Sistema $M/M/c$ e formula di Erlang C	29
2.6.9	Sistema $M/M/1/Y$	32
2.6.10	Sistemi $M/M/Y/Y$ e formula di Erlang B	34
2.6.11	Sistemi $M/M/1//N$	35
2.6.12	Sistemi $M/M/\infty//N$	37
2.7	Sistemi $M/G/1$	38
2.7.1	Numero medio di utenti in $M/G/1$, formula di Pollaczek Khinchine	40
2.7.2	Altri parametri caratteristici per sistemi $M/G/1$	42
2.7.3	Distribuzione dei clienti in $M/G/1$	42
2.7.4	Due casi particolari	44
2.8	Sistemi $G/M/1$	45
2.8.1	Probabilità di transizione	45
2.8.2	Distribuzione del numero dei clienti	48
2.8.3	Parametri caratteristici per sistemi $G/M/1$	50
2.8.4	Distribuzione del tempo di attesa in coda	50
2.8.5	Due casi particolari	51
3	Collegamenti punto-punto	55
3.1	Codifica	55
3.1.1	Codici a blocchi lineari	55
3.1.2	Codici ciclici	56
3.1.3	Capacità di rivelazione e/o correzione	60
3.2	Strategie ARQ	61
3.2.1	Parametri caratteristici	62
3.2.2	Stop and wait	62
3.2.3	Go back N	66
3.2.4	Selective repeat	69
4	Metodi di accesso	73
4.1	Assunzioni comuni e parametri caratteristici	73
4.2	Accesso deterministico	73
4.2.1	Sistema TDMA	74
4.2.2	Sistema FDMA	75
4.2.3	Confronti TDMA-FDMA	75
4.3	Accesso controllato	76
4.3.1	Tempo di ciclo	76
4.3.2	Tempo di attesa in coda	77
4.3.3	Tempo di accesso	80
4.4	Accesso casuale	81

4.4.1	Sistema ALOHA	81
4.4.2	Slotted Aloha	84
4.4.3	Instabilità	87
4.4.4	Effetto cattura	90
4.4.5	Carrier Sense Multiple Access, CSMA	94
4.5	Confronto tra schemi di accesso	101
4.6	Fattori che incidono sulle prestazioni	102
5	Instradamento e controllo della congestione	103
5.1	Reti di code	103
5.1.1	Indipendenza di Kleinrock	104
5.1.2	Teorema di Burke	104
5.1.3	Teorema di Jackson	105
5.1.4	Parametri caratteristici di una rete di code	107
5.2	Algoritmi di instradamento	108
5.2.1	Algoritmo di Dijkstra	109
5.2.2	Bellman-Ford	111
5.2.3	Distance Vector, DV	112
5.2.4	Link State, LS	113
5.2.5	Ad Hoc On-Demand Distance Vector, AODV	114
5.2.6	Deflection routing (hot-potato)	115
5.2.7	Bifurcating routing o Multipath routing	115
5.2.8	Cut through	117
5.3	Controllo di flusso e della congestione	118
5.3.1	Finestra mobile, introduzione	119
5.3.2	Reti di code chiuse	119
5.3.3	Finestra mobile, prestazioni	123
5.3.4	Blocco sull'ingresso	126
6	Reti locali e accesso al mezzo	131
6.1	Framing e controllo di collegamento	131
6.2	Ritardo di propagazione	131
6.2.1	Topologia a Bus	132
6.2.2	Topologia ad anello	133
6.2.3	Topologie ad albero	134
6.3	Architettura delle LAN IEEE 802	134
6.4	Ethernet e IEEE 802.3	134
6.4.1	Trama Ethernet	135
6.4.2	CSMA/CD come modello di mezzo condiviso	135
6.4.3	Prestazioni del protocollo CSMA/CD 1-persistente	136
6.5	Accesso a token	144
6.5.1	Prestazioni	144
6.6	Disciplina timed-token	148
6.7	IEEE 802.11 e WLAN	151
6.7.1	Accesso	151
6.7.2	Struttura della trama	152
6.7.3	Prestazioni: modello di Bianchi	153
7	Commutazione di circuito	159
7.1	Sistemi di commutazione	159

7.1.1	Reti a divisioni di spazio, tipo S	159
7.1.2	Reti a divisioni di tempo, tipo T	162
7.1.3	Reti ibride	163
7.1.4	Sistemi multistadio	164
7.2	Prestazioni dei sistemi a commutazione di circuito	165
7.2.1	Esempio di dimensionamento	168
7.3	Procedure di segnalazione	169
8	Internetworking	171
8.1	Indirizzi MAC	171
8.2	Bridge e switch	172
8.2.1	Loop di livello 2 e spanning tree	172
8.2.2	Switching full-duplex e aggregazione dei link	173
8.2.3	Virtual LAN, VLAN	174
8.3	Router	174
8.4	Qualità del servizio, scheduling e gestione delle code	175
8.4.1	Scheduling delle code	175
8.4.2	Regolazione del traffico	175
8.4.3	DiffServ	176
8.4.4	Active Queue Management ed ECN	176
8.5	IP versione 4	177
8.5.1	Struttura del pacchetto	178
8.5.2	Indirizzi IPv4	178
8.5.3	Forwarding locale, next hop e ARP	179
8.5.4	Internet Control Message Protocol, ICMP	180
8.5.5	Multicast	180
8.5.6	Dynamic Host Configuration Protocol, DHCP	181
8.6	IP versione 6	181
8.6.1	Struttura del pacchetto	182
8.6.2	Indirizzi IPv6	182
8.6.3	Extension header e frammentazione	183
8.6.4	ICMPv6 e Neighbor Discovery	183
8.7	Protocolli di instradamento	183
8.7.1	Routing Information Protocol, RIP	184
8.7.2	Open Shortest Path First, OSPF	184
8.7.3	Border Gateway Protocol, BGP	184
8.7.4	Reverse Path Forwarding, RPF	185
8.7.5	Protocol Independent Multicast, PIM	185
8.8	MPLS e Segment Routing	185
8.8.1	Segment Routing	186
8.9	Reti di data center: Clos, ECMP e BGP	186
8.9.1	Oversubscription	187
8.9.2	BGP nella fabric	187
8.10	Demultiplazione e porte di trasporto	187
8.10.1	Network Address Translation, NAT	188
8.11	Transmission Control Protocol, TCP	189
8.11.1	Struttura del segmento	190
8.11.2	Instaurazione e chiusura della connessione	190
8.11.3	Segmentazione, acknowledgement e piccoli trasferimenti	191

8.11.4	Rilevazione delle perdite e ritrasmissione	192
8.11.5	Controllo di flusso	194
8.11.6	Controllo di congestione	194
8.12	User Datagram Protocol, UDP	197
8.12.1	Struttura del pacchetto	197
8.13	Real-time Transport Protocol, RTP	197
8.14	Servizi Internet e risoluzione dei nomi	198
8.14.1	Domain Name System, DNS	198
8.15	Sicurezza in Internet	198
8.15.1	Crittografia simmetrica e autenticazione dei dati	199
8.15.2	Funzioni hash crittografiche	199
8.15.3	Crittografia a chiave pubblica	200
8.15.4	Gestione e derivazione delle chiavi	201
8.15.5	IPsec	201
8.15.6	Sicurezza di trasporto: TLS e QUIC	202

Premessa

Una rete di telecomunicazione non è soltanto un insieme di mezzi trasmissivi e apparati. È un sistema nel quale risorse fisiche, protocolli, algoritmi e meccanismi di controllo cooperano per trasferire informazione tra entità distribuite. Comprenderne il funzionamento richiede quindi due punti di vista complementari: quello *architettuale*, che descrive funzioni e relazioni tra i diversi componenti, e quello *quantitativo*, che permette di valutare prestazioni, limiti e condizioni di funzionamento.

Questo libro nasce da un lungo percorso didattico e professionale. All'esperienza dell'insegnamento si è affiancato nel tempo il confronto quotidiano con reti realmente progettate, costruite ed esercite. Questo punto di osservazione rende particolarmente evidente un fatto: le tecnologie cambiano rapidamente, mentre molti dei principi che ne governano il funzionamento restano sorprendentemente stabili.

Ho quindi scelto di non trasformare il testo in un catalogo di tecnologie destinate a invecchiare rapidamente. L'obiettivo è conservare solidi i concetti di base, rafforzarne le spiegazioni e le derivazioni e, allo stesso tempo, aggiornare architetture, protocolli ed esempi ogni volta che questo aiuta a comprendere le reti di oggi.

Il volume sviluppa entrambi gli aspetti. Le tecnologie vengono presentate quando servono a rendere concreto un principio generale, non come un catalogo di standard. L'obiettivo è costruire strumenti che restino validi anche al cambiare delle implementazioni: modelli stocastici e sistemi a coda, protocolli di collegamento, accesso a un mezzo condiviso, instradamento, controllo del flusso e della congestione, commutazione, internetworking e trasporto end-to-end.

Nelle parti matematiche le ipotesi del modello vengono rese esplicite e i risultati non immediati vengono derivati. È importante distinguere tra categorie che nel linguaggio tecnico vengono talvolta sovrapposte: un risultato esatto conseguente a un modello, un'approssimazione valida in determinate condizioni e una regola imposta dalla specifica di un protocollo. Questa distinzione è essenziale per capire che cosa possa essere generalizzato e che cosa appartenga invece a una particolare scelta progettuale.

La teoria delle code svolge un ruolo centrale perché una rete a pacchetto è, in larga misura, un sistema di risorse condivise. Quando più flussi competono per una stessa capacità si formano code; da esse dipendono ritardo, variabilità del ritardo, perdita e utilizzazione delle risorse. I modelli probabilistici introdotti all'inizio del volume vengono quindi ripresi nei capitoli successivi per collegare le scelte di protocollo alle conseguenze prestazionali.

Le figure quantitative sono costruite a partire dalle equazioni e dai parametri dichiarati nel testo. Gli schemi concettuali hanno invece il compito di rendere visibili relazioni che sarebbero difficili da seguire soltanto in forma discorsiva: sequenze temporali, topologie, strutture di pacchetto, macchine a stati e percorsi dei dati. In entrambi i casi la figura è parte della spiegazione e non un elemento decorativo.

Il percorso parte dai concetti generali di rete e dall'organizzazione a livelli. Introduce poi gli strumenti probabilistici necessari all'analisi, affronta i collegamenti punto-punto e i protocolli ARQ, studia i metodi di accesso multiplo e passa quindi all'instradamento, alle reti locali e radio, alla commutazione di circuito e all'internetworking. L'ultima parte riunisce questi elementi nel contesto delle reti IP, includendo qualità del servizio, MPLS e Segment Routing, reti di data center, TCP e UDP, servizi Internet e meccanismi fondamentali di sicurezza.

Il testo è pensato per essere letto in modo progressivo ma anche consultato per argomenti. Le sezioni più analitiche richiedono familiarità con probabilità, variabili aleatorie e algebra lineare elementare; i richiami necessari vengono comunque introdotti quando diventano funzionali allo sviluppo dei modelli.

L'intento complessivo è fornire una visione unitaria: non separare la descrizione dei protocolli dalla comprensione delle ragioni per cui sono costruiti in un certo modo, né separare le formule dal fenomeno che esse rappresentano. Una rete è interessante proprio nel punto in cui architettura e prestazioni si incontrano.

Voglio ringraziare innanzitutto il Prof. Oreste Andrisano, amico e maestro, che mi ha indirizzato con entusiasmo verso tante sfide della mia vita e mi ha trasmesso il gusto di affrontarle con curiosità, rigore e fiducia.

Gianluca Mazzini

Introduzione

Questo capitolo introduce i concetti utilizzati nel resto del volume: classificazione delle reti, tipi di servizio, tecniche di commutazione, modalità di accesso al mezzo, topologie e organizzazione a livelli. L'obiettivo non è descrivere un insieme di tecnologie specifiche, ma individuare le astrazioni che ricorrono nei diversi sistemi di telecomunicazione.

1.1 Classificazione delle reti

Una rete di telecomunicazione collega entità che producono, trasportano e ricevono informazione. La classificazione per estensione geografica resta utile, pur non determinando da sola né la tecnologia né il bit rate.

- **Personal e Body Area Network, PAN/BAN.** Collegano dispositivi appartenenti alla stessa persona o presenti nel suo immediato ambiente. Sono tipicamente caratterizzate da distanze ridotte, vincoli energetici e uso frequente di collegamenti radio.
- **Local Area Network, LAN.** Collega sistemi appartenenti a uno stesso ambiente organizzativo o fisico, ad esempio un edificio, un laboratorio o un campus. Ethernet e IEEE 802.11 costituiscono esempi fondamentali di tecnologie LAN.
- **Metropolitan Area Network e reti di accesso, MAN/access.** Le MAN interconnettono sedi o reti locali su scala urbana o regionale. Le reti di accesso collegano invece gli utenti o le reti terminali al primo punto di aggregazione dell'operatore. Quest'ultima categoria non è definita dalla sola estensione geografica: conta soprattutto la funzione svolta nell'architettura. In entrambi i casi diventano importanti l'aggregazione del traffico, la gerarchia e la condivisione delle infrastrutture.
- **Wide Area Network, WAN.** Interconnette reti e sistemi su grandi distanze. Il problema dominante non è soltanto il trasporto fisico dell'informazione, ma anche l'instradamento tra reti amministrativamente e tecnologicamente differenti. Internet è il principale esempio di internetwork globale.
- **Reti di data center.** Interconnettono un numero elevato di server e apparati di memorizzazione all'interno di infrastrutture concentrate. Pur avendo estensione geografica limitata, presentano caratteristiche di traffico, ridondanza e multipath che le rendono concettualmente differenti da una LAN tradizionale.
- **Reti cellulari e di accesso radio.** La copertura è ottenuta suddividendo il territorio in celle e riutilizzando in modo controllato le risorse radio. La rete di accesso si integra con una rete di trasporto a pacchetto e deve gestire mobilità, interferenza e condivisione del mezzo.

La classificazione precedente non deve essere interpretata come una sequenza rigida né come una partizione in categorie mutuamente esclusive. Una stessa infrastruttura può essere descritta secondo la sua estensione, la funzione che svolge e la tecnologia di accesso. Le stesse tecniche di commutazione, accodamento, instradamento e controllo della congestione compaiono quindi a scale molto diverse.

1.2 Localizzazione del traffico

In molte reti il traffico non è distribuito uniformemente tra tutte le coppie di utenti. Flussi appartenenti allo stesso servizio, alla stessa organizzazione o alla stessa area geografica tendono spesso a concentrarsi in sottoinsiemi della rete.

Questa proprietà, detta localizzazione del traffico, è importante per il dimensionamento. Una rete gerarchica cerca di mantenere localmente la quota di traffico che non ha necessità di attraversare livelli superiori, riducendo il carico sul resto dell'infrastruttura.

Il principio è analogo alla località dei riferimenti nei sistemi di elaborazione: quando traffico o dati presentano una struttura locale, una gerarchia ben progettata può sfruttarla per ridurre costo, ritardo e occupazione delle risorse.

1.3 Servizi

Una rete non viene caratterizzata soltanto dalla capacità dei collegamenti, ma anche dai servizi che deve sostenere. Dal punto di vista delle prestazioni è utile distinguere almeno:

- trasferimenti di dati elastici, nei quali il tempo di completamento dipende dalla capacità disponibile ma piccoli cambiamenti del ritardo non compromettono il servizio;
- comunicazioni interattive, sensibili al ritardo e alla sua variabilità;
- flussi audio e video in tempo reale, nei quali continuità, ritardo, jitter e perdita assumono importanza specifica;
- traffico di controllo e segnalazione, spesso di volume ridotto ma con requisiti stringenti di affidabilità o tempestività;
- trasferimenti massivi e comunicazioni tra sistemi distribuiti, che possono generare elevati volumi di traffico aggregato.

Servizi differenti condividono spesso la stessa rete. Da questa convivenza nascono i problemi di accodamento, controllo del flusso, congestione, qualità del servizio e allocazione delle risorse discussi nei capitoli successivi.

1.4 Tipi di comunicazione

1.4.1 Commutazione di circuito

Nella commutazione di circuito viene riservato un insieme di risorse lungo un percorso tra sorgente e destinazione. Una volta instaurato il circuito, la capacità riservata rimane disponibile per la comunicazione fino al suo rilascio.

Il principale vantaggio è la prevedibilità delle risorse. Lo svantaggio è che la capacità rimane impegnata anche nei periodi in cui la sorgente non trasmette informazione.

1.4.2 Commutazione di pacchetto

Nella commutazione di pacchetto l'informazione viene suddivisa in unità trasmesse e memorizzate temporaneamente nei nodi intermedi. Questa modalità consente a molti flussi di condividere dinamicamente gli stessi collegamenti.

La condivisione statistica aumenta l'efficienza, ma introduce accodamento. Il tempo di consegna non è quindi costante: dipende dal traffico, dalla capacità dei collegamenti, dagli algoritmi di instradamento e dalle politiche di servizio delle code.

La presenza delle code è il motivo per cui una parte rilevante del volume è dedicata ai processi stocastici e ai sistemi a coda.

1.4.3 Datagram e circuito virtuale

Una rete a pacchetto può trattare ogni pacchetto in modo indipendente. In questo caso si parla di servizio a datagram: il pacchetto contiene le informazioni necessarie per essere instradato e pacchetti appartenenti allo stesso flusso possono, in linea di principio, seguire percorsi differenti.

Un'alternativa consiste nell'associare preventivamente il traffico a uno stato o a un percorso logico. I pacchetti trasportano allora un identificatore del circuito virtuale anziché tutte le informazioni necessarie a determinare nuovamente il percorso.

Un circuito virtuale non implica, da solo, una prenotazione di banda: lo stato di inoltro e la riserva delle risorse sono proprietà distinte. Un percorso logico può dunque essere soggetto ad accodamento e congestione, a meno che il servizio non preveda anche un controllo di ammissione e un'allocazione adeguata delle risorse.

Le due idee non si escludono a livello architetturale: molte reti moderne utilizzano forwarding a pacchetto insieme a stati, identificatori o label mantenuti in particolari domini della rete.

1.5 Tecniche di accesso

Quando più stazioni condividono la stessa risorsa trasmissiva è necessario stabilire come viene assegnato il diritto di trasmettere. Nel capitolo 4 vengono analizzate tre famiglie fondamentali.

- **Accesso deterministico.** La risorsa è suddivisa secondo una regola prefissata, ad esempio nel tempo o nella frequenza. L'assenza di collisioni è ottenuta mediante una separazione preventiva degli utenti.
- **Accesso controllato.** Il diritto di trasmettere viene assegnato dinamicamente da una procedura di polling o tramite il passaggio di un token. L'allocazione resta coordinata, ma può adattarsi al traffico.
- **Accesso casuale.** Le stazioni competono per il mezzo senza un'assegnazione preventiva. La semplicità del meccanismo viene pagata con la possibilità di collisioni, ritrasmissioni e instabilità ad alto carico.

Queste tre famiglie costituiscono modelli di riferimento. Sistemi reali possono combinare più meccanismi oppure modificarne la disciplina in funzione del traffico e dello stato del canale.

1.6 Tipologie di canali

1.6.1 Simplex, half duplex e full duplex

Un canale simplex trasporta informazione in una sola direzione. In un collegamento half duplex entrambe le stazioni possono trasmettere, ma non contemporaneamente. In un collegamento full duplex le due direzioni sono disponibili contemporaneamente.

La realizzazione fisica del full duplex può utilizzare mezzi distinti oppure tecniche che separano le due direzioni sullo stesso mezzo.

1.6.2 Canali punto-punto

In un collegamento punto-punto la risorsa collega due entità. Una rete formata da collegamenti punto-punto può assumere topologie differenti.

La figura 1.1 mostra alcune topologie fondamentali. La topologia influenza ridondanza, numero di collegamenti, lunghezza dei percorsi e modalità con cui il traffico viene ripartito.

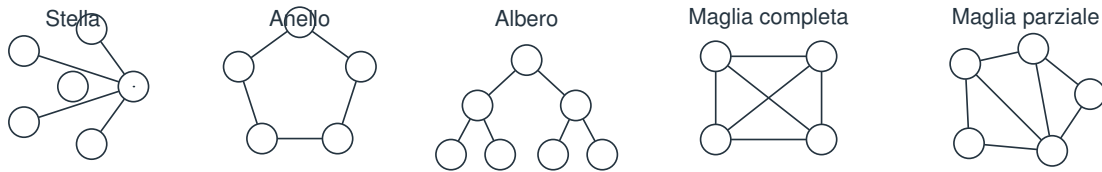


Figura 1.1. Esempi di topologie punto-punto: stella, anello, albero, rete completamente connessa e rete parzialmente connessa.

1.6.3 Canali condivisi

Un mezzo condiviso può essere utilizzato da più stazioni. Il segnale trasmesso da una stazione può essere fisicamente osservabile da più ricevitori e diventa quindi necessario un meccanismo di accesso al mezzo.

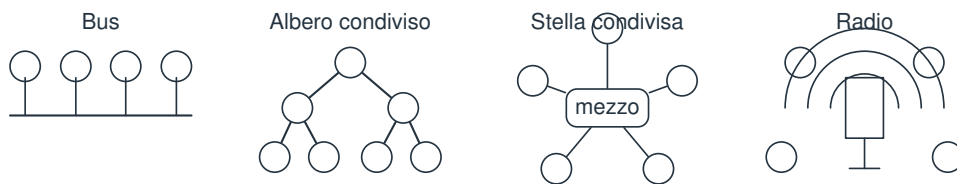


Figura 1.2. Esempi di topologie con mezzo condiviso: bus, albero, stella e mezzo radio.

La natura condivisa del mezzo non implica necessariamente una comunicazione broadcast a livello di servizio: indirizzamento e filtraggio possono selezionare il destinatario logico.

1.7 Architettura di rete e organizzazione a livelli

La complessità di una rete viene affrontata decomponendo le funzioni in livelli. Ogni livello offre servizi a quello superiore e usa i servizi del livello inferiore. Le entità dello stesso livello comunicano logicamente secondo un protocollo.

Questa separazione consente di modificare una parte dell'architettura senza dover riprogettare tutte le altre, a condizione che le interfacce e i servizi offerti restino coerenti.

Il modello ISO OSI organizza le funzioni in sette livelli e rimane utile come modello concettuale. Internet non implementa letteralmente i sette livelli OSI: utilizza una pila più compatta. Una corrispondenza utile, pur non perfetta, è la seguente:

Livello OSI	Funzione principale	Esempi Internet/rete
7-5	Applicazione, rappresentazione, sessione	DNS, HTTP, applicazioni
4	Trasporto end-to-end	TCP, UDP, QUIC
3	Instradamento e internetworking	IPv4, IPv6
2	Trame, accesso al mezzo, collegamento	Ethernet, WLAN
1	Segnali e trasmissione fisica	cavo, rame, fibra, radio

La tabella non deve essere letta come un'equivalenza normativa. Serve a distinguere le funzioni e a localizzare i meccanismi studiati nel volume.

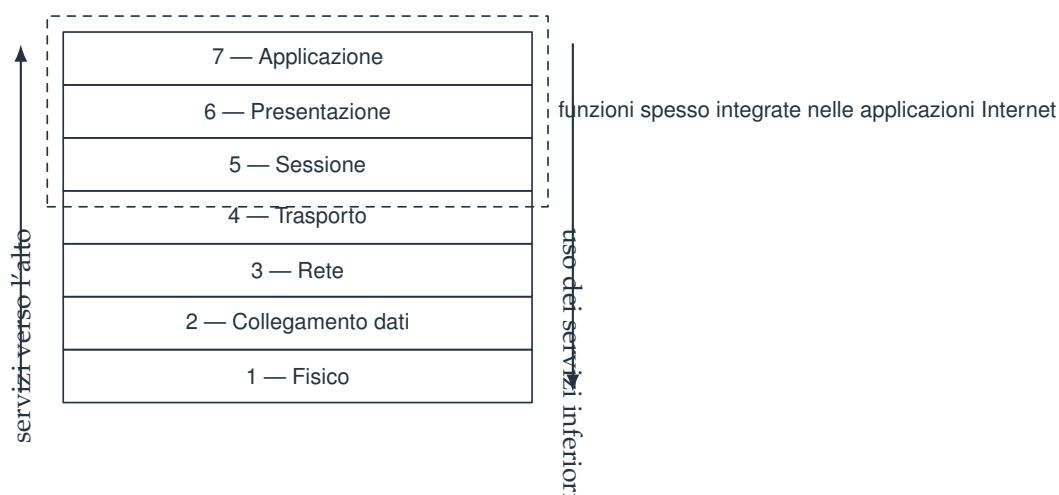


Figura 1.3. Modello di riferimento ISO OSI.

1.7.1 Piano dati, piano di controllo e gestione

Una seconda decomposizione, trasversale a quella per livelli, distingue le funzioni svolte da una rete in base al loro ruolo operativo. Questa distinzione è particolarmente utile nelle reti moderne perché separa il trattamento dei singoli pacchetti dalla costruzione e dalla supervisione dello stato che rende possibile tale trattamento.

Il *piano dati* comprende le operazioni applicate ai pacchetti durante il loro attraversamento dei nodi: classificazione, consultazione delle tabelle di inoltro, accodamento, scheduling, riscrittura di campi e trasmissione sull'interfaccia successiva. Sono operazioni che devono essere eseguite alla velocità del traffico e che, per questo motivo, sono normalmente basate su stato già calcolato.

Il *piano di controllo* costruisce e aggiorna quello stato. Ne fanno parte, per esempio, i protocolli di instradamento, i meccanismi con cui si apprendono raggiungibilità e topologie e le procedure che installano le decisioni risultanti nelle tabelle usate dal piano dati. Il piano di controllo opera quindi su una scala temporale diversa: non decide nuovamente il percorso completo per ogni pacchetto, ma prepara le informazioni che permettono al piano dati di inoltrarlo rapidamente.

Le funzioni di *gestione* configurano, osservano e amministrano la rete: politiche, provisioning, telemetria, allarmi e diagnostica. La separazione tra i tre piani è logica e non impone che essi risiedano in apparati fisicamente distinti. Anche nelle architetture software-defined, dove parte della logica di controllo può essere concentrata in controller, il piano dati continua a svolgere localmente l'inoltro dei pacchetti secondo lo stato installato.

1.7.2 Livello fisico

Il livello fisico descrive come i bit vengono rappresentati sul mezzo e comprende aspetti elettrici, ottici, radio, meccanici e temporali. Tra i suoi problemi rientrano codifica di linea, sincronismo, modulazione, connettività e caratteristiche del mezzo.

Una specifica di livello fisico deve quindi rendere non ambigue almeno la rappresentazione dei simboli, la temporizzazione, le direzioni di trasmissione e le caratteristiche dell'interfaccia verso il mezzo. Il modo concreto con cui queste funzioni vengono realizzate dipende dalla tecnologia; il principio generale è invece che due estremi possano interpretare nello stesso modo la sequenza fisica che trasporta i bit.

1.7.3 Livello di collegamento dati

Il livello 2 organizza il flusso di bit in trame, gestisce l'identificazione delle entità sul collegamento e può fornire rivelazione degli errori, ritrasmissioni, controllo di flusso e accesso a un mezzo condiviso.

Nei collegamenti reali non tutte queste funzioni sono necessariamente presenti nello stesso protocollo. Codifica, ARQ e accesso multiplo sono studiati separatamente nei capitoli successivi proprio per distinguere i principi generali dalle specifiche implementazioni.

1.7.4 Livello di rete

Il livello 3 determina come i pacchetti attraversano una rete composta da più nodi e collegamenti. Le funzioni fondamentali sono indirizzamento, forwarding e instradamento.

Quando più flussi condividono gli stessi collegamenti possono formarsi code e congestione. Instradamento e controllo della congestione non sono quindi problemi indipendenti: entrambi influenzano la distribuzione del traffico e l'occupazione delle risorse.

IP è l'esempio fondamentale di protocollo di internetworking studiato nel capitolo 8.

1.7.5 Livello di trasporto

Il livello di trasporto offre comunicazione end-to-end tra processi. Può fornire affidabilità, ordinamento, controllo di flusso, controllo della congestione e multiplexing di più applicazioni sulla stessa macchina.

TCP realizza un servizio affidabile orientato alla connessione; UDP offre invece un servizio a datagram minimale. QUIC realizza funzioni di trasporto affidabile nello spazio applicativo sopra UDP, integrando anche la sicurezza.

1.7.6 Livelli applicativi

Le funzioni tradizionalmente attribuite ai livelli OSI 5, 6 e 7 sono normalmente realizzate dalle applicazioni e dai protocolli applicativi: gestione del dialogo, rappresentazione dei dati, autenticazione, cifratura, nomi, trasferimento di oggetti e semantica del servizio.

La distinzione OSI rimane utile per separare concettualmente queste funzioni, ma nelle architetture Internet non corrisponde a tre strati di protocollo obbligatori.

1.7.7 Colloquio tra livelli

Quando un'unità informativa attraversa la pila protocollare, ogni livello può aggiungere informazioni di controllo necessarie al proprio peer. Il processo è detto incapsulamento; in ricezione avviene l'operazione inversa.

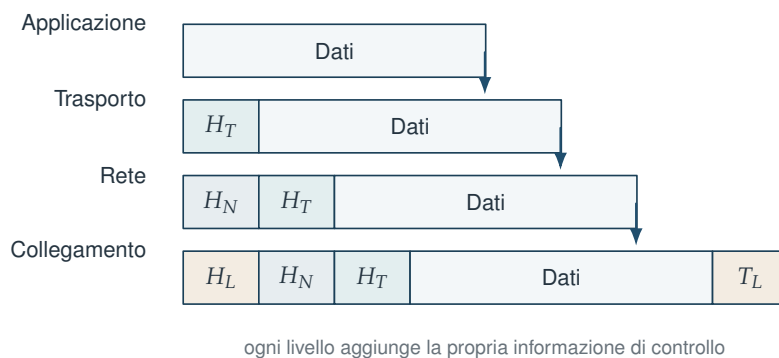


Figura 1.4. Incapsulamento e passaggio dei dati tra livelli.

Il principio è illustrato in figura 1.4: i dati prodotti da un livello superiore diventano

payload per il livello inferiore, che aggiunge il proprio header ed eventualmente un trailer.

Un servizio viene reso disponibile attraverso un punto di accesso. Nel modello OSI si usa il termine Service Access Point, SAP; nelle implementazioni concrete lo stesso principio assume forme differenti, ad esempio porte di trasporto, socket, identificatori di protocollo o interfacce tra moduli.

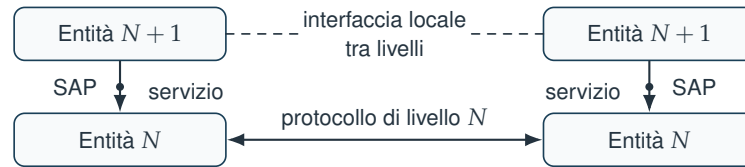


Figura 1.5. Relazione logica tra servizi, interfacce e protocolli di pari livello.

La distinzione essenziale è tra *servizio* e *protocollo*. Il servizio descrive ciò che un livello rende disponibile a quello superiore; il protocollo descrive invece le regole con cui due entità pari cooperano per realizzare quel servizio.

Nei modelli formali di servizio si usano spesso primitive quali request, indication, response e confirm. Il dettaglio delle primitive dipende dall'architettura, ma il principio resta generale: separare l'interfaccia locale tra livelli dal protocollo scambiato attraverso la rete.

Capitolo 2

Elementi di teoria delle code

Le reti a pacchetto condividono risorse finite: capacità trasmissiva, memoria e tempo di elaborazione. Quando il traffico offerto a una risorsa varia nel tempo, i pacchetti possono dover attendere prima di essere serviti. Nascono così le code, e con esse grandezze fondamentali come occupazione dei buffer, tempo di attesa, tempo di permanenza, probabilità di perdita e utilizzazione del servitore.

Questo capitolo introduce gli strumenti probabilistici necessari a studiare tali fenomeni. Il percorso parte da momenti e trasformate, prosegue con processi stocastici e catene di Markov a tempo continuo, e arriva ai principali modelli di coda. Le ipotesi di ogni modello vengono rese esplicite perché le formule hanno significato soltanto entro tali ipotesi: cambiare il processo degli arrivi, la distribuzione dei tempi di servizio, il numero di servitori o la capacità del buffer può cambiare in modo sostanziale il risultato.

In un collegamento di capacità costante C , un pacchetto di lunghezza F richiede un tempo di servizio $x = F/C$. Per questo motivo la variabilità della lunghezza dei pacchetti si traduce direttamente in variabilità dei tempi di servizio. Analogamente, il processo degli arrivi è descritto attraverso i tempi interarrivo. Arrivi e tempi di servizio costituiscono quindi le due sorgenti aleatorie fondamentali dei modelli sviluppati nel seguito.

2.1 Richiami utili: momenti e trasformate

Nel seguito vengono utilizzati frequentemente momenti, funzioni generatrici e trasformate di Laplace. Si richiamano qui le relazioni necessarie alle derivazioni dei sistemi a coda.

2.1.1 Variabili discrete non negative

Sia X una variabile aleatoria a valori in $\{0, 1, 2, \dots\}$, con funzione di massa $g_k = P[X = k]$. La funzione generatrice delle probabilità (PGF) è

$$G(z) = E[z^X] = \sum_{k=0}^{\infty} g_k z^k. \quad (2.1)$$

Per $0 \leq z < 1$ si può derivare la serie termine a termine. Facendo poi tendere z a 1 da sinistra, la convergenza monotona dà il momento fattoriale, eventualmente infinito. Se tale momento è finito, allora

$$\left. \frac{d^n G(z)}{dz^n} \right|_{z=1} = E[X(X-1) \cdots (X-n+1)]. \quad (2.2)$$

In particolare

$$G'(1) = E[X], \quad G''(1) = E[X(X-1)] = E[X^2] - E[X].$$

Ne segue

$$\text{Var}(X) = G''(1) + G'(1) - [G'(1)]^2. \quad (2.3)$$

2.1.2 Variabili continue non negative

Sia X una variabile aleatoria non negativa con funzione di ripartizione $A_X(x)$. La trasformata di Laplace-Stieltjes è $\mathcal{A}(s) = E[e^{-sX}] = \int_{[0,\infty)} e^{-sx} dA_X(x)$, per $s \geq 0$. Questa definizione vale

anche per distribuzioni discrete, miste o deterministiche. Quando esiste una densità $a(x)$, si riduce a

$$\mathcal{A}(s) = \int_0^\infty a(x)e^{-sx} dx = E[e^{-sX}]. \quad (2.4)$$

Per $s > 0$ il fattore esponenziale consente la derivazione sotto il segno di aspettazione. Se $E[X^n] < \infty$, la convergenza dominata permette poi di passare al limite $s \downarrow 0$:

$$\frac{d^n \mathcal{A}(s)}{ds^n} = E[(-X)^n e^{-sX}], \quad (2.5)$$

e quindi

$$E[X^n] = (-1)^n \left. \frac{d^n \mathcal{A}(s)}{ds^n} \right|_{s=0}. \quad (2.6)$$

Le derivate ai bordi si intendono quindi come limiti da destra per $s = 0$ e da sinistra per $s = 1$. Nel seguito, quando si scrive un integrale con densità, per una distribuzione generale va inteso il corrispondente integrale rispetto alla sua legge. Per esempio, se $X = x_0$ con probabilità uno, $\mathcal{A}(s) = e^{-sx_0}$. Queste relazioni verranno richiamate per i momenti dei tempi di servizio e delle distribuzioni di stato.

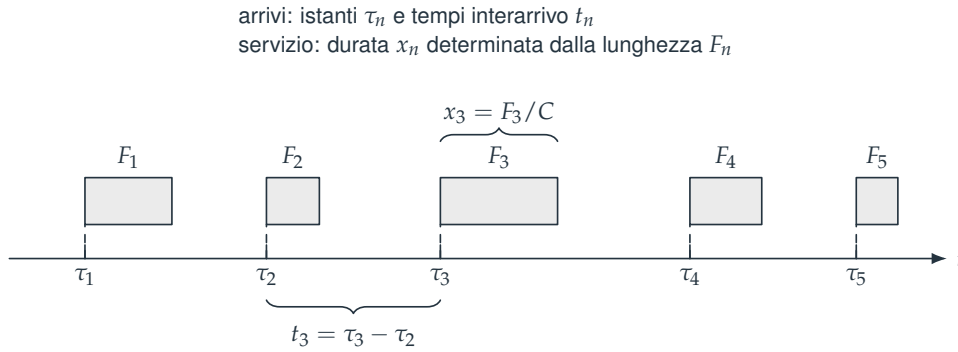


Figura 2.1. Esempio di tempo interarrivo e durata del pacchetto.

La figura 2.1 rende esplicite le due scale temporali che entrano nel modello: la distanza fra arrivi successivi e la durata del servizio richiesto da ciascun pacchetto. La loro distribuzione, e non soltanto il loro valore medio, determina l'andamento delle code.

Un nodo può avere più ingressi e più uscite. Quando ogni uscita dispone di una propria coda e di una propria capacità trasmissiva, come nella figura 2.2, ciascuna uscita può essere studiata inizialmente come un sistema a coda. Le interazioni tra più code verranno introdotte quando il modello lo richiederà.

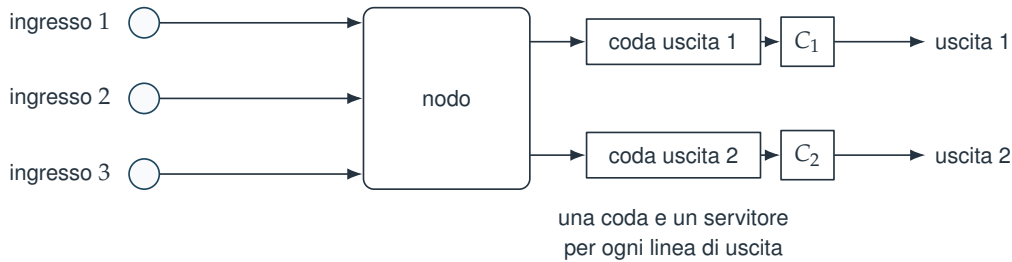


Figura 2.2. Esempio di nodo con tre ingressi e due uscite.

2.2 Generalità sui processi stocastici

Un processo stocastico è una famiglia di variabili aleatorie $\{X(t) : t \in T\}$ indicizzata da un parametro, qui normalmente il tempo. Per caratterizzarlo non basta conoscere la distribuzione di $X(t)$ a un singolo istante: occorre, in generale, specificare le distribuzioni congiunte a più istanti. Le proprietà seguenti ricorrono nel seguito e vanno tenute distinte.

- **Stazionarietà in senso stretto.** Per ogni n , ogni scelta di istanti $t_1, \dots, t_n$ e ogni traslazione τ compatibile con il dominio,

$$F_{X(t_1), \dots, X(t_n)}(x_1, \dots, x_n) = F_{X(t_1+\tau), \dots, X(t_n+\tau)}(x_1, \dots, x_n).$$

Le distribuzioni congiunte sono quindi invarianti rispetto a una traslazione dell'origine dei tempi. La stazionarietà in senso lato richiede invece, quando i secondi momenti esistono, media costante e autocorrelazione dipendente solo dalla differenza tra i due istanti. Le due nozioni non sono intercambiabili senza ipotesi aggiuntive.

- **Indipendenza tra campioni.** Per istanti distinti $t_1, \dots, t_n$, i campioni sono indipendenti se la distribuzione congiunta fattorizza nel prodotto delle marginali:

$$f_{X(t_1), \dots, X(t_n)}(x_1, \dots, x_n) = \prod_{i=1}^n f_{X(t_i)}(x_i).$$

È una proprietà molto più forte dell'assenza di correlazione.

- **Proprietà di Markov.** Il futuro, condizionato allo stato presente, non dipende dagli stati passati. Per $t_1 < \dots < t_n < t_{n+1}$,

$$P[X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n, \dots, X(t_1) = x_1] = P[X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n].$$

Quando lo spazio degli stati è numerabile si parla di catena di Markov.

- **Processi di nascita e morte.** Sono catene di Markov a stati interi non negativi nelle quali una transizione può modificare lo stato soltanto di una unità. Questa struttura descrive naturalmente numerosi modelli di popolazione e di coda.

2.3 Catene di Markov a tempo continuo

Consideriamo processi a salti con tassi di uscita finiti e senza esplosione: in ogni intervallo temporale finito avvengono quasi certamente soltanto finite transizioni. Le manipolazioni matriciali che seguono sono immediate per uno spazio degli stati finito, con tassi continui a tratti e limitati su intervalli compatti. Per uno spazio numerabile occorre inoltre giustificare la convergenza delle somme e gli scambi con i limiti; non si può trattare un generatore infinito non limitato come una matrice finita. Nei modelli di coda successivi useremo le equazioni di bilancio componente per componente su processi non esplosivi.

2.3.1 Sulla distribuzione dei tempi di permanenza

Per una catena di Markov a tempo continuo omogenea, la proprietà di Markov impone una struttura precisa anche al tempo trascorso in ciascuno stato. Il risultato seguente formalizza la proprietà di assenza di memoria del tempo di permanenza:

Teorema 1. *In una catena di Markov a tempo continuo omogenea, condizionatamente all'ingresso in uno stato E_i , il tempo di permanenza τ_i nello stato è distribuito esponenzialmente con parametro $q_i > 0$. Se $q_i = 0$, lo stato è assorbente e il tempo di permanenza è infinito con probabilità uno.*

Dimostrazione: sia

$$S_i(t) = P[\tau_i > t], \quad t \geq 0,$$

la funzione di sopravvivenza del tempo di permanenza nello stato E_i . Per la proprietà di Markov e per l'omogeneità temporale, una volta noto che al tempo s il processo si trova ancora nello stato E_i , la probabilità di rimanervi per ulteriori t unità di tempo non dipende dal tempo già trascorso. Pertanto

$$P[\tau_i > s + t \mid \tau_i > s] = P[\tau_i > t] = S_i(t).$$

Inoltre, dalla definizione di probabilità condizionata,

$$P[\tau_i > s + t \mid \tau_i > s] = \frac{P[\tau_i > s + t]}{P[\tau_i > s]},$$

perché $\{\tau_i > s + t\} \subseteq \{\tau_i > s\}$. Ne segue

$$S_i(s + t) = S_i(s)S_i(t), \quad s, t \geq 0. \quad (2.7)$$

La funzione S_i è non crescente, continua da destra e soddisfa $S_i(0) = 1$. Non può annullarsi a un tempo finito: altrimenti $S_i(t) = S_i(t/n)^n = 0$ implicherebbe $S_i(t/n) = 0$ anche per $n \rightarrow \infty$, contraddicendo la continuità in zero. Poniamo allora $g_i(t) = -\log S_i(t)$. La (2.7) diventa $g_i(s + t) = g_i(s) + g_i(t)$; per ogni razionale non negativo m/n segue $g_i(m/n) = (m/n)g_i(1)$. Continuità e densità dei razionali estendono l'identità a ogni $t \geq 0$. Con $q_i = g_i(1) \geq 0$ si ottiene

$$S_i(t) = e^{-q_i t}, \quad q_i \geq 0.$$

Se $q_i > 0$, derivando si ottiene la densità

$$f_{\tau_i}(t) = q_i e^{-q_i t}, \quad t \geq 0,$$

che è la densità esponenziale. Se $q_i = 0$, $S_i(t) = 1$ per ogni t e lo stato è assorbente.

2.3.2 Evoluzione dello stato nei sistemi a tempo continuo

Si definisce la probabilità di transizione dallo stato i allo stato j come:

$$p_{ij}(s, t) = P[X(t) = j \mid X(s) = i]$$

con $t > s$.

Se il processo passa da uno stato i all'istante s ad uno stato j all'istante t e se si considera un istante intermedio $s \leq u \leq t$, allora il processo deve passare attraverso un qualche stato intermedio k , come mostrato in figura 2.3.

Possiamo quindi esprimere $p_{ij}(s, t)$ come la somma delle probabilità di tutti i possibili stati intermedi:

$$p_{ij}(s, t) = \sum_k P[X(t) = j, X(u) = k \mid X(s) = i]$$

dove la somma è estesa a tutti gli stati del sistema. Ricordando che $P[A, B \mid C] = P[B \mid C]P[A \mid B, C]$ si ottiene:

$$\begin{aligned} p_{ij}(s, t) &= \sum_k P[X(u) = k \mid X(s) = i] P[X(t) = j \mid X(u) = k, X(s) = i] = \\ &= \sum_k P[X(u) = k \mid X(s) = i] P[X(t) = j \mid X(u) = k] = \\ &= \sum_k p_{ik}(s, u) p_{kj}(u, t) \end{aligned}$$

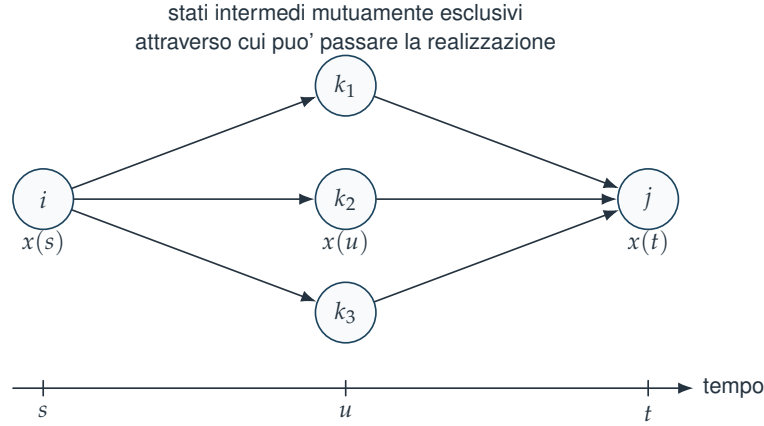


Figura 2.3. Possibili realizzazioni di un processo stocastico.

che è l'equazione di Chapman–Kolmogorov. È possibile dare una descrizione matriciale di questa equazione definendo la matrice delle probabilità di transizione:

$$\underline{H}(s, t) = [p_{ij}(s, t)]$$

per cui:

$$\underline{H}(s, t) = \underline{H}(s, u)\underline{H}(u, t)$$

2.3.3 Equazioni in avanti di Chapman Kolmogorov

Sia Δ_t il passo attraverso il quale si vuole frammentare l'asse temporale in modo da poter identificare gli istanti temporali tramite numeri interi. Definendo $s = m\Delta_t$ e $t = n\Delta_t$ si ha:

$$\begin{aligned}\underline{H}(s, t) &= \underline{H}(m\Delta_t, n\Delta_t) = \underline{H}(m\Delta_t, (n-1)\Delta_t)\underline{H}((n-1)\Delta_t, n\Delta_t) = \\ &= \underline{H}(m\Delta_t, (n-1)\Delta_t)\underline{V}((n-1)\Delta_t)\end{aligned}$$

dove è stato scelto $u = (n-1)\Delta_t$ e definita la seguente matrice

$$\underline{V}(n\Delta_t) = [p_{ij}(n\Delta_t, (n+1)\Delta_t)] = [p_{ij}(t, t + \Delta_t)]$$

Si procede valutando la differenza tra due matrici di transizione in istanti differenziati da un intervallo Δ_t :

$$\begin{aligned}&\underline{H}(m\Delta_t, n\Delta_t) - \underline{H}(m\Delta_t, (n-1)\Delta_t) = \\ &= \underline{H}(m\Delta_t, (n-1)\Delta_t)\underline{V}((n-1)\Delta_t) - \underline{H}(m\Delta_t, (n-1)\Delta_t) = \\ &= \underline{H}(m\Delta_t, (n-1)\Delta_t)[\underline{V}((n-1)\Delta_t) - \underline{I}]\end{aligned}$$

dove $\underline{I}$ è la matrice identità. Dividendo ambo i membri per Δ_t si ottiene:

$$\frac{\underline{H}(m\Delta_t, n\Delta_t) - \underline{H}(m\Delta_t, (n-1)\Delta_t)}{\Delta_t} = \underline{H}(m\Delta_t, (n-1)\Delta_t) \frac{\underline{V}((n-1)\Delta_t) - \underline{I}}{\Delta_t}$$

che facendo tendere Δ_t a zero si può scrivere come:

$$\frac{\partial \underline{H}(s, t)}{\partial t} = \underline{H}(s, t)\underline{Q}(t) \quad (2.8)$$

dove la matrice $\underline{\underline{Q}}(t)$ è detta matrice dei tassi o delle frequenze di transizione ed ha la seguente definizione:

$$\underline{\underline{Q}}(t) = \lim_{\Delta_t \rightarrow 0} \frac{\underline{\underline{V}}(t) - \underline{\underline{I}}}{\Delta_t} \quad (2.9)$$

L'equazione differenziale (2.8) viene detta equazione in avanti di Chapman Kolmogorov. La soluzione può essere determinata specificando le condizioni iniziali.

2.3.4 Osservazioni sui tassi di transizione

La matrice generatrice, formata dagli elementi $\underline{\underline{Q}}(t) = [q_{ij}(t)]$, descrive i tassi di uscita da un generico stato i verso un generico stato j . Si possono identificare due casi di interesse:

- $j = i$, cioè permanenza nello stesso stato,

$$q_{ii}(t) = \lim_{\Delta_t \rightarrow 0} \frac{p_{ii}(t, t + \Delta_t) - 1}{\Delta_t}$$

Quindi se in t il sistema si trova nello stato E_i la probabilità che avvenga una transizione verso un qualsiasi altro stato, diverso da i , nell'intervallo $[t, t + \Delta_t]$ è $1 - p_{ii}(t, t + \Delta_t) = -q_{ii}(t)\Delta_t + o(\Delta_t)$, e $-q_{ii}(t)$ è il tasso con cui si esce dallo stato E_i .

- $j \neq i$, cioè transizione di stato,

$$q_{ij}(t) = \lim_{\Delta_t \rightarrow 0} \frac{p_{ij}(t, t + \Delta_t)}{\Delta_t}$$

Quindi la probabilità di transizione da E_i a E_j è $q_{ij}(t)\Delta_t + o(\Delta_t)$ e $q_{ij}(t)$ rappresenta il tasso di spostamento da E_i ad E_j .

Per la conservazione della probabilità vale il seguente risultato:

Teorema 2. Per ogni stato i , la somma degli elementi della riga i della matrice generatrice è nulla:

$$\sum_j q_{ij}(t) = 0.$$

Equivalentemente, il termine diagonale è l'opposto della somma dei tassi di transizione verso gli altri stati.

Dimostrazione:

$$\sum_j q_{ij}(t) = \lim_{\Delta_t \rightarrow 0} \frac{p_{ii}(t, t + \Delta_t) - 1}{\Delta_t} + \sum_{j \neq i} \lim_{\Delta_t \rightarrow 0} \frac{p_{ij}(t, t + \Delta_t)}{\Delta_t} = \lim_{\Delta_t \rightarrow 0} \frac{\sum_j p_{ij}(t, t + \Delta_t) - 1}{\Delta_t} = 0$$

2.3.5 Equazioni in indietro di Chapman Kolmogorov

Si procede con la stessa metodologia utilizzata per ricavare la (2.8), ma si fissa $u = (m + 1)\Delta_t$:

$$\underline{\underline{H}}(m\Delta_t, n\Delta_t) = \underline{\underline{H}}(m\Delta_t, (m + 1)\Delta_t) \underline{\underline{H}}((m + 1)\Delta_t, n\Delta_t) = \underline{\underline{V}}(m\Delta_t) \underline{\underline{H}}((m + 1)\Delta_t, n\Delta_t)$$

Dividendo ambo i membri per Δ_t si ottiene:

$$\frac{\underline{\underline{H}}(m\Delta_t, n\Delta_t) - \underline{\underline{H}}((m + 1)\Delta_t, n\Delta_t)}{\Delta_t} = \frac{\underline{\underline{V}}(m\Delta_t) - \underline{\underline{I}}}{\Delta_t} \underline{\underline{H}}((m + 1)\Delta_t, n\Delta_t)$$

procedendo con il passaggio al limite rispetto a Δ_t si ottiene l'equazione in indietro di Chapman Kolmogorov:

$$\frac{\partial \underline{\underline{H}}(s, t)}{\partial s} = -\underline{\underline{Q}}(s) \underline{\underline{H}}(s, t) \quad (2.10)$$

La soluzione può essere determinata specificando le condizioni finali.

2.3.6 Soluzione generale per sistemi a tempo continuo

Nel caso generale, in cui la matrice generatrice dipende dal tempo, la matrice di transizione è la soluzione fondamentale di

$$\frac{\partial \underline{\underline{H}}(s, t)}{\partial t} = \underline{\underline{H}}(s, t) \underline{\underline{Q}}(t), \quad \underline{\underline{H}}(s, s) = \underline{\underline{I}}.$$

La soluzione si indica mediante l'esponenziale ordinato temporalmente. Poiché la matrice generatrice moltiplica a destra nella nostra convenzione per vettori riga, l'ordinamento va inteso coerentemente con la serie di Peano–Baker scritta qui sotto:

$$\underline{\underline{H}}(s, t) = \mathcal{T} \exp \left(\int_s^t \underline{\underline{Q}}(u) du \right).$$

Integrando l'equazione differenziale si ha $\underline{\underline{H}}(s, t) = \underline{\underline{I}} + \int_s^t \underline{\underline{H}}(s, u) \underline{\underline{Q}}(u) du$. Sostituendo ripetutamente questa stessa identità sotto l'integrale si ottiene la serie di Peano–Baker

$$\underline{\underline{H}}(s, t) = \underline{\underline{I}} + \int_s^t \underline{\underline{Q}}(u_1) du_1 + \int_s^t \int_s^{u_1} \underline{\underline{Q}}(u_2) \underline{\underline{Q}}(u_1) du_2 du_1 + \dots$$

Per una matrice finita con $\|\underline{\underline{Q}}(u)\| \leq M$ su $[s, t]$, la norma del termine di ordine n è al più $M^n (t-s)^n / n!$: la serie converge quindi assolutamente e soddisfa l'equazione integrale. Se le matrici generatrici commutano a tempi diversi, $\underline{\underline{Q}}(u) \underline{\underline{Q}}(v) = \underline{\underline{Q}}(v) \underline{\underline{Q}}(u)$ per ogni u, v , l'ordinamento temporale non è necessario e la soluzione si riduce a

$$\underline{\underline{H}}(s, t) = \exp \left(\int_s^t \underline{\underline{Q}}(u) du \right).$$

2.3.7 Probabilità di stato

Si definisce la probabilità di essere nello stato E_j all'istante di tempo t :

$$P_j(t) = P[X(t) = j].$$

Per definizione di matrice di transizione e sfruttando la soluzione generale delle equazioni di Chapman Kolmogorov, si ha:

$$\underline{\underline{P}}(t) = \underline{\underline{P}}(0) \underline{\underline{H}}(0, t).$$

Equivalentemente è possibile dare l'espressione della probabilità di stato del sistema tramite l'equazione differenziale:

$$\frac{d\underline{\underline{P}}(t)}{dt} = \underline{\underline{P}}(t) \underline{\underline{Q}}(t) \quad (2.11)$$

2.4 Catene di Markov a tempo continuo omogenee

Le catene di Markov si dicono omogenee quando la probabilità di transizione dipende soltanto dalla durata dell'intervallo, non dalla posizione dell'intervallo nel tempo. In particolare, non si intende che sia indipendente dal tempo trascorso.

Si procede allora con i seguenti cambi di notazione:

- $p_{ij}(s, s+t) = \hat{p}_{ij}(t)$ da cui segue $\underline{\underline{H}}(s, s+t) = \underline{\underline{\hat{H}}}(t)$;
- $q_{ij}(t) = \hat{q}_{ij}$ da cui segue $\underline{\underline{Q}}(t) = \underline{\underline{\hat{Q}}}$.

Di seguito vengono adattati i risultati ottenuti nella sezione precedente alla luce delle nuove notazioni.

Equazione di Chapman Kolmogorov:

$$\hat{H}(s+t) = \underline{H}(u, u+s+t) = \underline{H}(u, u+s) \underline{H}(u+s, u+s+t) = \hat{H}(s) \hat{H}(t)$$

Equazione in avanti di Chapman Kolmogorov:

$$\frac{d\hat{H}(t)}{dt} = \frac{d\underline{H}(s, s+t)}{dt} = \underline{H}(s, s+t) \underline{Q}(s+t) = \hat{H}(t) \hat{Q}$$

Equazione in indietro di Chapman Kolmogorov:

$$\frac{d\hat{H}(t)}{dt} = \frac{d\underline{H}(u-t, u)}{dt} = -\frac{d\underline{H}(s, u)}{ds} = \underline{Q}(s) \underline{H}(s, u) = \hat{Q} \hat{H}(t)$$

Soluzioni equazioni avanti ed indietro:

$$\hat{H}(0) = \underline{I}, \quad \hat{H}(t) = e^{\hat{Q}t}$$

Probabilità di stato:

$$\frac{d\underline{P}(t)}{dt} = \underline{P}(t) \hat{Q} \quad (2.12)$$

2.5 Processi di nascita e morte

Si consideri un processo di nascita e morte tempo continuo con un tasso di nascita $\lambda_k = q_{k,k+1}$ ed un tasso di morte $\mu_k = q_{k,k-1}$, dipendenti dallo stato k ma non dal tempo, così da identificare una catena omogenea.

Si supponga di identificare lo stato generico E_k con il numero degli utenti presenti nel sistema, per cui lo stato k implica k utenti.

Nel seguito, per alleggerire la notazione, si omette il cappello sulle grandezze introdotte nella sezione precedente.

2.5.1 Condizione di adiacenza

La definizione di processo di nascita e morte è basata sulla condizione di adiacenza, esprimibile come:

$$q_{kj} = 0 \quad \forall |k-j| > 1$$

e come conseguenza dell'azzeramento dei tassi uscenti da un nodo qualsiasi, si ha:

$$\sum_j q_{kj} = q_{k,k-1} + q_{k,k} + q_{k,k+1} = 0$$

da cui si ottiene l'importante relazione

$$q_{k,k} = -q_{k,k-1} - q_{k,k+1} = -(\lambda_k + \mu_k)$$

che comporta la seguente matrice dei tassi di transizione:

$$\underline{Q} = \begin{bmatrix} -\lambda_0 & \lambda_0 & 0 & \dots \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & \dots \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

È interessante osservare che $q_{00} = -\lambda_0$ in quanto $\mu_0 = q_{0,-1}$ deve essere nullo, dato che non ha senso uno stato con un numero di utenti negativo.

Una volta nota la matrice generatrice, è possibile esplicitare l'equazione differenziale (2.12), relativa alle probabilità di stato, per lo stato 0 ed il generico stato $k \geq 1$:

$$\begin{cases} \frac{dP_k(t)}{dt} = -(\lambda_k + \mu_k)P_k(t) + \lambda_{k-1}P_{k-1}(t) + \mu_{k+1}P_{k+1}(t) & k \geq 1 \\ \frac{dP_0(t)}{dt} = -\lambda_0P_0(t) + \mu_1P_1(t) & k = 0 \end{cases} \quad (2.13)$$

2.5.2 Analisi processi nascita e morte

Le equazioni di Chapman Kolmogorov descrivono in modo generale una qualunque catena di Markov tra cui anche i processi di nascita e morte. Di seguito si procede dimostrando le equazioni caratteristiche dei processi di nascita e morte con un approccio alternativo, verificando infine l'equivalenza delle due metodologie.

Si considerino le seguenti ipotesi locali, che definiscono i tassi del processo per $\Delta_t \rightarrow 0$:

- catena omogenea
- $P[\text{una nascita in } [t, t + \Delta_t] | k \text{ utenti}] = \lambda_k \Delta_t + o(\Delta_t)$ identificato come caso N1
- $P[\text{una morte in } [t, t + \Delta_t] | k \text{ utenti}] = \mu_k \Delta_t + o(\Delta_t)$ identificato come caso M1
- $P[\text{nessuna transizione in } [t, t + \Delta_t] | k \text{ utenti}] = 1 - (\lambda_k + \mu_k) \Delta_t + o(\Delta_t)$
- $P[\text{due o più transizioni} | k \text{ utenti}] = o(\Delta_t)$

Per procedere si segue lo schema riportato in figura 2.4, distinguendo il caso di popolazione nulla dal caso generico. Lo stato 0, infatti, può essere raggiunto per morte dallo stato 1 o per permanenza ma comunque mai per nascita, in quanto non ha senso uno stato negativo.

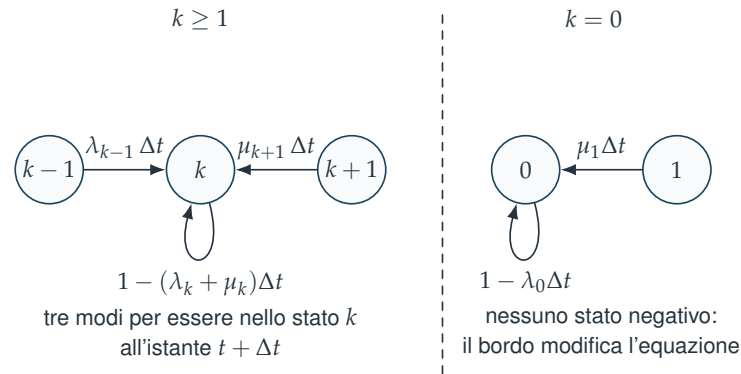


Figura 2.4. Evoluzione dello stato del sistema.

$$\begin{cases} P_k(t + \Delta_t) = P_k(t)p_{kk}(\Delta_t) + P_{k-1}(t)p_{k-1k}(\Delta_t) + P_{k+1}(t)p_{k+1k}(\Delta_t) + o(\Delta_t) & k \geq 1 \\ P_0(t + \Delta_t) = P_0(t)p_{00}(\Delta_t) + P_1(t)p_{10}(\Delta_t) + o(\Delta_t) & k = 0 \end{cases}$$

Inoltre occorre considerare che il sistema deve essere in uno degli stati possibili:

$$\sum_{k=0}^{\infty} P_k(t) = 1 \quad (2.14)$$

Si procede quindi esplicitando le relazioni di evoluzione della popolazione, attraverso le ipotesi fatte, per il caso $k \geq 1$:

$$P_k(t + \Delta_t) = \begin{cases} P_k(t)[1 - (\lambda_k + \mu_k)\Delta_t + o(\Delta_t)] & \text{nessuna transizione} \\ +P_{k-1}(t)[\lambda_{k-1}\Delta_t + o(\Delta_t)] & \text{N1} \\ +P_{k+1}(t)[\mu_{k+1}\Delta_t + o(\Delta_t)] + o(\Delta_t) & \text{M1} \end{cases}$$

e per il caso $k = 0$:

$$P_0(t + \Delta_t) = \begin{cases} P_0(t)[1 - \lambda_0\Delta_t + o(\Delta_t)] & \text{nessuna transizione} \\ +P_1(t)[\mu_1\Delta_t + o(\Delta_t)] + o(\Delta_t) & \text{M1} \end{cases}$$

Riassumendo:

$$\begin{cases} P_k(t + \Delta_t) = P_k(t) - (\lambda_k + \mu_k)\Delta_t P_k(t) \\ \quad + \lambda_{k-1}\Delta_t P_{k-1}(t) + \mu_{k+1}\Delta_t P_{k+1}(t) + o(\Delta_t) & k \geq 1 \\ P_0(t + \Delta_t) = P_0(t) - \lambda_0\Delta_t P_0(t) + \mu_1\Delta_t P_1(t) + o(\Delta_t) & k = 0 \end{cases}$$

Nell'ottica di ottenere delle equazioni differenziali per descrivere l'evoluzione del sistema, si procede identificando i rapporti incrementali per le funzioni che descrivono lo stato del sistema $P_k(t)$:

$$\begin{cases} \frac{P_k(t + \Delta_t) - P_k(t)}{\Delta_t} = -(\lambda_k + \mu_k)P_k(t) + \lambda_{k-1}P_{k-1}(t) + \mu_{k+1}P_{k+1}(t) + \frac{o(\Delta_t)}{\Delta_t} & k \geq 1 \\ \frac{P_0(t + \Delta_t) - P_0(t)}{\Delta_t} = -\lambda_0P_0(t) + \mu_1P_1(t) + \frac{o(\Delta_t)}{\Delta_t} & k = 0 \end{cases}$$

Facendo tendere Δ_t a zero si ottengono infine:

$$\begin{cases} \frac{dP_k(t)}{dt} = -(\lambda_k + \mu_k)P_k(t) + \lambda_{k-1}P_{k-1}(t) + \mu_{k+1}P_{k+1}(t) & k \geq 1 \\ \frac{dP_0(t)}{dt} = -\lambda_0P_0(t) + \mu_1P_1(t) & k = 0 \end{cases} \quad (2.15)$$

che risulta uguale alla (2.13).

Infine in figura 2.5 si riporta un diagramma esplicativo dell'evoluzione del sistema che permette di scrivere in modo immediato le equazioni (2.15). Infatti, se consideriamo il caso $k = 0$ abbiamo che la variazione della probabilità di essere nello stato E_0 , $dP_0(t)/dt$, aumenta quanto più si entra nello stato E_0 , con tasso μ_1 a partire da P_1 e diminuisce quanto più si esce dallo stato E_0 con tasso λ_0 a partire da P_0 . Simili considerazioni permettono di ricavare dalla figura 2.5 anche l'espressione valida per un qualunque $k \geq 1$.

2.5.3 Processi di Poisson

Si consideri un processo di sole nascite, con $\lambda_k = \lambda$ e $\mu_k = 0$. Le equazioni che caratterizzano il sistema si ottengono specializzando (2.15):

$$\begin{cases} \frac{dP_k(t)}{dt} = -\lambda P_k(t) + \lambda P_{k-1}(t) & k \geq 1 \\ \frac{dP_0(t)}{dt} = -\lambda P_0(t) & k = 0 \end{cases}$$

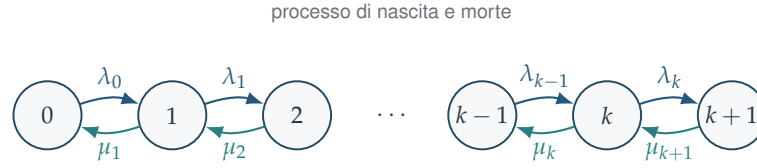


Figura 2.5. Diagramma delle frequenze di transizioni.

a cui bisogna aggiungere le condizioni iniziali:

$$P_k(0) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

Per trovare la soluzione a questo sistema di equazioni differenziali si procede nel dominio delle trasformate di Laplace.

Per il termine con $k = 0$ si ha:

$$s\mathcal{P}_0(s) - P_0(0) = -\lambda\mathcal{P}_0(s) \Rightarrow (s + \lambda)\mathcal{P}_0(s) = P_0(0) = 1 \Rightarrow \mathcal{P}_0(s) = \frac{1}{s + \lambda}$$

Mentre per $k > 0$:

$$s\mathcal{P}_k(s) - P_k(0) = -\lambda\mathcal{P}_k(s) + \lambda\mathcal{P}_{k-1}(s) \Rightarrow (s + \lambda)\mathcal{P}_k(s) = \lambda\mathcal{P}_{k-1}(s) \Rightarrow$$

$$\mathcal{P}_k(s) = \frac{\lambda}{s + \lambda} \mathcal{P}_{k-1}(s)$$

Combinando le due equazioni si ottiene:

$$\mathcal{P}_k(s) = \frac{\lambda^k}{(s + \lambda)^k} \mathcal{P}_0(s) = \frac{\lambda^k}{(s + \lambda)^{k+1}}$$

L'espressione di $P_k(t)$ nel dominio del tempo, ottenibile antitrasformando $\mathcal{P}_k(s)$, risulta:

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

per $k \geq 0$ e $t \geq 0$. Questa distribuzione, riportata in figura 2.6, è la distribuzione di Poisson, descrittiva di molti processi fisici. In particolare $P_k(t)$ rappresenta la probabilità di essere nello stato E_k all'istante t supponendo di essersi trovati nello stato E_0 all'istante $t = 0$; quindi rappresenta la probabilità di avere k nascite nell'intervallo di tempo t . La distribuzione di Poisson si ottiene anche come limite binomiale per un grande numero di prove indipendenti, quando la probabilità del singolo evento tende a zero mantenendo costante il numero medio atteso di eventi. Occorre distinguere due massimi. Per $k \geq 1$ fissato, $d \log P_k(t) / dt = k/t - \lambda$: come funzione di t , $P_k(t)$ è massima in $t = k/\lambda$; per $k = 0$ il massimo è in $t = 0$. Per $t > 0$ fissato, invece, $P_{k+1}(t)/P_k(t) = \lambda t / (k + 1)$: la moda intera è $\lfloor \lambda t \rfloor$, con due mode adiacenti $\lambda t - 1$ e λt quando λt è un intero positivo. Il primo confronto varia il tempo, il secondo il numero di arrivi.

2.5.4 Momenti della distribuzione di Poisson

Applicando la metodologia per il calcolo dei momenti di una variabile aleatoria discreta introdotta all'inizio del capitolo, si ha:

$$G(z) = \sum_{k=0}^{\infty} P_k(t) z^k = \sum_{k=0}^{\infty} \frac{(\lambda t)^k}{k!} e^{-\lambda t} z^k = e^{-\lambda t} \sum_{k=0}^{\infty} \frac{(\lambda t z)^k}{k!} = e^{-\lambda t} e^{\lambda t z} = e^{\lambda t(z-1)}$$

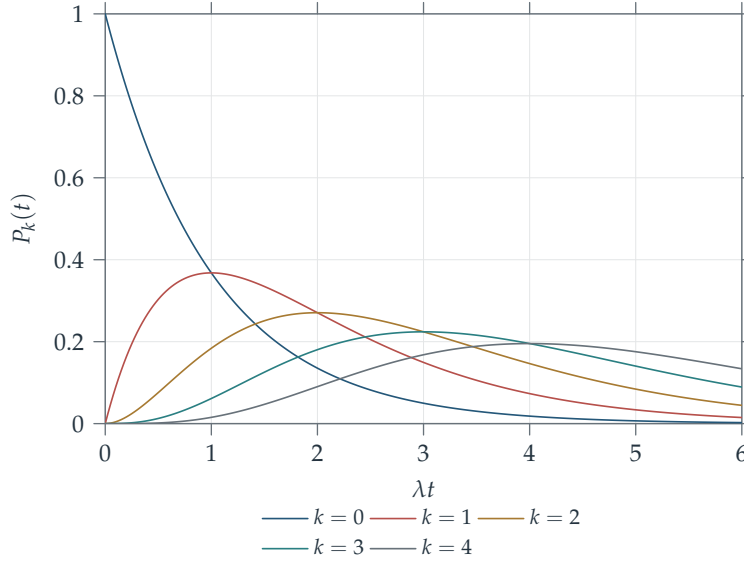


Figura 2.6. Distribuzione di Poisson.

quindi:

$$E[k] = \left. \frac{dG(z)}{dz} \right|_{z=1} = \left. \lambda t e^{\lambda t(z-1)} \right|_{z=1} = \lambda t$$

$$E[k^2] = \left. \frac{d^2G(z)}{dz^2} \right|_{z=1} + \left. \frac{dG(z)}{dz} \right|_{z=1} = (\lambda t)^2 e^{\lambda t(z-1)} \Big|_{z=1} + \lambda t = (\lambda t)^2 + \lambda t$$

$$\sigma_k^2 = E[k^2] - E^2[k] = (\lambda t)^2 + \lambda t - (\lambda t)^2 = \lambda t$$

2.5.5 Tempi interarrivo, distribuzione esponenziale

Si consideri la variabile $\tilde{t} \geq 0$ indicante il tempo intercorso tra due arrivi successivi:

$$A(t) = P[\tilde{t} \leq t] = 1 - P[\tilde{t} > t] = 1 - P_0(t) = 1 - e^{-\lambda t}$$

con $t \geq 0$. Derivando rispetto a t si ottiene la densità degli interarrivi risulta esponenziale:

$$a(t) = \frac{dA(t)}{dt} = \lambda e^{-\lambda t} \quad (2.16)$$

Come mostrato dal teorema 1, è interessante osservare che il futuro di una variabile aleatoria con densità esponenziale è indipendente dalla storia passata e che la sua densità rimane invariante nel tempo:

$$\begin{aligned} P[\tilde{t} \leq t + t_0 | \tilde{t} > t_0] &= \frac{P[t_0 \leq \tilde{t} \leq t_0 + t]}{P[\tilde{t} > t_0]} = \frac{P[\tilde{t} \leq t_0 + t] - P[\tilde{t} \leq t_0]}{P[\tilde{t} > t_0]} = \\ &= \frac{A(t_0 + t) - A(t_0)}{1 - A(t_0)} = \frac{1 - e^{-\lambda(t_0+t)} - 1 + e^{-\lambda t_0}}{1 - (1 - e^{-\lambda t_0})} = 1 - e^{-\lambda t} = A(t) \end{aligned}$$

Questo risultato permette di osservare che se si suppone di lasciar passare un certo intervallo t_0 , durante il quale non c'è nessun arrivo, e si è interessati alla probabilità che il prossimo arrivo avvenga entro un tempo t da t_0 , tale probabilità è uguale a quella di avere un arrivo

entro un tempo t dall'istante zero. Quindi la distribuzione del tempo residuo fino al prossimo arrivo non dipende da quanto tempo è già trascorso dall'ultimo arrivo. Per una variabile continua non si parla infatti della probabilità di assumere un singolo valore, che è nulla, ma della sua distribuzione.

2.5.6 Momenti della distribuzione esponenziale

Usando le relazioni introdotte all'inizio del capitolo per i momenti di una variabile aleatoria continua, per valutare i momenti della distribuzione esponenziale occorre considerare:

$$\mathcal{A}(s) = \int_0^\infty \lambda e^{-\lambda t} e^{-st} dt = \lambda \int_0^\infty e^{-t(\lambda+s)} dt = \frac{\lambda}{s + \lambda} \quad (2.17)$$

da cui risulta:

$$\begin{aligned} E_t[t] &= - \left. \frac{d\mathcal{A}(s)}{ds} \right|_{s=0} = \left. \frac{\lambda}{(s + \lambda)^2} \right|_{s=0} = \frac{1}{\lambda} \\ E_t[t^2] &= \left. \frac{d^2\mathcal{A}(s)}{ds^2} \right|_{s=0} = \left. \frac{2\lambda}{(s + \lambda)^3} \right|_{s=0} = \frac{2}{\lambda^2} \\ \sigma_t^2 &= E_t[t^2] - E_t^2[t] = \frac{2}{\lambda^2} - \frac{1}{\lambda^2} = \frac{1}{\lambda^2} \end{aligned} \quad (2.18)$$

2.5.7 Tempo per collezionare k arrivi, distribuzione di Erlang

Sia t_n il tempo tra l'arrivo n e l'arrivo $n - 1$. La densità di t_n è nota, in quanto precedentemente calcolata come tempo interarrivo, dalla (2.16). Sia x il tempo complessivo per collezionare k arrivi:

$$x = \sum_{n=1}^k t_n$$

Gli interarrivi del processo di Poisson sono indipendenti e hanno la stessa legge esponenziale. Per $k \geq 1$ la densità di x è pertanto la convoluzione delle loro densità e la trasformata è il prodotto:

$$\mathcal{F}(s) = [\mathcal{A}(s)]^k = \frac{\lambda^k}{(s + \lambda)^k} \Rightarrow f_x(x) = \frac{\lambda^k x^{k-1}}{(k-1)!} e^{-\lambda x} \quad (2.19)$$

e viene detta distribuzione di Erlang.

2.6 Processi di nascita e morte: soluzioni in equilibrio

Lo studio del regime di transitorio è espresso in termini generali dalle equazioni di Chapman Kolmogorov e per quanto riguarda la sottoclasse dei processi di nascita e morte dalle equazioni (2.15). Tale analisi risulta comunque complessa, per cui si considera il regime stazionario. Quando la catena è positiva ricorrente e ammette una distribuzione stazionaria, le probabilità di stato possono convergere, sotto le usuali condizioni di ergodicità, a valori indipendenti dal tempo. Si pone allora:

$$P_k = \lim_{t \rightarrow \infty} P_k(t) \quad (2.20)$$

cioè, in regime stazionario, una osservazione del sistema trova k utenti con probabilità P_k . Le equazioni di equilibrio si ottengono imponendo che le probabilità stazionarie non varino nel tempo, quindi:

$$\lim_{t \rightarrow \infty} \frac{dP_k(t)}{dt} = 0$$

Specializzando le equazioni (2.15) si ha:

$$\begin{aligned} 0 &= -(\lambda_k + \mu_k)P_k + \lambda_{k-1}P_{k-1} + \mu_{k+1}P_{k+1} & k \geq 1 \\ 0 &= -\lambda_0P_0 + \mu_1P_1 & k = 0 \end{aligned} \quad (2.21)$$

Vale il seguente risultato:

Teorema 3. *In equilibrio i processi di Markov di nascita e morte godono della seguente relazione di conservazione del flusso, per ogni $k \geq 1$:*

$$\lambda_{k-1}P_{k-1} = \mu_kP_k.$$

Infatti, introducendo:

$$g_k = \lambda_kP_k - \mu_{k+1}P_{k+1}$$

si può riscrivere (2.21) come: $g_{k-1} = g_k \forall k \geq 1$. Dal caso $k = 0$ si ha $g_0 = 0$ e quindi $g_k = 0 \forall k \geq 0$. In definitiva:

$$g_{k-1} = 0 = \lambda_{k-1}P_{k-1} - \mu_kP_k \Rightarrow \lambda_{k-1}P_{k-1} = \mu_kP_k$$

Un'interpretazione grafica di questa relazione è riportata in figura 2.7 dove il bilancio mostra che il flusso che entra in una sezione deve eguagliare il flusso che esce dalla sezione stessa.

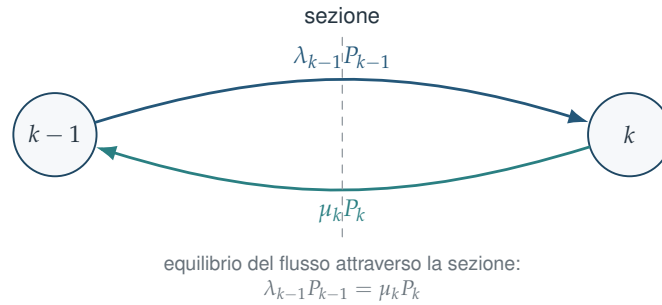


Figura 2.7. Teorema della conservazione del flusso.

2.6.1 Soluzione generale

Dal teorema precedente si ha:

$$P_k = \frac{\lambda_{k-1}}{\mu_k} P_{k-1}$$

per cui iterando fino a $k = 1$ si ottiene:

$$P_k = \frac{\lambda_0 \dots \lambda_{k-1}}{\mu_1 \dots \mu_k} P_0 = P_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \quad (2.22)$$

Considerando poi la condizione di normalizzazione (2.14). Per uno spazio degli stati infinito la distribuzione stazionaria esiste in questa forma solo se la serie di normalizzazione è finita:

$$1 + \sum_{k=1}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} < \infty.$$

In tal caso:

$$\sum_{k=0}^{\infty} P_k = 1 = P_0 \left[1 + \sum_{k=1}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \right] \Rightarrow P_0 = \left[1 + \sum_{k=1}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \right]^{-1} \quad (2.23)$$

2.6.2 Risultato di Little

Le distribuzioni di stato consentono di ricavare il numero medio di utenti presenti; per tradurre questa grandezza in un tempo medio di permanenza serve una relazione che non dipenda dalla struttura markoviana dei modelli precedenti. La relazione di Little fornisce proprio questo collegamento ed è molto più generale dei sistemi di nascita e morte:

Teorema 4. *Consideriamo un sistema conservativo per i clienti ammessi, con tasso medio $\lambda > 0$ e medie di lungo periodo finite $E[k]$ ed $E[T]$. Supponiamo che i contributi dei clienti presenti ai bordi dell'intervallo di osservazione siano $o(t)$, nel senso precisato nella dimostrazione. Allora vale*

$$E[k] = \lambda E[T]. \quad (2.24)$$

La relazione non dipende dalla particolare disciplina di servizio, purché valgano le condizioni enunciate. In regime stazionario ed ergodico tali medie di lungo periodo coincidono con le corrispondenti medie probabilistiche. Se il sistema rifiuta utenti, λ conta soltanto gli ammessi; se un ammesso abbandona il sistema, anche l'abbandono deve essere contato come uscita e nel suo tempo di permanenza.

Dimostrazione: indichiamo con a_n e d_n rispettivamente gli istanti di arrivo e di partenza dell' n -esimo utente ammesso e con $T_n = d_n - a_n$ il suo tempo di permanenza. Sia $N_a(t)$ il numero di utenti ammessi nell'intervallo $[0, t]$ e sia $N(u)$ il numero di utenti presenti all'istante u . L'area sotto la traiettoria del numero di utenti è

$$\Gamma(t) = \int_0^t N(u) du.$$

Supponendo, per semplicità, il sistema vuoto all'istante zero, ogni utente arrivato prima di t contribuisce a quest'area per la parte del proprio tempo di permanenza trascorsa entro $[0, t]$. Quindi

$$\Gamma(t) = \sum_{n=1}^{N_a(t)} T_n - R(t),$$

dove

$$R(t) = \sum_{n: a_n \leq t < d_n} (d_n - t) \geq 0$$

è il tempo residuo, successivo a t , degli utenti ancora presenti. La presenza di un numero finito di utenti iniziali aggiunge soltanto un ulteriore termine di bordo. Introduciamo, come nella derivazione classica, le medie empiriche sull'intervallo $[0, t]$:

$$\bar{\lambda}(t) = \frac{N_a(t)}{t},$$

$$\bar{T}(t) = \frac{1}{N_a(t)} \sum_{n=1}^{N_a(t)} T_n,$$

$$\bar{N}(t) = \frac{\Gamma(t)}{t}.$$

Dividendo l'identità dell'area per t si ottiene

$$\bar{N}(t) = \bar{\lambda}(t) \bar{T}(t) - \frac{R(t)}{t}.$$

Questa uguaglianza rende esplicito anche il termine di bordo che nella forma intuitiva della dimostrazione viene spesso sottinteso. L'ipotesi di bordo enunciata richiede precisamente

$R(t)/t \rightarrow 0$, oltre all'analogia trascurabilità dei clienti iniziali. Non è corretto sostituire questa condizione con la sola parola "stabilità" senza specificare le medie considerate. Usando i limiti empirici assunti,

$$\bar{N}(t) \rightarrow E[k], \quad \bar{\lambda}(t) \rightarrow \lambda, \quad \bar{T}(t) \rightarrow E[T].$$

Passando al limite per $t \rightarrow \infty$ segue

$$E[k] = \lambda E[T],$$

che dimostra il risultato.

2.6.3 Notazione di Kendall

Per identificare i sistemi a coda si utilizza solitamente la notazione di Kendall, che li descrive nel seguente formato:

$$A/B/c/Y/N/Z$$

con i seguenti significati:

- A : distribuzione dei tempi interarrivo;
- B : distribuzione dei tempi di servizio;
- c : numero dei servitori;
- Y : capacità del sistema, cioè massimo numero di utenti presenti;
- N : numero dei potenziali utenti nella popolazione;
- Z : disciplina di servizio della coda.

Alcuni di questi parametri possono non essere specificati. In particolare per quanto riguarda A e B sono classici i seguenti formalismi:

- G : distribuzione generale;
- M : distribuzione markoviana, cioè senza memoria: tempi interarrivo o di servizio esponenziali; nel caso degli interarrivi questo equivale a un processo di arrivo di Poisson;
- D : distribuzione deterministica.

2.6.4 Parametri caratteristici dei sistemi a coda

Il *fattore di utilizzazione* ρ misura la frazione della capacità del servitore richiesta dal lavoro ammesso. Per un singolo servitore si normalizza la capacità di lavoro a una unità per unità di tempo. Se $\bar{\lambda}$ è il tasso medio degli arrivi effettivamente ammessi ed $E[x]$ il tempo medio di servizio richiesto da ciascun cliente, il lavoro medio offerto per unità di tempo è

$$\rho = \bar{\lambda} E[x]. \quad (2.25)$$

Se $\rho > 1$, il lavoro ammesso in un intervallo lungo t supera la capacità t disponibile: il lavoro residuo è almeno $(\rho - 1)t + o(t)$ e non può rimanere stazionario. Nei modelli stocastici non degeneri sviluppati nel seguito si richiede $\rho < 1$. Non è però una condizione strettamente necessaria per ogni processo possibile: un sistema $D/D/1$ sincronizzato, con interarrivi e servizi identici, può funzionare con $\rho = 1$ senza accumulare attesa. Inoltre, l'esistenza di un regime stazionario non implica da sola un tempo medio d'attesa finito: in $M/G/1$ conta anche il secondo momento del servizio.

Teorema 5. *Per un sistema stabile $G/G/1$ a singolo servitore, stazionario ed ergodico, con disciplina work-conserving e non-idling, la frazione di tempo in cui il servitore è occupato è*

$$\rho = 1 - P_0,$$

dove P_0 è la probabilità stazionaria che il sistema sia vuoto.

Dimostrazione. In un intervallo lungo t , siano $N_a(t)$ gli arrivi ammessi e $x_1, \dots, x_{N_a(t)}$ i relativi tempi di servizio. Per stazionarietà ed ergodicità,

$$\frac{N_a(t)}{t} \longrightarrow \bar{\lambda}, \quad \frac{1}{N_a(t)} \sum_{n=1}^{N_a(t)} x_n \longrightarrow E[x].$$

Il lavoro richiesto per unità di tempo converge quindi a $\bar{\lambda}E[x] = \rho$. In un sistema stabile, i termini di bordo dovuti al lavoro presente agli estremi dell'intervallo sono trascurabili rispetto a t ; perciò la stessa quantità è la frazione di tempo durante la quale il servitore lavora. Essendo la disciplina non-idling e work-conserving, il servitore lavora se e solo se il sistema non è vuoto. La frazione di tempo occupata è dunque $1 - P_0$ e segue

$$\rho = \bar{\lambda}E[x] = 1 - P_0. \quad (2.26)$$

Equivalentemente, il bilancio del lavoro può essere scritto

$$\frac{1 - P_0}{E[x]} = \bar{\lambda}. \quad (2.27)$$

Altri parametri di interesse nei sistemi a coda sono:

- $E[k]$ il numero medio di utenti presenti nel sistema
- $E[q]$ il numero medio di utenti in coda
- $E[T]$ il tempo medio speso da un utente nel sistema
- $E[T_q]$ il tempo medio speso da un utente in coda

e risultano caratteristici per ogni sistema a coda.

Per comodità, nei modelli markoviani successivi si indica inoltre con $A = \lambda/\mu$ il traffico offerto adimensionale. In questo uso A è uno scalare e non va confuso con il primo campo della notazione di Kendall, dove la lettera A rappresenta genericamente la legge dei tempi interarrivo.

2.6.5 Sistema $M/M/1$

In questo sistema si ha un solo servitore, un processo di arrivo di Poisson di tasso λ e tempi di servizio esponenziali di parametro μ , quindi:

$$\begin{cases} \lambda_k = \lambda \\ \mu_k = \mu \end{cases}$$

Per l'esistenza del regime stazionario è necessario $\rho = \lambda/\mu < 1$. Sfruttando quindi la (2.22) e la (2.23):

$$\begin{cases} P_k = P_0 \left(\frac{\lambda}{\mu}\right)^k & k \geq 1 \\ P_0 = \left[1 + \sum_{k=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^k\right]^{-1} = \left[\sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k\right]^{-1} = 1 - \frac{\lambda}{\mu} & k = 0 \end{cases}$$

Considerando che il tasso degli arrivi è costante e pari a λ e che dalla (2.18) si ha $E[x] = 1/\mu$, il fattore di utilizzazione è dato da:

$$\rho = \lambda E[x] = \frac{\lambda}{\mu}$$

per cui:

$$P_k = (1 - \rho)\rho^k \quad (2.28)$$

Si osservi che in conseguenza del Teorema 5 sui sistemi $G/G/1$ la (2.28) può essere espressa come $P_k = P_0(1 - P_0)^k$.

Si procede valutando i parametri caratteristici definiti nella sezione precedente. Applicando la funzione generatrice introdotta all'inizio del capitolo si ha:

$$G(z) = \sum_k P_k z^k = (1 - \rho) \sum_{k=0}^{\infty} \rho^k z^k = \frac{1 - \rho}{1 - \rho z} \quad (2.29)$$

e quindi risulta agevole valutare il numero medio degli utenti nel sistema e la relativa varianza:

$$\begin{aligned} E[k] &= \left. \frac{dG(z)}{dz} \right|_{z=1} = \left. \frac{1 - \rho}{(1 - \rho z)^2} \rho \right|_{z=1} = \frac{\rho}{1 - \rho} \\ E[k^2] &= \left. \frac{d^2 G(z)}{dz^2} \right|_{z=1} + \left. \frac{dG(z)}{dz} \right|_{z=1} = - \left. \frac{(1 - \rho)\rho}{(1 - \rho z)^4} 2(1 - \rho z)(-\rho) \right|_{z=1} + \frac{\rho}{1 - \rho} = \frac{\rho^2 + \rho}{(1 - \rho)^2} \\ \sigma_k^2 &= E[k^2] - E^2[k] = \frac{\rho^2 + \rho}{(1 - \rho)^2} - \frac{\rho^2}{(1 - \rho)^2} = \frac{\rho}{(1 - \rho)^2} \end{aligned} \quad (2.30)$$

Il numero medio di utenti in servizio è $P[k > 0] = \rho$; di conseguenza il numero medio di utenti in coda è

$$E[q] = E[k] - \rho = \frac{\rho^2}{1 - \rho}.$$

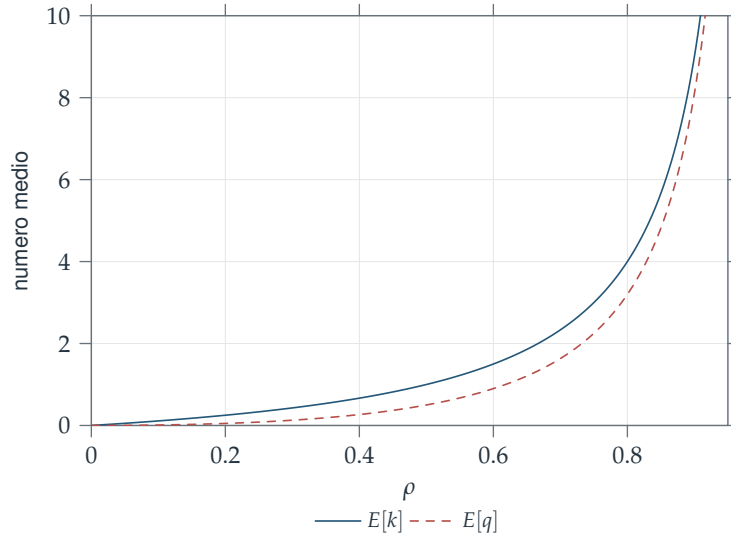


Figura 2.8. Numero medio di utenti nel sistema e in coda.

Sfruttando il risultato di Little si ottiene il tempo medio trascorso nel sistema:

$$E[T] = \frac{E[k]}{\lambda} = \frac{\rho}{1 - \rho} \frac{1}{\lambda} = \frac{1}{\mu(1 - \rho)}.$$

Il tempo medio trascorso in coda è il tempo complessivo meno il tempo medio di servizio:

$$E[T_q] = E[T] - \frac{1}{\mu} = \frac{\rho}{\mu(1 - \rho)}. \quad (2.31)$$

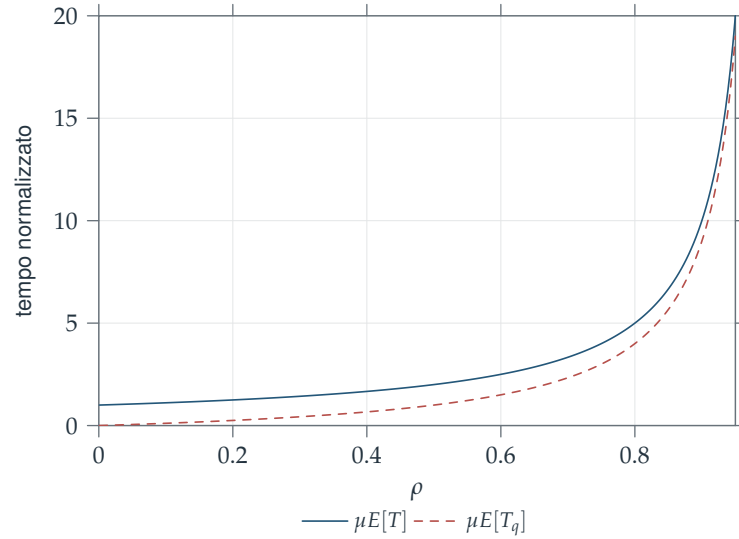


Figura 2.9. Tempo medio trascorso da un utente nel sistema e in coda.

La stessa quantità si verifica anche applicando Little alla sola coda: $E[q] = \lambda E[T_q]$.

Si osserva che quando il sistema ha un carico esiguo, $\rho \rightarrow 0$, si ha $E[T] \rightarrow 1/\mu$, $E[T_q] \rightarrow 0$, $E[k] \rightarrow 0$ ed $E[q] \rightarrow 0$: nel limite il tempo trascorso nel sistema coincide con il solo tempo di servizio.

Nelle figure 2.8 e 2.9 vengono riportati i parametri caratteristici in funzione del fattore di utilizzazione.

Infine un ulteriore parametro di interesse, che può essere agevolmente valutato per questo sistema, è la probabilità che vi sia un numero di utenti maggiore o uguale a N nel sistema:

$$\begin{aligned} P[k \geq N] &= \sum_{k=N}^{\infty} P_k = (1 - \rho) \sum_{k=N}^{\infty} \rho^k \\ &= (1 - \rho) \left[\sum_{k=0}^{\infty} \rho^k - \sum_{k=0}^{N-1} \rho^k \right] \\ &= (1 - \rho) \left[\frac{1}{1 - \rho} - \frac{1 - \rho^N}{1 - \rho} \right] = \rho^N. \end{aligned}$$

Quest'ultima equazione può essere riscritta come $P[k \geq N] = (1 - P_0)^N$.

2.6.6 Sistema $M/M/1$ con arrivi rallentati

In questo sistema gli arrivi tendono ad essere ritardati tanto più il sistema risulta affollato. In particolare:

$$\begin{cases} \lambda_k = \frac{\alpha}{k+1} \\ \mu_k = \mu \end{cases}$$

Sfruttando la (2.22) e la (2.23) si ha:

$$\begin{cases} P_k = P_0 \prod_{i=0}^{k-1} \frac{\alpha}{i+1} \frac{1}{\mu} = P_0 \left(\frac{\alpha}{\mu} \right)^k \frac{1}{k!} & k \geq 1 \\ P_0 = \left[1 + \sum_{k=1}^{\infty} \left(\frac{\alpha}{\mu} \right)^k \frac{1}{k!} \right]^{-1} = \left[\sum_{k=0}^{\infty} \left(\frac{\alpha}{\mu} \right)^k \frac{1}{k!} \right]^{-1} = e^{-\frac{\alpha}{\mu}} & k = 0 \end{cases}$$

quindi:

$$P_k = \frac{(\alpha/\mu)^k}{k!} e^{-\frac{\alpha}{\mu}}$$

che risulta essere una distribuzione di Poisson.

Dalle considerazioni generali fatte sul fattore di utilizzazione per sistemi $G/G/1$, tramite la (2.26), si ha:

$$\bar{\lambda}E[x] = \rho = 1 - P_0 = 1 - e^{-\frac{\alpha}{\mu}} \Rightarrow \bar{\lambda} = \frac{\rho}{E[x]} = \rho\mu = \mu(1 - e^{-\frac{\alpha}{\mu}})$$

Un modo alternativo per valutare il tasso medio effettivo $\bar{\lambda}$ è:

$$\begin{aligned} \bar{\lambda} &= \sum_{k=0}^{\infty} \lambda_k P_k = \sum_{k=0}^{\infty} \frac{\alpha}{k+1} \frac{(\alpha/\mu)^k}{k!} e^{-\alpha/\mu} = \alpha e^{-\alpha/\mu} \sum_{k=0}^{\infty} \frac{(\alpha/\mu)^{k+1}}{(k+1)!} \frac{1}{\alpha/\mu} = \\ &= \mu e^{-\alpha/\mu} \sum_{j=1}^{\infty} \frac{(\alpha/\mu)^j}{j!} = \mu e^{-\alpha/\mu} \left[\sum_{j=0}^{\infty} \frac{(\alpha/\mu)^j}{j!} - 1 \right] = \mu e^{-\alpha/\mu} [e^{\alpha/\mu} - 1] = \mu [1 - e^{-\frac{\alpha}{\mu}}] \end{aligned}$$

Per quanto riguarda la valutazione dei parametri caratteristici del sistema, considerando che P_k è una distribuzione di Poisson e tenendo conto del risultato di Little, si ha:

$$E[k] = \frac{\alpha}{\mu}$$

$$E[T] = \frac{E[k]}{\bar{\lambda}} = \frac{\alpha}{\rho\mu^2} = \frac{\alpha}{\mu^2(1 - e^{-\frac{\alpha}{\mu}})}$$

Per la coda invece si ha:

$$E[T_q] = E[T] - \frac{1}{\mu} = \frac{\alpha - \rho\mu}{\rho\mu^2}$$

$$E[q] = E[T_q]\bar{\lambda} = \frac{\alpha - \rho\mu}{\mu}$$

2.6.7 Sistema $M/M/\infty$

In questo sistema ogni nuovo cliente trova immediatamente un servitore libero. Il numero di servitori è idealmente illimitato e ciascun servitore completa il servizio con tasso μ ; quando sono presenti k clienti, il tasso complessivo di uscita è quindi $k\mu$. Si ha:

$$\begin{cases} \lambda_k = \lambda \\ \mu_k = k\mu \end{cases}$$

Sfruttando l'equazione (2.22) si ha:

$$P_k = P_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} = \frac{(\lambda/\mu)^k}{k!} e^{-\frac{\lambda}{\mu}}$$

che risulta analoga alla formula relativa al sistema con arrivi ritardati, sostituendo α con λ . Il numero di utenti mediamente presenti nel sistema è:

$$E[k] = \frac{\lambda}{\mu}$$

come per il sistema con arrivi ritardati, mentre il tempo speso nel sistema è sempre uguale al tempo di servizio in quanto nessun cliente entra in coda ($E[q] = 0$, $E[T_q] = 0$) dato che esiste sempre un servitore disponibile:

$$E[T] = \frac{1}{\mu}$$

Per questo sistema non esiste un singolo servitore la cui saturazione possa essere descritta dal consueto fattore di utilizzazione: il numero di risorse cresce idealmente con il numero di clienti presenti.

2.6.8 Sistema $M/M/c$ e formula di Erlang C

In questo sistema il numero di servitori è c , per cui il tasso di servizio può crescere fino a tale limite:

$$\begin{cases} \lambda_k = \lambda \\ \mu_k = \min[k\mu, c\mu] \end{cases}$$

Utilizzando la (2.22) si ha:

$$P_k = \begin{cases} P_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} = P_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} & k \leq c \\ P_0 \prod_{i=0}^{c-1} \frac{\lambda}{(i+1)\mu} \prod_{j=c}^{k-1} \frac{\lambda}{c\mu} = P_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{c!c^{k-c}} & k > c \end{cases}$$

Per quanto riguarda il fattore di utilizzazione si ha che $\lambda E[x] = \lambda/\mu$ ed il massimo tasso con cui il lavoro può essere smaltito è c , quindi:

$$\rho = \frac{\lambda}{c\mu}$$

da cui:

$$P_k = \begin{cases} P_0 \frac{(c\rho)^k}{k!} & k \leq c \\ P_0 \frac{(c\rho)^k}{c!c^{k-c}} = P_0 \frac{\rho^k c^c}{c!} & k > c \end{cases}$$

La condizione di stabilità è $\rho < 1$. Per quanto riguarda la condizione di normalizzazione (2.23) si ha:

$$\begin{aligned} P_0 &= \left[1 + \sum_{k=1}^{c-1} \frac{(c\rho)^k}{k!} + \sum_{k=c}^{\infty} \frac{\rho^k c^c}{c!} \right]^{-1} = \left[\sum_{k=0}^{c-1} \frac{(c\rho)^k}{k!} + \frac{c^c}{c!} \rho^c \sum_{k=c}^{\infty} \rho^{k-c} \right]^{-1} = \\ &= \left[\sum_{k=0}^{c-1} \frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} \right]^{-1} \end{aligned}$$

Un arrivo deve attendere se trova almeno c clienti. L'uguaglianza fra la distribuzione vista dagli arrivi e quella temporale usa la proprietà PASTA, dimostrata più avanti nella derivazione di Erlang B: vale perché gli arrivi esterni sono poissoniani e non anticipano lo stato del sistema. Con $A = c\rho$ segue:

$$\begin{aligned}
P[\text{coda}] &= \sum_{k=c}^{\infty} P_k = \sum_{k=c}^{\infty} P_0 \frac{c^c}{c!} \rho^k = P_0 \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} = \frac{\frac{(c\rho)^c}{c!} \frac{1}{1-\rho}}{\sum_{k=0}^{c-1} \frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!} \frac{1}{1-\rho}} = \\
&= \frac{\frac{A^c}{c!} \frac{1}{1-\frac{A}{c}}}{\sum_{k=0}^{c-1} \frac{A^k}{k!} + \frac{A^c}{c!} \frac{1}{1-\frac{A}{c}}} = C(c, A) = C(c, c\rho)
\end{aligned}$$

che viene detta formula di Erlang C.

Per quanto riguarda il numero medio di utenti in coda si procede scrivendo:

$$E[q] = \sum_{k=c}^{\infty} (k-c) P_k = \sum_{v=0}^{\infty} v P_{v+c} = \sum_{v=0}^{\infty} v P_0 \rho^{v+c} \frac{c^c}{c!} = P_0 \frac{c^c}{c!} \rho^c \sum_{v=0}^{\infty} v \rho^v$$

Il termine $\sum_{v=0}^{\infty} v \rho^v$ non è un valore medio, perché i pesi ρ^v non costituiscono una distribuzione normalizzata. Lo si valuta semplicemente derivando la serie geometrica:

$$G(z) = \sum_{k=0}^{\infty} \rho^k z^k = \frac{1}{1-\rho z} \Rightarrow \sum_{v=0}^{\infty} v \rho^v = \left. \frac{dG(z)}{dz} \right|_{z=1} = \frac{\rho}{(1-\rho)^2} \quad (2.32)$$

quindi:

$$E[q] = \left[P_0 \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} \right] \frac{\rho}{1-\rho} = C(c, c\rho) \frac{\rho}{1-\rho}$$

Il numero medio di utenti presenti nel sistema è dato dalla somma tra il numero medio di utenti in coda ed il numero medio di utenti in servizio $E[s]$, cioè:

$$E[k] = E[q] + E[s]$$

Per quanto riguarda la valutazione del numero medio di utenti in servizio:

$$\begin{aligned}
E[s] &= \sum_{k=0}^c k P_k + c \sum_{k=c+1}^{\infty} P_k = \sum_{k=1}^c k P_0 \frac{(c\rho)^k}{k!} + c \sum_{k=c+1}^{\infty} P_0 \frac{(c\rho)^k}{c! c^{k-c}} = \\
&= P_0 \sum_{k=1}^c \frac{(c\rho)^k}{(k-1)!} + P_0 \frac{c(c\rho)^c}{c!} \sum_{j=1}^{\infty} \frac{(c\rho)^j}{c^j} = P_0 \left[c\rho \sum_{j=0}^{c-1} \frac{(c\rho)^j}{j!} + \frac{c(c\rho)^c}{c!} \frac{\rho}{1-\rho} \right] = \\
&= P_0 c\rho \left[\sum_{j=0}^{c-1} \frac{(c\rho)^j}{j!} + \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} \right] = P_0 c\rho P_0^{-1} = c\rho
\end{aligned}$$

Il tempo medio trascorso in coda e nel sistema si desume dalle equazioni precedenti, tramite la formula di Little:

$$\begin{aligned}
E[T_q] &= \frac{E[q]}{\lambda} = \frac{\rho}{\lambda(1-\rho)} C(c, c\rho) = \frac{C(c, c\rho)}{c\mu(1-\rho)} = \frac{C(c, c\rho)}{\hat{\mu}(1-\rho)} \\
E[T] &= \frac{E[k]}{\lambda} = \frac{E[q]}{\lambda} + \frac{c\rho}{\lambda} = \frac{C(c, c\rho) + c(1-\rho)}{\hat{\mu}(1-\rho)}
\end{aligned}$$

dove, con lo scopo di confrontare direttamente i vari sistemi, si fa riferimento al tasso di servizio dell' $M/M/1$ come tasso complessivo dei c servitori, qui indicato con $\hat{\mu}$, per cui:

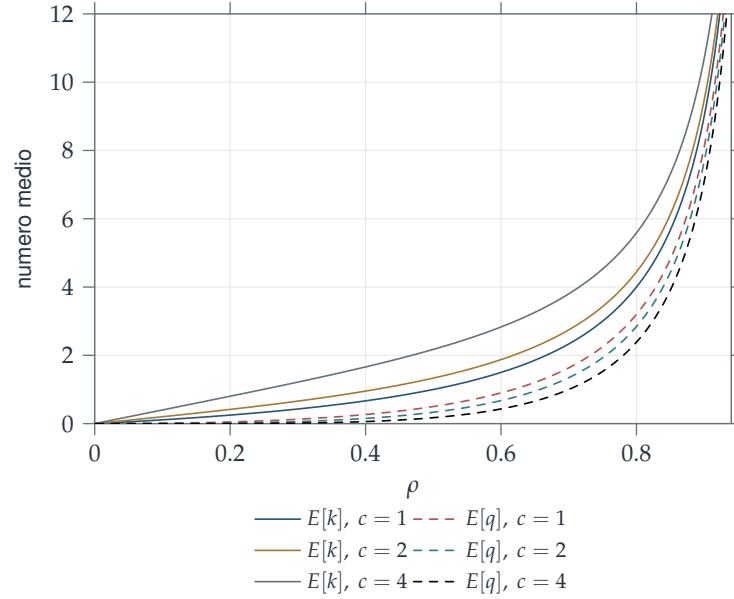


Figura 2.10. Numero medio di utenti nel sistema ed in coda al variare del numero dei servitori.

$\mu = \hat{\mu}/c$. Ponendo $c = 1$ si verifica che le prestazioni del sistema $M/M/c$ degenerano a quelle del $M/M/1$ con $c = 1$.

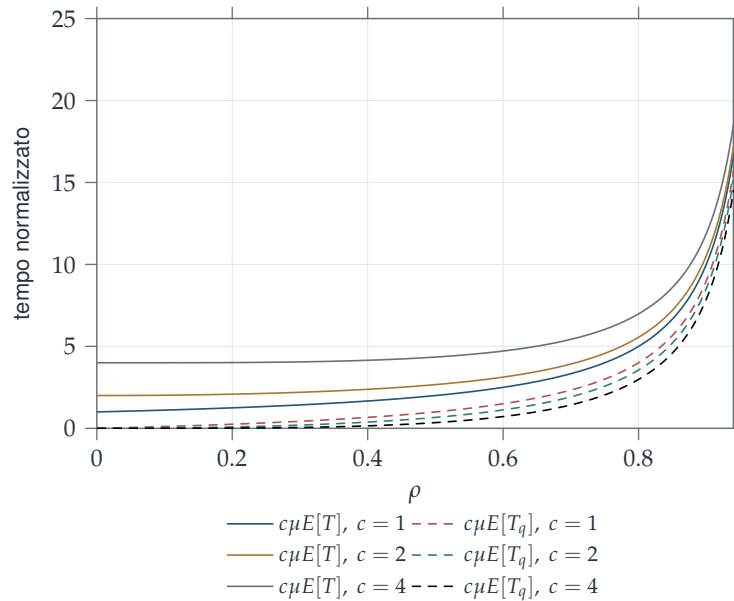


Figura 2.11. Tempo medio trascorso da un utente nel sistema ed in coda al variare del numero dei servitori.

Nelle figure 2.10 e 2.11 sono riportati i parametri caratteristici in funzione del fattore di utilizzazione. Le figure confrontano sistemi con la stessa capacità complessiva di servizio $\hat{\mu} = c\mu$. In questa particolare normalizzazione, aumentare c significa suddividere la stessa capacità totale tra un numero maggiore di servitori, ciascuno più lento. La probabilità di dover attendere e la componente di coda possono diminuire, mentre il tempo medio di servizio del singolo cliente cresce con c . Il numero medio complessivo di clienti può quindi aumentare anche se la coda si riduce.

Non esiste, in generale, una regola secondo cui siano sempre preferibili pochi servitori veloci o molti servitori lenti. La scelta dipende dal vincolo che si mantiene costante e dalla prestazione di interesse: ritardo totale, attesa in coda, parallelismo disponibile, probabilità di blocco, affidabilità e costo delle risorse. Il confronto delle figure 2.10 e 2.11 vale quindi per il modello e per la normalizzazione qui dichiarati.

2.6.9 Sistema $M/M/1/Y$

Rispetto al sistema $M/M/1$ si impongono delle limitazioni alla capacità del buffer. Supponendo che la massima dimensione della coda possa essere di $Y - 1$ celle e che quindi siano possibili al più $Y + 1$ stati (dallo stato 0 allo stato Y), risulta:

$$\begin{cases} \lambda_k = \begin{cases} \lambda & k < Y \\ 0 & k \geq Y \end{cases} \\ \mu_k = \mu \end{cases}$$

Applicando la (2.22) si ha:

$$P_k = \begin{cases} P_0 \prod_{i=0}^{k-1} \frac{\lambda}{\mu} = P_0 \left(\frac{\lambda}{\mu} \right)^k & k \leq Y \\ 0 & k > Y \end{cases}$$

Mentre per la condizione di normalizzazione (2.23):

$$P_0 = \left[1 + \sum_{k=1}^Y \left(\frac{\lambda}{\mu} \right)^k \right]^{-1}.$$

Ponendo $A = \lambda/\mu$, per $A \neq 1$ la somma geometrica fornisce

$$P_0 = \frac{1-A}{1-A^{Y+1}}, \quad P_k = \frac{1-A}{1-A^{Y+1}} A^k, \quad 0 \leq k \leq Y.$$

Per $A = 1$ la distribuzione stazionaria è invece uniforme, $P_k = 1/(Y+1)$. Poiché lo spazio degli stati è finito, il sistema ammette una distribuzione stazionaria per ogni $A \geq 0$; le formule che presentano apparentemente una singolarità per $A = 1$ vanno quindi interpretate per continuità.

La probabilità che il sistema sia pieno e che quindi un nuovo arrivo venga rifiutato coincide con P_Y , cioè con la probabilità di trovare Y clienti complessivamente presenti: uno eventualmente in servizio e fino a $Y - 1$ in coda.

Per la valutazione del numero medio degli utenti nel sistema si può partire direttamente dalla somma geometrica finita

$$F(A) = \sum_{i=0}^Y A^i = \frac{1-A^{Y+1}}{1-A}, \quad A \neq 1.$$

Poiché $AF'(A) = \sum_{i=0}^Y iA^i$, dalla definizione di valore medio segue

$$E[k] = P_0 AF'(A) = \frac{A}{1-A} - \frac{(Y+1)A^{Y+1}}{1-A^{Y+1}}, \quad A \neq 1.$$

Nel caso $A = 1$, usando la distribuzione uniforme, si ottiene

$$E[k] = \frac{1}{Y+1} \sum_{i=0}^Y i = \frac{Y}{2},$$

che coincide con il limite della formula precedente per $A \rightarrow 1$. In questo stesso caso, per $Y \geq 1$,

$$E[q] = \frac{Y(Y-1)}{2(Y+1)}, \quad \bar{\lambda} = \lambda \frac{Y}{Y+1} = \mu \frac{Y}{Y+1},$$

per cui, dalla formula di Little,

$$E[T] = \frac{Y+1}{2\mu}, \quad E[T_q] = \frac{Y-1}{2\mu}.$$

Per quanto riguarda il numero di utenti in coda si potrebbe ripercorrere la metodologia del caso $M/M/1$ ma si preferisce introdurre, anche a titolo esemplificativo, un approccio alternativo. Il numero medio di utenti in servizio risulta:

$$E[s] = \sum_{i=0}^1 iP_i + \sum_{i=2}^Y P_i = \sum_{i=1}^Y P_i = 1 - P_0$$

quindi:

$$\begin{aligned} E[q] &= E[k] - (1 - P_0) = \frac{A}{1-A} - \frac{(Y+1)A^{Y+1}}{1-A^{Y+1}} - 1 + \frac{1-A}{1-A^{Y+1}} = \\ &= \frac{A}{1-A} - A + A - \frac{A + YA^{Y+1}}{1-A^{Y+1}} = \frac{A^2}{1-A} - \frac{(Y+A)A^{Y+1}}{1-A^{Y+1}} \end{aligned}$$

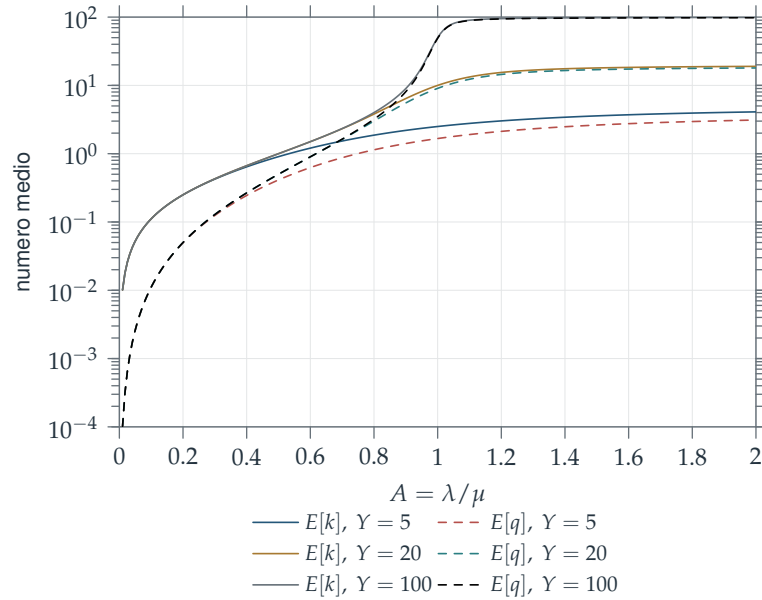


Figura 2.12. Numero medio di utenti nel sistema ed in coda.

Per le valutazioni dei tempi $E[T]$ ed $E[T_q]$ si utilizza la formula di Little notando che il tasso medio degli arrivi è dato da:

$$\bar{\lambda} = \sum_i \lambda_i P_i = \lambda \sum_{i=0}^{Y-1} P_i = \lambda(1 - P_Y) \quad (2.33)$$

Infine, per $Y \rightarrow \infty$ e $A < 1$, la distribuzione stazionaria e i parametri convergono a quelli del sistema $M/M/1$. Per $A \geq 1$ il sistema a capacità infinita non possiede invece una distribuzione stazionaria normalizzabile.

Nelle figure 2.12 e 2.13 vengono riportati i parametri caratteristici in funzione di A per diverse lunghezze della coda Y .

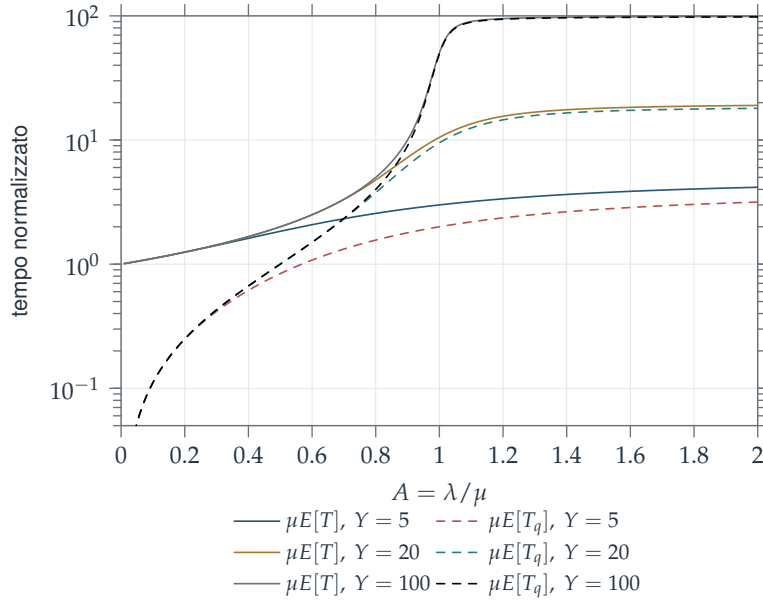


Figura 2.13. Tempo medio trascorso da un utente nel sistema ed in coda.

2.6.10 Sistemi $M/M/Y/Y$ e formula di Erlang B

In questo sistema, rispetto al caso $M/M/1/Y$, si aumenta a Y il numero dei servitori. Ogni cliente che prova ad entrare nel sistema, quando ve ne sono già Y presenti, viene rifiutato. I tassi sono esprimibili analiticamente come segue:

$$\begin{cases} \lambda_k = \begin{cases} \lambda & k < Y \\ 0 & k \geq Y \end{cases} \\ \mu_k = k\mu \end{cases}$$

L'equazione (2.22) diventa:

$$P_k = \begin{cases} P_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} = P_0 \left(\frac{\lambda}{\mu} \right)^k \frac{1}{k!} & k \leq Y \\ 0 & k > Y \end{cases}$$

Per quanto riguarda la condizione di normalizzazione (2.23) si ha:

$$P_0 = \left[1 + \sum_{k=1}^Y \left(\frac{\lambda}{\mu} \right)^k \frac{1}{k!} \right]^{-1} = \left[\sum_{k=0}^Y \frac{A^k}{k!} \right]^{-1}$$

e quindi risolvendo per $k \leq Y$, si ottiene:

$$P_k = \frac{\frac{A^k}{k!}}{\sum_{i=0}^Y \frac{A^i}{i!}}$$

Un parametro che risulta particolarmente interessante è la probabilità che tutti i servitori siano occupati (o la frazione di tempo in cui lo sono):

$$P_Y = \frac{\frac{A^Y}{Y!}}{\sum_{i=0}^Y \frac{A^i}{i!}} = B(Y, A) \quad (2.34)$$

questa espressione è la formula di Erlang B. Per interpretarla come probabilità di blocco vista dagli arrivi serve la proprietà PASTA (*Poisson Arrivals See Time Averages*), che qui si può derivare direttamente. Sia $I_k(t)$ l'indicatore dell'evento "il sistema è nello stato k " appena prima dell'istante t . In un intervallo infinitesimo dt , un arrivo poissoniano si verifica con probabilità $\lambda dt + o(dt)$, indipendentemente dallo stato corrente; pertanto

$$P[I_k(t) = 1 \text{ e arrivo in } (t, t + dt)] = P_k \lambda dt + o(dt).$$

Dividendo per la probabilità di un arrivo, $\lambda dt + o(dt)$, e passando al limite $dt \rightarrow 0$, la probabilità che un arrivo osservi lo stato k è P_k . Gli arrivi vedono quindi la stessa distribuzione delle osservazioni temporali stazionarie. Nel sistema $M/M/Y/Y$ un arrivo è bloccato se e solo se osserva lo stato Y , per cui la probabilità di blocco è proprio $P_Y = B(Y, A)$. La formula è quindi centrale nel dimensionamento dei sistemi a commutazione di circuito, trattati nel capitolo 7.

Il numero medio di utenti nel sistema è dato da:

$$E[k] = \frac{\sum_{i=0}^Y i \frac{A^i}{i!}}{\sum_{i=0}^Y \frac{A^i}{i!}} = A \frac{\sum_{i=0}^{Y-1} \frac{A^i}{i!}}{\sum_{i=0}^Y \frac{A^i}{i!}} = A [1 - B(Y, A)]$$

mentre nessun utente può essere in coda in quanto il numero dei servitori disponibili eguaglia quello dei clienti che possono essere presenti nel sistema. Come nel caso del sistema $M/M/1/Y$ il tasso medio degli arrivi è dato dalla (2.33) e quindi:

$$\bar{\lambda} = \lambda [1 - B(Y, A)]$$

da cui si desume che il tempo medio di permanenza nel sistema è: $E[T] = 1/\mu$.

Nella figura 2.14 viene riportato il numero medio di utenti presenti nel sistema al variare di A .

2.6.11 Sistemi $M/M/1//N$

In questo sistema il numero dei clienti potenziali è limitato ad N mentre in tutti i casi precedenti era infinito. Il tasso di nascita, quindi, deve diminuire con l'aumentare degli utenti presenti nel sistema:

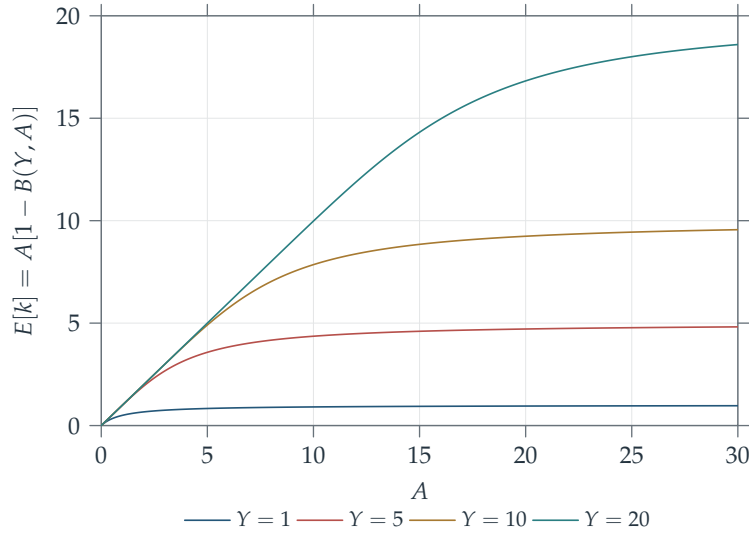


Figura 2.14. Numero di utenti nel sistema $M/M/Y/Y$.

$$\begin{cases} \lambda_k = \begin{cases} \lambda(N-k) & k < N \\ 0 & k \geq N \end{cases} \\ \mu_k = \mu \end{cases}$$

Sfruttando la (2.22) si ottiene:

$$P_k = \begin{cases} P_0 \prod_{i=0}^{k-1} \frac{\lambda(N-i)}{\mu} = P_0 \left(\frac{\lambda}{\mu}\right)^k \frac{N!}{(N-k)!} & k \leq N \\ 0 & k > N \end{cases}$$

e, per quanto riguarda la condizione di normalizzazione, dalla (2.23) segue che:

$$P_0 = \left[\sum_{k=0}^N \left(\frac{\lambda}{\mu}\right)^k \frac{N!}{(N-k)!} \right]^{-1}$$

Risolvendo per $k \leq N$ si ottiene:

$$P_k = \frac{A^k \frac{N!}{(N-k)!}}{\sum_{i=0}^N A^i \frac{N!}{(N-i)!}}.$$

Per ricavare in modo compatto i parametri caratteristici introduciamo la costante di normalizzazione

$$Z_N(A) = \sum_{k=0}^N A^k \frac{N!}{(N-k)!}, \quad P_0 = \frac{1}{Z_N(A)}.$$

Derivando rispetto ad A si ha

$$AZ'_N(A) = \sum_{k=0}^N k A^k \frac{N!}{(N-k)!},$$

e quindi, direttamente dalla definizione di valore medio,

$$E[k] = \frac{AZ'_N(A)}{Z_N(A)}. \quad (2.35)$$

Poiché vi è un solo servitore, il numero medio di clienti in servizio coincide con la probabilità che il sistema non sia vuoto:

$$E[s] = 1 - P_0.$$

Ne segue

$$E[q] = E[k] - (1 - P_0). \quad (2.36)$$

Il tasso di arrivo dipende dallo stato, perciò il tasso medio degli arrivi effettivi non è $N\lambda$. Mediando λ_k sulla distribuzione stazionaria si ottiene

$$\bar{\lambda} = \sum_{k=0}^N \lambda(N-k)P_k = \lambda(N - E[k]). \quad (2.37)$$

In equilibrio lo stesso tasso deve coincidere con il tasso medio delle partenze; il servitore completa lavori con tasso μ quando è occupato, quindi

$$\bar{\lambda} = \mu(1 - P_0),$$

relazione che costituisce anche un controllo della soluzione stazionaria. Applicando infine il risultato di Little agli utenti effettivamente ammessi,

$$E[T] = \frac{E[k]}{\bar{\lambda}}, \quad E[T_q] = \frac{E[q]}{\bar{\lambda}}. \quad (2.38)$$

In questo modo tutti i parametri principali sono determinati dalla sola normalizzazione $Z_N(A)$ e dalla sua derivata.

2.6.12 Sistemi $M/M/\infty/N$

Anche in questo sistema, come già nel precedente, il numero dei clienti potenziali è limitato ad N ma si è in presenza di un numero infinito di servitori che influisce sull'equazione del tasso di morte:

$$\begin{cases} \lambda_k = \begin{cases} \lambda(N-k) & k < N \\ 0 & k \geq N \end{cases} \\ \mu_k = k\mu \end{cases}$$

Sfruttando la (2.22) si ottiene:

$$P_k = \begin{cases} P_0 \prod_{i=0}^{k-1} \frac{\lambda(N-i)}{\mu(i+1)} = P_0 \left(\frac{\lambda}{\mu}\right)^k \frac{N!}{k!(N-k)!} = P_0 \left(\frac{\lambda}{\mu}\right)^k \binom{N}{k} & k \leq N \\ 0 & k > N \end{cases}$$

e, per quanto riguarda la condizione di normalizzazione, dalla (2.23) segue che:

$$P_0 = \left[\sum_{k=0}^N \left(\frac{\lambda}{\mu}\right)^k \binom{N}{k} \right]^{-1} = \left[\left(1 + \frac{\lambda}{\mu}\right)^N \right]^{-1} = (1 + A)^{-N}$$

Risolvendo per $k \leq N$ si ottiene:

$$P_k = \frac{A^k \binom{N}{k}}{(1+A)^N}.$$

Ponendo

$$p = \frac{A}{1+A}, \quad 1-p = \frac{1}{1+A},$$

la distribuzione si riscrive come

$$P_k = \binom{N}{k} p^k (1-p)^{N-k},$$

cioè come una distribuzione binomiale. Di conseguenza

$$E[k] = Np = \frac{NA}{1+A}. \quad (2.39)$$

Non esiste attesa, poiché ogni cliente dispone immediatamente di un servitore, e quindi

$$E[q] = 0, \quad E[T_q] = 0.$$

Il tasso medio degli arrivi effettivi è

$$\bar{\lambda} = \lambda(N - E[k]) = \frac{N\lambda}{1+A}. \quad (2.40)$$

D'altra parte il tasso medio delle partenze è $\mu E[k]$; usando $A = \lambda/\mu$ si verifica immediatamente che $\mu E[k] = \bar{\lambda}$. Il risultato di Little fornisce infine

$$E[T] = \frac{E[k]}{\bar{\lambda}} = \frac{1}{\mu},$$

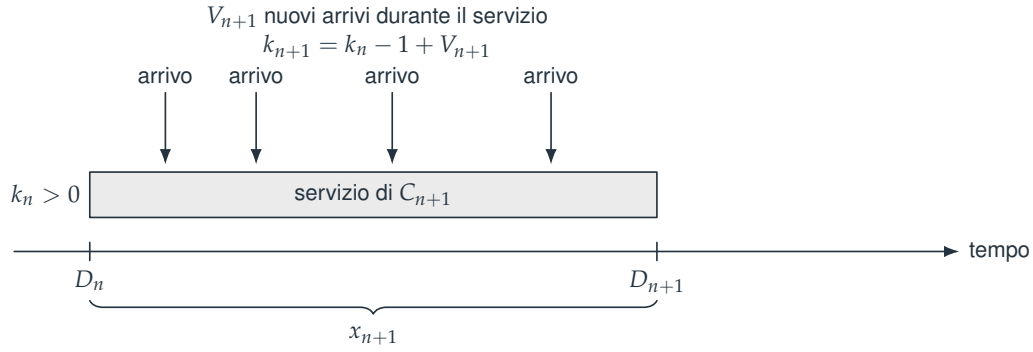
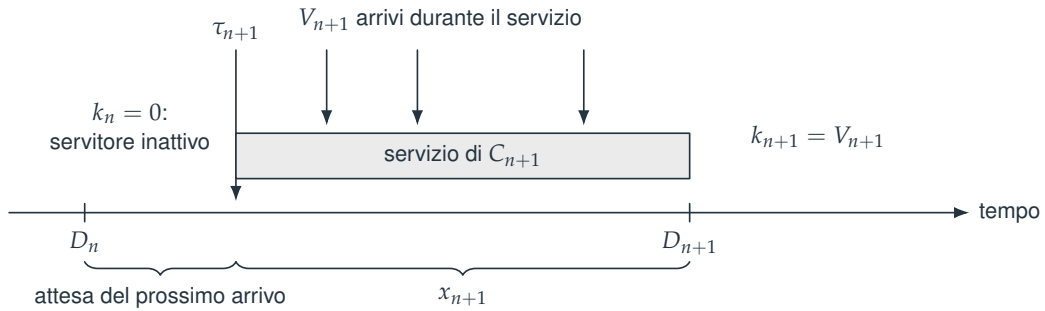
come deve essere: in assenza di coda, il tempo di permanenza coincide con il tempo medio di servizio.

2.7 Sistemi $M/G/1$

Consideriamo arrivi di Poisson di tasso $\lambda > 0$, servizi indipendenti e identicamente distribuiti, indipendenti dagli arrivi, con $0 < E[x] < \infty$, un solo servitore senza interruzioni o vacanze, buffer infinito e disciplina FIFO. Il regime stazionario studiato richiede $\rho = \lambda E[x] < 1$. Nei sistemi $M/G/1$ il tempo di servizio ha legge generale. In generale la conoscenza del numero degli utenti ad un determinato istante, $N(t)$, non costituisce una descrizione completa dello stato in quanto il processo di servizio può avere memoria; occorre quindi specificare la quantità di servizio, $X_0(t)$, già ricevuta dall'utente. Il processo $N(t)$ non è di Markov mentre lo è il processo descritto congiuntamente da $N(t)$ e $X_0(t)$.

Data la familiarità con i processi dipendenti da un solo parametro, si cerca di ricondurre il problema ad una veste monodimensionale; in tal senso si procede fissando gli istanti di osservazione in modo che dalla conoscenza del numero degli utenti in questi istanti si possa calcolare il numero degli utenti all'istante successivo. Se questi istanti coincidono con gli istanti di partenza del servizio non occorre specificare $X_0(t)$ in quanto risulta nullo per definizione. Si costruisce così una catena di Markov incorporata, osservata agli istanti di partenza, nei quali il servizio appena concluso non lascia alcun residuo da descrivere.

Sono utili le seguenti definizioni:

Figura 2.15. Sistema M/G/1 con $k_n > 0$.Figura 2.16. Sistema M/G/1 con $k_n = 0$.

- C_n n -esimo cliente che entra nel sistema
- τ_n istante di arrivo di C_n
- $t_n = \tau_n - \tau_{n-1}$ tempo interarrivo
- x_n tempo di servizio di C_n
- k_n numero di utenti dopo che C_n è uscito dal sistema
- V_n numero di arrivi durante il servizio di C_n

È interessante valutare la relazione che intercorre tra k_{n+1} e k_n : dato che durante il servizio di C_{n+1} arrivano V_{n+1} utenti ed alla fine C_{n+1} esce dal sistema, si ha $k_{n+1} = k_n + V_{n+1} - 1$. Questo comportamento è descritto graficamente dalla figura 2.15. È inoltre importante considerare il caso in cui $k_n = 0$, come mostrato in figura 2.16, dove C_{n+1} passa direttamente in servizio ed alla sua uscita gli utenti che rimangono nel sistema sono tutti quelli che sono arrivati dall'entrata in servizio di C_{n+1} , quindi $k_{n+1} = V_{n+1}$. Complessivamente si ha:

$$k_{n+1} = \begin{cases} k_n - 1 + V_{n+1} & k_n > 0 \\ V_{n+1} & k_n = 0 \end{cases} \quad (2.41)$$

Per costruire la ricorrenza si adotta una disciplina FIFO (First In First Out), così che l'ordine dei clienti coincida con l'ordine di servizio. La formula media di Pollaczek–Khinchine vale anche per altre discipline non preemptive e work-conserving che selezionino il prossimo cliente senza usare le sue esigenze di servizio. Quest'ultima restrizione è essenziale: una disciplina che conosca e privilegi i lavori brevi può modificare il tempo medio di attesa. Nel seguito manteniamo FIFO, necessaria anche per identificare gli indici dei clienti con l'ordine

di partenza. L'equazione (2.41) può essere resa compatta attraverso la definizione di:

$$\Delta_k = \begin{cases} 1 & k \geq 1 \\ 0 & k = 0 \end{cases}$$

da cui si ottiene:

$$k_{n+1} = k_n - \Delta_{k_n} + V_{n+1} \quad (2.42)$$

Si assume ora che il sistema sia stabile, $\rho < 1$, e che la catena incorporata agli istanti di partenza ammetta un regime stazionario con i momenti necessari finiti. La derivazione per elevamento al quadrato richiede in particolare il secondo momento del numero di clienti: non va confuso con il secondo momento del servizio. Una condizione sufficiente qui è $E[x^3] < \infty$. Dopo la (2.54) verrà completata la derivazione della media con la sola ipotesi $E[x^2] < \infty$, senza sottrarre due momenti infiniti:

$$\lim_{n \rightarrow \infty} E[k_n^j] = E[\tilde{k}^j], \quad j = 1, 2. \quad (2.43)$$

Dalla (2.42) e (2.43) si ha:

$$\begin{aligned} E[k_{n+1}] &= E[k_n] - E[\Delta_{k_n}] + E[V_{n+1}] \Rightarrow E[\tilde{k}] = E[\tilde{k}] - E[\Delta_{\tilde{k}}] + E[\tilde{V}] \Rightarrow \\ E[\Delta_{\tilde{k}}] &= E[\tilde{V}]. \end{aligned} \quad (2.44)$$

Condizionando sulla durata x di un servizio, il numero degli arrivi durante il servizio è Poisson di media λx . Pertanto

$$E[\tilde{V}] = E[E[\tilde{V} | x]] = \lambda E[x] = \rho.$$

D'altra parte, per definizione di Δ_k ,

$$E[\Delta_{\tilde{k}}] = \sum_{j=0}^{\infty} \Delta_j P[\tilde{k} = j] = \sum_{j=1}^{\infty} P[\tilde{k} = j] = P[\tilde{k} > 0] = \rho \quad (2.45)$$

Quindi il primo risultato di interesse è esprimibile tramite il seguente teorema:

Teorema 6. *In un sistema $M/G/1$ stabile, con arrivi di Poisson di tasso λ e tempi di servizio i.i.d., indipendenti dal processo degli arrivi e con media finita $E[x]$, il numero medio di arrivi durante un periodo di servizio è uguale al fattore di utilizzazione $\rho = \lambda E[x]$, cioè:*

$$E[\tilde{V}] = \rho \quad (2.46)$$

2.7.1 Numero medio di utenti in $M/G/1$, formula di Pollaczek Khinchine

Volendo valutare il numero medio degli utenti si procede elevando al quadrato la (2.42):

$$k_{n+1}^2 = k_n^2 + \Delta_{k_n}^2 + V_{n+1}^2 - 2k_n \Delta_{k_n} + 2k_n V_{n+1} - 2\Delta_{k_n} V_{n+1}$$

e si osserva che: $\Delta_{k_n}^2 = \Delta_{k_n}$ e $k_n \Delta_{k_n} = k_n$, quindi:

$$k_{n+1}^2 = k_n^2 + \Delta_{k_n} + V_{n+1}^2 - 2k_n + 2k_n V_{n+1} - 2\Delta_{k_n} V_{n+1}$$

Considerando che k_n e V_{n+1} sono indipendenti, applicando l'operatore di media statistica, sfruttando la (2.45) e passando al limite, si ha:

$$E[\tilde{k}^2] = E[\tilde{k}^2] + \rho + E[\tilde{V}^2] - 2E[\tilde{k}] + 2E[\tilde{k}]E[\tilde{V}] - 2\rho E[\tilde{V}]$$

utilizzando la (2.46):

$$\rho + 2E[\tilde{k}](\rho - 1) + E[\tilde{V}^2] - 2\rho^2 = 0$$

e ri assemblando i vari termini si ottiene infine:

$$E[\tilde{k}] = \frac{E[\tilde{V}^2] - 2\rho^2 + \rho}{2(1 - \rho)} = \frac{E[\tilde{V}^2] + 2\rho(1 - \rho) - \rho}{2(1 - \rho)} = \rho + \frac{E[\tilde{V}^2] - E[\tilde{V}]}{2(1 - \rho)} \quad (2.47)$$

Per ottenere una formulazione direttamente utilizzabile si procede valutando la differenza delle due medie statistiche presenti nel numeratore della (2.47). Osservando che V_n dipende solo dalla durata di x_n e non da n e che il processo degli arrivi è di Poisson, si ha:

$$P[\tilde{V} = j] = \int_0^\infty P_j(x)b(x)dx = \int_0^\infty \frac{(\lambda x)^j}{j!} e^{-\lambda x} b(x) dx$$

dove $b(x)$ è la densità dei tempi di servizio e $P_j(x)$ è la distribuzione di Poisson dovuta al processo degli arrivi. Usando la funzione generatrice per variabili discrete:

$$\begin{aligned} V(z) = E[z^{\tilde{V}}] &= \sum_{j=0}^{\infty} P[\tilde{V} = j] z^j = \sum_{j=0}^{\infty} \left[\int_0^\infty \frac{(\lambda x)^j}{j!} e^{-\lambda x} b(x) dx \right] z^j = \\ &= \int_0^\infty e^{-\lambda x} \left[\sum_{j=0}^{\infty} \frac{(\lambda x z)^j}{j!} \right] b(x) dx = \int_0^\infty e^{-\lambda x} e^{\lambda x z} b(x) dx = \\ &= \int_0^\infty e^{-(\lambda - \lambda z)x} b(x) dx = \mathcal{B}(\lambda - \lambda z) \end{aligned} \quad (2.48)$$

dove $\mathcal{B}(s)$ è la trasformata di Laplace di $b(x)$; ricordando che:

$$\left. \frac{d^2 V(z)}{dz^2} \right|_{z=1} = E[\tilde{V}^2] - E[\tilde{V}]$$

si ha:

$$\begin{aligned} \frac{dV(z)}{dz} &= \lambda \int_0^\infty x e^{-\lambda x(1-z)} b(x) dx \\ \frac{d^2 V(z)}{dz^2} &= \lambda^2 \int_0^\infty x^2 e^{-\lambda x(1-z)} b(x) dx \Rightarrow \left. \frac{d^2 V(z)}{dz^2} \right|_{z=1} = \lambda^2 E[x^2] \end{aligned}$$

Sostituendo in (2.47) si ottiene anzitutto la media osservata immediatamente dopo le partenze:

$$E[\tilde{k}] = \rho + \frac{\lambda^2 E[x^2]}{2(1 - \rho)}.$$

Resta da mostrare che questa media coincide con la media temporale del numero di clienti. Indichiamo con π_k la probabilità stazionaria di avere k clienti in un istante arbitrario. Per la proprietà PASTA dimostrata nella sezione precedente, un arrivo poissoniano trova k clienti con probabilità π_k ; gli attraversamenti verso l'alto del livello compreso tra k e $k + 1$ hanno quindi tasso $\lambda \pi_k$. In regime stazionario, su un intervallo lungo, il numero di attraversamenti verso l'alto e verso il basso dello stesso livello differisce al più per un termine di bordo. Il tasso dei passaggi verso il basso da $k + 1$ a k è dunque anch'esso $\lambda \pi_k$.

In un sistema stabile il tasso medio delle partenze è uguale al tasso degli arrivi, cioè λ . Una partenza lascia esattamente k clienti proprio quando realizza un attraversamento verso il basso da $k + 1$ a k ; pertanto

$$P[\tilde{k} = k] = \frac{\lambda \pi_k}{\lambda} = \pi_k.$$

La distribuzione osservata immediatamente dopo una partenza coincide quindi con la distribuzione temporale stazionaria e, in particolare, $E[\tilde{k}] = E[k]$. Si ottiene così

$$E[k] = \rho + \frac{\lambda^2 E[x^2]}{2(1 - \rho)}. \quad (2.49)$$

Per $\rho < 1$ ed $E[x^2] < \infty$ questa è la formula di Pollaczek–Khinchine per il numero medio di clienti nel sistema. Essa mostra in modo esplicito che, a parità di media del servizio, la variabilità del tempo di servizio incide sulle prestazioni attraverso il secondo momento $E[x^2]$.

2.7.2 Altri parametri caratteristici per sistemi $M/G/1$

Il numero medio di utenti in coda si calcola agevolmente dalla equazione:

$$\begin{aligned} E[q] &= \sum_{j=1}^{\infty} (j-1) P[\tilde{k} = j] = \sum_{j=0}^{\infty} j P[\tilde{k} = j] - \sum_{j=1}^{\infty} P[\tilde{k} = j] = \\ &= E[k] - P[\tilde{k} > 0] = E[k] - \rho \end{aligned}$$

Per quanto riguarda il tempo medio speso nel sistema, sfruttando la formula di Little si ha:

$$E[k] = \lambda E[T] = \rho + \frac{\lambda^2 E[x^2]}{2(1 - \rho)} \Rightarrow E[T] = E[x] + \frac{\lambda E[x^2]}{2(1 - \rho)}$$

che risulta essere composto dal tempo medio trascorso nel servizio, $E[x]$, e dal tempo medio speso in coda:

$$E[T_q] = \frac{\lambda E[x^2]}{2(1 - \rho)} \quad (2.50)$$

2.7.3 Distribuzione dei clienti in $M/G/1$

È interessante valutare la distribuzione dei clienti nel generico sistema $M/G/1$, calcolando la sua z -trasformata:

$$Q_n(z) = \sum_{j=0}^{\infty} P[k_n = j] z^j = E[z^{k_n}] \Rightarrow Q(z) = \lim_{n \rightarrow \infty} Q_n(z) = \sum_{j=0}^{\infty} P[\tilde{k} = j] z^j = E[z^{\tilde{k}}]$$

Dalla (2.42) si ha:

$$z^{k_{n+1}} = z^{k_n - \Delta_{k_n} + V_{n+1}} \Rightarrow E[z^{k_{n+1}}] = E[z^{k_n - \Delta_{k_n}} z^{V_{n+1}}] = Q_{n+1}(z)$$

Considerando l'indipendenza di $k_n - \Delta_{k_n}$ da V_{n+1} , l'indipendenza degli arrivi V_n da n e ricordando che $V(z) = E[z^{\tilde{V}}]$ come nella (2.48), si ha:

$$Q_{n+1}(z) = E[z^{k_n - \Delta_{k_n}}] E[z^{V_{n+1}}] = E[z^{k_n - \Delta_{k_n}}] E[z^{\tilde{V}}] = E[z^{k_n - \Delta_{k_n}}] V(z) \quad (2.51)$$

La media statistica presente in (2.51) può essere calcolata come:

$$E[z^{k_n - \Delta_{k_n}}] = \sum_{j=0}^{\infty} P[k_n = j] z^{j - \Delta_j} = P[k_n = 0] + \sum_{j=1}^{\infty} P[k_n = j] z^{j-1} \quad (2.52)$$

dove:

$$\begin{aligned} \sum_{j=1}^{\infty} P[k_n = j] z^{j-1} &= z^{-1} \sum_{j=1}^{\infty} P[k_n = j] z^j = z^{-1} \left[\sum_{j=0}^{\infty} P[k_n = j] z^j - P[k_n = 0] \right] = \\ &= z^{-1} [Q_n(z) - P[k_n = 0]] \end{aligned} \quad (2.53)$$

Utilizzando i risultati (2.51), (2.52) e (2.53) si ha:

$$Q_{n+1}(z) = V(z) \left[P[k_n = 0] + z^{-1} (Q_n(z) - P[k_n = 0]) \right]$$

Per ricavare la soluzione in equilibrio si procede calcolando il limite per n che tende all'infinito:

$$Q(z) = V(z) \left[P[\tilde{k} = 0] + z^{-1} (Q(z) - P[\tilde{k} = 0]) \right]$$

attraverso il risultato (2.45), $P[\tilde{k} = 0] = 1 - \rho$, si ha:

$$Q(z) = V(z) \left[(1 - \rho) + z^{-1} Q(z) - z^{-1} (1 - \rho) \right] \Rightarrow$$

$$Q(z)(1 - z^{-1}V(z)) = V(z)(1 - \rho)(1 - z^{-1}) \Rightarrow$$

$$Q(z) = V(z) \frac{(1 - \rho)(1 - \frac{1}{z})}{1 - \frac{V(z)}{z}}$$

Sfruttando la (2.48) si ottiene il risultato finale:

$$Q(z) = \mathcal{B}(\lambda - \lambda z) \frac{(1 - \rho)(1 - \frac{1}{z})}{1 - \frac{\mathcal{B}(\lambda - \lambda z)}{z}} \quad (2.54)$$

nota come relazione di Pollaczek–Khinchine per la distribuzione dei clienti. La forma senza potenze negative è

$$Q(z) = \frac{(1 - \rho)(1 - z)V(z)}{V(z) - z}.$$

Essa vale per $0 \leq z < 1$; in $z = 1$ si prende il limite $Q(1) = 1$. La stessa normalizzazione, applicata all'equazione stazionaria prima di sostituire $P[\tilde{k} = 0]$, dà $P[\tilde{k} = 0] = 1 - V'(1^-) = 1 - \rho$ usando solo $E[x] < \infty$. La distribuzione non richiede quindi che la media della coda sia finita.

Per giustificare la formula della media con il solo secondo momento del servizio finito, poniamo $u = 1 - z \downarrow 0$. Dalla (2.48) e dalla (2.6),

$$V(1 - u) = 1 - \rho u + \frac{\lambda^2 E[x^2]}{2} u^2 + o(u^2).$$

Sostituendo nella forma precedente di Q e dividendo numeratore e denominatore per u ,

$$Q(1 - u) = 1 - \left[\rho + \frac{\lambda^2 E[x^2]}{2(1 - \rho)} \right] u + o(u).$$

Quindi $Q'(1^-)$ è proprio la (2.49), senza assumere $E[\tilde{k}^2] < \infty$. Se invece $E[x^2] = \infty$, il regime può restare stazionario con $\rho < 1$, ma la media del numero di clienti e quella del tempo di attesa sono infinite.

Si osserva infine che il numero medio dei clienti (2.49) può essere ricavato a partire dalla (2.54):

$$E[k] = \left. \frac{dQ(z)}{dz} \right|_{z=1} = \frac{d}{dz} \left[\mathcal{B}(\lambda - \lambda z) \frac{(1-\rho)(1-\frac{1}{z})}{1 - \frac{\mathcal{B}(\lambda - \lambda z)}{z}} \right] \bigg|_{z=1}$$

2.7.4 Due casi particolari

In questa sezione si verificano due casi particolari del sistema $M/G/1$, utili anche come controllo della formula generale. Il primo sistema in esame è $M/M/1$ dove il servizio ha densità esponenziale caratterizzata dal parametro μ e quindi dalla (2.18) si ha: $E[x] = 1/\mu$ e $E[x^2] = 2/\mu^2$. I parametri caratteristici del sistema $M/M/1$, in accordo con quanto derivato precedentemente, risultano:

$$E[k] = \rho + \frac{\lambda^2 \frac{2}{\mu^2}}{2(1-\rho)} = \rho + \frac{\rho^2}{1-\rho} = \frac{\rho}{1-\rho}$$

$$E[q] = \frac{\lambda^2 \frac{2}{\mu^2}}{2(1-\rho)} = \frac{\rho^2}{1-\rho}$$

$$E[T] = \frac{1}{\mu} + \frac{\lambda \frac{2}{\mu^2}}{2(1-\rho)} = \frac{1-\rho+\rho}{\mu(1-\rho)} = \frac{1/\mu}{1-\rho}$$

$$E[T_q] = \frac{\lambda \frac{2}{\mu^2}}{2(1-\rho)} = \frac{\rho}{\mu(1-\rho)}$$

Per quanto riguarda la distribuzione degli utenti dalla (2.17) si ha:

$$\mathcal{B}(s) = \frac{\mu}{s + \mu}$$

quindi:

$$Q(z) = \frac{\mu}{\lambda - \lambda z + \mu} \frac{(1-\rho)(1-\frac{1}{z})}{1 - \frac{\mu}{\lambda - \lambda z + \mu} \frac{1}{z}} = \frac{1-\rho}{1-\rho z}$$

che è in perfetto accordo con il risultato (2.29).

Il secondo sistema in esame è $M/D/1$ in cui il tempo di servizio è costante e pari ad $1/\mu$ quindi $E[x] = 1/\mu$ e $E[x^2] = 1/\mu^2$. I parametri caratteristici, messi in una forma che ne permette la comparazione con il caso $M/M/1$, risultano:

$$E[k] = \rho + \frac{\lambda^2 \frac{1}{\mu^2}}{2(1-\rho)} = \rho + \frac{\rho^2}{2(1-\rho)} = \frac{\rho}{1-\rho} - \frac{\rho^2}{2(1-\rho)}$$

$$E[q] = E[k] - \rho = \frac{\rho^2}{2(1-\rho)}$$

$$E[T] = \frac{1}{\mu} + \frac{\lambda \frac{1}{\mu^2}}{2(1-\rho)} = \frac{2-2\rho+\rho}{2\mu(1-\rho)} = \frac{2-\rho}{2\mu(1-\rho)} = \frac{1/\mu}{1-\rho} - \frac{\rho}{2\mu(1-\rho)}$$

$$E[T_q] = \frac{\lambda \frac{1}{\mu^2}}{2(1-\rho)} = \frac{\rho}{2\mu(1-\rho)}$$

È interessante confrontare i tempi spesi nei due sistemi:

$$\frac{E[T_{M/D/1}]}{E[T_{M/M/1}]} = \frac{\frac{1/\mu}{1-\rho} - \frac{\rho}{2\mu(1-\rho)}}{\frac{1/\mu}{1-\rho}} = 1 - \frac{\rho}{2}$$

quindi il vantaggio di utilizzare un sistema con servizio deterministico cresce con l'aumentare dell'intensità del traffico fino alla situazione limite $\rho \rightarrow 1$ in cui $E[T_{M/D/1}] \rightarrow E[T_{M/M/1}]/2$.

Per quanto riguarda la distribuzione degli utenti è da notare che essendo il tempo di servizio costante e pari a $1/\mu$ si ha che $\mathcal{B}(s) = e^{-s/\mu}$ per cui:

$$Q(z) = e^{-\frac{\lambda-\lambda z}{\mu}} \frac{(1-\rho)(1-\frac{1}{z})}{1 - \frac{e^{-\frac{\lambda-\lambda z}{\mu}}}{z}} = \frac{(1-\rho)(1-z)}{1 - ze^{\rho(1-z)}}$$

2.8 Sistemi G/M/1

Consideriamo interarrivi indipendenti e identicamente distribuiti, strettamente positivi con media finita $E[t]$, servizi esponenziali indipendenti di parametro $\mu > 0$ e indipendenti dagli arrivi, un servitore work-conserving, disciplina FIFO e buffer infinito. Poniamo $\bar{\lambda} = 1/E[t]$ e assumiamo $0 < \rho = \bar{\lambda}/\mu < 1$. Il modello è anche indicato come GI/M/1 per rendere esplicito che gli arrivi formano un processo di rinnovo. Si scrive $a(t)$ quando la legge possiede densità; gli integrali di Stieltjes consentono anche il caso deterministico.

In modo duale al caso M/G/1 la descrizione dello stato del sistema G/M/1 non può essere basata solo sul numero degli utenti presenti in un certo istante, $N(t)$, ma deve comprendere il tempo trascorso dall'istante dell'ultimo arrivo, $Y_0(t)$, dato che il processo degli arrivi è generico e quindi in generale dotato di memoria. La probabilità che si verifichi un arrivo in un certo intervallo dipende infatti dal tempo trascorso dal momento dell'ultimo arrivo.

Per semplificare la descrizione dello stato del sistema, che risulta bidimensionale $(N(t), Y_0(t))$, si cerca di ricondurre il problema ad un caso monodimensionale fissando gli istanti di osservazione coincidenti con quelli in cui si ha un nuovo arrivo, così che il tempo trascorso dall'ultimo arrivo risulti sempre nullo. Si ottiene quindi una catena di Markov con transizioni in istanti stabiliti.

Definiamo:

- C_n n -esimo cliente che entra nel sistema
- k'_n numero di clienti presenti nel sistema subito **prima** dell'arrivo di C_n
- q_{n+1} numero di clienti **serviti** tra gli arrivi di C_n e C_{n+1}
- t_{n+1} intervallo di tempo tra l'arrivo di C_n e di C_{n+1}

La figura 2.17 mostra la sequenza di istanti di arrivo.

Dal bilancio dei clienti tra due arrivi consecutivi segue la relazione:

$$k'_{n+1} = k'_n + 1 - q_{n+1}$$

Sia $\bar{\lambda} = 1/E[t]$ il tasso medio del processo di rinnovo degli arrivi e sia μ il tasso di servizio. Il fattore di utilizzazione è $\rho = \bar{\lambda}/\mu$.

2.8.1 Probabilità di transizione

Per calcolare le probabilità di transizione associate alla catena di Markov monodimensionale considerata definiamo:

$$p_{ij} = P[k'_{n+1} = j | k'_n = i]$$

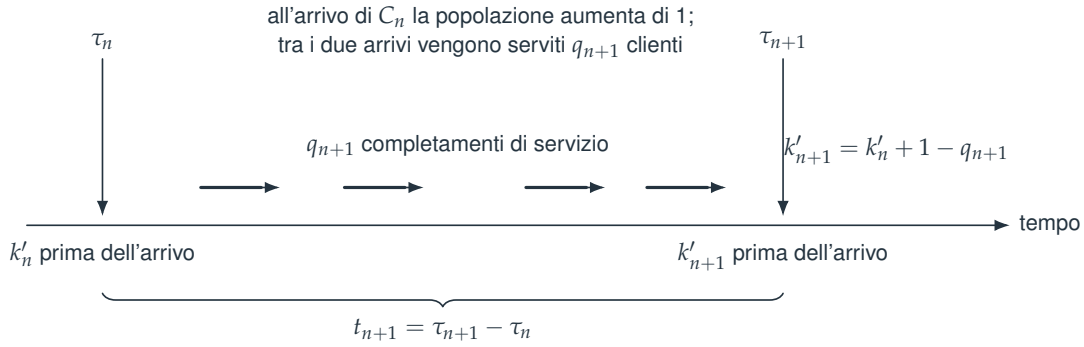
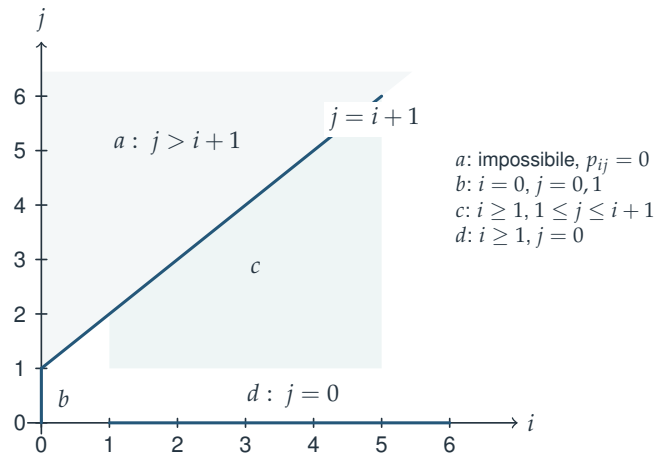


Figura 2.17. Sistema G/M/1.

che rappresenta la probabilità che durante un singolo intervallo fra due arrivi successivi vengano serviti $i + 1 - j$ clienti. Deve essere $p_{ij} = 0$ per $j > i + 1$ dal momento che ci sono al più $i + 1$ clienti nel sistema nel periodo di tempo fra l'arrivo di C_n e di C_{n+1} . Definiamo la matrice delle probabilità di transizione $\underline{\underline{P}} = [p_{ij}]$ e consideriamo quattro regioni del piano generato da i e j come in figura 2.18.

Figura 2.18. Zone di validità delle p_{ij} .

Nella regione a risulta $p_{ij} = 0$ dato che $i + 1 < j$.

La regione b è caratterizzata da $j \leq 1, i = 0$, che individua la situazione in cui l'utente C_n entrato nel sistema va direttamente in servizio e (poco prima che entri C_{n+1}) può aver terminato il servizio ($j = 0$) o meno ($j = 1$). Dato che i tempi di servizio sono distribuiti in modo esponenziale la probabilità che il cliente C_n esca dal sistema entro $t_{n+1} = t$ dal suo arrivo è uguale a $1 - e^{-\mu t}$ e la probabilità che non sia servito entro lo stesso tempo t è data da $e^{-\mu t}$. Quindi, la probabilità di avere j utenti nel sistema (con $j = 0, 1$) poco prima dell'ingresso di C_{n+1} , condizionata al fatto che non ce n'è nessuno quando entra C_n e che $t_{n+1} = t$, risulta:

$$P[k'_{n+1} = j | k'_n = 0, t_{n+1} = t] = [1 - e^{-\mu t}]^{1-j} [e^{-\mu t}]^j = \begin{cases} 1 - e^{-\mu t} & \text{se } j = 0 \\ e^{-\mu t} & \text{se } j = 1 \end{cases}$$

La probabilità di transizione si ottiene rimuovendo la condizione su t_{n+1} , per cui:

$$p_{ij} = \int_0^\infty [1 - e^{-\mu t}]^{1-j} [e^{-\mu t}]^j a(t) dt$$

dove $a(t)$ è la densità di probabilità dei tempi interarrivo.

La regione c è caratterizzata da $1 \leq j \leq i+1, i \geq 1$. Possiamo costruire un orologio poissoniano di completamenti *potenziali* $D_\mu(t)$, di tasso μ , ignorandone gli eventi quando il sistema è vuoto. Per assenza di memoria, questo realizza gli stessi servizi esponenziali. Fra due arrivi non entra nessun altro cliente e vale

$$k'_{n+1} = \max\{0, i+1 - D_\mu(t)\}.$$

Per $j \geq 1$ ciò equivale a $D_\mu(t) = i+1 - j$; pertanto

$$P[k'_{n+1} = j \mid k'_n = i, t_{n+1} = t] = \frac{(\mu t)^{i+1-j}}{(i+1-j)!} e^{-\mu t}.$$

Non si deve aggiungere un condizionamento sul fatto che il servitore rimanga occupato: esso selezionerebbe e rinormalizzerebbe la legge poissoniana.

Rimuovendo la condizione su t_{n+1} la probabilità di transizione risulta:

$$p_{ij} = \int_0^\infty \frac{(\mu t)^{i+1-j}}{(i+1-j)!} e^{-\mu t} a(t) dt = \beta_{i+1-j} \quad (2.55)$$

dove si è introdotta la quantità ad un solo indice β_{i+1-j} dato che gli indici i e j compaiono solo come differenza.

La regione d è caratterizzata da $j = 0, i \geq 1$, che corrisponde al caso in cui C_n arriva e trova il servitore occupato e $i-1$ utenti in coda, coda a cui lui si unisce.

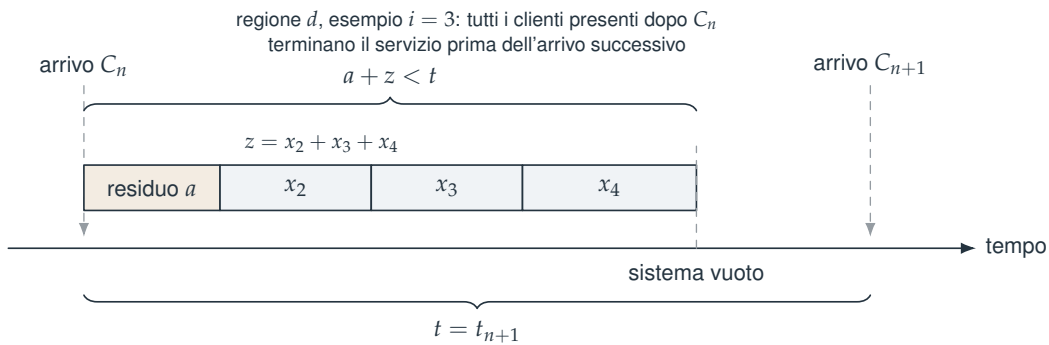


Figura 2.19. Evoluzione temporale del sistema nella regione d per $i = 3$.

L'evoluzione temporale del sistema è mostrata in figura 2.19 per $i = 3$. L'utente già in servizio esce dal sistema dopo un tempo a dall'ingresso di C_n e gli $i = 3$ utenti rimanenti escono dopo un tempo $a + z$ dove z è la somma dei tre tempi di servizio $z = x_2 + x_3 + x_4$. Dato che i tempi di servizio hanno una distribuzione esponenziale, z risulta avere una distribuzione di Erlang, come riportato in sezione 2.5.7, con densità di probabilità $f_z(z) = \mu^i z^{i-1} e^{-\mu z} / (i-1)!$.

Il residuo a del servizio in corso è esponenziale di parametro μ e indipendente dai servizi successivi. Per $z \leq t$, $P[a \leq t - z] = 1 - e^{-\mu(t-z)}$. La probabilità di svuotare il sistema prima dell'arrivo successivo è dunque:

$$p_{i0} = \int_0^\infty \left[\int_0^t [1 - e^{-\mu(t-z)}] f_z(z) dz \right] a(t) dt = \int_0^\infty \left[\int_0^t (1 - e^{-\mu(t-z)}) \frac{\mu^i z^{i-1} e^{-\mu z}}{(i-1)!} dz \right] a(t) dt \quad (2.56)$$

Equivalentemente, l'orologio dei completamenti potenziali deve produrre almeno $i+1$ eventi. Con la definizione di β_m segue

$$p_{i0} = 1 - \sum_{m=0}^i \beta_m.$$

Questa identità verifica anche che ogni riga della matrice di transizione sommi a uno; il caso $i = 0$ dà $p_{00} = 1 - \beta_0$.

La matrice delle probabilità di transizione risulta quindi:

$$\underline{\underline{P}} = \begin{bmatrix} p_{00} & p_{01} = \beta_0 & 0 & .. & .. & .. \\ p_{10} & \beta_1 & \beta_0 & 0 & .. & .. \\ p_{20} & \beta_2 & \beta_1 & \beta_0 & 0 & .. \\ p_{30} & \beta_3 & \beta_2 & \beta_1 & \beta_0 & .. \\ .. & .. & .. & .. & .. & .. \end{bmatrix}$$

2.8.2 Distribuzione del numero dei clienti

Definiamo la probabilità in condizione equilibrio di avere k clienti presenti nel sistema negli istanti che precedono un arrivo:

$$r_k = \lim_{n \rightarrow \infty} P[k'_n = k]$$

Indichiamo inoltre con P_k la probabilità in equilibrio di avere k clienti presenti nel sistema in un qualsiasi istante. Per $k \geq 1$, in regime stazionario il tasso degli attraversamenti del livello k verso l'alto deve coincidere con quello degli attraversamenti verso il basso. Gli attraversamenti verso l'alto sono prodotti dagli arrivi e hanno tasso $\bar{\lambda}r_{k-1}$; quelli verso il basso sono prodotti dai completamenti di servizio e hanno tasso μP_k . Ne segue:

$$P_k = (1 - P_0)r_{k-1} = \rho r_{k-1} \quad (2.57)$$

dove $1 - P_0 = \rho$ segue dal teorema 5.

In generale vale la relazione:

$$r_k = \sum_{i=0}^{\infty} r_i p_{ik} \quad (2.58)$$

che in forma matriciale può essere riscritta come:

$$\underline{r} = \underline{r} \underline{\underline{P}}$$

dove $\underline{\underline{P}} = [p_{ij}]$ è la matrice delle probabilità di transizione appena determinata e $\underline{r} = [r_0, r_1, r_2, \dots]$.

La soluzione dell'equazione (2.58) è data da:

$$r_k = L\sigma^k \quad k \geq 0 \quad (2.59)$$

con L e σ opportunamente scelti.

Quindi la distribuzione del numero di clienti trovati nel sistema da un cliente che arriva è una distribuzione geometrica di parametro σ . Per verificare la (2.59) possiamo sostituirla nella (2.58), considerando $k \geq 1$ e specializzando le probabilità di transizione ricavate in precedenza:

$$r_k = L\sigma^k = \sum_{i=0}^{\infty} r_i p_{ik} = \sum_{i=k-1}^{\infty} r_i p_{ik} = \sum_{i=k-1}^{\infty} L\sigma^i \beta_{i+1-k}$$

Semplificando la costante L e il fattore comune σ^{k-1} otteniamo:

$$\sigma = \sum_{i=k-1}^{\infty} \sigma^{i+1-k} \beta_{i+1-k} = \sum_{n=0}^{\infty} \sigma^n \beta_n$$

Sostituendo l'espressione (2.55) di β_n si ottiene:

$$\sigma = \sum_{n=0}^{\infty} \sigma^n \int_0^{\infty} \frac{(\mu t)^n}{n!} e^{-\mu t} a(t) dt = \int_0^{\infty} e^{-(\mu - \mu\sigma)t} a(t) dt = \mathcal{A}(\mu - \mu\sigma) \quad (2.60)$$

dove $\mathcal{A}(s)$ è la trasformata di Laplace della densità dei tempi interarrivo.

Perché la (2.59) sia soluzione della (2.58) è necessario che la (2.60) sia soddisfatta. Mostriamo che, nel caso stabile $\rho < 1$, esiste una sola radice normalizzabile nell'intervallo $0 < \sigma < 1$. Poniamo

$$F(s) = \mathcal{A}(\mu(1-s)) - s, \quad 0 \leq s \leq 1.$$

Poiché $\mathcal{A}(0) = 1$, si ha $F(1) = 0$: $s = 1$ è sempre una radice, ma non può produrre una distribuzione geometrica normalizzabile. Inoltre

$$F(0) = \mathcal{A}(\mu) > 0.$$

Per $s < 1$ la trasformata è derivabile e

$$F'(s) = \mu E[te^{-\mu(1-s)t}] - 1, \quad F''(s) = \mu^2 E[t^2 e^{-\mu(1-s)t}] > 0.$$

La funzione è quindi strettamente convessa. Al limite per $s \rightarrow 1^-$, usando $E[t] = 1/\bar{\lambda}$,

$$F'(1^-) = \frac{\mu}{\bar{\lambda}} - 1 = \frac{1}{\rho} - 1 > 0.$$

Di conseguenza $F(s) < 0$ per s sufficientemente vicino a 1 da sinistra, mentre $F(0) > 0$: per continuità esiste almeno una radice in $(0, 1)$. La stretta convessità impedisce che, oltre a questa radice e alla radice 1, ve ne sia una terza. Esiste dunque un'unica $\sigma \in (0, 1)$, ed è questa la radice che determina la distribuzione geometrica normalizzabile.

Si può dimostrare, con il procedimento che segue, che la (2.59) vale anche per $k = 0$. Infatti:

$$r_0 = \sum_{i=0}^{\infty} r_i p_{i0} = r_0 p_{00} + r_1 p_{10} + r_2 p_{20} \cdots$$

quindi:

$$r_0 = \frac{1}{(1-p_{00})} \sum_{i=1}^{\infty} r_i p_{i0} = \frac{1}{(1-p_{00})} \sum_{i=1}^{\infty} L \sigma^i p_{i0} = \frac{L}{(1-p_{00})} \sum_{i=1}^{\infty} \sigma^i p_{i0} \quad (2.61)$$

Sfruttando la (2.56) si ottiene:

$$\begin{aligned} \sum_{i=1}^{\infty} \sigma^i p_{i0} &= \int_0^{\infty} \left[\int_0^t [1 - e^{-\mu(t-z)}] \sum_{i=1}^{\infty} \frac{(\sigma\mu)^i}{(i-1)!} z^{i-1} e^{-\mu z} dz \right] a(t) dt = \\ &= \int_0^{\infty} \left[\int_0^t [1 - e^{-\mu(t-z)}] \sigma\mu \sum_{j=0}^{\infty} \frac{(\sigma\mu z)^j}{j!} e^{-\mu z} dz \right] a(t) dt = \\ &= \int_0^{\infty} \left[\int_0^t [1 - e^{-\mu(t-z)}] \sigma\mu e^{\sigma\mu z} e^{-\mu z} dz \right] a(t) dt = \\ &= \sigma\mu \int_0^{\infty} \left[\int_0^t \left(e^{-\mu z + \sigma\mu z} - e^{-\mu z + \sigma\mu z - \mu t + \mu z} \right) dz \right] a(t) dt = \\ &= \sigma\mu \int_0^{\infty} \left[\int_0^t \left(e^{-z(\mu - \sigma\mu)} - e^{-\mu t} e^{\sigma\mu z} \right) dz \right] a(t) dt = \\ &= \sigma\mu \int_0^{\infty} \left(\frac{1 - e^{-t(\mu - \sigma\mu)}}{\mu(1 - \sigma)} - e^{-\mu t} \frac{e^{\sigma\mu t} - 1}{\sigma\mu} \right) a(t) dt = \\ &= \sigma\mu \int_0^{\infty} \frac{\sigma - \sigma e^{-t(\mu - \sigma\mu)} - (1 - \sigma)e^{-t(\mu - \sigma\mu)} + (1 - \sigma)e^{-\mu t}}{\sigma\mu(1 - \sigma)} a(t) dt = \\ &= \frac{1}{1 - \sigma} \left[\int_0^{\infty} \sigma a(t) dt - \int_0^{\infty} e^{-t(\mu - \sigma\mu)} a(t) dt + \int_0^{\infty} e^{-\mu t} (1 - \sigma) a(t) dt \right] = \\ &= \frac{1}{1 - \sigma} \left[\sigma - \sigma + \int_0^{\infty} e^{-\mu t} (1 - \sigma) a(t) dt \right] = \int_0^{\infty} e^{-\mu t} a(t) dt = 1 - p_{00} \end{aligned}$$

dove si è utilizzata la definizione di σ della (2.60). Dalla (2.61) si ottiene quindi $r_0 = L$, che dimostra la validità della (2.58) anche per $k = 0$.

Inoltre:

$$\sum_{k=0}^{\infty} r_k = \sum_{k=0}^{\infty} L\sigma^k = 1 \Rightarrow L = 1 - \sigma$$

Quindi dalla (2.57) e dalla (2.59) risulta:

$$\begin{aligned} r_k &= (1 - \sigma)\sigma^k & k &= 0, 1, 2, \dots \\ P_k &= \rho(1 - \sigma)\sigma^{k-1} & k &= 1, 2, 3, \dots \\ P_0 &= 1 - \rho & k &= 0 \end{aligned} \quad (2.62)$$

2.8.3 Parametri caratteristici per sistemi $G/M/1$

Il numero medio di utenti presenti nel sistema è dato da:

$$E[k] = \sum_{i=0}^{\infty} iP_i = \sum_{i=1}^{\infty} i\rho(1 - \sigma)\sigma^{i-1} = \frac{\rho(1 - \sigma)}{\sigma} \sum_{i=0}^{\infty} i\sigma^i = \frac{\rho}{1 - \sigma}$$

Il numero medio degli utenti in coda è valutabile come:

$$E[q] = \sum_{i=1}^{\infty} (i - 1)P_i = \sum_{i=1}^{\infty} (i - 1)\rho(1 - \sigma)\sigma^{i-1} = \rho \sum_{j=0}^{\infty} j(1 - \sigma)\sigma^j = \frac{\rho\sigma}{1 - \sigma}$$

Sfruttando il risultato di Little è possibile ricavare il tempo medio di permanenza nel sistema, $E[T]$, e in coda, $E[T_q]$:

$$\begin{aligned} E[T] &= \frac{E[k]}{\bar{\lambda}} = \frac{\rho}{\bar{\lambda}(1 - \sigma)} = \frac{1}{\mu(1 - \sigma)} \\ E[T_q] &= \frac{E[q]}{\bar{\lambda}} = \frac{\rho\sigma}{\bar{\lambda}(1 - \sigma)} = \frac{\sigma}{\mu(1 - \sigma)} \end{aligned} \quad (2.63)$$

2.8.4 Distribuzione del tempo di attesa in coda

La probabilità che un cliente che arriva nel sistema entri in coda prima di ricevere servizio (valutata nell'istante che precede il suo arrivo) è:

$$P[\text{coda}] = \sum_{k=1}^{\infty} r_k = \sum_{k=1}^{\infty} (1 - \sigma)\sigma^k = \sigma$$

La probabilità che la coda sia lunga n condizionata al fatto che un utente entri in coda è:

$$P[n \text{ utenti in coda} \mid \text{coda}] = \frac{r_{n+1}}{P[\text{coda}]} = \frac{(1 - \sigma)\sigma^{n+1}}{\sigma} = (1 - \sigma)\sigma^n \quad (2.64)$$

Quindi la lunghezza condizionata della coda è distribuita geometricamente con parametro σ .

Un cliente che entra in coda trova davanti a sé $n + 1$ utenti con una probabilità data dalla (2.64) e in tal caso deve attendere che $n + 1$ clienti lascino il sistema prima che possa ricevere servizio. Pertanto la distribuzione del tempo d'attesa è quella di un intervallo che è la somma di $n + 1$ variabili casuali indipendenti distribuite esponenzialmente, ognuna con parametro μ . Si procede come nella sezione 2.5.7 e si indica con $\mathcal{W}(s)$ la trasformata di Laplace della densità di probabilità del tempo di attesa. Tale trasformata, condizionata al fatto di avere una coda lunga n , è:

$$\mathcal{W}(s|n) = \left(\frac{\mu}{s + \mu}\right)^{n+1}$$

La trasformata della densità del tempo di attesa, condizionata al fatto di attendere in coda, risulta quindi:

$$\begin{aligned} \mathcal{W}(s| \text{coda}) &= \sum_{n=0}^{\infty} \mathcal{W}(s|n) P[n \text{ utenti in coda} | \text{coda}] = \\ &= \sum_{n=0}^{\infty} (1 - \sigma) \sigma^n \left(\frac{\mu}{s + \mu}\right)^{n+1} = (1 - \sigma) \frac{\mu}{s + \mu - \mu\sigma} \end{aligned}$$

La cui antitrasformata è:

$$w(y| \text{coda}) = \mu(1 - \sigma)e^{-\mu(1-\sigma)y} \quad y \geq 0 \quad (2.65)$$

Quindi il tempo di attesa condizionato risulta distribuito in modo esponenziale, con parametro $\mu(1 - \sigma)$.

La (2.65) permette di ottenere la distribuzione non condizionata del tempo di attesa, infatti, se definiamo B come l'evento in cui un cliente che arriva deve attendere e B^c l'evento complementare, segue che:

$$\begin{aligned} W(y) &= P[\text{tempo speso in coda} \leq y] = \\ &= 1 - P[\text{tempo speso in coda} > y | B]P[B] - P[\text{tempo speso in coda} > y | B^c]P[B^c] \end{aligned} \quad (2.66)$$

L'ultimo termine di questa equazione è zero, per cui dalla (2.65) risulta:

$$W(y) = 1 - P[\text{coda}] \int_y^{\infty} w(z| \text{coda}) dz = 1 - \sigma \int_y^{\infty} (1 - \sigma) \mu e^{-\mu(1-\sigma)z} dz$$

da cui:

$$W(y) = 1 - \sigma e^{-\mu(1-\sigma)y} \quad y \geq 0$$

Questa non è una distribuzione esponenziale pura: presenta una massa atomica in $y = 0$ pari a $1 - \sigma$, corrispondente agli utenti che non trovano coda, e una componente continua esponenziale per $y > 0$. Per $y > 0$ la densità della componente non condizionata è quindi:

$$w(y) = \sigma \mu (1 - \sigma) e^{-\mu(1-\sigma)y} \quad y > 0 \quad (2.67)$$

che permette di ritrovare il risultato (2.63) sul tempo medio di permanenza in coda.

L'equazione (2.67) poteva essere anche ricavata notando che:

$$w(y) = w(y| \text{coda}) P[\text{coda}]$$

2.8.5 Due casi particolari

In questa sezione si verificano due casi particolari del sistema G/M/1, utili come controllo dei risultati generali.

Il primo sistema in esame è l'M/M/1, nel quale la distribuzione dei tempi interarrivo $A(t)$ è esponenziale, per cui $\mathcal{A}(s) = \lambda / (s + \lambda)$. Utilizzando la (2.60) si ottiene:

$$\sigma = \frac{\lambda}{\mu - \mu\sigma + \lambda} \Rightarrow (\sigma - 1)(\mu\sigma - \lambda) = 0$$

la cui soluzione significativa è:

$$\sigma = \frac{\lambda}{\mu} = \rho$$

Questo risultato sostituito nella (2.62) porta a:

$$r_k = (1 - \rho)\rho^k = P_k \quad k \geq 0$$

Si ritrovano inoltre i parametri caratteristici:

$$E[k] = \frac{\rho}{1 - \rho}$$

$$E[q] = \frac{\rho^2}{1 - \rho}$$

$$E[T] = \frac{1}{\mu(1 - \rho)}$$

$$E[T_q] = \frac{\rho}{\mu(1 - \rho)}$$

Il secondo sistema in esame è il $D/M/1$, nel quale il tempo interarrivo è costante e pari a $1/\lambda$, per cui $\mathcal{A}(s) = e^{-s/\lambda}$. Utilizzando la (2.60) si ottiene:

$$\sigma = e^{-\frac{(\mu - \mu\sigma)}{\lambda}} = e^{-\frac{(1 - \sigma)}{\rho}} \quad (2.68)$$

che è un'equazione trascendente in σ .

Nelle figure 2.20 e 2.21 è riportato un confronto dei parametri caratteristici $E[k]$, $E[q]$ ed $E[T]$, $E[T_q]$, rispettivamente, per i sistemi $M/M/1$, $M/D/1$ e $D/M/1$.

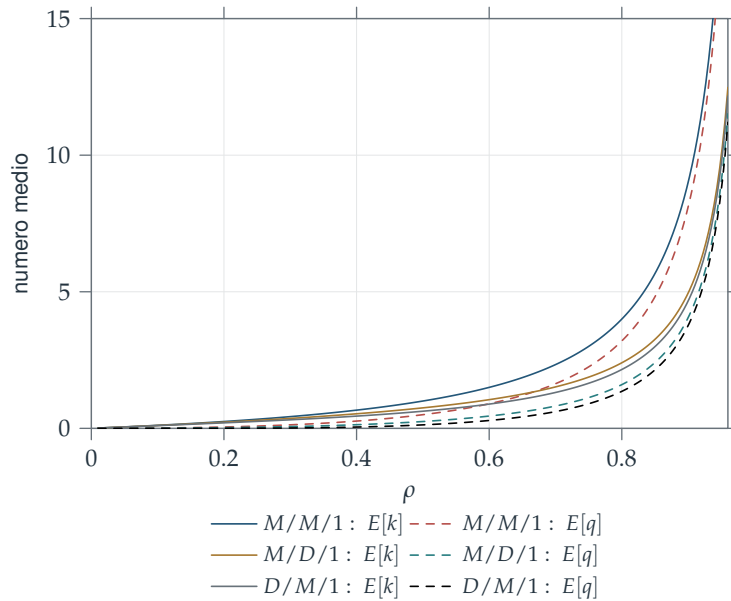


Figura 2.20. Confronto di $E[k]$, $E[q]$ per i sistemi $M/M/1$, $M/D/1$, $D/M/1$.

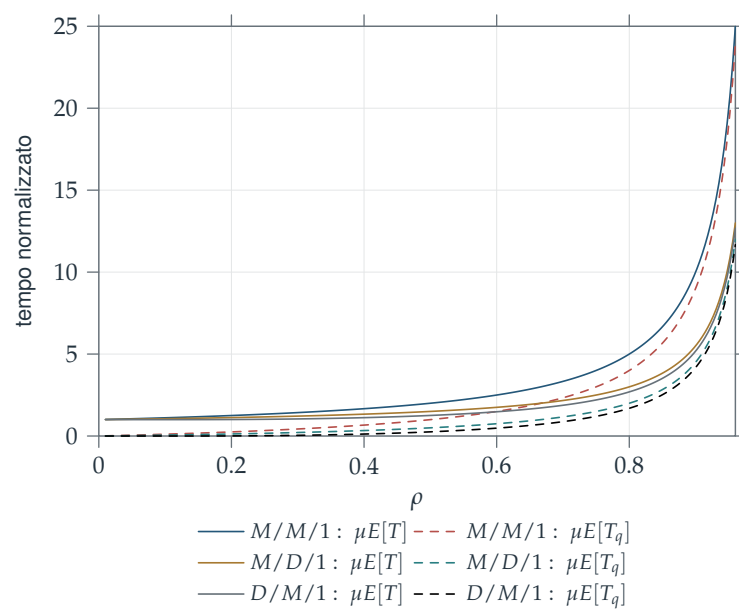


Figura 2.21. Confronto di $E[T]$, $E[T_q]$ per i sistemi $M/M/1$, $M/D/1$, $D/M/1$.

Collegamenti punto-punto

Questo capitolo studia i meccanismi con cui un collegamento digitale può rendere affidabile il trasferimento dell'informazione in presenza di errori. L'affidabilità non significa assenza assoluta di errori: dipende dal modello di canale, dalla capacità del codice di rivelare o correggere le alterazioni e, nei protocolli a ritrasmissione, dalla correttezza delle ipotesi su riscontri e timeout.

Si introducono dapprima i codici a blocchi lineari e i codici ciclici, mostrando come la ridondanza renda possibile la rivelazione o la correzione. Si passa poi alle strategie ARQ, nelle quali un errore rivelato provoca una nuova trasmissione. Per stop-and-wait, Go-Back-N e Selective Repeat vengono distinti i risultati esatti del modello dalle approssimazioni usate per ottenere espressioni analitiche semplici.

3.1 Codifica

Un collegamento digitale deve poter riconoscere, e quando il codice lo consente correggere, le alterazioni introdotte dal canale. Il principio comune è aggiungere ridondanza in modo controllato: il ricevitore non conosce a priori i bit errati, ma conosce l'insieme delle parole valide e può verificare se la sequenza ricevuta vi appartiene.

Le operazioni algebriche di questa sezione sono effettuate nel campo binario $GF(2) = \{0, 1\}$. L'addizione coincide con XOR e la moltiplicazione con il prodotto binario. In particolare $1 + 1 = 0$ e, per ogni elemento, l'opposto coincide con l'elemento stesso. Questa convenzione è essenziale nelle relazioni tra matrice generatrice, matrice di controllo e polinomi dei codici ciclici.

3.1.1 Codici a blocchi lineari

Si consideri l'informazione raggruppata in pacchetti di lunghezza k , detti blocchi, identificati dal vettore $\underline{X}$. Il codificatore lineare genera un vettore di lunghezza n , $\underline{C}$, con $n > k$ (ridondanza di $n - k$ bit), secondo la relazione:

$$\underline{C} = \underline{X} \underline{G}$$

dove $\underline{G}$ è detta matrice di generazione ed ha dimensione $k \cdot n$. Al fine di non perdere informazione occorre che $\underline{G}$ sia di rango k , cioè la matrice di generazione deve essere composta da k vettori indipendenti (base). Dato che esistono più basi di dimensione k si ha che $\underline{G}$ non è unica. Un codice così definito viene indicato come codice (n, k) .

Forma sistematica

Un codice è detto *sistematico* quando i k bit informativi compaiono senza modifica in posizioni prefissate della parola di codice e gli altri $n - k$ bit contengono la ridondanza. La separazione rende immediata l'estrazione dell'informazione quando la parola è ritenuta valida e semplifica molte realizzazioni del codificatore.

Si ha allora che la matrice $\underline{G}$ assume la forma:

$$\underline{G} = [\underline{I}_k | \underline{P}] \quad (3.1)$$

dove $\underline{I}_k$ è la matrice identità di ordine k e $\underline{P}$ è una matrice $k \cdot (n - k)$ detta matrice di parità.

Codice duale $(n, n - k)$

Il codice duale è quel particolare codice i cui vettori sono ortogonali a quelli del codice (n, k) . In questo caso la matrice di generazione $\underline{H}$ ha dimensione $(n - k) \cdot n$.

Sia $\underline{C}_H$ la parola del codice duale e $\underline{Y}$ la parola non codificata di lunghezza $n - k$, allora:

$$\underline{C}_H = \underline{Y} \underline{H} \Rightarrow \underline{C} \underline{C}_H^T = \underline{C} \underline{H}^T \underline{Y}^T = 0 \Rightarrow \underline{C} \underline{H}^T = \underline{0} \quad \forall \underline{Y}$$

quindi:

$$\underline{C} \underline{H}^T = \underline{0}$$

Al fine di costruire la matrice di generazione del codice duale si ha:

$$\underline{X} \underline{G} \underline{H}^T = \underline{0} \quad \forall \underline{X} \Rightarrow \underline{G} \underline{H}^T = \underline{0} \Rightarrow \underline{H} = [-\underline{P}^T | \underline{I}_{n-k}]$$

infatti

$$[\underline{I}_k | \underline{P}] [-\underline{P}^T | \underline{I}_{n-k}]^T = [\underline{I}_k | \underline{P}] \begin{bmatrix} -\underline{P} \\ \underline{I}_{n-k} \end{bmatrix} = \underline{0}$$

dove il $-$ in questo caso coincide con il $+$ dato che la tabella del $+$ coincide con quella dello XOR.

Sindrome di errore

In ricezione occorre verificare la correttezza dell'informazione e questa operazione si realizza tramite la ridondanza.

Si consideri la generica parola ricevuta $\underline{r}$ affetta dal vettore di errori $\underline{e}$:

$$\underline{r} = \underline{C} + \underline{e}$$

allora

$$\underline{r} \underline{H}^T = \underline{C} \underline{H}^T + \underline{e} \underline{H}^T = \underline{e} \underline{H}^T = \underline{S}$$

dove il vettore $\underline{S}$ è detto sindrome di errore, ha dimensione $n - k$ e dipende dal vettore di errore attraverso $\underline{S} = \underline{e} \underline{H}^T$. Una sindrome non nulla segnala che la parola ricevuta non appartiene al codice, ma i singoli bit della sindrome non costituiscono, in generale, una maschera dei bit errati. La sindrome può identificare univocamente un particolare errore soltanto quando la struttura e la distanza minima del codice lo consentono; per esempio, nei codici di Hamming a correzione di singolo errore le sindromi non nulle identificano la posizione del bit errato. Esistono inoltre configurazioni di errore non rivelabili, precisamente quelle per cui $\underline{e} \underline{H}^T = \underline{0}$, cioè quando il vettore di errore trasforma una parola di codice in un'altra parola di codice.

3.1.2 Codici ciclici

Si tratta di una particolare famiglia di codici a blocchi la cui generazione risulta semplice da realizzare in hardware, per esempio mediante registri a scorrimento. La proprietà caratteristica è la *chiusura* rispetto alle traslazioni cicliche: se $[C_{n-1}, \dots, C_0]$ è una parola di codice, allora anche $[C_{n-2}, \dots, C_0, C_{n-1}]$ e tutte le sue ulteriori traslazioni cicliche appartengono al codice. Ciò non significa che tutte le parole del codice debbano essere rotazioni di una singola parola.

Si definisce il seguente polinomio di grado $\leq n - 1$ associato alla parola di codice $[C_{n-1}, \dots, C_0]$:

$$C(p) = C_{n-1}p^{n-1} + C_{n-2}p^{n-2} + \dots + C_0$$

Supponendo di considerare il polinomio:

$$pC(p) = C_{n-1}p^n + C_{n-2}p^{n-1} + \dots + C_0p$$

si nota che questo non rappresenta una parola di codice in quanto il suo grado può essere uguale ad n . Se si procede dividendo $pC(p)$ per il polinomio $p^n + 1$ si ottiene:

$$\frac{pC(p)}{p^n + 1} = C_{n-1} + \frac{C_1(p)}{p^n + 1} \quad (3.2)$$

come si verifica immediatamente per sostituzione, avendo definito $C_1(p)$ come il polinomio associato alla parola di codice $\underline{C}$ ruotata verso sinistra di una posizione:

$$C_1(p) = C_{n-2}p^{n-1} + C_{n-3}p^{n-2} + \dots + C_0p + C_{n-1}$$

Il termine C_{n-1} della (3.2) rappresenta il quoziente della divisione a primo membro e $C_1(p)$ il resto, che è ancora una parola di codice in conseguenza alla definizione di codice ciclico.

Iterando questo procedimento si ha:

$$p^i C(p) = Q(p)(p^n + 1) + C_i(p)$$

dove $C_i(p)$ è il polinomio associato alla parola di codice $\underline{C}$ ruotata verso sinistra di i posizioni.

Viste le proprietà delle parole dei codici ciclici, è interessante analizzare come si realizza il codificatore per un codice a blocchi (n, k) ; si definiscono:

$$X(p) = X_{k-1}p^{k-1} + X_{k-2}p^{k-2} + \dots + X_0$$

$$g(p) = p^{n-k} + g_{n-k-1}p^{n-k-1} + \dots + g_1p + 1$$

che sono rispettivamente il polinomio messaggio di grado $\leq k-1$ ed il polinomio generatore di grado $n-k$, fattore di $p^n + 1$. Si nota che il polinomio generatore è caratterizzato da $g_{n-k} = g_0 = 1$, quindi è sicuramente di grado $n-k$.

Il prodotto $X(p)g(p)$ è un polinomio di grado $\leq n-1$ che può rappresentare una parola di codice, quindi, in generale:

$$C(p) = X(p)g(p) \quad (3.3)$$

Chiaramente vi sono 2^k diversi possibili vettori di informazione $\underline{X}$ a cui competono 2^k distinte parole di codice $\underline{C}$.

Occorre ora mostrare che le parole di codice generate tramite la (3.3) soddisfano la proprietà di ciclicità e che quindi la (3.3) rappresenta un possibile meccanismo per la generazione di un codice ciclico. A tal fine si riscrive la (3.2) come:

$$C_1(p) = pC(p) + C_{n-1}(p^n + 1) \quad (3.4)$$

e si osserva che $g(p)$ è sia un divisore di $C(p)$ (in conseguenza della (3.3)) sia un divisore di $p^n + 1$ per definizione, quindi $g(p)$ è un divisore di entrambi i termini a secondo membro della (3.4) ed allora deve essere anche un divisore di $C_1(p)$. Complessivamente $C_1(p) = X_1(p)g(p)$: esiste dunque un messaggio $X_1(p)$ che genera la parola di codice ruotata. Non si sta affermando che anche il messaggio sia una rotazione di $X(p)$.

Infatti:

$$C_1(p) = X_1(p)g(p) = pC(p) + C_{n-1}(p^n + 1) = pX(p)g(p) + C_{n-1}(p^n + 1)$$

da cui, ponendo

$$h(p) = \frac{p^n + 1}{g(p)},$$

si ricava correttamente

$$X_1(p) = pX(p) + C_{n-1}h(p).$$

Il polinomio $X_1(p)$ è il polinomio informativo che genera la parola di codice ciclicamente traslata $C_1(p)$; in generale non coincide con una semplice rotazione dei k bit del messaggio originario.

Iterando il procedimento si ottiene:

$$C_i(p) = X_i(p)g(p)$$

dove $X_i(p)$ è il polinomio informativo corrispondente alla parola di codice $C_i(p)$ ottenuta mediante i traslazioni cicliche.

Le caratteristiche di ciclicità impongono dei vincoli alla matrice di generazione $\underline{G}$ che deve essere costruita a partire da $g(p)$, come verrà illustrato in seguito con un esempio.

Codice ciclico duale

Sia un codice ciclico binario di lunghezza n generato da $g(p)$ e sia

$$p^n + 1 = g(p)h(p),$$

dove, su $GF(2)$, $p^n + 1$ coincide con $p^n - 1$. Il polinomio $h(p)$, di grado k , è detto polinomio di controllo. Non va interpretato come “ortogonale” a $g(p)$ nel senso del prodotto scalare: la relazione $g(p)h(p) = p^n + 1$ esprime invece che i due polinomi sono fattori complementari del polinomio che definisce la ciclicità.

Il codice duale è ancora ciclico. Con la convenzione di ordinamento dei coefficienti adottata qui, un suo polinomio generatore è il reciproco di $h(p)$:

$$g_{\perp}(p) = p^k h(p^{-1}).$$

Se

$$h(p) = p^k + h_{k-1}p^{k-1} + \dots + h_1p + 1,$$

allora

$$g_{\perp}(p) = 1 + h_{k-1}p + \dots + h_1p^{k-1} + p^k.$$

Per verificare l’ortogonalità, rappresentiamo il prodotto scalare tra due vettori mediante i coefficienti dei rispettivi polinomi. Una parola del codice ha forma $c(p) = a(p)g(p)$ e una parola generata da $g_{\perp}$ ha forma $d(p) = b(p)g_{\perp}(p)$. Nel prodotto ciclico tra $c(p)$ e $d(p^{-1})$ compare il fattore

$$g(p)h(p) = p^n + 1 \equiv 0 \pmod{p^n + 1}.$$

Il coefficiente che rappresenta il prodotto scalare è quindi nullo per ogni coppia di parole. Poiché il codice generato da $g_{\perp}$ ha dimensione $n - k$, esso coincide con il duale del codice originario.

Esempio di codice ciclico

A titolo di esempio si consideri la fattorizzazione:

$$p^7 + 1 = (p + 1)(p^3 + p^2 + 1)(p^3 + p + 1)$$

volendo costruire un codice $(7,4)$ si possono selezionare due diversi polinomi generatori di grado $7 - 4 = 3$. Procedendo con la scelta:

$$g(p) = p^3 + p^2 + 1$$

si ha il corrispondente codice duale $(7,3)$ basato sul polinomio di parità $h(p) = p^4 + p^3 + p^2 + 1$ e sul corrispondente polinomio reciproco:

$$p^4 h(p^{-1}) = 1 + p + p^2 + p^4 = (p+1)(p^3 + p^2 + 1)$$

La costruzione delle matrici di generazione $\underline{\underline{G}}$ ed $\underline{\underline{H}}$ segue dalle caratteristiche di ciclicità:

$$\underline{\underline{G}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$\underline{\underline{H}} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

La prima riga di $\underline{\underline{G}}$ coincide con i coefficienti del polinomio generatore seguiti da tre elementi nulli. Le altre righe si ottengono come shift a destra della prima. Analogamente per $\underline{\underline{H}}$ considerando il polinomio reciproco.

Forma sistematica del codice ciclico

Anche per i codici ciclici, come per i codici a blocchi in generale, è possibile costruire una forma sistematica. Per ottenere l'equivalente della (3.1) si può considerare un polinomio nella forma $p^{n-l} + R_l(p)$ con $l = 1, 2, \dots, k$, dove $R_l(p)$ è un polinomio di grado $< n - k$. Questa forma può essere ottenuta dividendo p^{n-l} per $g(p)$:

$$p^{n-l} = Q_l(p)g(p) + R_l(p) \Rightarrow p^{n-l} + R_l(p) = Q_l(p)g(p)$$

che risulta una parola di codice.

Ad esempio, considerando come polinomio generatore di un codice $(7,4)$ $g(p) = p^3 + p + 1$, si ha:

$$p^6 = (p^3 + p + 1)g(p) + p^2 + 1$$

$$p^5 = (p^2 + 1)g(p) + p^2 + p + 1$$

$$p^4 = pg(p) + p^2 + p$$

$$p^3 = g(p) + p + 1$$

quindi si costruisce la matrice di generazione $\underline{\underline{G}}$ composta da un primo blocco formato dalla matrice identità seguito da un secondo blocco le cui righe sono i coefficienti di $R_l(p)$ con $l = 1, 2, 3$ e 4.

$$\underline{\underline{G}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Per quanto riguarda la generazione diretta di parole di codice in forma sistematica, partendo dal polinomio messaggio $X(p)$, si ha:

$$p^{n-k}X(p) = Q(p)g(p) + R(p) \Rightarrow p^{n-k}X(p) + R(p) = Q(p)g(p)$$

CRC e scelta del polinomio generatore

Un Cyclic Redundancy Check (CRC) applica la costruzione polinomiale appena descritta. Nella forma algebrica elementare, se $g(p)$ ha grado r , il trasmettitore aggiunge prima r posizioni nulle al messaggio, calcolando il resto $R(p)$ della divisione di $p^r X(p)$ per $g(p)$. Trasmette quindi

$$C(p) = p^r X(p) + R(p).$$

Poiché $p^r X(p) = Q(p)g(p) + R(p)$ e in $GF(2)$ vale $R + R = 0$, $C(p) = Q(p)g(p)$ è divisibile per $g(p)$. Dividere il messaggio *non traslato* e poi accodare il resto non realizza in generale questa proprietà. Il ricevitore verifica il resto previsto dalla costruzione. Le specifiche concrete possono includere inizializzazioni, riflessione dei bit e complementi finali: il residuo atteso va allora interpretato secondo tali convenzioni e non è necessariamente lo zero della forma algebrica elementare.

La capacità di rivelazione non dipende soltanto dal numero di bit del CRC, ma dalla scelta del polinomio generatore e dalla lunghezza dei blocchi sui quali esso viene applicato. Per questo il polinomio è parte della specifica del protocollo: due CRC con lo stesso numero di bit possono avere proprietà diverse rispetto a errori singoli, multipli o a burst.

3.1.3 Capacità di rivelazione e/o correzione

Sia $\underline{C}_i$ la i -esima parola di codice. La distanza di Hamming tra due parole è il numero di posizioni nelle quali esse differiscono:

$$d_H(\underline{C}_i, \underline{C}_j) = w_H(\underline{C}_i + \underline{C}_j),$$

dove w_H indica il peso di Hamming e la somma è effettuata in $GF(2)$. Se i vettori binari sono immersi in $\mathbb{R}^n$, questa quantità coincide anche con $\|\underline{C}_i - \underline{C}_j\|^2$. La distanza minima del codice è quindi

$$d_{min} = \min_{i \neq j} d_H(\underline{C}_i, \underline{C}_j).$$

Per un codice lineare essa coincide con il minimo peso di Hamming tra le parole di codice non nulle. Infatti, per linearità, $\underline{C}_i + \underline{C}_j$ è ancora una parola di codice e $d_H(\underline{C}_i, \underline{C}_j) = w_H(\underline{C}_i + \underline{C}_j)$; al variare di due parole distinte la loro somma percorre le parole non nulle del codice. Viceversa ogni parola non nulla è la differenza, che in $GF(2)$ coincide con la somma, tra quella parola e la parola nulla.

La figura 3.1 rappresenta schematicamente due sfere di Hamming centrate su parole di codice distinte. I punti interni rappresentano vettori ricevuti ottenibili con un numero di errori non superiore al raggio di correzione. La figura è concettuale: la geometria effettiva è quella dello spazio discreto di Hamming, non una geometria euclidea.

Per quanto riguarda la rivelazione occorre solo verificare se i disturbi hanno cambiato la parola di codice in un'altra. Se il numero degli errori è minore della distanza minima, la sindrome di errore è necessariamente non nulla. Il massimo numero di errori di cui il codice garantisce la rivelazione, indipendentemente dalle posizioni coinvolte, è quindi:

$$d_{min} - 1 \tag{3.5}$$

infatti un numero d_{min} di errori potrebbe portare una parola di codice in un'altra e l'errore non verrebbe rilevato.

Per quanto riguarda la correzione, invece, occorre che i disturbi facciano rimanere la parola di codice all'interno del cerchio centrato sulla parola di codice corretta. Al fine della correzione i cerchi non devono né intersecarsi né toccarsi, quindi il numero di errori correggibili risulta:

$$\left\lfloor \frac{d_{min} - 1}{2} \right\rfloor \tag{3.6}$$

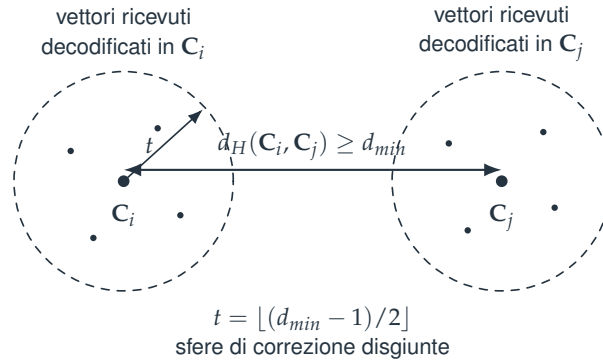


Figura 3.1. Sfere di Hamming attorno a due parole di codice e distanza minima.

La giustificazione nello spazio discreto non richiede l'analogia con i cerchi. Se un vettore ricevuto r fosse a distanza al più t da due parole distinte c_1, c_2 , la disuguaglianza triangolare darebbe $d_H(c_1, c_2) \leq d_H(c_1, r) + d_H(r, c_2) \leq 2t$. Pertanto $2t < d_{\min}$ rende univoca la decodifica entro raggio t . Viceversa, se $d_{\min} \leq 2t$, si prendano due parole alla distanza minima e si dividano le posizioni in cui differiscono in due gruppi, ciascuno di cardinalità al più t . Scegliendo in un gruppo i bit della prima parola e nell'altro quelli della seconda si costruisce un vettore a distanza al più t da entrambe. La garanzia universale di correzione è dunque esattamente la (3.6). La distanza minima è legata ai parametri n e k secondo una relazione che dipende dal tipo di codice utilizzato; in generale si ha:

$$d_{\min} \leq n - k + 1 \quad (3.7)$$

Questo limite, noto come limite di Singleton, segue direttamente da un argomento di conteggio. Si eliminino $d_{\min} - 1$ coordinate da ogni parola di un codice (n, k) . Due parole distinte non possono diventare uguali dopo l'eliminazione: se accadesse, prima dell'eliminazione differirebbero in al più $d_{\min} - 1$ posizioni, in contraddizione con la definizione di distanza minima. Le 2^k parole del codice devono quindi rimanere tutte distinte nelle $n - d_{\min} + 1$ coordinate residue. Poiché queste coordinate possono rappresentare al massimo $2^{n-d_{\min}+1}$ vettori distinti,

$$2^k \leq 2^{n-d_{\min}+1},$$

da cui $k \leq n - d_{\min} + 1$ e quindi $d_{\min} \leq n - k + 1$.

3.2 Strategie ARQ

Per consegnare correttamente i pacchetti al destinatario si può operare sia cercando di correggere direttamente i pacchetti in arrivo che rivelando i pacchetti errati e richiedendone la ritrasmissione. Queste due tecniche vengono chiamate rispettivamente FEC (Forward Error Correction) ed ARQ (Automatic Repeat Request).

FEC e ARQ spostano il costo dell'affidabilità in punti diversi. Il FEC aggiunge ridondanza sufficiente a correggere determinati errori senza attendere una nuova trasmissione; l'ARQ può usare un codice destinato principalmente alla rivelazione, ma paga eventuali errori con ritardo e traffico di ritrasmissione. La scelta dipende quindi da probabilità e struttura degli errori, latenza del collegamento, disponibilità del canale di ritorno e overhead accettabile.

Nelle strategie ARQ il ricevitore può inviare un riscontro positivo, **ACK** (acknowledgment), oppure un riscontro negativo, **NACK**, quando il protocollo lo prevede. Un ACK conferma che il ricevitore ha accettato il pacchetto o una sequenza di pacchetti; un NACK segnala esplicitamente la necessità di recuperare dati mancanti o errati. Inoltre può accadere che il

pacchetto venga distrutto o perso e che quindi non vi siano né conferme né segnalazioni di errore. Per evitare situazioni di stallo (deadlock) viene previsto un meccanismo di timeout in grado di forzare la ritrasmissione del pacchetto dopo un certo tempo.

Supponiamo, nella trattazione che segue, che la comunicazione sia full duplex, così che ACK e NACK possano viaggiare nella direzione inversa mentre il flusso dati continua nella direzione diretta. Non è necessario che le due direzioni usino mezzi fisici separati: è sufficiente che il modello consenta il loro uso contemporaneo.

Un protocollo ARQ deve anche riconoscere i duplicati: se un pacchetto è arrivato ma il suo ACK si perde, la ritrasmissione non deve essere consegnata due volte. A questo servono numeri di sequenza e memoria locale del ricevitore. Nello Stop-and-Wait è sufficiente alternare due valori, purché una vecchia copia non possa sopravvivere fino al riuso dello stesso numero.

Con numerazione modulo M , una finestra di trasmissione di W_s valori e una finestra di ricezione di W_r valori richiedono, nel modello ordinario senza ulteriori identificatori di generazione, $W_s + W_r \leq M$. Infatti i W_s numeri ancora riproponibili dal mittente e i W_r numeri che il ricevitore può accettare come nuovi devono occupare intervalli disgiunti nello spazio ciclico; se la somma superasse M , almeno un numero rappresenterebbe entrambe le categorie. Per Go-Back-N il ricevitore accetta soltanto il prossimo pacchetto, quindi $W_r = 1$ e $W_s \leq M - 1$. Per Selective Repeat con finestre uguali, $2W_s \leq M$. Questi vincoli non sostituiscono il requisito sulla vita massima delle copie: un duplicato arbitrariamente antico richiederebbe ulteriori meccanismi per essere distinto da dati nuovi.

Nei modelli prestazionali seguenti si assume che ogni pacchetto corrotto venga riconosciuto come errato. Una probabilità residua di errore non rivelato va analizzata separatamente e non viene eliminata dalle ritrasmissioni.

3.2.1 Parametri caratteristici

Lo studio dei sistemi ARQ viene compiuto attraverso tre parametri caratteristici:

- t_T : tempo totale per la trasmissione di un pacchetto. Questo tempo può essere maggiore del tempo di trasmissione del singolo pacchetto a causa delle ritrasmissioni dovute agli errori.
- ρ : fattore di utilizzazione, definito dal rapporto tra il tempo di trasmissione del singolo pacchetto ed il tempo totale speso per consegnarlo correttamente.
- η : efficienza, definita dal rapporto tra il tempo di trasmissione della sola parte utile (senza overhead) del pacchetto ed il tempo totale speso per consegnarlo correttamente.

Per le prestazioni medie si usano i rapporti $t_F/E[t_T]$ e $t_I/E[t_T]$, non le medie dei rapporti casuali. Nei modelli a pipeline t_T rappresenta il tempo di canale imputato, in media, a un pacchetto consegnato; non coincide con la sua latenza individuale. Il simbolo ρ indica qui la frazione normalizzata di trasmissioni riuscite, non la probabilità che il mezzo sia fisicamente occupato: un mezzo può essere sempre occupato da tentativi, anche quando molte trasmissioni falliscono. Questa utilizzazione utile va distinta da quella del servitore definita nel capitolo 2.

3.2.2 Stop and wait

Nello Stop-and-Wait, mostrato in figura 3.2, il trasmettitore mantiene al più un pacchetto non confermato. Il pacchetto successivo F_{i+1} viene trasmesso solo dopo l'ACK relativo a F_i ; F_i viene invece ritrasmesso in seguito a un NACK o alla scadenza del timeout. Dopo la trasmissione di un pacchetto si attende sempre il verificarsi di un evento, da cui il nome Stop-and-Wait.

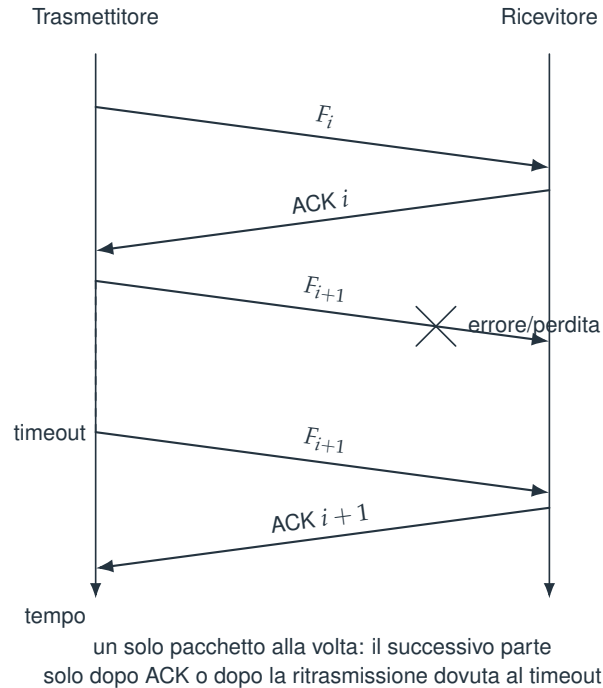


Figura 3.2. Procedura ARQ Stop-and-Wait.

Si assume la seguente notazione:

- t_F tempo di trasmissione del pacchetto, costante
- t_o timeout
- τ_p ritardo di propagazione
- t_A tempo di trasmissione ACK o NACK
- t_G tempo complessivo per la trasmissione di un solo pacchetto indipendentemente dagli eventuali errori che si presentano, comprensivo del tempo di trasmissione del pacchetto, dell'ACK e dei ritardi di propagazione

si assume inoltre che sia trascurabile il tempo di processamento del pacchetto. Allora:

$$t_G = t_F + \tau_p + t_A + \tau_p = t_F + t_A + 2\tau_p \quad (3.8)$$

Per quanto riguarda il timeout, dalla fine della trasmissione occorre attendere almeno che il messaggio arrivi al destinatario, che questo abbia il tempo di trasmettere l'ACK o il NACK e che tale segnalazione ritorni alla sorgente, quindi:

$$t_o \geq t_A + 2\tau_p$$

Si procede con le ulteriori ipotesi:

- $p = P[\text{pacchetto errato}]$, con errori indipendenti sui bit e tentativi di trasmissione indipendenti e identicamente distribuiti
- t_i tempo addizionale speso quando si verificano i tentativi falliti prima della consegna corretta
- P_i probabilità che i primi i tentativi falliscano e che il successivo, cioè il tentativo $i + 1$, abbia successo
- ack e nack privi di errori

Supponendo di considerare la specifica più stringente possibile per il timeout:

$$t_o = t_A + 2\tau_p \Rightarrow t_G = t_F + t_o$$

si può unificare il caso del timeout con i casi di ACK o NACK. Essendo i il numero delle ritrasmissioni si ha:

$$\begin{cases} t_i = i t_G \\ P_i = (1 - p)p^i \end{cases}$$

Per calcolare il tempo medio per la trasmissione corretta di un pacchetto occorre considerare il tempo per la trasmissione nel caso non vi siano errori ed il tempo speso per ogni eventuale ritrasmissione:

$$E[t_T] = t_G + \sum_{i=0}^{\infty} t_i P_i = t_G \left[1 + (1 - p) \sum_{i=0}^{\infty} i p^i \right]$$

Ricordando la (2.29) e la (2.30) si ha:

$$E[t_T] = t_G \left[1 + \frac{p}{1 - p} \right] = \frac{t_G}{1 - p} = \frac{t_F + t_o}{1 - p} = \frac{t_F + t_A + 2\tau_p}{1 - p} = \frac{t_F}{1 - p} \left(1 + \frac{A}{F} + 2\alpha \right) \quad (3.9)$$

dove assumendo la velocità di trasmissione pari a C (costante), si ha $A = Ct_A$ la lunghezza del messaggio di ACK o NACK, $F = Ct_F$ la lunghezza del messaggio informativo e $\alpha = \frac{\tau_p}{E[t_F]} = \frac{\tau_p}{t_F}$ il ritardo di propagazione normalizzato.

Si noti che, dato che t_G è il tempo totale per trasmettere il singolo pacchetto in assenza di errori, il termine $1/(1 - p)$ nella (3.9) rappresenta il numero medio di trasmissioni affinché il pacchetto arrivi a buon fine. Il numero medio di ritrasmissioni risulta quindi pari a $1/(1 - p) - 1 = p/(1 - p)$.

Il fattore di utilizzazione risulta:

$$\rho = \frac{t_F}{E[t_T]} = \frac{t_F(1 - p)}{t_F + t_o}$$

che ammette i seguenti limiti di interesse:

$$\lim_{p \rightarrow 0} \rho = \frac{t_F}{t_F + t_o}$$

$$\lim_{t_o \rightarrow 0} \rho = 1 - p$$

Si osservi che quando la probabilità di errore p tende a 0 il fattore di utilizzazione non tende al caso ideale 1; ciò è dovuto all'attesa ineliminabile di ack, nack o timeout, prima di poter effettuare una qualunque nuova operazione, anche se il canale è privo di errori.

Valutiamo ora il fattore di utilizzazione in funzione dei parametri caratteristici della trasmissione. Sia $I = Ct_I$ la lunghezza del pacchetto informazione, $O = Ct_O$ la parte del pacchetto relativa all'overhead assunta uguale, per semplicità, al messaggio di ACK o NACK: $O = A$; allora:

$$\rho = \frac{\frac{F}{C}(1 - p)}{\frac{F}{C} + \frac{A}{C} + 2\tau_p} = \frac{F(1 - p)}{F + O + 2C\tau_p} \quad (3.10)$$

Se gli errori sui bit sono indipendenti e hanno probabilità P_b , la probabilità esatta che un pacchetto di F bit contenga almeno un errore è

$$p = 1 - (1 - P_b)^F. \quad (3.11)$$

Per $FP_b \ll 1$ si può usare l'approssimazione del primo ordine $p \simeq FP_b$. In presenza di errori correlati o a burst la (3.11) non è in generale valida: occorre specificare il modello di errore del canale. Dalla (3.10) e dalla relazione esatta (3.11), con $F = I + O$, si ottiene

$$\rho = \frac{F(1-p)}{F+O+2C\tau_p} = \frac{(I+O)(1-P_b)^{I+O}}{I+2O+2C\tau_p}.$$

Per quanto riguarda l'efficienza η :

$$\eta = \frac{t_I}{E[t_T]} = \frac{t_F}{E[t_T]} \frac{t_I}{t_F} = \rho \frac{I}{I+O} = \frac{I(1-P_b)^{I+O}}{I+2O+2C\tau_p}.$$

Le curve della figura 3.3 sono rigenerate mediante queste espressioni esatte; l'approssimazione $p \simeq FP_b$ viene usata solo quando serve ottenere una forma analitica semplice.

In figura 3.3 vengono riportati gli andamenti di ρ e η per due possibili valori di ritardo al variare della lunghezza dell'informazione utile I avendo fissato l'overhead $O = 50$ e la probabilità di errore per bit $P_b = 10^{-5}$. È interessante osservare la presenza di un massimo che rappresenta il punto di ottimo.

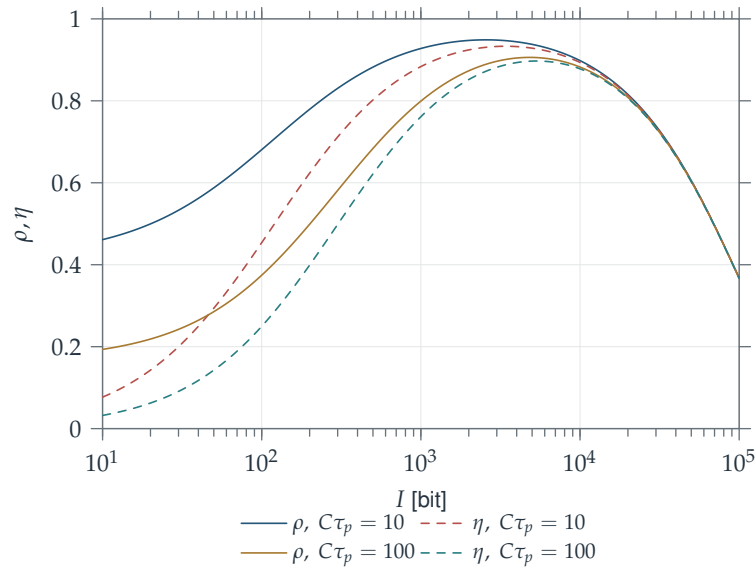


Figura 3.3. Fattore di utilizzazione ed efficienza per Stop-and-Wait con $O = 50$ e $P_b = 10^{-5}$.

Usando, in questa sola derivazione, l'approssimazione $p \simeq FP_b$, la massima efficienza si ottiene quando:

$$\frac{d\eta}{dI} = \frac{1 - 2IP_b - OP_b}{I + 2O + 2C\tau_p} - \frac{I[1 - (I + O)P_b]}{(I + 2O + 2C\tau_p)^2} = 0$$

semplificando:

$$I - 2I^2P_b - OIP_b + 2O - 4OIP_b - 2O^2P_b + 2C\tau_p - 4IP_bC\tau_p - 2C\tau_pOP_b = I - I^2P_b - IOP_b$$

Ponendo $D = 2O + 2C\tau_p$, l'equazione precedente si riduce a

$$P_b I^2 + 2DP_b I - D(1 - OP_b) = 0.$$

Per $OP_b < 1$ la radice positiva del modello linearizzato è

$$I_{\text{opt}} = -D + \sqrt{D^2 + \frac{D(1 - OP_b)}{P_b}}.$$

Se inoltre $D \ll I_{\text{opt}}$ e $(I_{\text{opt}} + O)P_b \ll 1$, si può trascurare il termine lineare e si ritrova

$$I_{\text{opt}} \simeq \sqrt{\frac{2(O + C\tau_p)}{P_b}}.$$

La lunghezza effettiva è intera e deve rispettare i limiti del protocollo; si confrontano quindi i valori ammissibili vicini all'ottimo continuo.

3.2.3 Go back N

La trattazione che segue è un modello analitico idealizzato: si assume traffico sempre disponibile, pacchetti di durata costante, errori indipendenti tra i tentativi, ACK/NACK privi di errore e una finestra scelta in modo da riempire il tempo di andata e ritorno. Se W è la finestra intera, l'assenza di tempi morti senza errori richiede $Wt_F \geq t_F + t_A + 2\tau_p$, cioè $W \geq \lceil 1 + (t_A + 2\tau_p)/t_F \rceil$. Nel calcolo continuo seguente si usa invece $N_* = (t_F + t_A + 2\tau_p)/t_F$ come costo temporale normalizzato di un ciclo di recupero, anche quando non è intero. Esso coincide con una finestra fisica N soltanto quando l'allineamento è esatto. In implementazioni generali del Go-Back-N il throughput dipende anche dalla dimensione della finestra e dalla temporizzazione effettiva degli ACK.

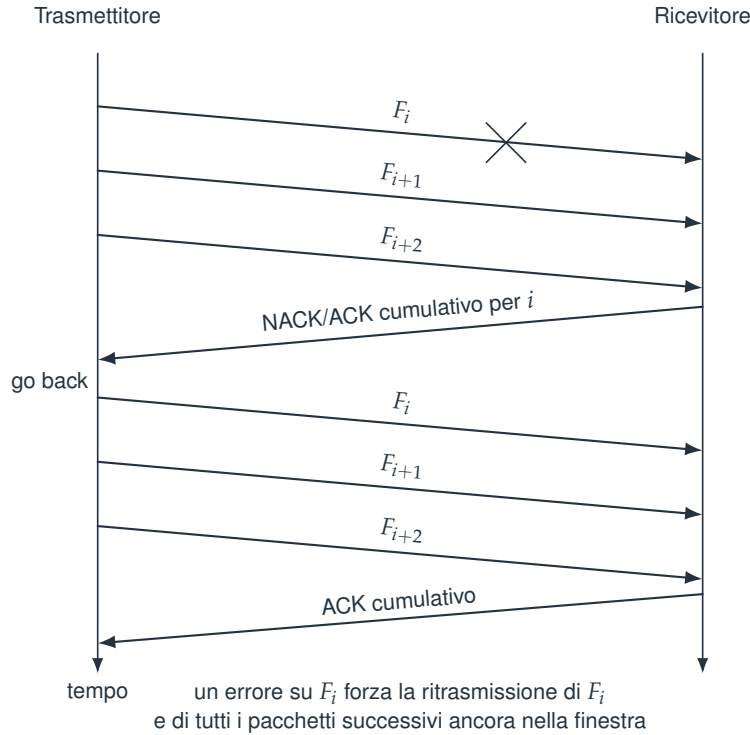


Figura 3.4. Procedura ARQ Go-Back-N.

Nel Go-Back-N, illustrato in figura 3.4, più pacchetti possono essere in volo contemporaneamente entro una finestra di trasmissione. Un errore sul pacchetto F_i (con conseguente ricezione di un NACK) o lo scadere del timeout implica la ritrasmissione di tutti i pacchetti a partire da F_i , da cui la nomenclatura Go-Back-N. Nel modello di recupero adottato si attribuisce a ogni tentativo fallito un ciclo di durata Nt_F prima del successivo tentativo utile. Il timeout, misurato dalla fine del pacchetto, viene posto pari al tempo di $N - 1$ ulteriori trasmissioni; questa è una scelta analitica, non una definizione generale del timeout Go-Back-N:

$$t_o = (N - 1)t_F$$

Anche per il Go-Back-N si suppone di utilizzare un timeout stringente, alla stregua di quanto fatto nello Stop-and-Wait, quindi:

$$t_o = t_A + 2\tau_p$$

Il valore di N deve essere tale che:

$$(N - 1)t_F = t_A + 2\tau_p$$

Definiamo anche in questo caso un tempo t_G che rappresenta il tempo totale per trasmettere N pacchetti. Si ha:

$$t_G = Nt_F = t_F + t_A + 2\tau_p$$

che risulta coerente con la definizione data nella tecnica Stop-and-Wait (equazione (3.8)).

In questo modello saturo, a differenza della tecnica Stop-and-Wait, il tempo di canale attribuito a un pacchetto consegnato senza errori è t_F , perché la stazione può continuare a trasmettere mentre attende l'ACK. Il ritardo di riscontro non scompare fisicamente: determina la finestra necessaria a mantenere piena la pipeline e contribuisce alla latenza del singolo pacchetto, ma non introduce tempo morto sul canale quando la finestra assunta dal modello è sufficiente. Mentre, con le assunzioni descritte in precedenza, i valori di t_i e P_i sono uguali al caso Stop-and-Wait. Infatti il tempo necessario per ritrasmettere il pacchetto F_i include anche la ritrasmissione degli $N - 1$ pacchetti successivi, quindi $t_i = iNt_F = it_G$.

Il calcolo del tempo medio totale è quindi:

$$\begin{aligned} E[t_T] &= t_F + \sum_{i=0}^{\infty} t_i P_i = t_F + t_G(1 - p) \sum_{i=0}^{\infty} i p^i = t_F + t_G \frac{p}{1 - p} = \\ &= t_F \left(1 + \frac{p}{1 - p} \right) + t_o \frac{p}{1 - p} = \frac{t_F + p t_o}{1 - p} = \frac{t_F}{1 - p} \left(1 + \frac{A}{F} p + 2\alpha p \right) \end{aligned}$$

Si osserva anche che:

$$E[t_T] = t_F + \frac{N t_F p}{1 - p} = \frac{t_F}{1 - p} + \frac{p t_F (N - 1)}{1 - p}$$

quindi $1/(1 - p)$ rappresenta il numero medio di trasmissioni per consegnare il pacchetto F_i corretto e $p/(1 - p)$ il numero medio di ritrasmissioni che coinvolgono gli $N - 1$ pacchetti che seguono F_i .

Il coefficiente di utilizzazione risulta:

$$\rho = \frac{t_F}{E[t_T]} = \frac{t_F(1 - p)}{t_F + p t_o}$$

che ammette i seguenti limiti di interesse:

$$\lim_{p \rightarrow 0} \rho = 1$$

$$\lim_{t_o \rightarrow 0} \rho = 1 - p$$

Si nota che, a differenza del caso Stop-and-Wait, al tendere di p a zero il fattore di utilizzazione tende al valore massimo 1, grazie all'assenza di tempi morti sul canale. La latenza di propagazione e di riscontro rimane presente per il singolo pacchetto.

Sostituendo i parametri definiti nella sezione precedente e ponendo $p = 1 - (1 - P_b)^{I+O}$ si ottiene, senza linearizzare la probabilità di errore di pacchetto,

$$\rho = \frac{F(1-p)}{F + p[O + 2C\tau_p]} = \frac{(I+O)(1-P_b)^{I+O}}{I+O + [1 - (1-P_b)^{I+O}](O + 2C\tau_p)}.$$

Per quanto riguarda l'efficienza:

$$\eta = \rho \frac{I}{I+O}.$$

Le curve della figura 3.5 sono ottenute da queste espressioni esatte per il bilancio temporale idealizzato descritto, con il costo continuo N_* e tentativi utili indipendenti. Non sono formule esatte per qualunque implementazione Go-Back-N. Se $FP_b \ll 1$, ponendo $p \simeq FP_b$ si ritrovano le forme approssimate utilizzate per la derivazione analitica che segue.

In figura 3.5 vengono riportati gli andamenti di ρ e η per due possibili valori di ritardo al variare della lunghezza dell'informazione utile I avendo fissato l'overhead $O = 50$ e la probabilità di errore per bit $P_b = 10^{-5}$. Le curve presentano un massimo, che individua la lunghezza ottima nel modello considerato. Nell'approssimazione $p \simeq FP_b$ la posizione dell'ottimo rispetto a I risulta indipendente dal ritardo di propagazione; la curva completa viene comunque valutata con la probabilità di errore esatta.

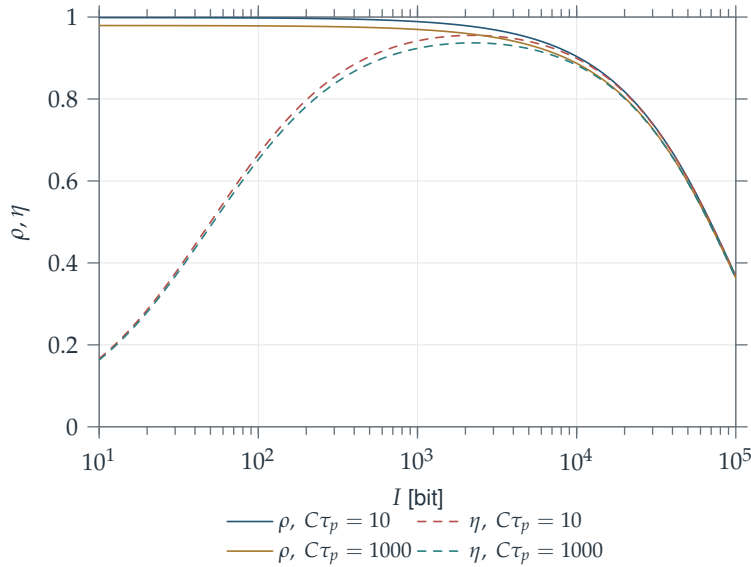


Figura 3.5. Fattore di utilizzazione ed efficienza per Go-Back-N con $O = 50$ e $P_b = 10^{-5}$.

Usando l'approssimazione $p \simeq FP_b$, l'efficienza assume la forma

$$\eta \simeq \frac{I[1 - (I+O)P_b]}{(I+O)[1 + P_b(O + 2C\tau_p)]}.$$

Il secondo fattore al denominatore non dipende da I ; la posizione dell'ottimo si ottiene quindi derivando soltanto $I[1 - (I+O)P_b]/(I+O)$. La condizione di stazionarietà è

$$O - P_b(I+O)^2 = 0,$$

da cui, nel modello approssimato,

$$I_{\text{opt}} = \sqrt{\frac{O}{P_b}} - O.$$

Se inoltre $O \ll I_{\text{opt}}$ si ritrova la forma semplificata

$$I_{\text{opt}} \simeq \sqrt{\frac{O}{P_b}}.$$

Il ritardo di propagazione modifica il valore dell'efficienza ma non la posizione di questo ottimo, conseguenza specifica delle ipotesi adottate per la finestra e per il timeout.

3.2.4 Selective repeat

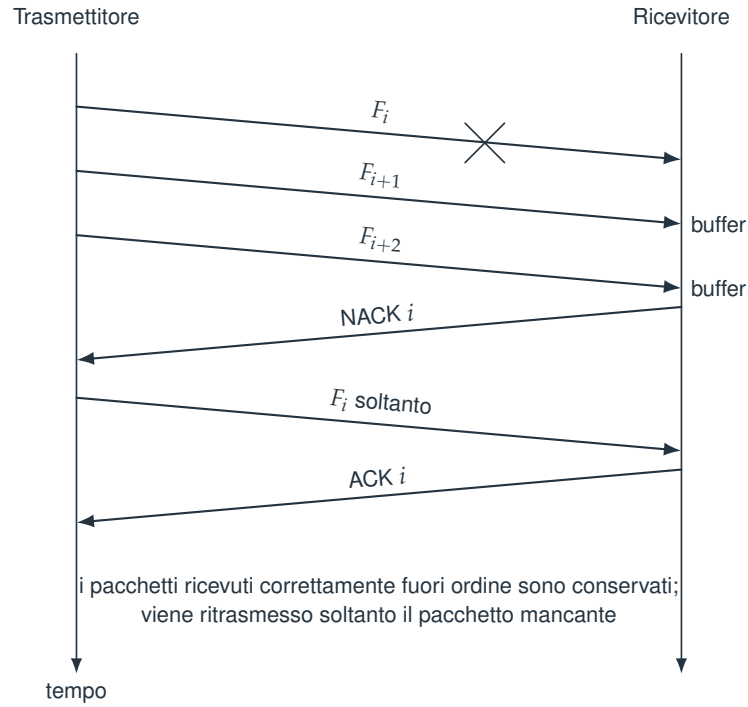


Figura 3.6. Procedura ARQ Selective Repeat.

Il Selective Repeat evita di ritrasmettere i pacchetti successivi che il ricevitore ha già acquisito correttamente: viene recuperato soltanto il pacchetto mancante o errato. Il prezzo è una maggiore complessità di stato e di buffering, perché il ricevitore deve conservare e riordinare pacchetti ricevuti fuori sequenza. Nel modello idealizzato a pipeline satura, con finestra abbastanza ampia da non lasciare inattivo il trasmettitore, si contabilizza il *tempo di trasmissione sul canale diretto* consumato dal pacchetto. Se sono necessarie i ritrasmissioni selettive, il lavoro addizionale sul canale è

$$t_i = it_F.$$

Questa quantità non è la latenza end-to-end del pacchetto: un NACK o un timeout può essere ricevuto soltanto dopo il relativo ritardo di propagazione. La distinzione consente di interpretare correttamente le formule seguenti come utilizzazione/efficienza del canale in regime continuo.
quindi

$$E[t_T] = t_F + \sum_{i=0}^{\infty} t_i P_i = t_F + t_F(1-p) \sum_{i=0}^{\infty} ip^i = \frac{t_F}{1-p}$$

ottenuto per analogia con le valutazioni del caso Stop-and-Wait. Per quanto riguarda il fattore di utilizzazione, usando la stessa probabilità esatta di errore di pacchetto introdotta in precedenza,

$$\rho = \frac{t_F}{E[t_T]} = 1 - p = (1 - P_b)^{I+O}.$$

L'efficienza riferita alla sola parte utile del pacchetto risulta:

$$\eta = \frac{I}{I+O}(1 - P_b)^{I+O}.$$

Soltanto nell'approssimazione di primo ordine $(I+O)P_b \ll 1$ si ha

$$\rho \simeq 1 - (I+O)P_b, \quad \eta \simeq \frac{I[1 - (I+O)P_b]}{I+O}.$$

Nel modello idealizzato considerato, con pipeline continua e senza vincoli ulteriori sulla finestra, fattore di utilizzazione ed efficienza non dipendono dal ritardo di propagazione. In figura 3.7 vengono riportati gli andamenti di ρ e η al variare della lunghezza dell'informazione utile I avendo fissato l'overhead $O = 50$ e la probabilità di errore per bit $P_b = 10^{-5}$. In questo caso è l'efficienza η a presentare un massimo: $\rho = (1 - P_b)^{I+O}$ diminuisce invece con I per $0 < P_b < 1$. L'ottimo dell'efficienza è analogo a quello del Go-Back-N nel modello assunto.

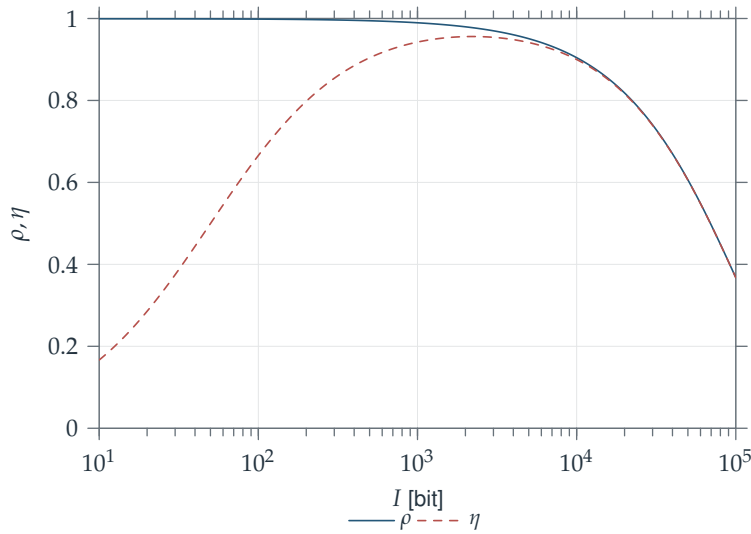


Figura 3.7. Fattore di utilizzazione ed efficienza per Selective Repeat con $O = 50$ e $P_b = 10^{-5}$.

Usando l'approssimazione di primo ordine appena indicata, la massima efficienza si ottiene quando:

$$\frac{d\eta}{dI} = \frac{1 - 2IP_b - OP_b}{I+O} - \frac{I[1 - (I+O)P_b]}{(I+O)^2} = 0.$$

semplificando si trova una formulazione analoga a quella del Go-Back-N:

$$I - 2I^2P_b - OIP_b + O - 2IOP_b - O^2P_b = I - I^2P_b - IOP_b.$$

Raccogliendo i termini si ottiene

$$P_b(I+O)^2 = O,$$

per cui, nel modello approssimato di primo ordine,

$$I_{\text{opt}} = \sqrt{\frac{O}{P_b}} - O.$$

Quando $O \ll I_{\text{opt}}$, la forma si riduce a

$$I_{\text{opt}} \simeq \sqrt{\frac{O}{P_b}},$$

coerentemente con l'approssimazione impiegata nel confronto con il Go-Back-N. Questo risultato riguarda il modello di errori indipendenti adottato qui e non è una proprietà universale del Selective Repeat.

Metodi di accesso

Quando più stazioni condividono la stessa risorsa trasmissiva occorre decidere chi possa usarla e in quale istante. Il problema di accesso al mezzo è quindi un problema di allocazione di una risorsa comune, nel quale efficienza, ritardo, equità e stabilità dipendono dal modo in cui vengono coordinati i tentativi di trasmissione.

In questo capitolo si studiano tre famiglie di meccanismi. Nell'accesso deterministico la risorsa viene suddivisa a priori; nell'accesso controllato il diritto di trasmettere viene assegnato esplicitamente; nell'accesso casuale le stazioni tentano l'accesso senza una preventiva assegnazione e devono gestire le collisioni. TDMA e FDMA, polling, ALOHA e CSMA vengono usati come modelli fondamentali dai quali ricavare prestazioni e limiti.

Per isolare il comportamento del metodo di accesso si assume, salvo indicazione contraria, che il canale non introduca errori fisici. Le perdite considerate nel modello derivano quindi dalla contesa. Questa è un'ipotesi analitica, non una proprietà dei collegamenti reali: se il canale fisico ha errori significativi, il relativo meccanismo di rivelazione, correzione o ritrasmissione deve essere aggiunto al modello e può modificare ritardo e throughput.

4.1 Assunzioni comuni e parametri caratteristici

Si suppone che:

- N sia il numero di stazioni che vogliono accedere al mezzo
- la capacità di trasmissione sia pari a C bit/s
- gli arrivi alla singola stazione formino un processo di Poisson di tasso λ pacchetti/s e i processi delle diverse stazioni siano indipendenti
- F sia la lunghezza dei pacchetti, espressa in bit, che può essere in generale aleatoria.

I parametri caratteristici medi sono:

- S : throughput, definito dal rapporto tra il numero medio di bit, appartenenti ad un pacchetto corretto, che passano in un determinato punto della rete in un secondo, rispetto alla velocità di trasmissione misurabile in quel punto. Il throughput è dunque normalizzato e adimensionale. Coincide con il carico smaltito ρ nei modelli senza sprechi di servizio, ma non con la frazione di tempo fisicamente occupata da collisioni, ritrasmissioni o controllo.
- t_{acc} : tempo medio di accesso, tempo trascorso dall'arrivo del pacchetto ad una stazione fino a quando tutto il pacchetto non è stato ricevuto correttamente dal destinatario.
- G : traffico offerto, definito dal rapporto tra il numero medio dei bit che si prova a trasmettere sul canale in un secondo e la velocità di trasmissione. Questo parametro viene utilizzato nei sistemi a contesa dove più stazioni possono tentare di accedere al canale in modo non disciplinato.

4.2 Accesso deterministico

Si considerano due diverse tecniche di accesso deterministico: Time Division Multiple Access (TDMA) e Frequency Division Multiple Access (FDMA). Nel TDMA l'asse temporale viene diviso in N slot tutti uguali, uno per ogni utente, mentre nell'FDMA è l'asse delle frequenze

che viene diviso in N bande distinte tutte uguali, assegnate ai vari clienti. Si noti che nel sistema TDMA si suppone di trasmettere un singolo pacchetto su ogni slot; questo implica l'utilizzo di pacchetti di lunghezza F costante. Per uniformità di trattazione e per rendere possibile un confronto, questa ipotesi è assunta anche nel caso FDMA.

Si osserva che in questi sistemi l'accesso al singolo canale (slot temporale o slot frequenziale) avviene, in un certo istante, sempre e solo da parte di un singolo utente. Mediamente questo utente, indipendentemente dalla tecnica usata, percepisce una capacità di trasmissione pari a C/N .

Perciò il tasso di servizio medio, riferito ad un singolo utente, risulta:

$$\mu = \frac{C}{NF}$$

e il throughput può essere scritto come:

$$S = \rho = \frac{\lambda}{\mu} = \lambda \frac{NF}{C}$$

dove si è considerato il tasso di arrivo sulla singola stazione λ e il tasso medio di smaltimento percepito da questa stazione.

Poniamo $H = NF/C$ e assumiamo $S = \lambda H < 1$. Per FDMA ogni stazione è esattamente un sistema $M/D/1$ con servizio H , nel modello ideale di capacità C/N costante e senza bande di guardia. Per TDMA il servizio è invece disponibile una sola volta ogni H : non è corretto identificare direttamente questo server periodico con un server sempre disponibile. La derivazione seguente mostrerà che nell'attesa TDMA compare lo stesso contributo algebrico

$$E[T_q^{\text{eq}}] = \frac{\lambda H^2}{2(1-S)} = \frac{SN}{2(1-S)} \frac{F}{C}. \quad (4.1)$$

Per conservare una notazione compatta, nel confronto indichiamo questo contributo con $E[T_q]$; per TDMA l'attesa completa include anche la fase fino allo slot. Applicando Pollaczek-Khinchine al caso FDMA,

$$E[T_q] = \frac{\lambda E[x^2]}{2(1-S)} = \frac{\lambda N^2 F^2}{2C^2(1-S)} = \frac{SN}{2(1-S)} \frac{F}{C} \quad (4.2)$$

4.2.1 Sistema TDMA

In questo sistema quando un utente ha a disposizione il canale lo utilizza completamente, cioè con velocità massima C , per cui il tempo di servizio, condizionato al fatto di avere a disposizione lo slot utile, è $x = F/C$.

Assumiamo che un pacchetto possa iniziare soltanto all'inizio dello slot assegnato e che uno slot vuoto non venga recuperato. Sia L_m il numero di clienti immediatamente prima dello slot della stazione. Fra due suoi slot arrivano $A_m \sim \text{Poisson}(S)$ clienti, indipendentemente da L_m , per cui

$$L_{m+1} = (L_m - 1)^+ + A_m.$$

Il bilancio della media dà $P[L_m > 0] = S$. Elevando al quadrato e usando $E[A_m^2] = S + S^2$ si ricava

$$2(1-S)E[L_m] = 2S - S^2, \quad E[(L_m - 1)^+] = \frac{S^2}{2(1-S)}.$$

Un arrivo poissoniano vede una fase uniforme del periodo H . I clienti in attesa, escluso quello eventualmente scelto per il servizio, sono in media $E[(L_m - 1)^+] + \lambda H/2 = S/[2(1-S)]$. Da

Little segue l'attesa media fino all'inizio del servizio:

$$E[W_{\text{TDMA}}] = \frac{H}{2(1-S)} = \frac{\lambda H^2}{2(1-S)} + \frac{H}{2}.$$

Questo dimostra la scomposizione nel contributo della (4.2) e nel residuo di fase. Quest'ultimo è:

$$E[T_{\text{slot}}] = \frac{NF}{2C}$$

Considerando inoltre il ritardo di propagazione per arrivare a destinazione, τ_p , si ottiene:

$$\begin{aligned} t_{\text{acc}|TDMA} &= E[T_q] + x + E[T_{\text{slot}}] + \tau_p = \frac{SN}{2(1-S)} \frac{F}{C} + \frac{F}{C} + \frac{NF}{2C} + \tau_p = \\ &= \frac{F}{C} \left[\frac{SN}{2(1-S)} + 1 + \frac{N}{2} + \alpha \right] \end{aligned} \quad (4.3)$$

4.2.2 Sistema FDMA

In questo sistema il tempo di servizio risulta $x = FN/C$ in quanto un utente ha sempre a disposizione solo una frazione della capacità complessiva assegnata. Non vi sono attese dovute al sincronismo di slot in quanto l'informazione non è strutturata temporalmente:

$$\begin{aligned} t_{\text{acc}|FDMA} &= E[T_q] + x + \tau_p = \frac{SN}{2(1-S)} \frac{F}{C} + \frac{NF}{C} + \tau_p = \\ &= \frac{F}{C} \left[\frac{SN}{2(1-S)} + N + \alpha \right] \end{aligned} \quad (4.4)$$

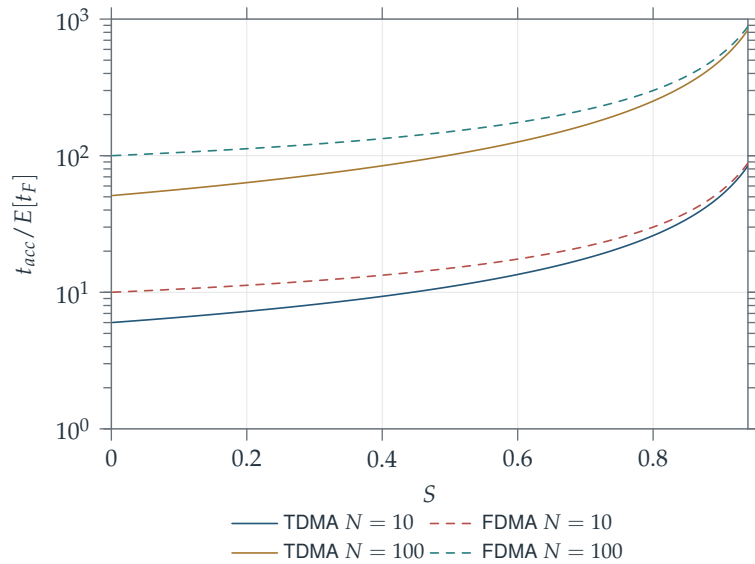


Figura 4.1. Tempo di accesso rispetto al throughput per TDMA e FDMA con pacchetti aventi lunghezza costante.

4.2.3 Confronti TDMA-FDMA

Mettendo a confronto i risultati generali per i sistemi TDMA e FDMA riportati in (4.3) e (4.4) si ha:

$$t_{acc|FDMA} = t_{acc|TDMA} + \frac{F}{C} \left[\frac{N}{2} - 1 \right]$$

e quindi il sistema FDMA ha tempi di accesso superiori rispetto al sistema TDMA per $N > 2$; per $N = 2$ i due modelli forniscono lo stesso valore. Per $N = 1$ il TDMA qui definito conserva comunque il vincolo di iniziare ai confini degli slot e quindi paga ancora $F/(2C)$ di allineamento rispetto a un server continuo. Solo rimuovendo anche quel vincolo i due sistemi coincidono. I grafici di confronto fra le prestazioni del TDMA e del FDMA sono riportati in figura 4.1. È da notare che, dato lo scarso impatto del ritardo di propagazione in questa metodologia di accesso, le figure sono riportate per $\alpha = 0$.

4.3 Accesso controllato

Nel caso dell'accesso controllato il permesso di accedere alla risorsa comune viene fornito sequenzialmente alle varie stazioni tramite il passaggio di un testimone (TOKEN); le stazioni possono occupare il canale solo quando è il loro turno. La distribuzione dell'accesso può essere coordinata da un'entità centralizzata preposta a tale funzione oppure può avvenire sfruttando un meccanismo distribuito in cui sono le stazioni stesse che si passano l'accesso. La sequenza con cui avvengono i vari accessi, o sequenza di polling, può essere variata in modo adattativo per soddisfare particolari richieste di traffico da parte di alcune stazioni.

Nel seguito non si fa distinzione formale tra il caso con accesso distribuito e centralizzato e si fa riferimento ad un polling bilanciato tra tutte le N stazioni.

4.3.1 Tempo di ciclo

Un parametro fondamentale è il tempo di ciclo, t_c , cioè il tempo necessario a servire sequenzialmente una volta tutte le stazioni.

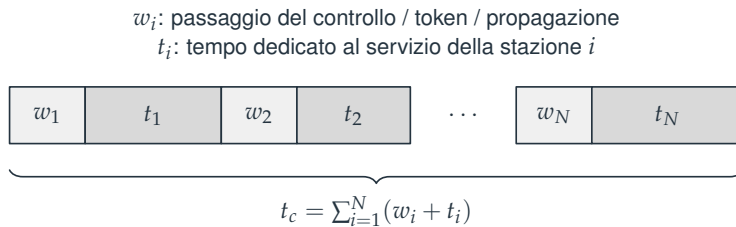


Figura 4.2. Composizione del tempo medio di ciclo.

Si consideri la figura 4.2 in cui il tempo di ciclo è dato dalla successione dei tempi t_i necessari per la trasmissione dei dati e dei tempi w_i necessari per trasferire il controllo da una stazione all'altra.

È interessante osservare che w_i è costituito da diversi contributi tra cui si mette in rilievo: il tempo per la trasmissione del token, il tempo di propagazione del segnale, il tempo per la sincronizzazione della stazione.

Focalizzando l'attenzione su un sistema omogeneo dove $E[w_i] = E[w]$, il numero medio di pacchetti che arrivano nel tempo di ciclo alla singola stazione, risulta:

$$E[v] = E[\lambda]E[t_c] = \lambda E[t_c]$$

Analogamente al caso di accesso deterministico, in un certo istante può accedere alla risorsa solo una singola stazione e quest'ultima, in una condizione di traffico uniforme, percepisce mediamente una capacità pari a C/N .

Il throughput risulta:

$$S = \lambda E[F] \frac{N}{C} = \lambda N E[t_F]$$

ed il tempo medio di servizio:

$$E[t_i] = \frac{E[v]E[F]}{C} = \lambda E[t_c]E[t_F] = \frac{S}{N}E[t_c] \quad (4.5)$$

Il tempo medio di ciclo può essere riscritto come:

$$E[t_c] = \sum_{i=1}^N E[w_i] + \sum_{i=1}^N E[t_i] = NE[w] + SE[t_c]$$

da cui si ottiene infine:

$$E[t_c] = \frac{NE[w]}{1 - S}$$

Si osserva che in condizione di traffico nullo, $S = 0$, si ha $E[t_c] = NE[w]$, cioè il tempo di ciclo è dovuto solo al tempo per il passaggio del controllo.

4.3.2 Tempo di attesa in coda

Nel sistema simmetrico considerato ogni stazione riceve una frazione S/N del carico complessivo e viene servita con disciplina *exhaustive*: quando riceve il controllo, la stazione continua a trasmettere fino a quando la propria coda risulta vuota. Il modello è quindi un polling system ciclico con arrivi di Poisson.

Per valutare il tempo medio di attesa consideriamo un polling ciclico con arrivi poissoniani indipendenti, servizi indipendenti e identicamente distribuiti per ciascuna coda, indipendenti dagli arrivi, e disciplina FIFO non preemptive. I tempi di trasferimento w_i dalla coda i alla successiva sono indipendenti fra loro, dai servizi e dagli arrivi; la loro legge non dipende dallo stato. Il server continua a percorrere il ciclo anche quando tutte le code sono vuote. Supponiamo esistenti le medie stazionarie utilizzate e finiti i secondi momenti di servizi e trasferimenti.

Poniamo $\rho_i = \lambda_i E[B_i]$, $S = \sum_i \rho_i < 1$, $s_i = E[w_i]$, $s = \sum_i s_i > 0$ e $b_2 = \sum_i \lambda_i E[B_i^2]$. Indichiamo con $V(t)$ il lavoro totale ancora da eseguire, espresso in unità di tempo di servizio, e con $E[W_i]$ l'attesa media prima del servizio alla coda i . Per Little, i clienti in attesa contribuiscono con $\sum_i \lambda_i E[W_i] E[B_i]$. Durante un servizio di durata b , l'area sotto il residuo è $\int_0^b (b - u) du = b^2/2$; i servizi della classe i si completano con tasso λ_i . Quindi

$$E[V] = \sum_i \rho_i E[W_i] + \frac{b_2}{2}. \quad (4.6)$$

L'indipendenza fra la durata del servizio di un cliente in attesa e la sua attesa è garantita dalla disciplina scelta; non vale in generale se il server sceglie i clienti in base alla loro dimensione.

Sia C_c la durata di un ciclo e $c = E[C_c]$. Il tempo medio di visita alla coda i è $\rho_i c$: su un numero grande di cicli il lavoro eseguito deve uguagliare quello arrivato. La conservazione del tempo dà

$$c = s + Sc, \quad c = \frac{s}{1 - S}.$$

Per calcolare il costo del polling separiamo gli intervalli di servizio da quelli di trasferimento. Durante il servizio, la derivata di $V(t)^2/2$ è $-V(t)$. All'arrivo di un lavoro B_i , l'incremento è $V(t^-)B_i + B_i^2/2$. In equilibrio, mediando questi contributi,

$$0 = -E[V \mathbf{1}_{\text{servizio}}] + SE[V] + \frac{b_2}{2}.$$

Poiché servizio e trasferimenti partizionano il tempo, ne segue

$$E[V] = \frac{b_2}{2(1-S)} + E[V \mid \text{trasferimento}]. \quad (4.7)$$

Il passaggio si può prima eseguire quando il momento quadratico di V è finito. Per non imporre questa ipotesi aggiuntiva, si applica lo stesso bilancio alla funzione con derivata $\min\{V, L\}$ e poi si manda L a infinito. Gli incrementi sono dominati da $VB_i + B_i^2/2$; le medie finite dichiarate permettono il passaggio al limite. Non si sottraggono quindi due aspettative infinite di V^2 .

Sia V_i^+ il workload all'inizio del trasferimento w_i . Durante tale intervallo il server non esegue lavoro e gli arrivi ne aggiungono in media Su dopo un tempo u . Integrando sul trasferimento e pesando ogni tipo di intervallo per la sua durata si ottiene

$$E[V \mid \text{trasferimento}] = \frac{1}{s} \sum_i \left(s_i E[V_i^+] + \frac{SE[w_i^2]}{2} \right). \quad (4.8)$$

Con servizio exhaustive la coda j è vuota quando termina la sua visita. Alla fine della successiva visita alla coda $i \neq j$, il lavoro accumulato in j è dunque il suo tasso ρ_j moltiplicato per il tempo medio trascorso. Usando intervalli di indici in ordine ciclico, $[j, i)$ comprende i trasferimenti da j fino a quello che precede i , mentre $(j, i]$ comprende le visite successive a j fino alla visita a i . Pertanto

$$E[V_i^+] = \sum_{j \neq i} \rho_j \left(\sum_{k \in [j, i)} s_k + c \sum_{k \in (j, i]} \rho_k \right).$$

I termini possono essere raccolti in coppie non ordinate:

$$\sum_i s_i E[V_i^+] = S \sum_{i < j} s_i s_j + sc \sum_{i < j} \rho_i \rho_j.$$

Per verificare la prima somma, si fissi la coda j e si leggano i trasferimenti a partire da j : ciascuna coppia di trasferimenti appare una sola volta, con il secondo successivo al primo. Per la seconda, si fissi i e si leggano le code nell'ordine ciclico che termina in i : ogni coppia di carichi compare una sola volta. Questi conteggi spiegano i coefficienti senza assumere indipendenza fra le durate delle visite.

Poiché i trasferimenti sono indipendenti, $E[T_s^2] = \sum_i E[w_i^2] + 2 \sum_{i < j} s_i s_j$, dove $T_s = \sum_i w_i$. Sostituendo nella (4.8),

$$E[V \mid \text{trasferimento}] = \frac{SE[T_s^2]}{2s} + \frac{s}{2(1-S)} \left(S^2 - \sum_i \rho_i^2 \right).$$

La (4.7) fornisce quindi

$$E[V] = \frac{\sum_i \lambda_i E[B_i^2]}{2(1-S)} + \frac{SE[T_s^2]}{2E[T_s]} + \frac{E[T_s]}{2(1-S)} \left(S^2 - \sum_i \rho_i^2 \right). \quad (4.9)$$

Sottraendo il servizio residuo $b_2/2$ della (4.6),

$$\sum_i \rho_i E[W_i] = \frac{S}{2(1-S)} \sum_i \lambda_i E[B_i^2] + \frac{SE[T_s^2]}{2E[T_s]} + \frac{E[T_s]}{2(1-S)} \left(S^2 - \sum_i \rho_i^2 \right). \quad (4.10)$$

La stessa dimostrazione permette un'estensione che servirà per l'accesso 1-limited. Se la visita a j lascia un lavoro medio $E[Z_{jj}]$ nella propria coda, al termine della visita a i esso si aggiunge agli arrivi avvenuti nel frattempo. Ogni $E[V_i^+]$ contiene quindi il termine aggiuntivo $\sum_j E[Z_{jj}]$. Nella (4.8) il suo coefficiente è $\sum_i s_i/s = 1$: il membro destro della (4.10) va aumentato di $\sum_j E[Z_{jj}]$. Per exhaustive questi residui sono nulli.

Nel caso simmetrico $\lambda_i = \lambda$, $B_i = t_F$, $\rho_i = S/N$ e $E[W_i] = E[T_q]$. Dividendo la (4.10) per S si ricava la formula usata per il modello del capitolo.

Sia

$$T_s = \sum_{i=1}^N w_i$$

il tempo complessivo speso, durante un ciclo, per trasferire il controllo fra le stazioni. Nel caso generale T_s può essere una variabile aleatoria. Per un sistema simmetrico con servizio exhaustive, specializzando la relazione di conservazione appena derivata si ottiene

$$E[T_q] = \frac{N\lambda E[t_F^2]}{2(1-S)} + \frac{E[T_s^2]}{2E[T_s]} + \frac{E[T_s]S}{2(1-S)} \left(1 - \frac{1}{N}\right), \quad (4.11)$$

dove $t_F = F/C$ e $S = N\lambda E[t_F] < 1$.

Il primo termine della (4.11) coincide con il contributo che si avrebbe nella corrispondente coda $M/G/1$ ottenuta accorpendo il carico delle N stazioni:

$$E[T_{q2}] = \frac{N\lambda E[t_F^2]}{2(1-S)} = \frac{N\lambda E[F^2]}{2C^2(1-S)} = \frac{S}{C} \frac{E[F^2]}{2E[F](1-S)}. \quad (4.12)$$

I due termini restanti rappresentano il costo introdotto dal polling e dai trasferimenti del controllo. Il termine

$$\frac{E[T_s^2]}{2E[T_s]}$$

è il tempo residuo medio associato al tempo complessivo di trasferimento del controllo; compare quindi il secondo momento e non, in generale, soltanto la metà del valore medio.

Nel caso particolare usato nei grafici di questo capitolo si assume che ogni trasferimento richieda un tempo deterministico w e che le N stazioni siano identiche. Allora $T_s = Nw$ e la parte della (4.11) dovuta al polling diventa

$$\begin{aligned} E[T_{q1}] &= \frac{Nw}{2} + \frac{NwS}{2(1-S)} \left(1 - \frac{1}{N}\right) \\ &= \frac{(1-S/N)Nw}{2(1-S)}. \end{aligned} \quad (4.13)$$

Questa è quindi una conseguenza esatta del modello simmetrico con servizio exhaustive e tempi di trasferimento deterministici; non viene usata come approssimazione.

Se invece i tempi di trasferimento sono aleatori, la (4.11) mostra esplicitamente l'effetto della loro variabilità. Infatti

$$\frac{E[T_s^2]}{2E[T_s]} = \frac{E[T_s]}{2} + \frac{\text{Var}(T_s)}{2E[T_s]},$$

per cui, a parità di valore medio, una maggiore varianza di T_s aumenta il tempo medio di attesa.

Si noti che nel caso di accesso deterministico TDMA, il tempo medio di attesa dello slot utile è stato valutato considerando la scansione di tutti gli N slot. Nel polling exhaustive,

invece, una stazione che ha già il controllo può servire anche i pacchetti che arrivano durante la visita, fino allo svuotamento della propria coda. È questa diversa disciplina di servizio, insieme al tempo di trasferimento del controllo, a determinare la struttura della (4.11).

Ponendo ora $w = t_A + \tau_p$, con t_A tempo di trasmissione del controllo, nel caso deterministico si ottiene

$$E[T_q] = E[t_F] \left[\frac{(1 - S/N)N}{2(1 - S)}(\beta + \alpha) + \frac{S}{2(1 - S)} \frac{E[F^2]}{E^2[F]} \right],$$

dove $\beta = t_A/E[t_F]$ e $\alpha = \tau_p/E[t_F]$.

4.3.3 Tempo di accesso

Il tempo di accesso è dato dalla somma del tempo di attesa in coda, del tempo medio di trasmissione e del ritardo di propagazione. Si osserva che il tempo di propagazione figura anche all'interno del termine $E[w]$ in quanto è essenziale per la determinazione del tempo di ciclo.

$$t_{acc} = E[T_q] + E[t_F] + \tau_p = E[t_F] \left[\frac{(1 - \frac{S}{N})N}{2(1 - S)}(\beta + \alpha) + \frac{S}{2(1 - S)} \frac{E[F^2]}{E^2[F]} + 1 + \alpha \right] \quad (4.14)$$

Nel caso di lunghezza deterministica o esponenzialmente distribuita dei pacchetti, il tempo di accesso risulta:

$$t_{acc|det} = t_F \left[\frac{(1 - \frac{S}{N})N}{2(1 - S)}(\beta + \alpha) + \frac{S}{2(1 - S)} + 1 + \alpha \right]$$

$$t_{acc|exp} = E[t_F] \left[\frac{(1 - \frac{S}{N})N}{2(1 - S)}(\beta + \alpha) + \frac{S}{(1 - S)} + 1 + \alpha \right]$$

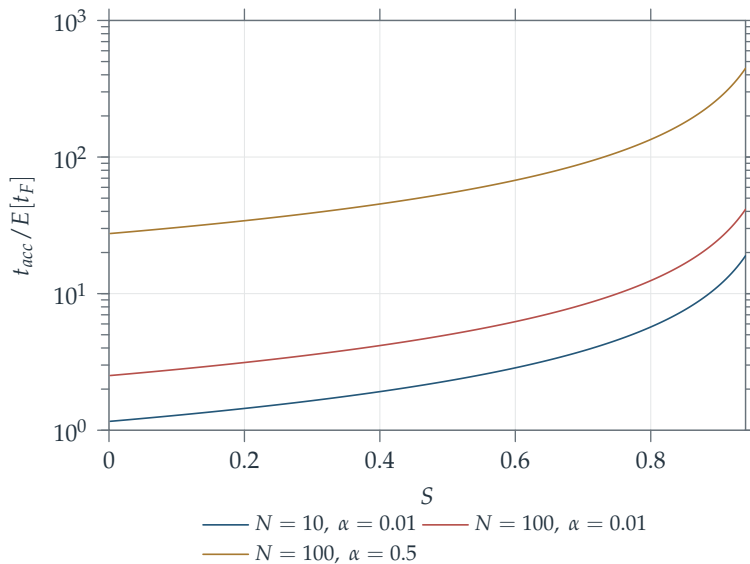


Figura 4.3. Tempo di accesso rispetto al throughput per sistema con accesso controllato con pacchetti aventi lunghezza distribuita esponenzialmente.

Le differenze tra i tempi di accesso per il controllo distribuito e quello centralizzato ricadono sui diversi tempi di propagazione e sulla diversa lunghezza del pacchetto di controllo e quindi

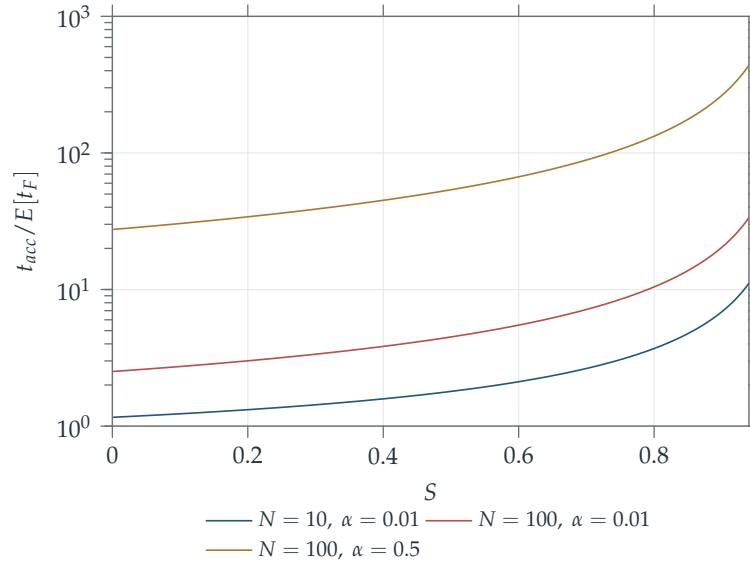


Figura 4.4. Tempo di accesso rispetto al throughput per sistema con accesso controllato con pacchetti aventi lunghezza costante.

riguardano esclusivamente i termini $E[w]$ e τ_p . Le figure 4.3 e 4.4 riportano gli andamenti dei tempi di accesso in funzione del throughput per sistemi con pacchetti aventi lunghezza distribuita esponenzialmente e costante, avendo fissato $\beta = 0.02$ e facendo variare, per alcuni casi, il ritardo di propagazione normalizzato α .

4.4 Accesso casuale

Nei sistemi ad accesso casuale le stazioni operano indipendentemente. Non esiste un controllore che assegni preventivamente il canale: ogni stazione decide quando tentare la trasmissione e il protocollo gestisce le eventuali collisioni. ALOHA è il modello di riferimento più semplice per studiare quantitativamente questo comportamento.

4.4.1 Sistema ALOHA

La filosofia di funzionamento dell'Aloha è la seguente: quando un utente ha un pacchetto da trasmettere lo trasmette indipendentemente da quello che stanno facendo gli altri utenti. Anche considerando un canale libero da errori, si possono comunque verificare degli errori a causa delle potenziali sovrapposizioni temporali tra i pacchetti trasmessi; è allora necessario introdurre dei pacchetti di ack per confermare la corretta ricezione delle informazioni e prevedere un meccanismo di timeout nel caso in cui i pacchetti vengano persi.

Dopo una collisione le stazioni possono trovarsi in condizioni simili. Se ritrasmettessero con gli stessi ritardi deterministici, potrebbero mantenere la sincronizzazione e collidere nuovamente. Non è però vero che ogni ritrasmissione immediata debba necessariamente collidere in qualunque configurazione asincrona. Si inserisce allora un ritardo nella ritrasmissione del pacchetto, ritardo che solitamente viene scelto casualmente dalla stazione trasmittente. Il pacchetto da ritrasmettere viene tenuto in uno stato di sospensione dalla stazione trasmittente, stato che solitamente viene detto di backoff. Si intuisce comunque che in questo tipo di sistema si possono avere situazioni di instabilità dove i pacchetti continuano a collidere.

Si suppone che tutti i pacchetti contenenti informazioni abbiano una lunghezza deterministica pari ad F , mentre si assume che i pacchetti di ack, di lunghezza A , siano caratterizzati da $A \ll F$, in modo da poter considerare trascurabile il loro apporto nel traffico generato e nel

ritardo introdotto. Il modello trascura quindi la contesa degli ACK; ciò può rappresentare sia un canale di controllo separato sia un overhead sullo stesso mezzo sufficientemente piccolo da non modificare le grandezze considerate.

Seguendo le definizioni di throughput e di traffico offerto si ha:

$$S = N\lambda \frac{F}{C} = N\lambda t_F$$

$$G = N\lambda' \frac{F}{C} = N\lambda' t_F$$

L'identità $S = N\lambda t_F$ richiede un regime stazionario in cui ogni pacchetto ammesso sia infine consegnato e non vi siano abbandoni: λ deve contare i pacchetti freschi ammessi, non richieste rifiutate prima dell'accesso. In generale S va definito direttamente dal tasso delle consegne riuscite.

I pacchetti ritrasmessi vanno ad incrementare il traffico ed a priori formano, con i pacchetti di nuova generazione, un processo che non è più di Poisson dato che una ritrasmissione è direttamente correlata ad una trasmissione o ritrasmissione precedente. Per ottenere il modello classico trattabile si approssima il processo aggregato dei tentativi, nuove trasmissioni più ritrasmissioni, con un processo di Poisson. Questa è un'ipotesi di decoupling, non un risultato esatto: il backoff riduce le correlazioni tra tentativi e rende l'approssimazione più plausibile, ma la sua accuratezza dipende dalla regola di ritrasmissione e dal regime di carico. Nella espressione del traffico offerto il tasso di arrivo, λ' , è dovuto alla somma degli arrivi dei nuovi pacchetti e dei pacchetti da ritrasmettere, quindi $\lambda' \geq \lambda$.

Throughput

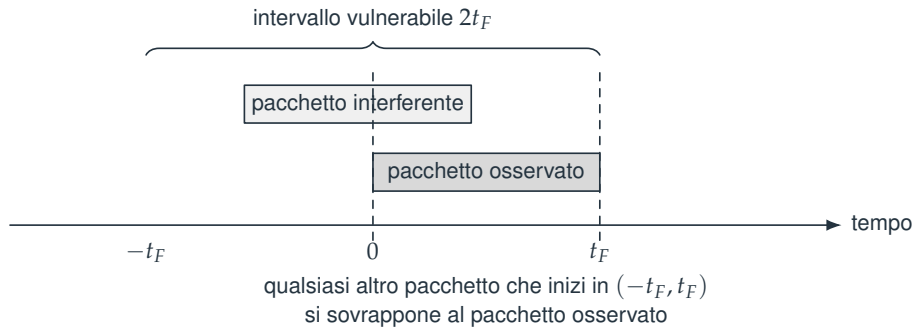


Figura 4.5. Esempio di collisione in un sistema Aloha.

Considerando i pacchetti a lunghezza costante, il rapporto S/G rappresenta la frazione dei pacchetti trasmessi senza collisioni. In figura 4.5 è riportato un esempio di collisione. Per non avere collisioni occorre che non vi sia nessun arrivo in un intervallo di estensione $2t_F$ e dato che il processo degli arrivi è modellato come un processo di Poisson, si ha:

$$P_0(2t_F) = e^{-N\lambda'2t_F} = e^{-2G} = \frac{S}{G}$$

Si ottiene quindi una formula molto semplice che lega il throughput al traffico offerto:

$$S = Ge^{-2G} \quad (4.15)$$

Il massimo throughput si ottiene quando:

$$\frac{dS}{dG} = e^{-2G}(1 - 2G) = 0 \Rightarrow G = \frac{1}{2} \Rightarrow S_{max} = \frac{1}{2e} \simeq 0.184.$$

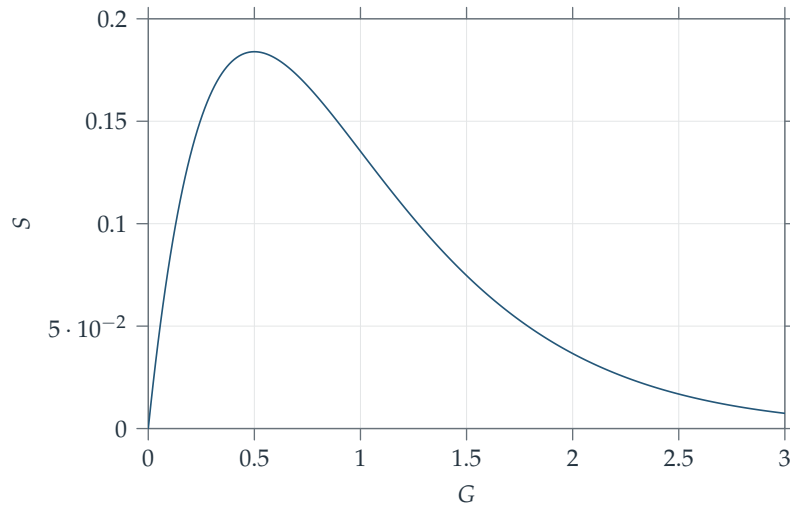


Figura 4.6. Throughput sistema Aloha.

In figura 4.6 viene riportata la curva del throughput in funzione del traffico offerto.

Tempo di accesso

In generale lo studio analitico del tempo di accesso nelle metodologie casuali è piuttosto complicato, di seguito si cerca di proporre una analisi semplificata e sufficientemente intuitiva.

In questa analisi il tempo parte quando il pacchetto diventa il prossimo candidato alla trasmissione presso il livello MAC e termina alla sua ricezione corretta. Non include quindi l'attesa dietro ad altri pacchetti nella coda della sorgente. Il tempo complessivo dalla generazione si ottiene aggiungendo tale attesa, se il modello della sorgente la prevede. Escluderla dal cronometro non equivale a introdurre un buffer di un solo pacchetto con scarti: quest'ultima scelta cambia anche gli arrivi ammessi e il traffico.

Date le definizioni di G ed S si ha che G/S rappresenta il numero di trasmissioni necessario per consegnare al destinatario un pacchetto corretto, quindi il numero medio di tentativi per pacchetto consegnato è G/S e il numero medio di ritrasmissioni è quindi

$$\frac{G}{S} - 1.$$

Questa identità è una relazione di conservazione tra tentativi e successi in regime stazionario, con pacchetti di uguale lunghezza e senza scarti definitivi. Non richiede che ogni pacchetto subisca lo stesso numero di collisioni. Con un limite ai tentativi, il rapporto include anche il lavoro consumato dai pacchetti abbandonati e non dà da solo il numero medio di tentativi condizionato alla consegna.

In generale il tempo di accesso può essere calcolato valutando il tempo speso in ogni ritrasmissione, moltiplicato per il numero delle ritrasmissioni e sommato al tempo speso per l'ultima (corretta) trasmissione.

Nel caso dell'Aloha prima di ogni ritrasmissione a seguito della verifica tramite timeout della distruzione di un pacchetto, viene inserito un ritardo t_R di backoff che ha lo scopo di rendere stabile il meccanismo di accesso. La trasmissione di ogni pacchetto impone un impegno temporale pari a $t_F + \tau_p$. Inoltre, considerando che si assume trascurabile il tempo per la trasmissione dell'ack, è sufficiente mettere in conto un tempo di timeout superiore a solo due volte il ritardo di propagazione, cioè $t_o = 2\tau_p$. Ogni ritrasmissione necessita di un tempo pari a:

$$t_F + t_o + t_R \geq t_F + 2\tau_p + t_R$$

Dalla formula del throughput dell'Aloha si ottiene immediatamente il numero delle trasmissioni necessarie per consegnare correttamente il pacchetto: $G/S = e^{2G}$. L'algoritmo di backoff può operare scegliendo il tempo di ritrasmissione in modo casuale seguendo una distribuzione uniforme nell'intervallo $[0, Kt_F]$, con media $E[t_R] = K t_F/2$; complessivamente il tempo di accesso risulta:

$$\begin{aligned} t_{acc} &= (t_F + 2\tau_p + E[t_R]) \left(\frac{G}{S} - 1 \right) + t_F + \tau_p = \\ &= t_F \left[\left(1 + 2\alpha + \frac{K}{2} \right) (e^{2G} - 1) + 1 + \alpha \right] \end{aligned}$$

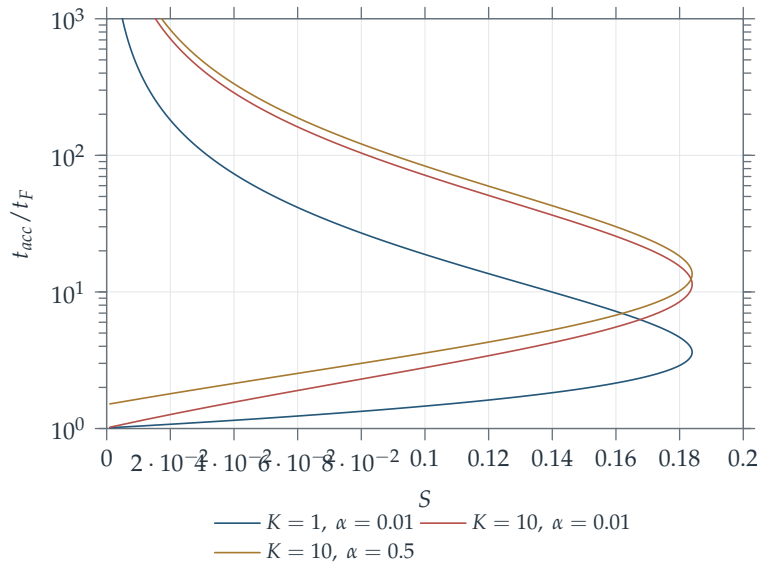


Figura 4.7. Tempo di accesso in funzione del throughput per il sistema Aloha.

La relazione (4.15) non è globalmente iniettiva e quindi non ammette un'unica inversa reale. Tuttavia può essere invertita ramo per ramo mediante la funzione di Lambert W :

$$G = -\frac{1}{2} W_k(-2S), \quad k \in \{0, -1\},$$

per $0 < S \leq 1/(2e)$. I due rami reali corrispondono rispettivamente alla regione a basso traffico e a quella oltre il punto di massimo del throughput. In figura 4.7 il tempo di accesso viene comunque riportato in forma parametrica, che rende visibili entrambi i rami. È interessante osservare che all'aumentare di K aumenta il ritardo, ma il sistema è potenzialmente più stabile, anche se questo tipo di analisi esula dagli scopi di questa trattazione. Inoltre all'aumentare di K il sistema risente meno del ritardo di propagazione.

4.4.2 Slotted Aloha

Per aumentare il throughput del sistema Aloha, che è stato dimostrato essere al più 0.18, occorre inserire nel sistema una qualche sorta di coordinamento.

Una prima idea consiste nel sincronizzare tra loro tutte le stazioni in modo da dividere l'asse dei tempi in slot di lunghezza pari a quella della trasmissione di un singolo pacchetto t_F . Quindi una stazione può iniziare la trasmissione solo all'inizio di uno slot e viene così dimezzato il tempo in cui può avvenire una collisione. Un sistema di questo tipo viene detto SLOTTED ALOHA.

Throughput

Nel modello slotted gli inizi di trasmissione sono concentrati ai confini degli slot; non formano quindi un processo di Poisson a tempo continuo. Si approssima invece il numero J di tentativi nello slot con una variabile di Poisson di media G . Lo slot consegna un pacchetto se e solo se $J = 1$, quindi $S = P[J = 1] = Ge^{-G}$. Equivalentemente, la frazione dei tentativi riusciti è $S/G = e^{-G}$:

$$P_0(t_F) = e^{-N\lambda' t_F} = e^{-G} \Rightarrow \frac{S}{G} = e^{-G} \Rightarrow S = Ge^{-G} \quad (4.16)$$

Si ottiene un legame tra il throughput ed il traffico offerto analogo a quello ricavato per l'Aloha (4.15) dove però l'esponente non presenta il fattore due. Lo slotted Aloha permette di ottenere un throughput massimo doppio rispetto all'Aloha.

Infatti:

$$\frac{dS}{dG} = e^{-G}(1 - G) = 0 \Rightarrow G = 1 \Rightarrow S_{max} = e^{-1} \simeq 0.368$$

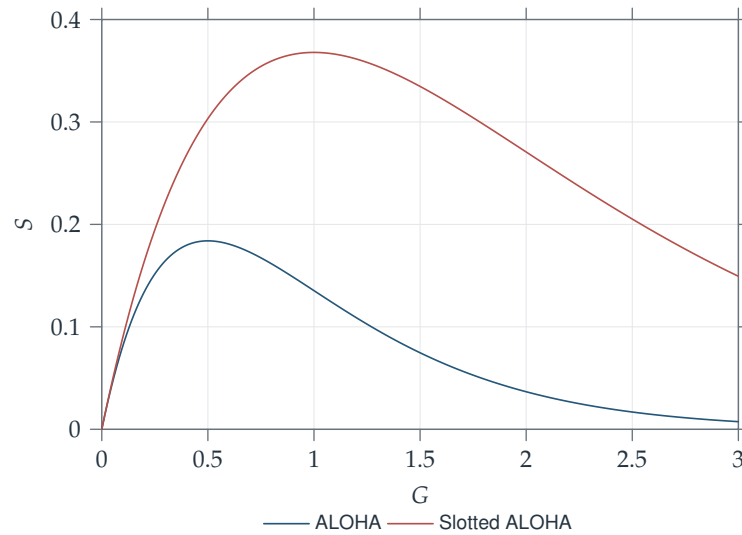


Figura 4.8. Throughput per sistema Aloha e Slotted Aloha.

In figura 4.8 vengono riportate le curve del throughput in funzione del traffico offerto per i sistemi Aloha e Slotted Aloha.

La valutazione classica del throughput, sia nel caso Aloha che Slotted Aloha, è basata sull'ipotesi di traffico di Poisson. Nel caso di una popolazione finita si può ottenere una forma che converge alla stessa espressione dello Slotted Aloha assumendo, slot per slot, tentativi indipendenti tra le N stazioni e uguale probabilità di trasmissione. Non è quindi necessario assumere direttamente un processo di Poisson a popolazione finita, ma resta essenziale l'ipotesi di indipendenza dei tentativi.

Infatti, coerentemente con la definizione di throughput e di traffico offerto, nel caso in cui tutte le N stazioni siano indipendenti ed abbiano lo stesso carico, $S_i = S/N$ è la probabilità che un pacchetto venga trasmesso correttamente dalla stazione i e $G_i = G/N$ è la probabilità che la stazione i acceda ad uno slot. La probabilità che la generica stazione j riesca a trasmettere con successo dipende dal fatto che quando prova a trasmettere tutte le altre stazioni rimangono in silenzio:

$$S_j = G_j \prod_{i=1, i \neq j}^N (1 - G_i)$$

Quindi:

$$S = G \left(1 - \frac{G}{N}\right)^{N-1}$$

La relazione vale per $0 \leq G \leq N$ e tende alla (4.16) per $N \rightarrow \infty$ a G fissato. La derivazione del massimo interno che segue assume $N \geq 2$; con una sola stazione $S = G$ e il massimo è 1, ottenuto trasmettendo in ogni slot. È possibile valutare il throughput massimo:

$$\frac{dS}{dG} = \left(1 - \frac{G}{N}\right)^{N-1} - G \left(1 - \frac{G}{N}\right)^{N-2} \frac{N-1}{N} = 0 \Rightarrow G = 1 \Rightarrow S_{max} = \left(1 - \frac{1}{N}\right)^{N-1}$$

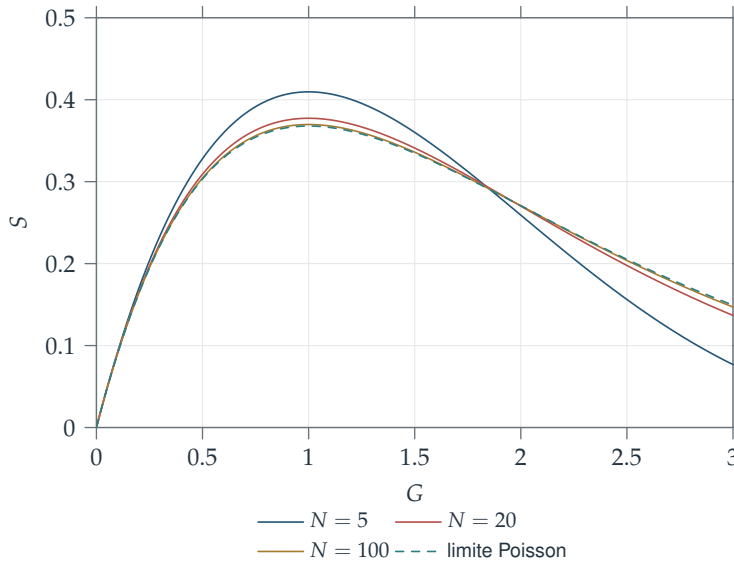


Figura 4.9. Throughput Slotted Aloha con popolazione finita.

In figura 4.9 vengono riportati gli andamenti del throughput al variare del numero delle stazioni che compongono la popolazione. A titolo di confronto viene anche riportato il caso con popolazione infinita.

Tempo di accesso

L'analisi del tempo di accesso è analoga a quella svolta per il sistema Aloha ma occorre considerare che il sincronismo di slot vincola l'inizio della trasmissione o della ritrasmissione.

Nel modello di ritardo usato per il confronto, ogni nuovo tentativo o ritentativo può iniziare soltanto al confine dello slot successivo. Se l'istante in cui il pacchetto diventa pronto è uniforme rispetto alla trama, il residuo medio fino al confine è $t_F/2$. Si adotta inoltre la convenzione conservativa di sincronizzare alla stessa trama anche il ciclo di riscontro che precede una ritrasmissione. Questa è una scelta del modello: con un canale ACK separato e asincrono il relativo termine di allineamento non sarebbe presente. Si approssimano inoltre le fasi come uniformi e indipendenti: con propagazione deterministica e backoff in un numero intero di slot queste fasi possono essere correlate. La formula seguente è quindi il bilancio del modello di ritardo adottato, non una conseguenza esatta del solo throughput:

$$\begin{aligned}
t_{acc} &= (2t_F + 2\tau_p + E[t_R]) \left(\frac{G}{S} - 1 \right) + t_F + \frac{t_F}{2} + \tau_p = \\
&= t_F \left[\left(2 + 2\alpha + \frac{K}{2} \right) (e^G - 1) + \frac{3}{2} + \alpha \right]
\end{aligned}$$

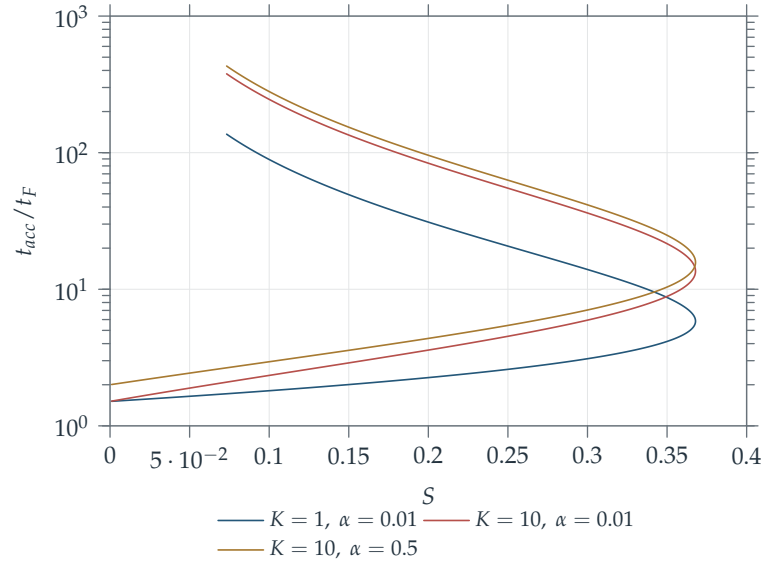


Figura 4.10. Tempo di accesso in funzione del throughput per il sistema Slotted Aloha.

In figura 4.10 viene riportato il tempo di accesso in funzione del throughput al variare di α e di K , ottenuto in modo parametrico. Valgono le stesse considerazioni fatte per il caso Aloha.

Confrontando le figure 4.7 e 4.10 si osserva che per bassi valori di throughput, dove le collisioni risultano poco probabili ed il traffico offerto tende a coincidere con il throughput, lo Slotted Aloha ha prestazioni più scadenti dell'Aloha, in termini di tempo di accesso, a causa del tempo che perdono i pacchetti per sincronizzarsi con la trama degli slot.

4.4.3 Instabilità

Un elemento importante nella valutazione delle prestazioni di un sistema ad accesso casuale è la sua stabilità. In questa sezione vengono riportate alcune considerazioni sulla stabilità di un sistema Slotted Aloha, considerazioni che possono comunque essere estese anche ad altri protocolli casuali.

Si consideri di avere N stazioni, capaci di gestire un solo pacchetto alla volta. Quando un pacchetto trova il sistema occupato viene scartato e viene successivamente riproposto tramite i meccanismi forniti dai livelli superiori. Sulla singola stazione si ipotizza che il processo di nascita sia di Poisson e si definisce

$$p = 1 - e^{-\lambda t_F}$$

come probabilità che arrivi almeno un pacchetto durante uno slot. Per $\lambda t_F \ll 1$ vale l'approssimazione $p \simeq \lambda t_F$.

Nel funzionamento classico dello Slotted Aloha una stazione attende un numero casuale di slot prima di provare a ritrasmettere un pacchetto colliso; nel seguito, per semplicità, si assume che un pacchetto colliso venga ritrasmesso con probabilità χ in ogni slot successivo, fino a quando la trasmissione non avviene con successo.

Lo stato del sistema può essere descritto mediante il numero delle stazioni che stanno effettuando ritrasmissioni, k . Nello stato k indichiamo con r_k il numero aleatorio di ritrasmissioni nello slot, con $0 \leq r_k \leq k$, e con n_k il numero aleatorio di nuovi tentativi, con $0 \leq n_k \leq N - k$. Le rispettive medie sono $E[r_k] = k\chi$ ed $E[n_k] = (N - k)p$. Dato che su uno slot si ha una trasmissione corretta solo se viene trasmesso un solo pacchetto, indipendentemente dal fatto che si tratti di una nuova trasmissione o di una ritrasmissione, la probabilità di avere una trasmissione corretta nello stato k risulta:

$$\begin{aligned}\Pi_k &= P\{n_k = 1\}P\{r_k = 0\} + P\{n_k = 0\}P\{r_k = 1\} = \\ &= (N - k)p(1 - p)^{N-k-1}(1 - \chi)^k + (1 - p)^{N-k}k\chi(1 - \chi)^{k-1}\end{aligned}$$

dove, secondo la distribuzione binomiale utilizzabile data l'indipendenza degli arrivi, $P\{n_k = x\} = \binom{N-k}{x}p^x(1 - p)^{N-k-x}$ e $P\{r_k = x\} = \binom{k}{x}\chi^x(1 - \chi)^{k-x}$ rappresentano rispettivamente le probabilità di avere x arrivi in uno slot nel caso di nuovi pacchetti o di ritrasmissioni.

La probabilità Π_k coincide con il numero medio di successi per slot nello stato k , essendo possibile al più una trasmissione corretta per slot. Il termine $S_k = (N - k)p$ rappresenta invece il numero medio di nuovi pacchetti ammessi nello slot dalle stazioni non in ritrasmissione. Ogni nuovo pacchetto aumenta di uno la popolazione pendente e ogni successo la diminuisce di uno. Nel modello finito questo fornisce la deriva condizionata esatta

$$E[k_{n+1} - k_n \mid k_n = k] = S_k - \Pi_k.$$

Interpretarne gli zeri come punti di equilibrio di un sistema continuo è invece un'approssimazione deterministica alla catena di Markov. Nel caso di un elevato numero di stazioni e ricordando che $p \ll 1$, si ha $P\{n_k = x\} \approx S_k^x / x! e^{-S_k}$, quindi:

$$\Pi_k = S_k e^{-S_k} (1 - \chi)^k + e^{-S_k} k \chi (1 - \chi)^{k-1}$$

Il sistema è in equilibrio quando il tasso di uscita e quello di ingresso si eguagliano, cioè $\Pi_k = S_k$. La curva del tasso di successi Π_k è riportata in figura 4.11 per $N = 200$, $p = 0.001$ e $\chi = 0.02, 0.03, 0.05$. Sotto la curva di equilibrio il tasso di uscita è maggiore del tasso di ingresso e quindi nello slot successivo il numero delle stazioni in ritrasmissione tende a calare. Sopra alla curva risulta $\Pi_k < S_k$, cioè il carico supera la capacità di sistema. Al crescere di k la curva di equilibrio tende a decrescere a causa dell'elevato numero di ritrasmissioni. Con il calare di χ l'area sotto la curva tende a crescere e quindi aumenta la stabilità del sistema, che è in grado di smaltire il carico anche con un elevato numero di stazioni in ritrasmissione. D'altra parte aumenta il numero medio di slot necessari per consegnare correttamente il pacchetto e quindi aumenta il tempo medio di accesso.

Una volta fissati N e p , che caratterizzano il traffico generato dal sistema, la rete è in una condizione di carico stazionario che permette di studiare la stabilità del sistema mediante la definizione di una retta di carico.

La retta di carico è rappresentata dalla funzione S_k nel piano (k, S_k) . Un esempio di retta di carico e di curva di equilibrio è mostrato in figura 4.12 dove si è considerato $N = 150$, $p = 0.001$ e $\chi = 0.035$. Le intersezioni tra la retta di carico e la curva di equilibrio sono punti in cui il tasso di ingresso ed il tasso di uscita sono uguali e quindi rappresentano possibili punti di lavoro o di equilibrio del sistema. Tali punti possono essere **stabili** o **instabili**.

Per i parametri della figura 4.12, l'interpolazione della deriva mostra un solo zero vicino a $k = 0.80$. Non è esattamente $k = 0$: per $N > 1$, in $k = 0$ possono già collidere due nuovi pacchetti e $S_0 - \Pi_0 = Np[1 - (1 - p)^{N-1}] > 0$. Sotto lo zero la deriva è positiva, sopra è negativa, perciò il punto è attrattivo nel modello medio. La popolazione reale resta intera e fluttua; questo zero non è uno stato assorbente della catena.

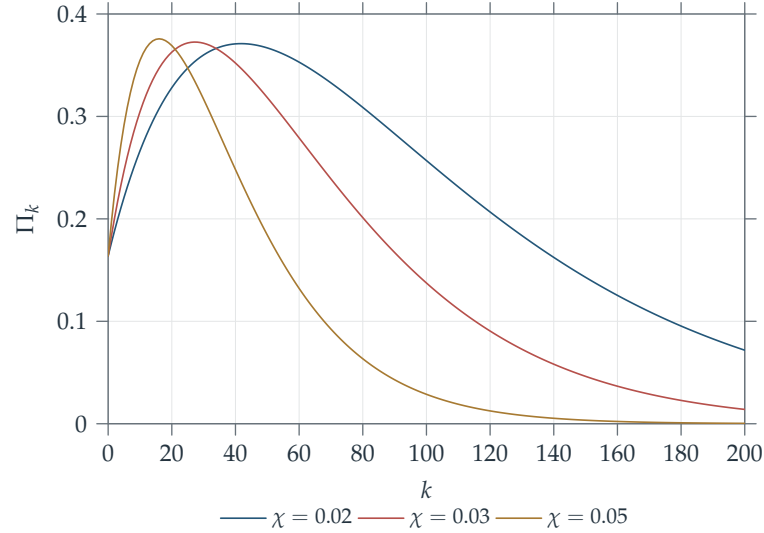


Figura 4.11. Tasso medio di successi per slot in funzione del numero di stazioni in ritrasmissione al variare di $\chi = 0.02, 0.03, 0.05$, con $N = 200$ e $p = 0.001$.

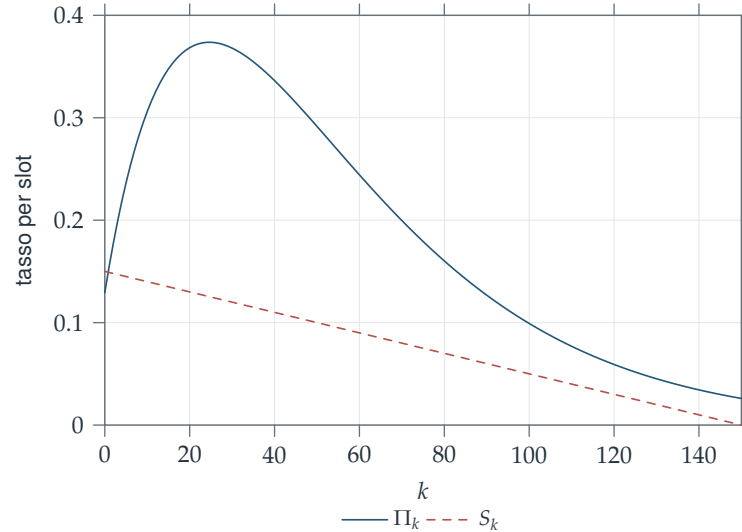


Figura 4.12. Curva di equilibrio e retta di carico in funzione del numero di stazioni in ritrasmissione, con $N = 150$, $p = 0.001$ e $\chi = 0.035$.

Per $N = 200$, $p = 0.001$ e $\chi = 0.035$, figura 4.13, la stessa interpolazione fornisce tre zeri circa in $k = 1.59, 97.06$ e 192.77 . Il segno della deriva passa rispettivamente da positivo a negativo, da negativo a positivo e ancora da positivo a negativo: il primo e il terzo sono attrattivi, quello centrale è repulsivo nel modello deterministico. Le regioni attrattive hanno prestazioni molto diverse: nella prima pochi pacchetti sono pendenti, nell'ultima quasi tutte le stazioni competono nelle ritrasmissioni.

Con $N = 450$, figura 4.14, lo zero rimasto è prossimo a 450: il tasso di consegna è molto piccolo, ma non è matematicamente nullo. La denominazione "instabilità" descrive qui la perdita di un regime operativo efficiente, non la divergenza di una coda illimitata. Infatti $0 \leq k \leq N$ e, per $N \geq 2$ e $0 < p, \chi < 1$, la catena finita può raggiungere tutti gli stati e dispone di una distribuzione stazionaria. Le fluttuazioni possono farla passare fra regioni a bassa e alta congestione. L'eventuale instabilità di code esterne alle stazioni richiederebbe un modello

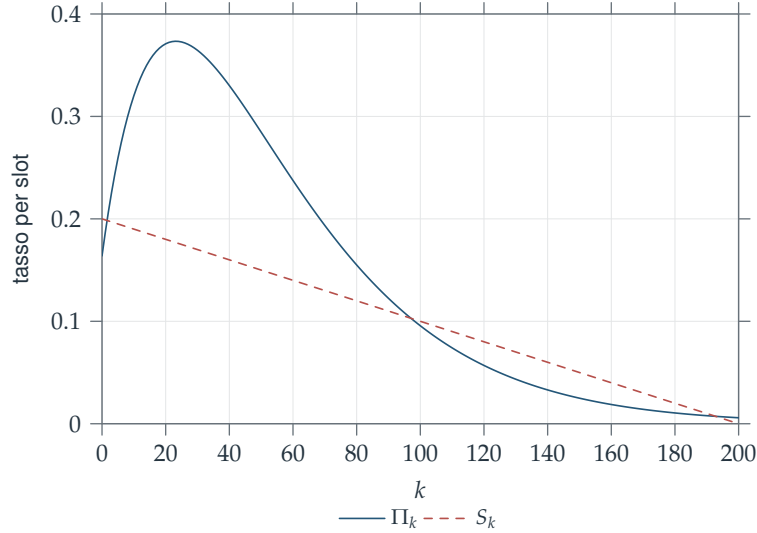


Figura 4.13. Curva di equilibrio e retta di carico in funzione del numero di stazioni in ritrasmissione, con $N = 200$, $p = 0.001$ e $\chi = 0.035$.

aggiuntivo.

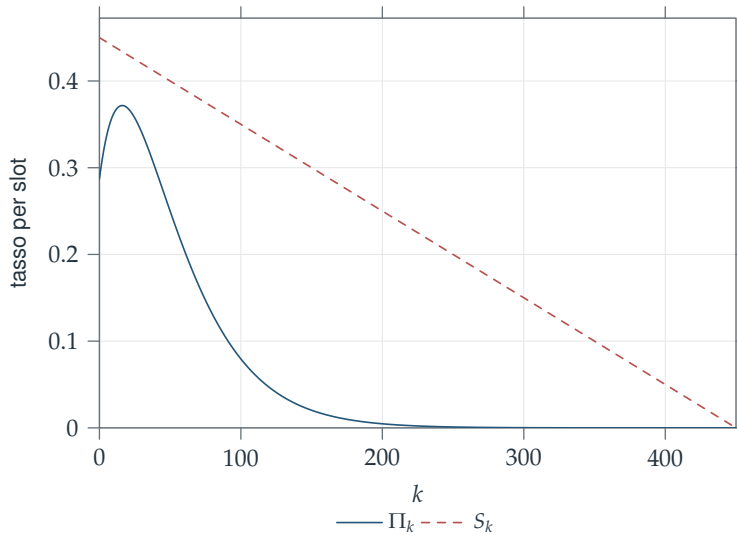


Figura 4.14. Curva di equilibrio e retta di carico in funzione del numero di stazioni in ritrasmissione, con $N = 450$, $p = 0.001$ e $\chi = 0.035$.

4.4.4 Effetto cattura

Supponendo che i livelli di potenza ricevuti da due diverse stazioni possano essere fortemente differenti, situazione tipica di una rete radio, è possibile che il pacchetto ricevuto con potenza maggiore, pur andando a collidere con quello a potenza minore, sia ricevuto correttamente. Questa particolare circostanza viene detta EFFETTO CATTURA.

Adottiamo un modello geometrico di esclusione: per una sorgente a distanza r , un tentativo riesce se non vi sono interferenti attivi entro il raggio ar , con $a \geq 1$. L'effetto aggregato degli interferenti esterni è trascurato. È un'approssimazione a soglia, non una proprietà generale della propagazione radio. Le quantità $G(r)$ e $S(r)$ saranno densità spaziali di traffico normalizzato, per unità di area, non i carichi totali G ed S delle sezioni precedenti. Si assume

un processo poissoniano di tentativi nello spazio e nel tempo, radialmente simmetrico; $N(r)$ indica la densità di stazioni. La figura 4.15 fissa la geometria.

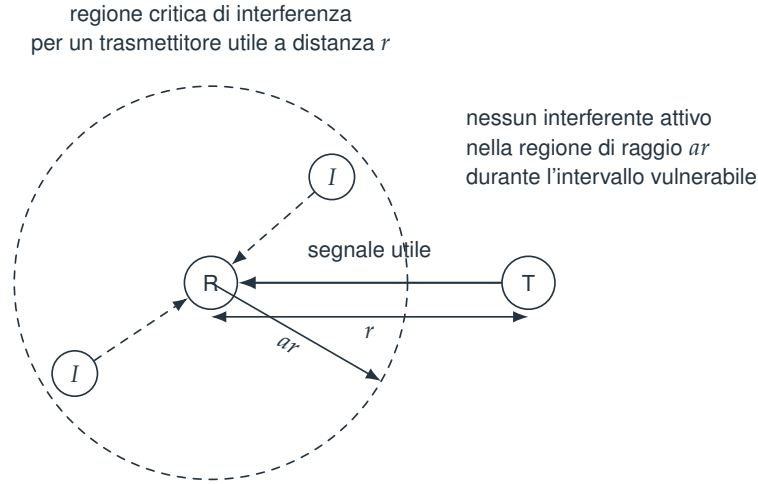


Figura 4.15. Geometria di riferimento per l'analisi dell'effetto cattura.

Il throughput ed il traffico offerto vengono assunti funzione della posizione radiale, considerando la geometria circolare del sistema di riferimento. Il traffico offerto, sull'anello di spessore dr , generato dalle stazioni a distanza r è: $G(r)2\pi r dr$. Essendo i pacchetti a lunghezza costante, il rapporto $S(r)/G(r)$ rappresenta la frazione di pacchetti trasmessi da stazioni a distanza r che arrivano corretti.

Prendendo come riferimento il sistema Aloha si ha che i pacchetti spediti da stazioni a distanza r arrivano corretti se non si hanno trasmissioni, nell'intervallo temporale $2t_F$, da parte di stazioni all'interno dell'area di raggio ar . Quindi:

$$\frac{S(r)}{G(r)} = P[0 \text{ tentativi in } 2t_F \text{ entro il raggio } ar] = \quad (4.17)$$

$$e^{-2 \int_0^{ar} 2\pi x N(x) \lambda'(x) t_F dx} \Rightarrow S(r) = G(r) e^{-4\pi \int_0^{ar} x G(x) dx}$$

L'equazione (4.17) è esatta per il processo poissoniano e la regola geometrica appena assunti. Il campo $G(x)$ deve essere definito anche oltre la regione di cui si misura il throughput. Se esistono stazioni soltanto in un disco di raggio R , l'estremo superiore dell'integrale va invece sostituito con $\min\{ar, R\}$. Un'equazione simile può essere trovata per il caso Slotted Aloha, qualora si consideri un intervallo di vulnerabilità pari a t_F invece che $2t_F$. Di seguito vengono riportate due possibili specializzazioni della (4.17) ai casi con traffico offerto costante e throughput costante.

Traffico offerto costante

Con l'ipotesi $G(r) = G_0$ la (4.17) diventa:

$$S(r) = G_0 e^{-4\pi \int_0^{ar} x G_0 dx} = G_0 e^{-2\pi G_0 a^2 r^2}$$

ed il throughput assume una forma gaussiana rispetto ad r , il cui andamento è motivabile osservando che i pacchetti lontani dal ricevitore sono ricevuti con più collisioni rispetto a quelli vicini. Integrando sull'area circoscritta dalla circonferenza di raggio R si ottiene il throughput complessivo per il sistema prodotto dalle stazioni nell'area considerata, mantenendo il campo uniforme di interferenti anche all'esterno del disco:

$$S_R = \int_0^R S(x) 2\pi x dx = G_0 \int_0^R e^{-2\pi G_0 a^2 x^2} 2\pi x dx = \frac{1}{2a^2} (1 - e^{-2\pi G_0 a^2 R^2})$$

Considerando, al limite, tutto lo spazio:

$$S_\infty = \frac{1}{2a^2}$$

che tende a $\frac{1}{2}$ quando $a = 1$. Il sistema con $a = 1$ è detto a cattura perfetta ed ha un throughput $S_\infty = 0.5$ superiore al massimo del modello ALOHA senza cattura. Il valore 0.5 è però un limite del modello su un'area crescente con traffico offerto totale illimitato, non una prestazione garantita a carico finito o in una rete radio qualunque.

Throughput costante

Per studiare il sistema a throughput costante si procede differenziando l'equazione (4.17):

$$S'(r) = G'(r) e^{-4\pi \int_0^{ar} xG(x)dx} + G(r) e^{-4\pi \int_0^{ar} xG(x)dx} [-4\pi ar G(ar)a]$$

e sfruttando ancora la (4.17) si ha:

$$S'(r) = G'(r) \frac{S(r)}{G(r)} + G(r) \frac{S(r)}{G(r)} [-4\pi a^2 r G(ar)] \Rightarrow \quad (4.18)$$

$$S'(r)G(r) = G'(r)S(r) - 4\pi r a^2 S(r)G(r)G(ar)$$

L'equazione (4.18) costituisce, al pari della (4.17), una descrizione analitica per il sistema Aloha con effetto cattura. Ponendo $S(r) = S_0$ nell'equazione differenziale (4.18), si ha:

$$G'(r) = 4\pi r a^2 G(r)G(ar)$$

Se il sistema è a cattura perfetta ($a = 1$), si ha:

$$G'(r) = 4\pi r G^2(r)$$

che, considerando la condizione iniziale $G(0) = S_0$, ammette soluzione:

$$G(r) = \frac{S_0}{1 - 2\pi r^2 S_0}$$

Nella regione di convergenza, cioè quella in cui $G(r) > 0$, definita da $r < (2\pi S_0)^{-1/2} = R$, si ha che $G(r)$ cresce al crescere di r . Si osserva infine che il throughput medio, nella regione di convergenza, risulta:

$$S_R = \int_0^R S(r) 2\pi r dr = \pi R^2 S_0 = \pi \frac{1}{2\pi S_0} S_0 = \frac{1}{2}$$

Il valore $1/2$ si intende come limite da raggi inferiori a $R = (2\pi S_0)^{-1/2}$. Infatti, per $r_0 < R$, il traffico offerto complessivo necessario è

$$G_{r_0} = 2\pi \int_0^{r_0} \frac{x S_0}{1 - 2\pi S_0 x^2} dx = -\frac{1}{2} \log(1 - 2\pi S_0 r_0^2),$$

che diverge quando $r_0 \uparrow R$. Con traffico finito il throughput integrato resta strettamente minore di $1/2$.

Note sul parametro a di cattura

La possibilità di ricevere correttamente un pacchetto quando questo collide, totalmente o parzialmente, con un altro è conseguenza delle diverse potenze con cui i pacchetti vengono ricevuti. Un'idea quantitativa può essere ottenuta considerando un sistema DPSK ed analizzando la collisione tra due pacchetti, assumendo assente il rumore termico e trattando l'interferenza alla stregua di rumore secondo l'ipotesi gaussiana.

Per DPSK binaria con rivelazione differenziale a due simboli, rumore gaussiano bianco complesso e fase del canale costante sui due simboli, la relazione pertinente usa il rapporto fra energia per bit e densità spettrale del rumore: $P_b = \frac{1}{2}e^{-E_b/N_0}$. La si può derivare considerando due simboli uguali $r_0 = \sqrt{E_b} + n_0$, $r_1 = \sqrt{E_b} + n_1$, con rumori complessi indipendenti di varianza N_0 . Le variabili $u = (r_1 + r_0)/\sqrt{2}$ e $v = (r_1 - r_0)/\sqrt{2}$ sono indipendenti, con medie $\sqrt{2E_b}$ e zero e la stessa varianza N_0 . L'errore $\text{Re}(r_1 r_0^*) < 0$ equivale a $|v|^2 > |u|^2$. Poiché $|v|^2$ è esponenziale di media N_0 ,

$$\begin{aligned} P_b &= E[e^{-|u|^2/N_0}] \\ &= \frac{1}{\pi N_0} \int_{\mathbb{C}} e^{-(|u|^2 + |u - \sqrt{2E_b}|^2)/N_0} d^2u = \frac{1}{2}e^{-E_b/N_0}. \end{aligned}$$

L'ultimo passaggio completa il quadrato: $|u|^2 + |u - m|^2 = 2|u - m/2|^2 + |m|^2/2$.

Se W è invece il rapporto fra potenza utile e potenza interferente $P_I = N_0 B_{eq}$, e R_b è il bit rate, allora $E_b/N_0 = (B_{eq}/R_b)W$. Nel seguito si adotta esplicitamente la normalizzazione $B_{eq}/R_b = 1$ e si tratta l'interferenza come rumore gaussiano equivalente. Solo sotto questa ipotesi si scrive $P_b = \frac{1}{2}e^{-W}$. Supponiamo di considerare I interferenti e inoltre che le potenze trasmesse P_i siano uguali per tutte le stazioni. Allora:

$$W = \frac{\frac{P_t}{A_u}}{\sum_{i=1}^I \frac{P_t}{A_i}}$$

dove A_u è l'attenuazione del segnale utile e A_i quella del segnale interferente prodotto dalla stazione i . Supponendo che l'attenuazione sia proporzionale a r^γ , che la stazione utile sia sulla circonferenza di raggio r e gli interferenti disposti sulla circonferenza di raggio ar si ha che:

$$W = \frac{(ar)^\gamma}{I r^\gamma} = \frac{a^\gamma}{I}$$

Quindi:

$$P_b = \frac{1}{2}e^{-\frac{a^\gamma}{I}} \quad (4.19)$$

Si consideri ora una collisione parziale. Nel modello semplificato si fissa una sovrapposizione di $F/2$ bit del pacchetto utile, con F pari; sui restanti bit si trascura il rumore termico. La quantità P_b della (4.19) è quindi la probabilità di errore condizionata ai bit effettivamente interessati dall'interferenza.

Assumendo inoltre indipendenti gli errori sui bit coinvolti, condizionatamente a questa sovrapposizione, la probabilità di errore nel modello a sovrapposizione fissata è

$$P_e = 1 - (1 - P_b)^{F/2}.$$

Se la sovrapposizione fosse invece un numero aleatorio M di bit, la relazione sarebbe $P_e = 1 - E[(1 - P_b)^M]$, non in generale $1 - (1 - P_b)^{E[M]}$. Anche l'indipendenza degli errori sui bit è

un'ulteriore idealizzazione, distinta dal modello di rivelazione a due simboli. Per $FP_b \ll 1$ si ottiene l'approssimazione del primo ordine

$$P_e \simeq \frac{F}{2}P_b = \frac{F}{4}e^{-a^\gamma/I},$$

da cui, nel dominio $P_e/F < 1/4$ in cui il logaritmo è positivo,

$$a \simeq \left[I \ln \frac{F}{4P_e} \right]^{1/\gamma}.$$

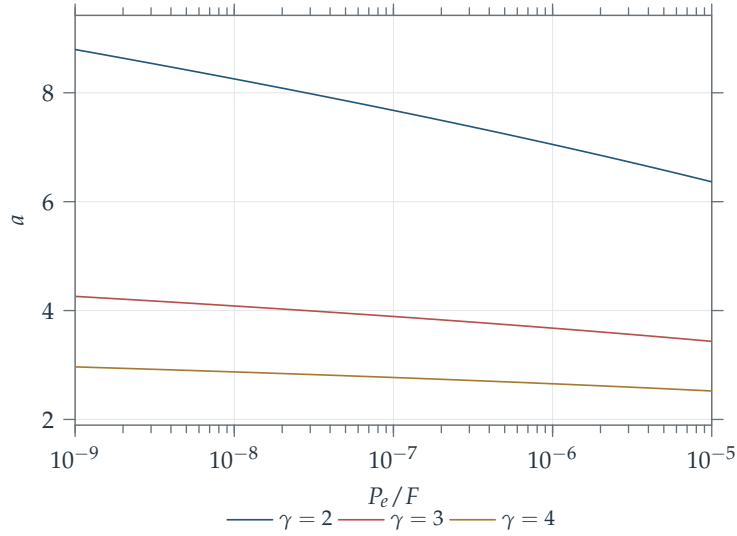


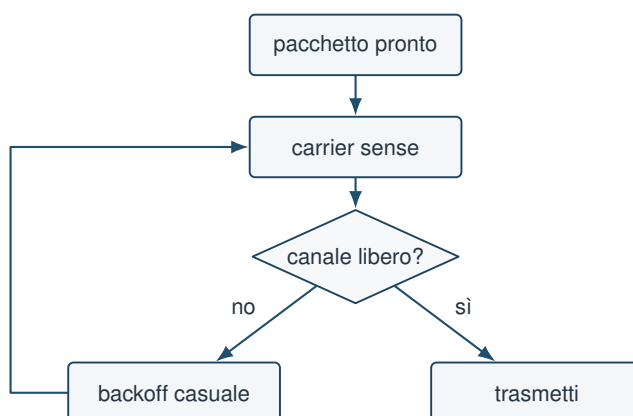
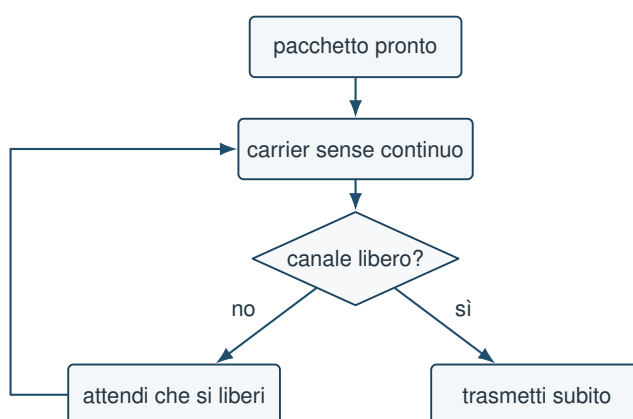
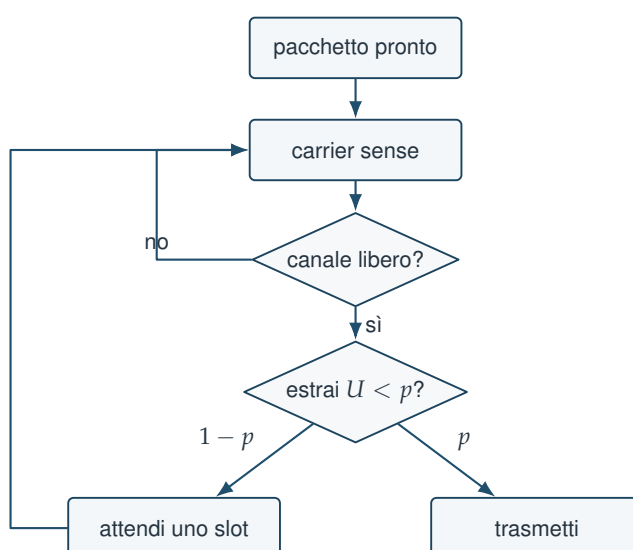
Figura 4.16. Parametro di cattura al variare del rapporto tra la probabilità che il pacchetto sia errato e la lunghezza del pacchetto.

In figura 4.16 è riportato l'andamento del parametro di cattura a al variare del rapporto P_e/F per tre diversi valori di γ , con $I = 4$ interferenti e $F = 1000$ bit. Il dominio mostrato, $10^{-9} \leq P_e/F \leq 10^{-5}$, corrisponde a $10^{-6} \leq P_e \leq 10^{-2}$ e mantiene piccola la probabilità di pacchetto usata nella linearizzazione. Non si estende l'approssimazione fino a rapporti incompatibili con queste ipotesi.

4.4.5 Carrier Sense Multiple Access, CSMA

Rispetto al sistema Aloha si cerca di coordinare le stazioni attraverso un test preposto a verificare lo stato di occupazione del canale prima della trasmissione. In questo modo si evita la collisione con i pacchetti che stanno occupando il canale. La possibilità che si verifichi una collisione, comunque, rimane a causa dei ritardi di propagazione. Esistono tre algoritmi base di CSMA, così riassumibili:

- **NON PERSISTENTE**, come riportato in figura 4.17, dove si attende che il canale sia libero ed immediatamente si procede trasmettendo il pacchetto; altrimenti si pone il pacchetto nello stato di backoff, al termine del quale si procede verificando nuovamente il canale.
- **1-PERSISTENTE**, come riportato in figura 4.18, dove si trasmette non appena il canale risulta libero e per ogni pacchetto in attesa viene testato continuamente il canale in attesa di trovarlo libero.
- **p -PERSISTENTE**, come riportato in figura 4.19, dove il canale viene testato in continuazione ed ogni volta che risulta libero si trasmette con probabilità p ; altrimenti si attende

**Figura 4.17.** CSMA non persistente.**Figura 4.18.** CSMA 1-persistente.**Figura 4.19.** CSMA p-persistente.

un determinato tempo (ad esempio pari al massimo ritardo di propagazione, necessario al fine che tutte le stazioni avvertano una nuova trasmissione) e si ripete la procedura. È evidente che il caso 1-persistente si ottiene come caso particolare di questo schema quando $p = 1$.

Throughput CSMA non persistente

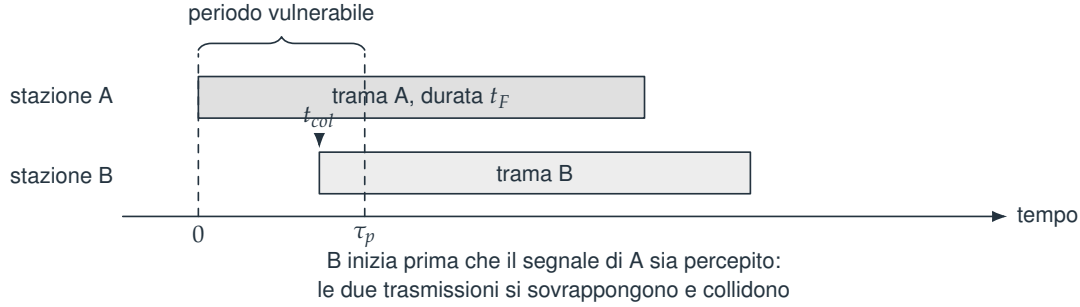


Figura 4.20. Tempistiche relative al CSMA non persistente.

Si consideri la figura 4.20 relativa alla collisione tra i pacchetti di due diversi utenti. t_{col} è il tempo in cui inizia la collisione e se ci sono più pacchetti che collidono si fa riferimento all'ultimo arrivato. Il periodo di vulnerabilità, definito come il periodo in cui una stazione ha emesso un pacchetto ma questo non viene ancora sentito da un'altra stazione, è pari al ritardo di propagazione. Una collisione può avvenire solo se si ha l'arrivo di un nuovo pacchetto nel periodo di vulnerabilità. Il caso senza collisione è immediatamente derivabile dalla figura 4.20, ponendo $t_{col} = 0$ e considerando un solo utente.

In questa analisi λ' conta le prove di accesso: un evento che trova il canale occupato viene rinviato senza emettere una trama. Quindi $G = N\lambda't_F$ normalizza le prove di sensing e non soltanto i tentativi effettivamente trasmessi. Il processo delle prove è approssimato come poissoniano; la propagazione è rappresentata da un unico ritardo τ_p , non dalla matrice reale delle distanze.

Sia t_U il tempo utile medio per ciclo, cioè la durata di una trasmissione riuscita moltiplicata per la probabilità di successo:

$$t_U = t_F P[\text{no trasmissioni in } \emptyset_p] = t_F P_0(\tau_p) = t_F e^{-N\lambda'\tau_p} = t_F e^{-N\lambda't_F \frac{\tau_p}{t_F}} = t_F e^{-G\alpha} \quad (4.20)$$

Si noti che dopo la trasmissione del pacchetto, esiste sempre un intervallo di tempo in cui il canale è libero, t_I , in quanto il sensing del canale, ogni volta in cui questo è occupato, viene ripetuto dopo un tempo di attesa casuale (backoff). Quindi, la probabilità che si trasmetta immediatamente dopo la fine di un'altra trasmissione risulta un evento con probabilità nulla.

t_I coincide con il tempo medio interarrivo, distribuito in modo esponenziale e per la sua valutazione si considera la teoria sviluppata nel paragrafo 2.5.6, finalizzata al calcolo dei momenti di una distribuzione esponenziale. Considerando che il tasso di arrivo complessivo è pari a $N\lambda'$, si ha:

$$t_I = \frac{1}{N\lambda'} = \frac{t_F}{G} \quad (4.21)$$

La determinazione statistica dell'estensione della collisione va interpretata come una distribuzione mista. Con probabilità $e^{-G\alpha}$ non avviene alcuna collisione e si pone $t_{col} = 0$. Per

$0 < t_{col} < \tau_p$ la parte continua ha funzione cumulativa complessiva e densità

$$F_{col}(t_{col}) = P[T_{col} \leq t_{col}] = e^{-N\lambda'(\tau_p - t_{col})} = e^{-G\left(\alpha - \frac{t_{col}}{t_F}\right)},$$

$$f_{col}(t_{col}) = \frac{G}{t_F} e^{-G\left(\alpha - \frac{t_{col}}{t_F}\right)}.$$

Si noti che $F_{col}(0) = e^{-G\alpha}$ è proprio la massa di probabilità concentrata in zero.

Quindi l'aspettazione di t_{col} risulta:

$$E[t_{col}] = \int_0^{\tau_p} t f_{col}(t) dt = \frac{G}{t_F} e^{-G\alpha} \int_0^{\tau_p} t e^{G \frac{t}{t_F}} dt = e^{-G\alpha} \left[\left[t e^{G \frac{t}{t_F}} \right]_0^{\tau_p} - \int_0^{\tau_p} e^{G \frac{t}{t_F}} dt \right] =$$

$$= e^{-G\alpha} \left[\tau_p e^{G\alpha} - \frac{t_F}{G} e^{G \frac{\tau_p}{t_F}} + \frac{t_F}{G} \right] = \tau_p - \frac{t_F}{G} (1 - e^{-G\alpha})$$

Il tempo medio in cui avviene la collisione risulta:

$$E[t_{col}] = t_F \left[\alpha - \frac{1 - e^{-G\alpha}}{G} \right]$$

Il tempo t_B , in cui il sistema è occupato da trasmissioni corrette o collisioni, ha espressione:

$$t_B = t_F + \tau_p + E[t_{col}] = t_F \left[1 + 2\alpha - \frac{1}{G} (1 - e^{-G\alpha}) \right]$$

Alla luce delle grandezze determinate, il throughput del canale è dato dal rapporto tra il tempo utile, t_U , ed il tempo di ciclo, costituito dal tempo in cui il canale è libero, t_I , e dal tempo in cui il canale è occupato, t_B ; complessivamente:

$$S = \frac{t_U}{t_B + t_I} = \frac{e^{-\alpha G}}{\left[1 + 2\alpha - \frac{1}{G} + \frac{e^{-\alpha G}}{G} \right] + \frac{1}{G}} = \frac{G e^{-\alpha G}}{G(1 + 2\alpha) + e^{-\alpha G}}$$

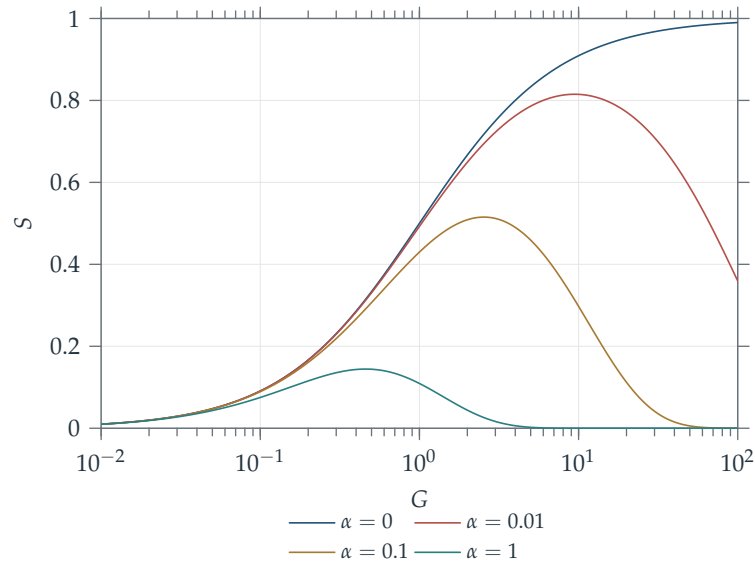


Figura 4.21. Throughput CSMA non persistente.

In figura 4.21 viene riportato il throughput in funzione del traffico offerto al variare di α .

Si osserva che quando $\alpha = 0$ il throughput risulta:

$$S(\alpha = 0) = \frac{G}{1 + G} \rightarrow 1 \text{ con } G \rightarrow \infty$$

Quindi con questa metodologia di accesso casuale si può arrivare a sfruttare pienamente il canale.

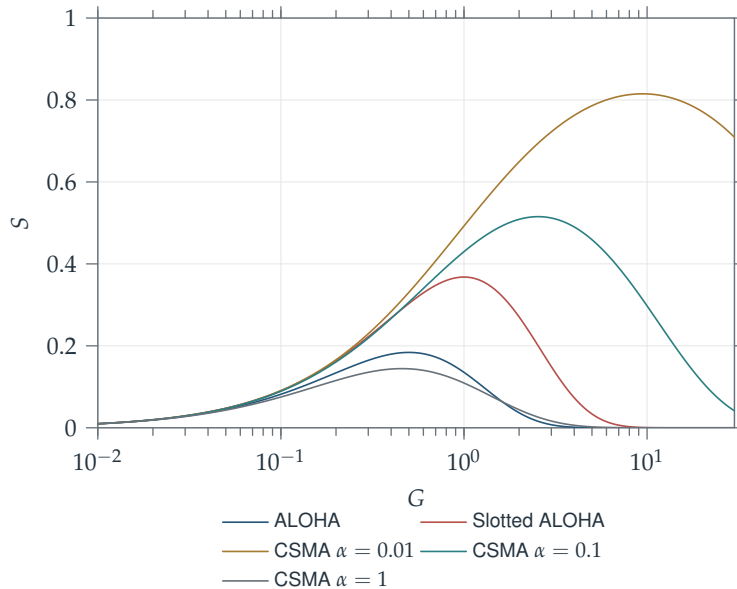


Figura 4.22. Throughput per sistema Aloha, Slotted Aloha e CSMA non persistente.

Il sistema CSMA non persistente è stato valutato a titolo di esempio. In figura 4.22 è mostrato il throughput in funzione del traffico offerto relativo alle tecniche ad accesso casuale considerate (Aloha, Slotted Aloha, CSMA non persistente), al variare di α .

È possibile valutare in modo analogo, ma con maggiori difficoltà analitiche, il caso CSMA 1-persistente ed in generale il caso CSMA p -persistente. Si osserva che anche per il sistema CSMA è possibile prevedere una implementazione slotted, che a fronte di una maggiore complessità del sistema, consente di ottenere migliori prestazioni.

Infine, si vogliono mettere in evidenza due effetti tipici delle reti radio basate su un carrier sense: il problema del TERMINALE NASCOSTO e del TERMINALE ESPOSTO.

Questi due effetti, contrariamente all'effetto cattura, determinano un peggioramento delle prestazioni comportando un calo del throughput, rispetto al caso ideale appena trattato analiticamente.

Si faccia riferimento alla figura 4.23. Il terminale A sta trasmettendo verso B. Il terminale C è anch'esso in grado di raggiungere B, ma A e C sono fuori dalla reciproca area di carrier sense. C può quindi percepire il canale libero e iniziare una trasmissione verso B, provocando una collisione al ricevitore. Il problema non nasce dal solo ritardo di propagazione: nasce dal fatto che due trasmettitori che interferiscono sullo stesso ricevitore non riescono a rilevarsi reciprocamente.

Il problema del terminale esposto è illustrato in figura 4.24. B sta trasmettendo verso A, mentre C vorrebbe trasmettere verso D. C sente la trasmissione di B e rinvia il proprio accesso, sebbene D sia fuori dall'area di interferenza di B e A sia fuori da quella di C. Le due trasmissioni potrebbero quindi avvenire contemporaneamente senza collisione: il carrier sense risulta in questo caso eccessivamente conservativo.

Per effettuare, nel caso di rete radio, un'analisi accurata del throughput in funzione del

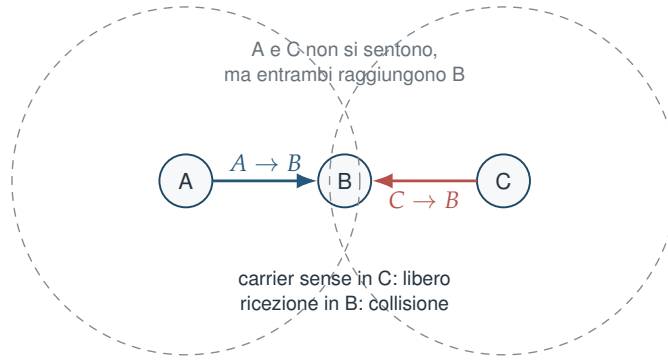


Figura 4.23. Problema del terminale nascosto.

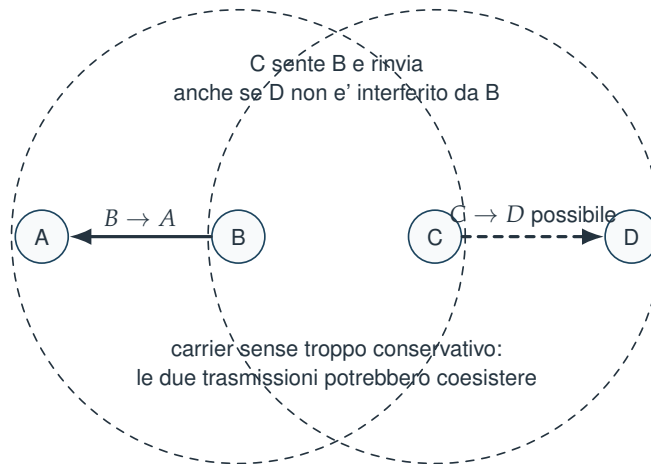


Figura 4.24. Problema del terminale esposto.

traffico offerto occorre modellare i vari effetti che si sovrappongono: problemi dei terminali nascosti ed esposti, che determinano un calo di S , ed effetto cattura, che determina un incremento di S . Un modello quantitativo completo deve quindi specificare insieme geometria, propagazione, carrier sense, cattura e processo di traffico. Il modello ideale precedente serve a isolare il ruolo del protocollo; non va interpretato come previsione universale di una rete radio reale.

Tempo di accesso CSMA non persistente

Per la valutazione del tempo di accesso si segue l'approccio proposto nella sezione 4.4.1 per l'Aloha. Nel CSMA non persistente si trasmette solo quando il canale è libero, mentre si innesca una procedura di backoff ogni volta che il canale risulta occupato.

Si procede valutando la probabilità che il canale risulti occupato, P_o , osservando che nel periodo t_B i primi τ_p secondi vengono rivelati liberi dal meccanismo di sensing; sulla base delle grandezze determinate per throughput si ha:

$$P_o = \frac{t_B - \tau_p}{t_B + t_I} = \frac{1 + \alpha - \frac{1}{G}(1 - e^{-\alpha G})}{1 + 2\alpha + \frac{e^{-\alpha G}}{G}}$$

Nel CSMA non persistente il backoff è presente ogni volta che il canale è occupato, quindi prima di poter effettuare un qualsiasi accesso occorre attendere un determinato tempo, qui indicato con t_{lib} . Si noti che questo tempo rappresenta l'intervallo temporale da quando il pacchetto è arrivato al livello di accesso per essere trasmesso, a quando si trova il canale libero.

Detto t_R il tempo di backoff, la valutazione di t_{lib} può essere compiuta in modo analogo a quanto visto per le ritrasmissioni dei casi ARQ nel capitolo 3, cioè definendo $t_i = it_R$ e $P_i = P_o^i(1 - P_o)$, quindi:

$$t_{lib} = \sum_{i=0}^{\infty} t_i P_i = E[t_R](1 - P_o) \sum_{i=0}^{\infty} i P_o^i = E[t_R] \frac{P_o}{1 - P_o}$$

La somma geometrica appena usata assume che i successivi istanti di sensing vedano stati indipendenti con la medesima probabilità P_o . È una chiusura approssimata: una scelta di backoff non rende automaticamente indipendenti gli stati osservati.

Il tempo di backoff può essere scelto, analogamente al caso analizzato per l'Aloha, uniformemente nell'intervallo $[0, Kt_F]$ con una media $E[t_R] = \frac{K}{2}t_F$. Non tutte le prove conteggiate da G producono una trama: la frazione che trova il canale libero è $1 - P_o$. Il carico effettivamente trasmesso è quindi $G_{tx} = G(1 - P_o)$ e il numero medio di trasmissioni per consegna è

$$R_{tx} = \frac{G(1 - P_o)}{S} = (1 + \alpha G)e^{\alpha G}.$$

L'ultima identità segue sostituendo le espressioni di P_o e S . Usare G/S come numero di trasmissioni conterebbe erroneamente anche le prove rinviate dal carrier sense, già pagate in t_{lib} .

Manteniamo ACK di durata trascurabile su un ritorno senza contesa, come nel modello ALOHA. Con timeout stringente misurato dalla fine trama, un fallimento costa $t_{lib} + t_F + 2\tau_p + E[t_R]$; l'ultima trasmissione costa $t_{lib} + t_F + \tau_p$. Definendo $L = t_{lib}/t_F = (K/2)P_o/(1 - P_o)$, il modello di ritardo è dunque

$$\frac{t_{acc}}{t_F} \simeq \left(L + 1 + 2\alpha + \frac{K}{2} \right) (R_{tx} - 1) + L + 1 + \alpha. \quad (4.22)$$

Per $\alpha = 0$ si ha $R_{tx} = 1$: non vi sono collisioni e rimangono soltanto attesa di accesso e trasmissione. Un ACK che contendesse sullo stesso mezzo richiederebbe invece di includerne anche il traffico nel modello, non soltanto aggiungere un secondo ritardo di sensing.

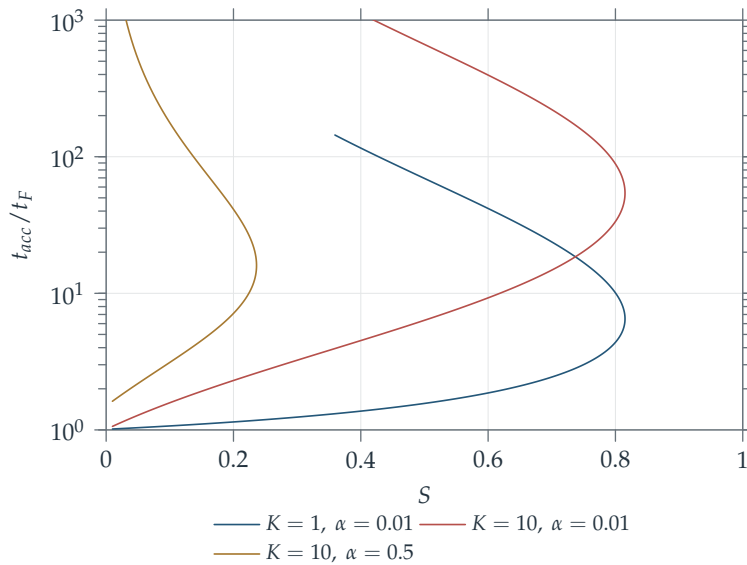


Figura 4.25. Tempo di accesso in funzione del throughput per il sistema CSMA non persistente.

Procedendo per via numerica si ottiene la prestazione riportata nella figura 4.25 che riporta il tempo di accesso in funzione del throughput al variare di α e di K .

4.5 Confronto tra schemi di accesso

In questa sezione si confrontano gli schemi di accesso analizzati con pacchetti di lunghezza costante e $N = 100$ stazioni. Per i metodi casuali si mantiene l'approssimazione poissoniana dei tentativi già dichiarata: il valore di N da solo non la rende esatta, ma rende ragionevole il limite quando i tentativi individuali sono rari e sufficientemente indipendenti. Nella figura 4.26 è riportato il caso con $K = 10$ e $\alpha = 0.01$. Nella figura 4.27 si ha $K = 20$ e $\alpha = 0.01$. Infine la figura 4.28 riporta il caso $K = 10$ e $\alpha = 0.5$.

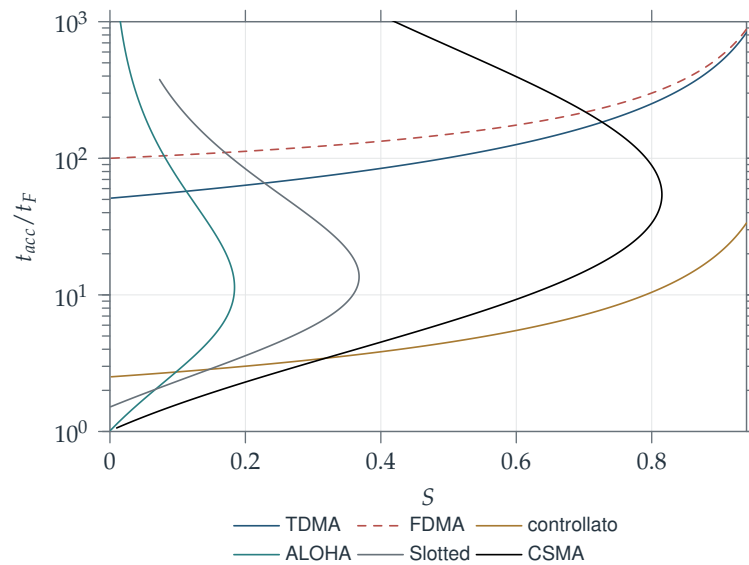


Figura 4.26. Tempo di accesso in funzione del throughput, $N = 100$, $K = 10$, $\alpha = 0.01$, $\beta = 0.02$.

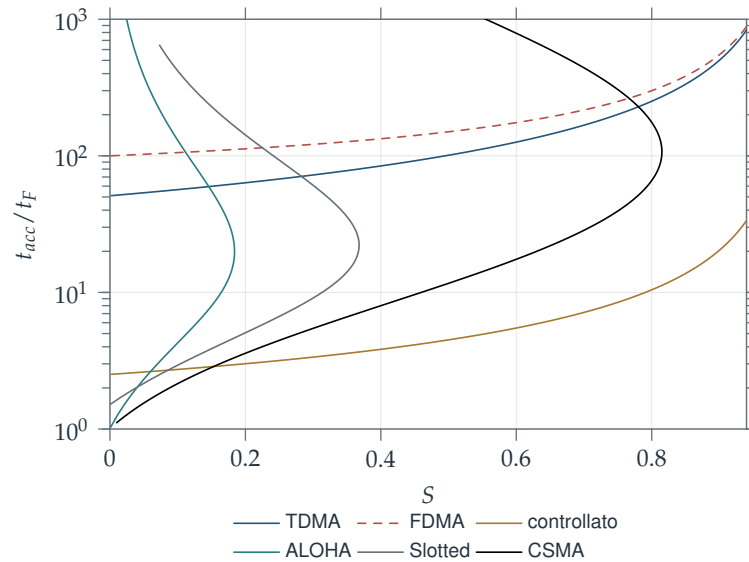


Figura 4.27. Tempo di accesso in funzione del throughput, $N = 100$, $K = 20$, $\alpha = 0.01$, $\beta = 0.02$.

Il confronto non rappresenta prestazioni end-to-end sotto identiche ipotesi di accodamento: TDMA, FDMA e polling includono la coda della sorgente, mentre per l'accesso casuale si è calcolato il tempo dal momento in cui il pacchetto è pronto al MAC. Per confrontare sistemi reali con lo stesso traffico occorre aggiungere anche ai casi casuali il corrispondente modello

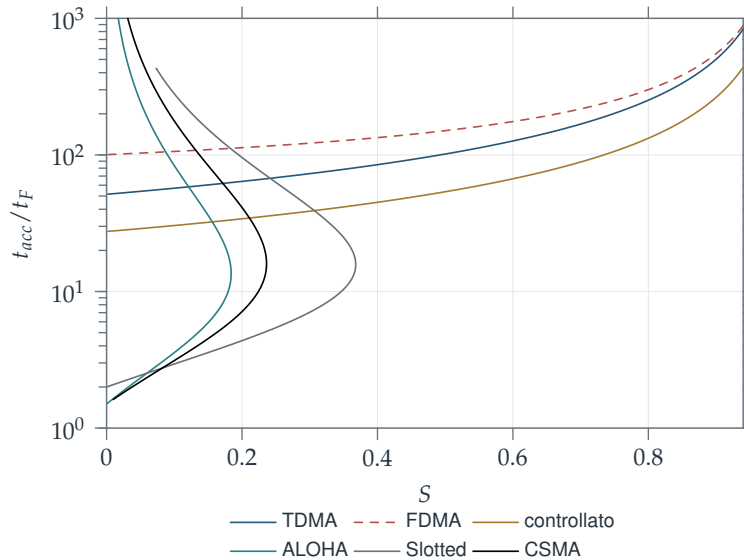


Figura 4.28. Tempo di accesso in funzione del throughput, $N = 100$, $K = 10$, $\alpha = 0.5$, $\beta = 0.02$.

di coda e verificare la sua stabilità.

Nel confronto condizionato alle ipotesi adottate, i protocolli deterministici sono poco sensibili al ritardo di propagazione nel modello adottato, mentre accesso controllato e CSMA ne risentono in modo più marcato. I metodi casuali sono particolarmente competitivi a basso carico, dove la probabilità di collisione è piccola. Non esiste però un metodo migliore in ogni regime: la scelta dipende dal carico medio e di picco, dal ritardo di propagazione, dall'equità richiesta, dalla robustezza in sovraccarico e dalla complessità che si è disposti a sostenere.

4.6 Fattori che incidono sulle prestazioni

In generale i fattori che incidono sulle prestazioni di un protocollo di accesso al mezzo sono i seguenti:

- *Topologia*, legata al ritardo di propagazione, qui assunto costante, ma che in generale varia da stazione a stazione. La topologia incide inoltre sulla potenza ricevuta e quindi sul margine di immunità ai disturbi e sulla probabilità di errore per bit o pacchetto.
- *Traffico*: la distribuzione dei tempi interarrivo e dei tempi di servizio determina direttamente i risultati analitici. Il modello poissoniano è utile perché rende trattabili molti casi, ma traffico correlato o a burst può produrre code e collisioni molto diverse a parità di carico medio.
- *Tecniche di modulazione*, legate direttamente alla probabilità di errore per bit o pacchetto, eventualmente coadiuvate da tecniche intelligenti di codifica.
- *Effetti di propagazione*, quali attenuazione, cammini multipli, shadowing e fading, fortemente dipendenti dall'ambiente ed in grado di produrre un notevole impatto sulle prestazioni in termini di probabilità di errore per bit o pacchetto.

Per avere prestazioni realistiche occorre trovare modelli che tengano conto degli effetti congiunti di topologia, traffico, tecniche di modulazione, fenomeni di propagazione, metodi di accesso. Data l'ampia possibilità di scelta riguardo ai diversi fattori e la complessità dello studio corrispondente, in questo capitolo sono stati trattati casi semplificati atti a dare un andamento di massima delle prestazioni e a proporre una metodologia di studio valida per un'ampia casistica.

È importante distinguere due concetti. Il *controllo di flusso* impedisce che un mittente sovraccarichi il ricevitore o una risorsa a valle; il *controllo della congestione* limita invece il carico immesso quando la rete, considerata nel suo insieme o in una sua porzione, non riesce a smaltirlo. I due meccanismi possono cooperare, ma non sono sinonimi.

Un sistema a commutazione di pacchetto è costituito da una moltitudine di nodi di tipo store and forward interconnessi in vario modo, con lo scopo di formare i percorsi tra sorgente e destinatario che gli utenti richiedono dinamicamente. Potenzialmente, ed in via di principio, ogni nodo può ricevere e generare traffico da e rispetto a tutti i nodi della rete.



103

evita che un pacchetto destinato ad una certa linea di uscita debba aspettare inutilmente lo smaltimento di un pacchetto destinato ad un'altra linea. Quindi l'instradamento sulle varie linee di uscita viene effettuato prima dell'accodamento.

Assumendo come riferimento il generico nodo i , la sua realizzazione fisica viene riportata schematicamente in figura 5.2.

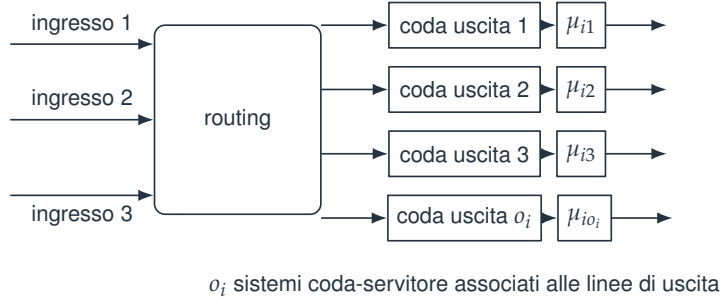


Figura 5.2. Esempio di realizzazione del generico nodo i .

Nel seguito si considera una rete costituita da M sistemi coda-servitore anche se i nodi della rete sono N con $N < M$ e detto o_i il numero delle linee di uscita del nodo i si ha il seguente legame tra N ed M :

$$M = \sum_{i=1}^N o_i$$

5.1.1 Indipendenza di Kleinrock

Un sistema costituito da più sistemi coda-servitore risulta particolarmente complesso da studiare a causa della forte correlazione tra i tempi interarrivo e le lunghezze dei pacchetti dopo il passaggio per un primo servitore (si pensi ad esempio ad un sistema $M/D/1$). Per procedere si sfrutta allora:

ASSUNZIONE DI INDIPENDENZA DI KLEINROCK: quando un collegamento aggrega molti flussi debolmente dipendenti, si approssimano come indipendenti il processo degli arrivi e le lunghezze dei pacchetti. Questa è un'ipotesi di modellazione, non un teorema: la sua accuratezza dipende dal grado di aggregazione, dalla dipendenza tra i flussi e dalle caratteristiche del traffico.

Inoltre, la sovrapposizione di molti flussi indipendenti e ciascuno relativamente poco intenso può, sotto opportune condizioni, essere approssimata da un processo di Poisson. Questa motivazione conduce al modello classico in cui gli arrivi ai nodi sono poissoniani e il nodo i può essere rappresentato da o_i code. Per usare $M/M/1$ occorre però anche assumere tempi di servizio esponenziali indipendenti: questa ipotesi non segue dall'aggregazione degli arrivi. Inoltre un pacchetto reale conserva normalmente la propria lunghezza ai vari salti; i suoi tempi di trasmissione successivi non sono quindi campioni indipendenti. L'approssimazione di rete di code deve dichiarare anche questa separazione fra modello e traffico reale.

5.1.2 Teorema di Burke

Dato che in un sistema a coda le partenze coincidono in generale con gli arrivi del sistema a coda successivo, vediamo come si caratterizza in un sistema $M/M/1$ la distribuzione delle partenze. Il risultato è noto come teorema di Burke:

Teorema 7. In un sistema $M/M/1$ in regime stazionario, con tasso di arrivo λ , tasso di servizio μ e

$\lambda < \mu$, il processo di partenza è di Poisson con tasso λ .

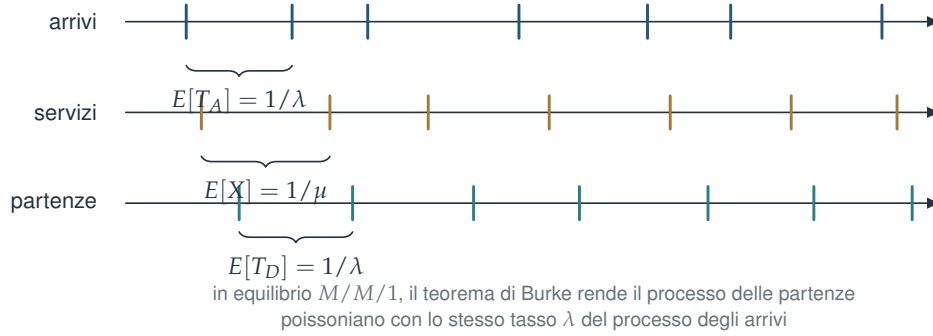


Figura 5.3. Scale temporali di arrivi, servizi e partenze in un sistema $M/M/1$ stazionario.

La figura 5.3 mette a confronto tre scale temporali: gli interarrivi, con media $1/\lambda$; i tempi di servizio esponenziali, con media $1/\mu$ quando il servitore è occupato; e gli intertempi fra partenze, che il teorema mostra essere esponenziali con media $1/\lambda$. I tempi di servizio non costituiscono un processo di partenze autonomo quando la coda è vuota: il tasso μ è il tasso condizionato al fatto che vi sia un cliente in servizio.

Per dimostrare il teorema di Burke consideriamo il sistema $M/M/1$ in regime stazionario, con $\lambda < \mu$ e $\rho = \lambda/\mu$. Il processo $N(t)$ del numero di utenti nel sistema è una catena di nascita e morte con distribuzione stazionaria

$$\pi_n = (1 - \rho)\rho^n, \quad n \geq 0.$$

Per ogni coppia di stati adiacenti vale la condizione di bilancio dettagliato

$$\pi_n \lambda = \pi_{n+1} \mu,$$

poiché

$$(1 - \rho)\rho^n \lambda = (1 - \rho)\rho^{n+1} \mu.$$

Per la formula di Bayes, in un intervallo h piccolo la probabilità che il predecessore di uno stato n fosse $n + 1$ è $\pi_{n+1}(\mu h + o(h))/\pi_n$. Dividendo per h e passando al limite si ricava il tasso della transizione inversa $n \rightarrow n + 1$ è

$$q_{n,n+1}^* = \frac{\pi_{n+1}\mu}{\pi_n} = \lambda,$$

indipendente da n . Nel processo temporalmente invertito le partenze del processo diretto diventano dunque transizioni verso l'alto generate da un orologio esponenziale di parametro costante λ : costituiscono un processo di Poisson di tasso λ . Analogamente il tasso inverso verso il basso è μ per $n > 0$: il generatore inverso coincide con quello diretto. La riflessione temporale conserva la legge di un processo di Poisson stazionario; le partenze nel verso diretto hanno quindi tasso λ e incrementi indipendenti. Questo dimostra il risultato.

5.1.3 Teorema di Jackson

Una rete di Jackson aperta è costituita da M nodi a singolo servitore. Gli arrivi esterni al nodo i sono poissoniani di tasso γ_i , i tempi di servizio sono esponenziali di parametro μ_i , e al termine del servizio un utente lascia la rete con probabilità p_{i0} oppure viene instradato al nodo j con probabilità p_{ij} . Le scelte di instradamento sono indipendenti dalla storia del sistema. Per semplicità assumiamo $p_{ii} = 0$; un auto-instradamento non è necessario alla

trattazione. Si assumono buffer illimitati, servizi e flussi esterni mutuamente indipendenti, tassi $\mu_i > 0$, disciplina non anticipante e assenza di blocco fra nodi. Da ogni nodo rilevante un cliente deve poter lasciare la rete con probabilità uno: la matrice di routing interna è transiente. Le equazioni di traffico hanno allora una soluzione finita univoca. Nella dimostrazione consideriamo i nodi raggiunti dal traffico esterno, per cui $\lambda_i > 0$; eventuali nodi con tasso nullo sono vuoti nel regime stazionario e vanno separati prima di dividere per ρ_i .

I tassi complessivi λ_i che attraversano i singoli nodi sono le soluzioni delle equazioni di traffico

$$\lambda_i = \gamma_i + \sum_{j=1}^M \lambda_j p_{ji}, \quad i = 1, \dots, M. \quad (5.1)$$

Vale il seguente teorema di Jackson:

Teorema 8. *Se le equazioni di traffico ammettono una soluzione non negativa e $\rho_i = \lambda_i / \mu_i < 1$ per ogni nodo, la rete ammette un regime stazionario e la probabilità dello stato $\underline{n} = (n_1, \dots, n_M)$ è*

$$P(\underline{n}) = \prod_{i=1}^M (1 - \rho_i) \rho_i^{n_i}. \quad (5.2)$$

La distribuzione ha quindi forma prodotto: in regime stazionario le distribuzioni marginali dei numeri di utenti coincidono con quelle di M code $M/M/1$ aventi tassi λ_i e μ_i .

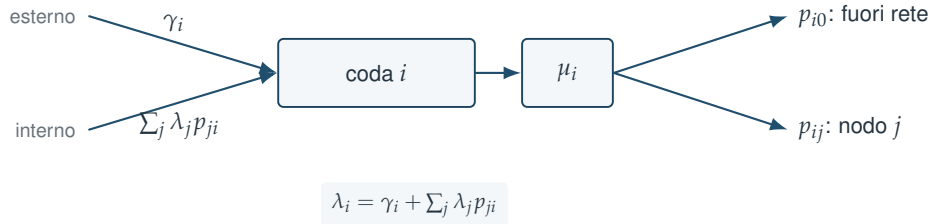


Figura 5.4. Esempio di nodo $M/M/1$ in una rete di Jackson.

Dimostrazione: indichiamo con $\underline{1}_i$ il vettore con un unico 1 in posizione i . Proponiamo come distribuzione stazionaria

$$\pi(\underline{n}) = \prod_{i=1}^M (1 - \rho_i) \rho_i^{n_i}.$$

Poiché $\rho_i < 1$, la distribuzione è normalizzata. Verifichiamo le equazioni di bilancio globale. Dallo stato $\underline{n}$ si esce per un arrivo esterno oppure per un completamento di servizio in ciascun nodo non vuoto; il tasso complessivo di uscita è quindi

$$\left(\sum_{i=1}^M \gamma_i + \sum_{i:n_i>0} \mu_i \right) \pi(\underline{n}). \quad (5.3)$$

Lo stato $\underline{n}$ può invece essere raggiunto in tre modi: mediante un arrivo esterno al nodo i , mediante una partenza dalla rete dal nodo i , oppure mediante l'instradamento di un utente dal nodo j al nodo i . Il tasso complessivo di ingresso è

$$\begin{aligned} & \sum_{i:n_i>0} \gamma_i \pi(\underline{n} - \underline{1}_i) + \sum_{i=1}^M \mu_i p_{i0} \pi(\underline{n} + \underline{1}_i) \\ & + \sum_{i:n_i>0} \sum_{j=1}^M \mu_j p_{ji} \pi(\underline{n} - \underline{1}_i + \underline{1}_j). \end{aligned} \quad (5.4)$$

Per la forma prodotto proposta valgono, quando gli stati indicati esistono,

$$\frac{\pi(\underline{n} - \underline{1}_i)}{\pi(\underline{n})} = \frac{1}{\rho_i}, \quad \frac{\pi(\underline{n} + \underline{1}_i)}{\pi(\underline{n})} = \rho_i,$$

e, per $i \neq j$,

$$\frac{\pi(\underline{n} - \underline{1}_i + \underline{1}_j)}{\pi(\underline{n})} = \frac{\rho_j}{\rho_i}.$$

Dividendo la (5.4) per $\pi(\underline{n})$, i termini associati a un nodo i non vuoto diventano

$$\frac{1}{\rho_i} \left(\gamma_i + \sum_{j=1}^M \mu_j \rho_j p_{ji} \right) = \frac{1}{\rho_i} \left(\gamma_i + \sum_{j=1}^M \lambda_j p_{ji} \right) = \frac{\lambda_i}{\rho_i} = \mu_i,$$

dove è stata usata l'equazione di traffico (5.1). I termini di uscita dalla rete forniscono invece

$$\sum_{i=1}^M \mu_i \rho_i p_{i0} = \sum_{i=1}^M \lambda_i p_{i0}.$$

Sommando le equazioni di traffico su tutti i nodi e usando $p_{i0} + \sum_j p_{ij} = 1$, si ottiene la conservazione del flusso

$$\sum_{i=1}^M \gamma_i = \sum_{i=1}^M \lambda_i p_{i0}.$$

Il tasso di ingresso (5.4) coincide dunque con il tasso di uscita (5.3) per ogni stato $\underline{n}$. La distribuzione proposta soddisfa le equazioni di bilancio globale ed è pertanto la distribuzione stazionaria della rete. L'indipendenza della forma prodotto riguarda i numeri di clienti osservati nello stesso istante. Non implica indipendenza dei processi nel tempo, né che ogni flusso interno in una rete con retroazione sia un processo di Poisson indipendente dagli altri: la prova usa il bilancio del processo congiunto, non questa affermazione più forte.

La stessa forma prodotto può essere scritta separando il termine di normalizzazione:

$$P(\underline{n}) = P(\underline{0}) \prod_{i=1}^M \left(\frac{\lambda_i}{\mu_i} \right)^{n_i}. \quad (5.5)$$

Poiché $\rho_i = \lambda_i / \mu_i < 1$, la normalizzazione si fattorizza in prodotti di serie geometriche:

$$P(\underline{0}) = \left[\prod_{i=1}^M \sum_{n_i=0}^{\infty} \rho_i^{n_i} \right]^{-1} = \left[\prod_{i=1}^M \frac{1}{1 - \rho_i} \right]^{-1} = \prod_{i=1}^M (1 - \rho_i). \quad (5.6)$$

Sostituendo la (5.6) nella (5.5) si ritrova la (5.2).

5.1.4 Parametri caratteristici di una rete di code

Si può procedere nell'analisi dei parametri caratteristici della rete valutando il numero degli utenti presenti; ricordando la (2.30) si ha:

$$E[k] = \sum_{i=1}^M E[n_i] = \sum_{i=1}^M \frac{\rho_i}{1 - \rho_i} = \sum_{i=1}^M \frac{\lambda_i}{\mu_i - \lambda_i} \quad (5.7)$$

Il tempo medio speso da un pacchetto per percorrere la successione di sistemi che costituiscono la rete (tempo medio di attraversamento o permanenza di un pacchetto qualsiasi nella rete) può essere calcolato sfruttando il risultato di Little. A tale scopo conoscendo il numero

medio di utenti $E[k]$ ricavato nella (5.7) ed il tasso medio di arrivo complessivo dall'esterno, λ_s , ricavabile dalla somma dei singoli tassi di arrivo verso la rete, si ha:

$$E[T] = \frac{E[k]}{\lambda_s} = \frac{1}{\lambda_s} \sum_{i=1}^M \frac{\lambda_i}{\mu_i - \lambda_i}$$

Per includere i ritardi di propagazione occorre pesare ogni collegamento con il numero medio di visite effettuate dai pacchetti. Sia τ_i il ritardo di propagazione della linea in uscita dal sistema i . Poiché il flusso medio che attraversa quel sistema è λ_i , il contributo medio del collegamento al ritardo di un pacchetto esterno è $\lambda_i \tau_i / \lambda_s$. Si ottiene quindi

$$E[T] = \frac{1}{\lambda_s} \sum_{i=1}^M \lambda_i \left[\frac{1}{\mu_i - \lambda_i} + \tau_i \right] = \frac{1}{\lambda_s} \sum_{i=1}^M f_i \left[\frac{1}{C_i - f_i} + \hat{\tau}_i \right], \quad (5.8)$$

dove si assume la stessa lunghezza media $E[F]$ per i pacchetti ai nodi considerati, e $C_i = \mu_i E[F]$ è la capacità della linea in bit/s, $f_i = \lambda_i E[F]$ è il flusso medio in bit/s e $\hat{\tau}_i = \tau_i / E[F]$, espresso in s/bit. La seconda forma è semplicemente la prima riscritta in termini di flussi di bit.

Per descrivere l'instradamento indichiamo con I_{ji} la frazione del flusso uscente dal nodo j inoltrata al nodo i , con I_{si} la frazione del traffico esterno f_s che entra nel nodo i e con I_{id} la frazione del flusso del nodo i che lascia la rete. La conservazione del flusso impone

$$\begin{aligned} f_i &= I_{si} f_s + \sum_{j=1}^M I_{ji} f_j, \quad i = 1, \dots, M, \\ f_s &= \sum_{i=1}^M I_{id} f_i. \end{aligned} \quad (5.9)$$

La seconda uguaglianza esprime il regime stazionario: il traffico che entra dall'esterno deve, in media, essere uguale a quello che esce. Per ogni origine le frazioni di instradamento verso i nodi successivi e verso l'esterno sommano a uno; gli elementi relativi a collegamenti inesistenti sono nulli.

A partire dall'indipendenza mostrata dalla (5.2) si possono poi valutare altri parametri caratteristici, come la varianza del numero di utenti presenti nel sistema o statistiche di ordine superiore.

5.2 Algoritmi di instradamento

Dentro ogni nodo della rete deve essere presente un meccanismo in grado di decidere su quale sistema coda-servitore deve essere inviata l'informazione in transito per poter raggiungere il destinatario. Il modo con cui viene presa questa decisione viene detto politica di instradamento ed è implementato tramite un opportuno algoritmo.

Le politiche di instradamento possono essere descritte secondo più assi indipendenti:

- **Centralizzate o distribuite.** In una soluzione centralizzata un'entità logica calcola le decisioni usando una vista globale della topologia e distribuisce lo stato necessario ai nodi; la centralizzazione può essere soltanto logica, perché il controllo può essere fisicamente replicato. In una soluzione distribuita ogni nodo costruisce invece la propria informazione cooperando con altri nodi. Il secondo approccio richiede protocolli di convergenza e traffico di controllo, ma non dipende da una singola vista globale.
- **Statiche o dinamiche.** Una politica statica viene configurata esplicitamente e non reagisce autonomamente alle variazioni della rete. Una politica dinamica ricalcola le decisioni

quando cambiano raggiungibilità, costi o vincoli. Se la metrica dipende dal carico, l'adattamento deve essere sufficientemente smorzato: reazioni troppo rapide possono produrre oscillazioni invece di ridurre la congestione.

- **Proattive o reattive.** Nel caso proattivo le informazioni vengono mantenute aggiornate prima che un pacchetto ne abbia bisogno; si riduce quindi la latenza di instradamento al prezzo di un overhead continuo. Nel caso reattivo la ricerca del percorso viene avviata su richiesta; si riduce l'overhead quando il traffico è scarso, ma il primo pacchetto può subire il ritardo di scoperta del percorso.
- **Piatte o gerarchiche.** In una rete piatta i nodi partecipano secondo regole sostanzialmente uniformi. In una rete gerarchica indirizzi, informazioni topologiche o responsabilità di controllo vengono aggregate per aree o gruppi. La gerarchia riduce la quantità di stato e l'overhead di segnalazione, al prezzo di una vista meno dettagliata oltre il proprio dominio.
- **Hop-by-hop o source routing.** Nel primo caso ogni nodo sceglie il prossimo salto sulla base dello stato locale e della destinazione. Nel source routing la sorgente impone invece il percorso, o una sua rappresentazione, nel pacchetto o nello stato associato al flusso. Il source routing è concettualmente distinto dal circuito virtuale: un circuito virtuale può inoltrare tramite identificatori di stato senza che la sorgente elenchi i singoli nodi del percorso.

L'algoritmo di instradamento è basato sulla ottimizzazione di funzioni di costo quali il numero dei salti, il costo fisico associato a ciascun percorso, il ritardo complessivo della rete, il throughput o da una loro composizione.

Nella costruzione dell'algoritmo occorre considerare aspetti di carattere generale quali: la correttezza, la semplicità, la stabilità, l'equità, la robustezza, l'affidabilità, la sicurezza.

Alcuni tipi di pacchetti possono essere considerati prioritari rispetto ad altri e questo influenza sia l'algoritmo di instradamento che la gestione della coda. Inoltre, per considerare le diverse esigenze degli utenti, le funzioni di costo possono essere differenti da pacchetto a pacchetto.

I nodi devono disporre delle informazioni necessarie alla politica scelta. Tali informazioni possono essere configurate, distribuite da un controllore, scambiate soltanto con i vicini oppure propagate in una porzione più ampia della rete. Il meccanismo di distribuzione dello stato è parte integrante dell'algoritmo perché ne determina overhead, rapidità di convergenza e robustezza ai guasti.

Per lo studio degli algoritmi di instradamento la rete può essere vista come un grafo con M nodi e diversi collegamenti; ad ogni collegamento è assegnato un costo calcolato in base alla metrica o funzione di costo scelta. Lo scopo dell'algoritmo è la minimizzazione del costo del collegamento tra nodo sorgente e nodo destinazione, ossia la somma dei costi di tutte le linee attraversate.

5.2.1 Algoritmo di Dijkstra

Consideriamo un grafo con costi di arco non negativi. L'algoritmo di Dijkstra determina il percorso di costo minimo dalla sorgente a tutti i nodi raggiungibili. Sia $D(v)$ la distanza o costo del destinatario v dalla sorgente, indicata come nodo 1. Sia $l(i, j)$ il costo dell'arco diretto dal nodo i al nodo j ($+\infty$ se l'arco non esiste); allora:

- 1) $\mathcal{F} = \{1\}$, $D(1) = 0$ e $D(v) = l(1, v)$ per $v \neq 1$
- 2) $w \notin \mathcal{F} \mid \forall z \notin \mathcal{F}, w \neq z, D(w) \leq D(z)$
- 3) $\mathcal{F} = \mathcal{F} + w$

- 4) $\forall z \notin \mathcal{F} \quad D(z) = \min[D(z), D(w) + l(w, z)]$
- 5) se $|\mathcal{F}| < M$
 - allora vai al passo 2
 - altrimenti STOP

L'insieme $\mathcal{F}$ contiene i nodi la cui distanza è già definitiva. Il passo 2 sceglie, fra i nodi non ancora selezionati, quello con etichetta temporanea minima; se tutte le etichette residue sono $+\infty$, tali nodi sono irraggiungibili e si può terminare; il passo 4 rilassa gli archi uscenti da tale nodo aggiornando le distanze dei nodi ancora non selezionati.

La correttezza segue da un invariante. Prima di ogni iterazione, $D(v)$ è il costo minimo fra i cammini dalla sorgente a v i cui nodi interni appartengono a $\mathcal{F}$. Sia w il nodo esterno a $\mathcal{F}$ con etichetta minima. Supponiamo esista un cammino verso w di costo inferiore a $D(w)$ e sia y il primo nodo di tale cammino non appartenente a $\mathcal{F}$. Il predecessore di y è in $\mathcal{F}$, quindi per l'invariante il rilassamento ha già imposto $D(y)$ non superiore al costo del prefisso del cammino fino a y . Poiché tutti i costi degli archi sono non negativi, quel prefisso non può costare più del cammino completo verso w ; ne seguirebbe $D(y) < D(w)$, in contraddizione con la scelta di w . Dunque $D(w)$ è la distanza minima definitiva. Il rilassamento conserva l'invariante e, per induzione, al termine tutte le etichette sono distanze minime.

Per ottenere una tabella completa da ogni possibile sorgente, la procedura viene ripetuta scegliendo a turno ciascun nodo come sorgente.

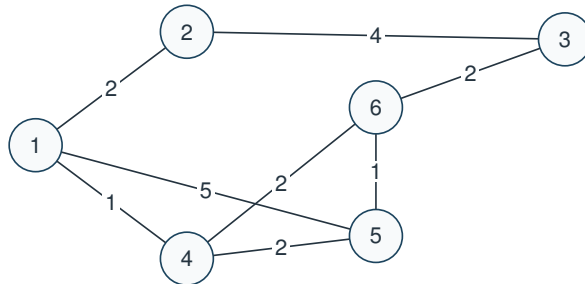


Figura 5.5. Rete di riferimento.

A titolo di esempio, per l'applicazione dell'algoritmo di Dijkstra, si consideri la rete riportata in figura 5.5. La tabella 5.1 mostra i vari passi compiuti dall'algoritmo per arrivare a selezionare tutti i percorsi più brevi per connettere il nodo 1 ad un qualsiasi altro nodo; i numeri sottolineati una volta indicano il nodo che presenta costo minore tra quelli non selezionati mentre i numeri sottolineati due volte indicano i nodi già selezionati nei passi precedenti.

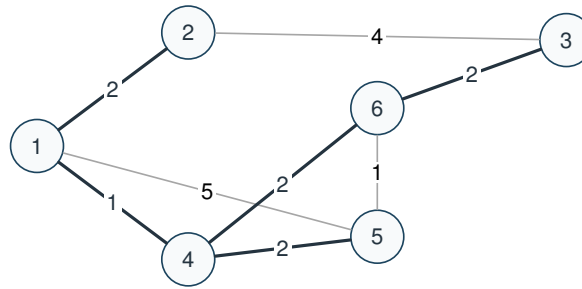
Le connessioni selezionate nella tabella 5.1 sono riportate nella figura 5.6.

È importante non attribuire a Dijkstra limiti che appartengono invece al modo in cui lo si usa nel piano di controllo. Una singola esecuzione con una metrica scalare costruisce un albero dei cammini minimi e, se si conserva un solo predecessore per nodo, seleziona un solo percorso per destinazione. In presenza di più cammini di uguale costo è però possibile conservarne più di uno; l'uso effettivo di multipath richiede poi una politica di inoltro coerente.

Analogamente, l'adattamento alle variazioni di traffico non dipende dall'algoritmo di cammino minimo in sé, ma da come vengono misurati e distribuiti i costi e da quanto spesso il calcolo viene ripetuto. Metriche che reagiscono troppo rapidamente al carico possono produrre oscillazioni: più traffico viene spostato su un percorso, il suo costo aumenta e il traffico può essere spostato nuovamente altrove. Per questo i protocolli reali separano il

Passo	$\mathcal{F}$	$D(2)$	$D(3)$	$D(4)$	$D(5)$	$D(6)$
0	{1}	2	∞	<u>1</u>	5	∞
1	{1,4}	<u>2</u>	∞	<u>1</u>	3	3
2	{1,4,2}	<u>2</u>	6	<u>1</u>	<u>3</u>	3
3	{1,4,2,5}	<u>2</u>	6	<u>1</u>	<u>3</u>	3
4	{1,4,2,5,6}	<u>2</u>	<u>5</u>	<u>1</u>	<u>3</u>	<u>3</u>
5	{1,4,2,5,6,3}	<u>2</u>	<u>5</u>	<u>1</u>	<u>3</u>	<u>3</u>

Tabella 5.1. Evoluzione di Dijkstra dalla sorgente 1. Una sottolineatura indica il minimo temporaneo; la doppia sottolineatura un costo già reso definitivo.



albero dei cammini minimi dalla sorgente 1

Figura 5.6. Connessioni selezionate con l'algoritmo di Dijkstra.

calcolo del cammino dai meccanismi di misura, filtraggio e temporizzazione delle metriche.

5.2.2 Bellman-Ford

Bellman-Ford è un secondo algoritmo per il percorso minimo. A differenza di Dijkstra, la sua ricorrenza si presta direttamente a una realizzazione distribuita e può trattare costi negativi purché non esistano cicli di costo negativo raggiungibili. Supponiamo che la destinazione sia il nodo 1 e sia $l(i, j)$ il costo dell'arco diretto dal nodo i al nodo j , con $l(i, i) = 0$ e $l(i, j) = +\infty$ quando l'arco non esiste. Sia D_i^h il costo minimo, al passo h , di un cammino dal nodo i alla destinazione 1:

- 1) $D_1^h = 0, \forall h; D_i^0 = \infty, \forall i \neq 1$
- 2) $D_i^{h+1} = \min_j [l_{ij} + D_j^h], \forall i \neq 1$
- 3) Se $D_i^{h+1} = D_i^h, \forall i$
 - allora STOP
 - altrimenti vai al passo 2

La ricorrenza ha un significato preciso: D_i^h è il costo minimo di un cammino da i alla destinazione che utilizza al massimo h archi. La proprietà si dimostra per induzione. Per $h = 0$ soltanto la destinazione ha costo zero e gli altri nodi hanno costo infinito. Se la proprietà vale al passo h , ogni cammino con al massimo $h + 1$ archi parte con un arco (i, j) e prosegue con un cammino da j di al più h archi; minimizzando $l(i, j) + D_j^h$ su j si ottiene quindi esattamente il minimo cercato. In un grafo con M nodi, un cammino minimo privo di cicli usa al massimo $M - 1$ archi. In assenza di cicli di costo negativo raggiungibili, dopo al più $M - 1$ rilassamenti le distanze sono quindi definitive. Un ulteriore rilassamento che riduca ancora una distanza segnala un ciclo di costo negativo nella parte del grafo che può raggiungere la destinazione. Un'implementazione deve imporre questo controllo dopo $M - 1$ passi invece di attendere

indefinitamente la convergenza. Per un dato nodo il costo può diventare illimitatamente negativo solo se da quel nodo si può raggiungere tale ciclo e poi la destinazione.

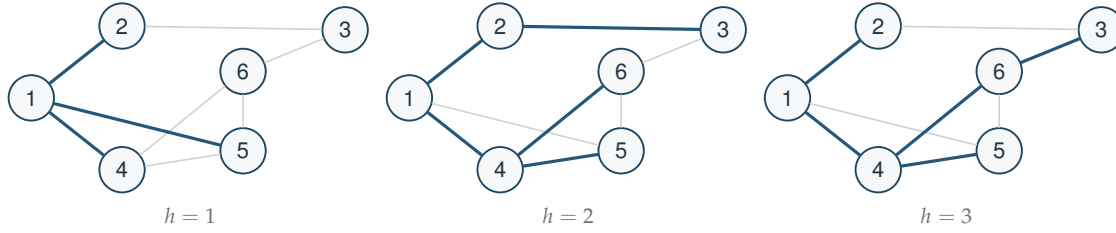


Figura 5.7. Connessioni selezionate con l'algoritmo Bellman-Ford con $h=1,2,3$.

A titolo di esempio, in figura 5.7 si riporta una applicazione di questo algoritmo alla rete in figura 5.5. La figura mostra i casi con $h = 1, 2$ e 3 . Si osserva che con $h = 3$ si raggiunge la configurazione definitiva, uguale a quella ottenuta con l'algoritmo di Dijkstra. L'algoritmo, comunque, necessita di un ulteriore passo per verificare che tutti i nodi siano stati raggiunti con costo minimo.

5.2.3 Distance Vector, DV

Il Distance Vector è la realizzazione distribuita della ricorrenza di Bellman-Ford. Ogni nodo i mantiene, per ogni destinazione x , una stima $D_i(x)$ del costo minimo e il vicino da usare come prossimo salto. Se $c(i, j)$ è il costo del collegamento diretto tra i e il vicino j , la relazione fondamentale è

$$D_i(x) = \min_{j \in \mathcal{N}(i)} \{c(i, j) + D_j(x)\}.$$

Il nodo non deve conoscere l'intera topologia: gli basta conoscere i costi dei propri collegamenti e le distanze annunciate dai vicini.

Quando i riceve il distance vector $DV_x^{(j)}$ dal vicino j , per ogni destinazione x calcola il candidato

$$c(i, j) + DV_x^{(j)}.$$

Se questo valore è inferiore alla stima corrente $DV_x^{(i)}$, la tabella viene aggiornata ponendo

$$DV_x^{(i)} = c(i, j) + DV_x^{(j)}$$

e scegliendo j come prossimo salto. In una realizzazione completa devono essere trattati anche gli aumenti di costo e la perdita di un vicino: gli annunci successivi possono quindi peggiorare, oltre che migliorare, una rotta. Gli scambi possono essere periodici, attivati da una variazione oppure combinare entrambe le modalità.

La figura 5.8 mostra il principio sulla rete di riferimento di figura 5.5: il nodo 1 confronta le proprie stime con quelle annunciate dai vicini 2, 4 e 5 e applica la ricorrenza di Bellman-Ford destinazione per destinazione.

Una difficoltà caratteristica è il *count-to-infinity*. Si considerino tre nodi in cascata A-B-C con costi unitari. Prima del guasto del collegamento B-C, B raggiunge C con costo 1 e A con costo 2 tramite B. Se B perde il collegamento ma riceve da A una vecchia stima pari a 2, può erroneamente concludere di raggiungere C tramite A con costo 3. A, a sua volta, può apprendere da B un costo 3 e portare la propria stima a 4; il processo continua finché una metrica massima o un'altra informazione interrompe il conteggio. Durante la convergenza può formarsi anche un anello di inoltro.

Tecniche come split horizon, poison reverse, aggiornamenti innescati e un valore finito che rappresenta l'infinito riducono durata e probabilità di questi fenomeni, ma non cambiano il principio: ogni nodo conosce direttamente soltanto ciò che gli viene annunciato dai vicini.

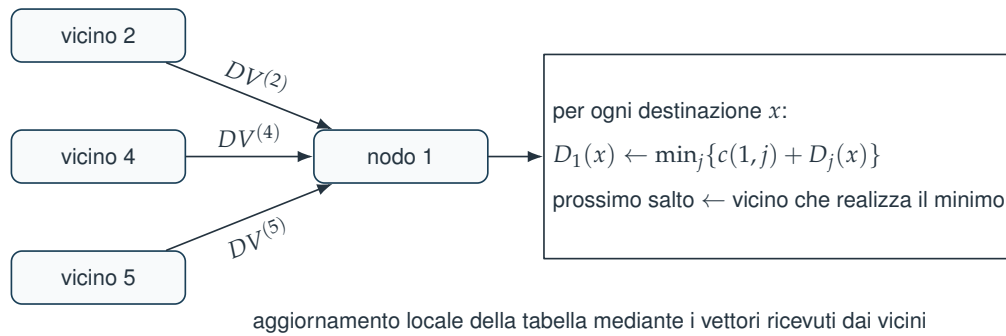


Figura 5.8. Esempio di aggiornamento di una tabella mediante distance vector.

5.2.4 Link State, LS

Il Link State distribuisce informazioni diverse dal Distance Vector: ogni nodo descrive i propri collegamenti locali e tutti i nodi di un dominio di instradamento ricostruiscono, dopo la convergenza, la stessa base dati topologica. La procedura concettuale comprende quattro operazioni.

- **Scoperta delle adiacenze.** Un nodo scambia messaggi di hello con i vicini direttamente raggiungibili e determina con quali di essi può formare un'adiacenza. Il contenuto esatto dei messaggi, le condizioni per formare l'adiacenza e i temporizzatori dipendono dal protocollo concreto.
- **Attribuzione delle metriche.** A ogni collegamento viene associato un costo, configurato o ricavato da caratteristiche della risorsa. Il costo usato dall'algoritmo non coincide necessariamente con un ritardo fisico istantaneo.
- **Origine e flooding dello stato dei link.** Ogni nodo genera descrizioni delle proprie adiacenze e delle relative metriche, dotate di informazioni di freschezza. Tali descrizioni vengono propagate con flooding controllato, così che aggiornamenti nuovi sostituiscano quelli vecchi e le duplicazioni non producano una propagazione indefinita.
- **Calcolo locale dei cammini.** Dalla base dati dello stato dei link ogni nodo costruisce il grafo del dominio e calcola i cammini minimi, tipicamente con Dijkstra. Le tabelle di inoltro risultanti sono quindi calcolate localmente a partire da una vista topologica condivisa.

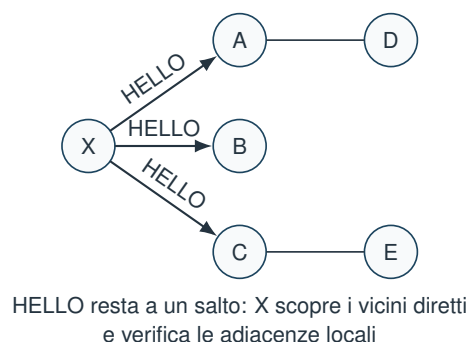


Figura 5.9. Scambio schematico di messaggi di hello per la scoperta delle adiacenze.

La figura 5.9 rappresenta soltanto il primo di questi passi: la scoperta dei vicini. Non prescrive il formato di un particolare protocollo Link State.

Il Link State non soffre del count-to-infinity tipico del Distance Vector, perché le decisioni

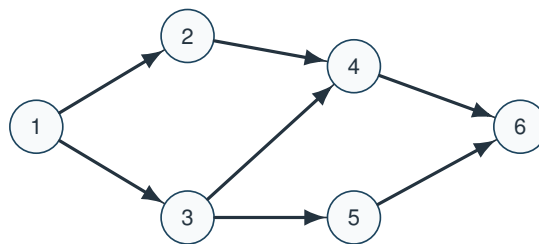
vengono ricalcolate da una descrizione della topologia anziché concatenando soltanto distanze annunciate. Durante la propagazione di un cambiamento, tuttavia, nodi diversi possono temporaneamente possedere basi dati non ancora allineate e quindi produrre cammini transitori differenti. Freschezza degli aggiornamenti, flooding affidabile e temporizzazione sono perciò parte essenziale del meccanismo.

5.2.5 Ad Hoc On-Demand Distance Vector, AODV

AODV è un protocollo di instradamento reattivo: una rotta viene scoperta quando serve, invece di mantenere continuamente un cammino verso ogni possibile destinazione. Questa scelta riduce lo stato e il traffico di controllo quando le comunicazioni sono sporadiche, ma introduce un ritardo di scoperta prima dell'invio dei primi dati.

Quando una sorgente non possiede una rotta valida genera una *Route Request* (RREQ) e la diffonde ai vicini. Il RREQ identifica l'origine e la specifica richiesta, così che una coppia origine-identificatore permetta ai nodi di riconoscere e scartare copie già elaborate. Contiene inoltre numeri di sequenza associati alle rotte: essi consentono di confrontarne la freschezza e di evitare che informazioni obsolete prevalgano su informazioni più recenti.

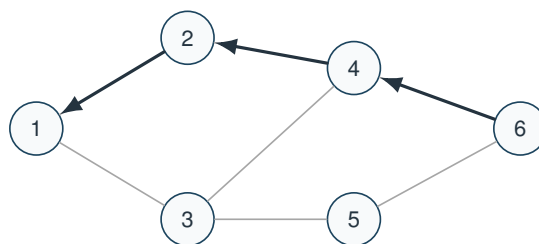
Mentre inoltra per la prima volta il RREQ, ogni nodo memorizza il prossimo salto necessario a tornare verso l'origine. Si costruisce così, implicitamente, un percorso inverso.



RREQ: diffusione dalla sorgente 1 verso la destinazione 6

Figura 5.10. Flooding schematico di una Route Request.

La destinazione, oppure un nodo intermedio che possiede una rotta sufficientemente fresca quando la politica lo consente, genera una *Route Reply* (RREP). Il RREP viene inviato in unicast lungo il percorso inverso creato dai RREQ; attraversando i nodi installa o aggiorna lo stato di inoltra verso la destinazione, compresi prossimo salto, metrica, freschezza e tempo di validità.



RREP: risposta unicast lungo il percorso inverso selezionato

Figura 5.11. Ritorno schematico della Route Reply sul percorso inverso.

Le rotte non restano valide indefinitamente: l'assenza di utilizzo o la perdita di un collegamento ne provoca la scadenza o l'invalidazione. Quando una rotta attiva si interrompe, messaggi di errore di rotta consentono di informare le sorgenti interessate, che possono avviare una nuova scoperta.

Le figure 5.10 e 5.11 mostrano il solo principio RREQ–RREP per una ricerca dal nodo 1 al nodo 6; identificatori, numeri di sequenza, temporizzatori e gestione degli errori sono parte dello stato di protocollo e non sono rappresentati nello schema.

5.2.6 Deflection routing (hot-potato)

Quando un nodo dispone di memoria molto limitata, un'alternativa all'accodamento è inoltrare immediatamente un pacchetto su una linea disponibile anche se non appartiene al cammino preferito. Questo principio è detto *deflection routing* o, in questo significato, *hot-potato routing*: il pacchetto viene tenuto in movimento anziché conservarlo in un buffer.

Il vantaggio è la riduzione della memoria richiesta e del tempo di attesa locale. Il costo viene trasferito alla rete: un pacchetto può percorrere più collegamenti del necessario, arrivare fuori ordine o essere deflesso ripetutamente. La politica deve quindi privilegiare, quando possibile, uscite che avvicinino alla destinazione e deve impedire permanenze indefinite, per esempio mediante limiti di vita o altre regole di progresso.

Il termine *hot-potato* viene usato anche con un significato diverso nell'instradamento interdominio: scegliere l'uscita più vicina dal proprio dominio verso una destinazione esterna. Quel comportamento non è il deflection routing descritto qui e non implica assenza di buffer.

5.2.7 Bifurcating routing o Multipath routing

Le metodologie presentate fino ad ora sono caratterizzate dallo sfruttare una sola linea per la trasmissione di un pacchetto. È comunque pensabile di utilizzare più linee contemporaneamente, se queste sono disponibili, e cercare di minimizzare il tempo medio di permanenza di un pacchetto nella rete.

Supponendo di poter trascurare i ritardi di propagazione rispetto ai tempi medi necessari per la sola trasmissione, il problema può essere identificato a partire dalla (5.8) sfruttando la (5.9):

$$E[T] = \frac{1}{\lambda_s} \sum_{i=1}^M \frac{f_i}{C_i - f_i} = \frac{1}{\lambda_s} \sum_{i=1}^M \frac{f_s I_{si} + \sum_{j=1}^M I_{ji} f_j}{C_i - f_s I_{si} - \sum_{j=1}^M I_{ji} f_j} = \mathcal{F}(\underline{C}, \underline{f}, \underline{I}) \quad (5.10)$$

dove il vettore $\underline{f}$ ha $M + 1$ componenti ed è così composto:

$$\underline{f} = (f_1, \dots, f_M, f_s)$$

e la matrice di instradamento $\underline{I}$ ha dimensione $(M + 1) \cdot (M + 1)$:

$$\underline{I} = \begin{pmatrix} I_{11} & \dots & I_{1M} & I_{1d} \\ I_{21} & \dots & I_{2M} & I_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ I_{M1} & \dots & I_{MM} & I_{Md} \\ I_{s1} & \dots & I_{sM} & 0 \end{pmatrix}$$

L'algoritmo di instradamento prevede la determinazione della matrice $\underline{I}$ attraverso la minimizzazione della (5.10) sotto i seguenti vincoli:

- il tasso complessivo f_s degli arrivi è fissato
- alcuni collegamenti non sono presenti in quanto la rete in generale non è completamente connessa. Questo significa che la matrice di instradamento ha alcuni elementi nulli, fissati a priori.
- i collegamenti con il mondo esterno sono definiti a priori e quindi risultano vincolati gli elementi I_{id} e I_{si} per ogni i .

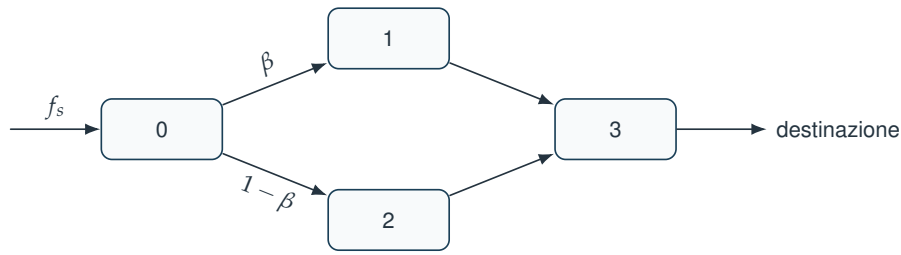
- le velocità di trasmissione dei singoli collegamenti sono fissate.
- i legami introdotti dalla (5.9) devono essere rispettati per garantire la conservazione del flusso. Se si usa il vettore aumentato $(f_1, \dots, f_M, f_s)$ e si identifica, come artificio di chiusura contabile, il flusso esterno uscente con il flusso esterno entrante, gli stessi vincoli possono essere raccolti nella forma matriciale $\underline{f} = \underline{f} \underline{I}$. La forma componente per componente della (5.9) resta però quella che ne esplicita il significato fisico.

Il problema consiste quindi nel minimizzare $\mathcal{F}$ nel dominio ammissibile definito dai vincoli di topologia, capacità e conservazione del flusso. È un problema di ottimizzazione del traffico, non un algoritmo shortest-path: una stessa domanda può essere suddivisa su più collegamenti. Per questo viene indicato come bifurcating o multipath routing. Nel modello considerato l'obiettivo è il ritardo medio, per cui si può parlare di instradamento ottimo rispetto a quella specifica funzione di costo; obiettivi diversi, come robustezza, equità o consumo energetico, conducono in generale a una soluzione diversa.

I ritardi di propagazione possono essere mantenuti nella funzione obiettivo usando la forma completa della (5.8); nell'esempio che segue vengono trascurati per isolare l'effetto della suddivisione del flusso.

Esempio di biforcazione

A titolo di esempio è possibile studiare analiticamente una rete composta da quattro sistemi $M/M/1$ con un solo ingresso ed una sola uscita come riportato in figura 5.12.



biforcazione su due percorsi paralleli e successiva ricombinazione

Figura 5.12. Esempio di biforcazione, sistema a 4 code.

Si osservi che: $I_{s0} = 1$, dato che il traffico in ingresso alla rete è tutto diretto al nodo 0, $I_{13} = I_{23} = 1$, dato che il traffico uscente dai nodi 1 e 2 è vincolato ad entrare nel nodo 3, e che $I_{3d} = 1$, dato che il traffico uscente dalla nostra rete coincide con il traffico smaltito dal nodo 3.

La matrice di instradamento $\underline{I}$ di dimensione 5×5 risulta:

$$\underline{I} = \begin{pmatrix} 0 & \beta & 1-\beta & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.11)$$

dove si è posto $I_{01} = \beta$ e $I_{02} = 1 - I_{01} = 1 - \beta$. La somma delle probabilità sulle righe è uguale a 1, dato che a partire dall'esterno o da un certo nodo il traffico deve essere instradato in un qualche altro nodo, o verso l'esterno.

Partendo dalla (5.10) e sfruttando la matrice di transizione (5.11) si ha:

$$\lambda_s E[T] = \frac{f_s}{C_0 - f_s} + \frac{\beta f_s}{C_1 - \beta f_s} + \frac{(1-\beta)f_s}{C_2 - (1-\beta)f_s} + \frac{f_s}{C_3 - f_s}$$

dove si è considerato $f_0 = f_s$ e $f_1 + f_2 = f_s(I_{01} + I_{02}) = f_s$.

Definendo le velocità di trasmissione normalizzate $C_i = f_s \hat{C}_i$, si ha:

$$\lambda_s E[T] = \frac{1}{\hat{C}_0 - 1} + \frac{\beta}{\hat{C}_1 - \beta} + \frac{1 - \beta}{\hat{C}_2 - 1 + \beta} + \frac{1}{\hat{C}_3 - 1} \quad (5.12)$$

Lo scopo dell'algoritmo è determinare il parametro ottimo di suddivisione del flusso β , cioè il valore di β che minimizza il tempo medio di permanenza nel sistema.

Si procede quindi derivando la (5.12):

$$\frac{d\lambda_s E[T]}{d\beta} = \frac{\hat{C}_1}{(\hat{C}_1 - \beta)^2} - \frac{\hat{C}_2}{(\hat{C}_2 - 1 + \beta)^2} = 0 \Rightarrow \frac{\hat{C}_2 - 1 + \beta}{\hat{C}_1 - \beta} = \sqrt{\frac{\hat{C}_2}{\hat{C}_1}}$$

da cui si ottiene il valore ottimo di β :

$$\beta \left(1 + \sqrt{\frac{\hat{C}_2}{\hat{C}_1}} \right) = \sqrt{\hat{C}_2 \hat{C}_1} + 1 - \hat{C}_2 \Rightarrow \beta = \left[\sqrt{\hat{C}_2 \hat{C}_1} + 1 - \hat{C}_2 \right] \left[1 + \sqrt{\frac{\hat{C}_2}{\hat{C}_1}} \right]^{-1} \quad (5.13)$$

La fattibilità richiede prima di tutto $C_0 > f_s$, $C_3 > f_s$ e $C_1 + C_2 > f_s$: i nodi in serie portano tutto il traffico e i rami paralleli devono poterlo smaltire insieme. Per $C > 0$,

$$\frac{d^2}{df^2} \frac{f}{C - f} = \frac{2C}{(C - f)^3} > 0, \quad 0 \leq f < C.$$

L'obiettivo è quindi strettamente convesso nella suddivisione e il punto stazionario ammissibile è l'unico minimo. Il valore di β deve quindi appartenere non soltanto a $[0, 1]$, ma anche al dominio di stabilità delle due code parallele:

$$\beta < \hat{C}_1, \quad 1 - \beta < \hat{C}_2.$$

In altri termini l'intervallo ammissibile è

$$\max(0, 1 - \hat{C}_2) < \beta < \min(1, \hat{C}_1),$$

con gli estremi chiusi soltanto quando non saturano una coda. Se il punto stazionario (5.13) cade fuori dall'intervallo ammissibile, l'ottimo vincolato si trova sul bordo ammissibile corrispondente.

Si nota che l'ottimo non dipende dalle caratteristiche dei nodi 0 e 3 (che infatti non hanno possibilità di scegliere l'instradamento) e che non dipende dal tasso degli arrivi ai singoli nodi, ma solo dal tasso di arrivo dall'esterno e dalle velocità sulle linee (si ricorda la normalizzazione introdotta $C_i = f_s \hat{C}_i$).

In figura 5.13 viene riportato il coefficiente di instradamento (o di biforcazione) ottimo in funzione delle velocità di trasmissione normalizzate dei primi due nodi come valutato nella (5.13). Si osserva che il flusso viene incanalato nel nodo 1 con il crescere della velocità di trasmissione del nodo 1 e con il calare della velocità di trasmissione del nodo 2.

5.2.8 Cut through

Normalmente un pacchetto viene completamente ricevuto da un nodo prima di essere trasmesso al successivo. Per aumentare le prestazioni del sistema si può pensare di iniziare a trasmettere il pacchetto non appena l'analisi della parte ricevuta consente di prendere una decisione sulla linea di uscita. Questa politica viene detta CUT THROUGH. In tal senso le informazioni

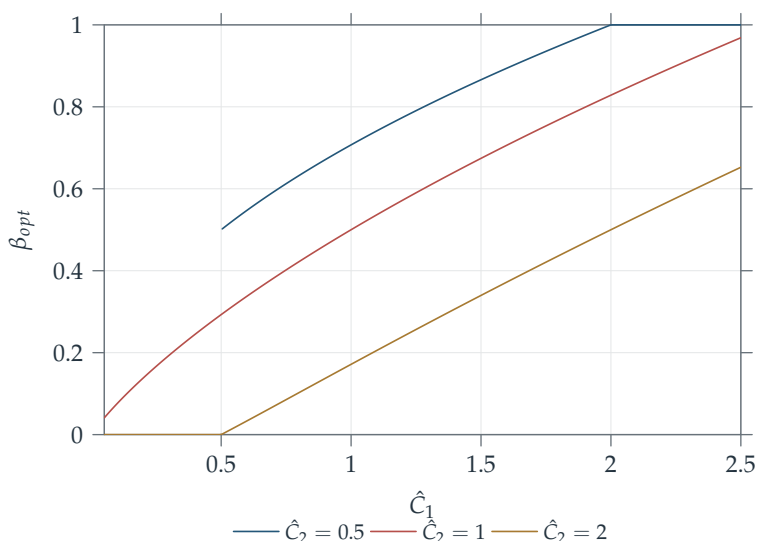


Figura 5.13. Coefficiente di biforcazione ottimo vincolato nel modello a quattro code.

relative al destinatario (indirizzo o circuito virtuale associato) devono essere collocate nella parte iniziale del pacchetto.

In assenza di altri pacchetti instradati sulle stesse linee il pacchetto rimane nella rete per un tempo dovuto a tre fattori: la trasmissione sulla linea più lenta, la ricezione della parte iniziale del pacchetto qualificante la sua destinazione ad ogni nodo per poter prendere la decisione sull'instradamento ed il ritardo di propagazione. In questo senso si tratta della tecnica che consente l'accesso, almeno per bassi volumi di traffico, più veloce possibile rispetto a quelle presentate precedentemente.

Il cut-through non definisce una politica di instradamento: riduce la latenza sovrapponendo ricezione e trasmissione. La contropartita è che un nodo può iniziare a inoltrare una trama prima di averne verificato l'integrità completa; eventuali errori scoperti alla fine richiedono quindi una gestione compatibile con questa scelta.

5.3 Controllo di flusso e della congestione

Quando il traffico offerto supera per un tempo sufficientemente lungo la capacità delle risorse attraversate, le code crescono, il ritardo aumenta e i buffer finiti iniziano a perdere pacchetti. Se la reazione alla perdita consiste in ulteriori ritrasmissioni non coordinate, il carico può aumentare proprio mentre la capacità utile diminuisce. Si entra allora in congestione.

La figura 5.14 rappresenta qualitativamente tre comportamenti. Nel caso ideale il throughput segue il carico offerto fino alla capacità e poi satura. Un controllo efficace cerca di avvicinarsi a questo comportamento limitando il carico prima che le code diventino persistentemente sature. Senza controllo, invece, la crescita di perdita e ritrasmissioni può ridurre il traffico utile consegnato anche mentre il traffico immesso continua a crescere. La figura è concettuale e non rappresenta una legge quantitativa universale.

Il controllo di flusso e il controllo della congestione agiscono su cause differenti. Il primo regola quanto un mittente può avere in volo rispetto alla capacità del ricevitore o di una risorsa a valle. Il secondo usa informazioni sullo stato della rete per limitare o ridistribuire il carico complessivo. I meccanismi possono essere *end-to-end*, quando la decisione è presa agli estremi della comunicazione, oppure locali e *hop-to-hop*, quando un nodo esercita pressione verso monte perché non riesce a smaltire il traffico. Questa propagazione all'indietro prende il nome di backpressure.

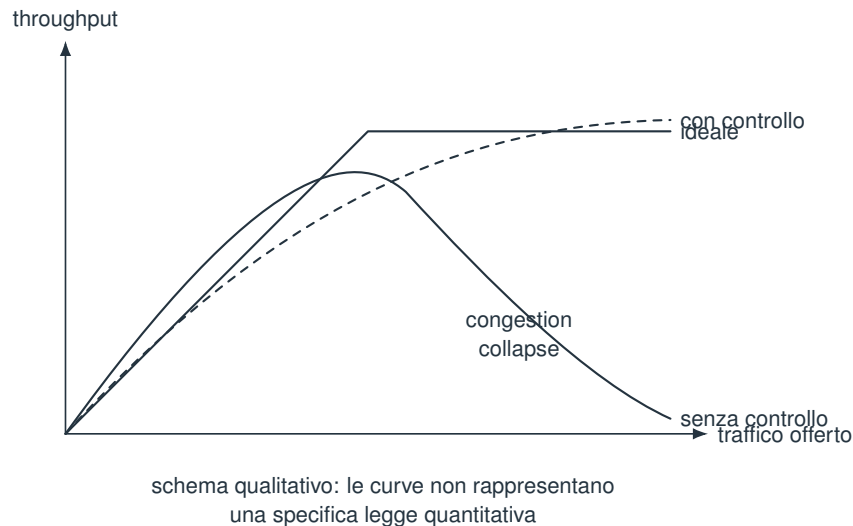


Figura 5.14. Comportamento qualitativo del throughput al crescere del carico offerto.

Una politica di controllo può agire in più modi: limitare il numero di pacchetti contemporaneamente in volo, ridurre il tasso di emissione, rifiutare nuovi flussi o nuovi pacchetti quando una soglia è superata, o deviare traffico verso risorse meno cariche. Il criterio corretto dipende dal tipo di servizio: throughput, ritardo, equità e perdita non sono in generale ottimizzabili tutti contemporaneamente.

Nel seguito vengono analizzati due modelli quantitativi. Il primo limita a H il numero di pacchetti di una comunicazione presenti nella rete e conduce naturalmente a una rete di code chiusa. Il secondo protegge il traffico già ammesso riservando parte del buffer e bloccando prima il traffico nuovo. Non sono prescrizioni di protocollo, ma modelli utili a mostrare come un vincolo sul carico possa impedire la saturazione incontrollata.

5.3.1 Finestra mobile, introduzione

Una finestra mobile limita a H il numero massimo di pacchetti non ancora confermati appartenenti a una comunicazione. La numerazione dei pacchetti e i riscontri permettono alla finestra di avanzare quando i dati vengono consegnati. Il principio è indipendente da una specifica tecnologia di rete: H rappresenta semplicemente un limite al lavoro contemporaneamente in volo.

Per studiarne l'effetto sulle prestazioni si considera un modello idealizzato nel quale sono presenti esattamente H pacchetti che circolano fra M servitori. Il sistema diventa così una rete di code chiusa; il modello non descrive i dettagli di un particolare protocollo di riscontro, ma isola il legame tra finestra, throughput e ritardo.

Prendendo in esame una singola comunicazione si può pensare di schematizzare il sistema come riportato in figura 5.15, dove vengono evidenziati gli M sistemi a coda chiusi che devono ospitare H pacchetti.

Si procede quindi con l'analisi generale delle reti di code chiuse, rimandando alla sezione 5.3.3 l'analisi quantitativa della finestra mobile.

5.3.2 Reti di code chiuse

In una rete di code chiusa non vi sono arrivi dall'esterno né partenze verso l'esterno: un numero fissato H di clienti circola tra gli M nodi. Il modello è utile per rappresentare un limite al lavoro in volo, perché sostituisce la creazione e la rimozione dei pacchetti con la

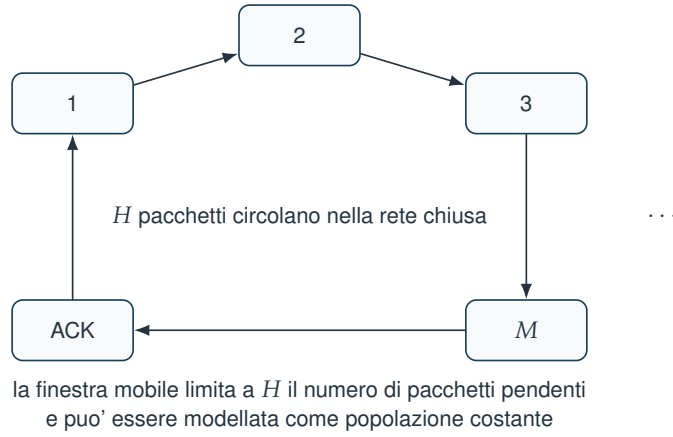


Figura 5.15. Schematizzazione di un sistema tramite reti di code chiuse.

circolazione di H entità indistinguibili. Il contenuto dei pacchetti non è rilevante ai fini del modello prestazionale.

Si considerino M servitori esponenziali con tassi μ_i . Dopo il servizio al nodo i , un cliente viene inviato al nodo j con probabilità p_{ij} , con $\sum_j p_{ij} = 1$. Si assumono tempi di servizio indipendenti a ogni visita, routing indipendente dallo stato, nessun blocco e una matrice di routing irriducibile. I clienti possono così circolare fra tutti i nodi e la classe di stati a popolazione H ha un'unica distribuzione stazionaria. Siano $\lambda_i > 0$ coefficienti che soddisfano le equazioni relative di traffico

$$\lambda_j = \sum_{i=1}^M \lambda_i p_{ij}, \quad j = 1, \dots, M. \quad (5.14)$$

Poiché non esiste un flusso esterno che ne fissi la scala, se $\{\lambda_i\}$ è una soluzione lo è anche $\{c\lambda_i\}$ per ogni $c > 0$. Solo i rapporti tra questi coefficienti hanno significato.

Teorema 9. Per una rete chiusa esponenziale con H clienti e coefficienti λ_i che soddisfano la (5.14), la distribuzione stazionaria è

$$P(\underline{n}) = \frac{1}{g(H, M)} \prod_{i=1}^M \left(\frac{\lambda_i}{\mu_i} \right)^{n_i}, \quad \sum_{i=1}^M n_i = H, \quad (5.15)$$

dove

$$g(H, M) = \sum_{\substack{\underline{n}: n_i \geq 0 \\ \sum_i n_i = H}} \prod_{i=1}^M \left(\frac{\lambda_i}{\mu_i} \right)^{n_i}. \quad (5.16)$$

Per dimostrare il risultato, indichiamo con $\pi(\underline{n})$ il membro destro della (5.15). Se $p_{ii} > 0$, i completamenti che tornano allo stesso nodo non cambiano lo stato: possono essere inclusi come eventi virtuali in entrambi i membri del bilancio, dove si cancellano. Con questa convenzione Dallo stato $\underline{n}$ è possibile uscire quando termina un servizio in un nodo non vuoto; il tasso complessivo di uscita è

$$\pi(\underline{n}) \sum_{i: n_i > 0} \mu_i.$$

Lo stato $\underline{n}$ può essere raggiunto mediante un completamento al nodo j seguito dall'instradamento verso un nodo i con $n_i > 0$. Lo stato precedente è $\underline{n} - \underline{1}_i + \underline{1}_j$, e quindi il tasso

complessivo di ingresso è

$$\sum_{i:n_i>0} \sum_{j=1}^M \mu_j p_{ji} \pi(\underline{n} - \underline{1}_i + \underline{1}_j).$$

Per la forma prodotto proposta,

$$\frac{\pi(\underline{n} - \underline{1}_i + \underline{1}_j)}{\pi(\underline{n})} = \frac{\lambda_j / \mu_j}{\lambda_i / \mu_i}.$$

Dividendo il tasso di ingresso per $\pi(\underline{n})$ si ottiene

$$\sum_{i:n_i>0} \frac{\mu_i}{\lambda_i} \sum_{j=1}^M \lambda_j p_{ji}.$$

La (5.14) trasforma la somma interna in λ_i ; il tasso di ingresso diventa pertanto

$$\sum_{i:n_i>0} \mu_i,$$

uguale al tasso di uscita. Le equazioni di bilancio globale sono quindi soddisfatte. La costante $g(H, M)$ normalizza la distribuzione sullo spazio finito degli stati con $\sum_i n_i = H$, completando la dimostrazione.

Il throughput del nodo i è il numero medio di completamenti di servizio per unità di tempo. Poiché il servitore lavora con tasso μ_i quando $n_i \geq 1$,

$$S_i = \mu_i P[n_i \geq 1].$$

È una media incondizionata nel regime stazionario, non il tasso condizionato al fatto che il nodo sia occupato. Qui S_i si misura in completamenti/s, mentre il throughput S del capitolo 4 era adimensionale; per ottenere un valore normalizzato si divide S_i per il tasso nominale pertinente.

La funzione $g(H, M)$ può essere calcolata senza enumerare tutti gli stati mediante la ricorrenza seguente.

Teorema di Buzen

Teorema 10. *La funzione g , costante di normalizzazione della rete chiusa, è ottenibile in modo ricorsivo mediante l'algoritmo di convoluzione di Buzen, a partire dalle condizioni iniziali:*

$$\begin{aligned} g(h, 1) &= \left(\frac{\lambda_1}{\mu_1} \right)^h & h = 0, 1, \dots, H \\ g(0, m) &= 1 & m = 1, 2, \dots, M \end{aligned} \tag{5.17}$$

secondo la formula:

$$g(h, m) = g(h, m-1) + \frac{\lambda_m}{\mu_m} g(h-1, m)$$

con $g(-1, m) = 0$, $g(h, 0) = 0$ per $h \geq 1$ e $g(0, 0) = 1$.

Per quanto concerne le condizioni iniziali, queste si ricavano immediatamente dalla (5.16). Per la determinazione della formula ricorsiva si consideri l'insieme degli stati $\underline{n}$ dovuti ai primi m sistemi degli M presenti per un totale di h utenti:

$$\mathcal{S}(h, m) = \left\{ \underline{n} \mid \sum_{i=1}^m n_i = h \right\}$$

allora:

$$g(h, m) = \sum_{\underline{n} \in \mathcal{S}(h, m)} \prod_{i=1}^m \left(\frac{\lambda_i}{\mu_i} \right)^{n_i} \quad (5.18)$$

Si procede distinguendo il caso con $n_m = 0$ dal caso $n_m > 0$:

$$g(h, m) = \sum_{\substack{\underline{n} \in \mathcal{S}(h, m) \\ n_m = 0}} \prod_{i=1}^m \left(\frac{\lambda_i}{\mu_i} \right)^{n_i} + \sum_{\substack{\underline{n} \in \mathcal{S}(h, m) \\ n_m > 0}} \prod_{i=1}^m \left(\frac{\lambda_i}{\mu_i} \right)^{n_i}$$

Il primo termine non contiene utenti relativi all' m -esimo sistema mentre per il secondo termine si mette in luce il contributo λ_m / μ_m relativo proprio all' m -esimo sistema:

$$g(h, m) = \sum_{\underline{n} \in \mathcal{S}(h, m-1)} \prod_{i=1}^{m-1} \left(\frac{\lambda_i}{\mu_i} \right)^{n_i} + \frac{\lambda_m}{\mu_m} \sum_{\substack{\underline{n} \in \mathcal{S}(h, m) \\ n_m > 0}} \left(\frac{\lambda_m}{\mu_m} \right)^{n_m-1} \prod_{i=1}^{m-1} \left(\frac{\lambda_i}{\mu_i} \right)^{n_i}$$

Il primo termine, secondo la definizione (5.18), risulta $g(h, m-1)$, mentre per il secondo è possibile trovare una forma compatta osservando che n_m deve essere calato di 1 come conseguenza del raccoglimento λ_m / μ_m e questo calo si ripercuote direttamente su h . Infatti, la scomposizione dei due termini li rende disgiunti e quindi una variazione nel numero degli utenti h nel secondo termine è relativa solo a n_m .

$$g(h, m) = g(h, m-1) + \frac{\lambda_m}{\mu_m} \sum_{\underline{n} \in \mathcal{S}(h-1, m)} \prod_{i=1}^m \left(\frac{\lambda_i}{\mu_i} \right)^{n_i} = g(h, m-1) + \frac{\lambda_m}{\mu_m} g(h-1, m)$$

Questo dimostra il teorema di Buzen.

Parametri caratteristici delle reti di code chiuse

La probabilità che la coda i abbia almeno k utenti è:

$$P[n_i \geq k] = \sum_{\substack{\underline{n} \in \mathcal{S}(H, M) \\ n_i \geq k}} P(\underline{n}) = \sum_{\substack{\underline{n} \in \mathcal{S}(H, M) \\ n_i \geq k}} \frac{1}{g(H, M)} \prod_{j=1}^M \left(\frac{\lambda_j}{\mu_j} \right)^{n_j}$$

Raccogliendo $(\lambda_i / \mu_i)^k$ si ha:

$$P[n_i \geq k] = \frac{\left(\frac{\lambda_i}{\mu_i} \right)^k}{g(H, M)} \sum_{\substack{\underline{n} \in \mathcal{S}(H, M) \\ n_i \geq k}} \left(\frac{\lambda_i}{\mu_i} \right)^{n_i-k} \prod_{j=1; j \neq i}^M \left(\frac{\lambda_j}{\mu_j} \right)^{n_j}$$

si procede cambiando variabile con la sostituzione $n'_i = n_i - k$ che porta a $n'_i \geq 0$; inoltre, considerando il numero degli utenti abbassato di k unità:

$$P[n_i \geq k] = \frac{\left(\frac{\lambda_i}{\mu_i}\right)^k}{g(H, M)} \sum_{\underline{n} \in S(H-k, M)} \prod_{j=1}^M \left(\frac{\lambda_j}{\mu_j}\right)^{n_j}$$

Quindi, la probabilità che la coda i abbia almeno k utenti è:

$$P[n_i \geq k] = \left(\frac{\lambda_i}{\mu_i}\right)^k \frac{g(H-k, M)}{g(H, M)} \quad (5.19)$$

Questa relazione implica che lo stato di ogni sistema è completamente descritto tramite la funzione g , avendo specificato i tassi di arrivo e di servizio.

Il numero medio degli utenti nel sistema i si ottiene osservando che:

$$E[n_i] = \sum_{k=0}^H k P[n_i = k] = \sum_{k=0}^H k (P[n_i \geq k] - P[n_i \geq k+1]) = \sum_{k=1}^H P[n_i \geq k]$$

dove l'ultima uguaglianza si verifica grazie alle cancellazioni dei termini che si presentano successivamente ed alla condizione $P[n_i \geq H+1] = 0$. Sostituendo la (5.19) si ottiene:

$$E[n_i] = \frac{1}{g(H, M)} \sum_{k=1}^H \left(\frac{\lambda_i}{\mu_i}\right)^k g(H-k, M)$$

Il throughput del sistema i , cioè il numero medio di completamenti di servizio per unità di tempo, si ottiene moltiplicando il tasso di servizio μ_i per la probabilità che il servitore sia occupato, $P[n_i \geq 1]$:

$$S_i = \mu_i P[n_i \geq 1] = \lambda_i \frac{g(H-1, M)}{g(H, M)}$$

Si osservi che se $M = 1$, cioè nella rete è presente un solo sistema a coda che si auto-alimenta, sfruttando le condizioni iniziali (5.17), si ha:

$$S_i = S_1 = \lambda_1 \frac{g(H-1, 1)}{g(H, 1)} = \lambda_1 \frac{\left(\frac{\lambda_1}{\mu_1}\right)^{H-1}}{\left(\frac{\lambda_1}{\mu_1}\right)^H} = \mu_1$$

cioè con un unico sistema a coda il throughput coincide con il tasso di smaltimento μ_1 .

Se $H = 0$, cioè non vi sono pacchetti circolanti nel sistema:

$$S_i = \lambda_i \frac{g(-1, M)}{g(0, M)} = 0 \quad \forall i$$

cioè il throughput su ogni nodo è nullo dato che non c'è nulla da smaltire.

5.3.3 Finestra mobile, prestazioni

In questa sezione, alla luce della teoria svolta sulle reti di code chiuse, viene completata l'analisi del controllo di flusso a finestra mobile. Si considera la rete simmetrica di figura 5.15, con M servitori aventi tutti tasso μ . Nella rete chiusa i coefficienti λ_i delle equazioni di traffico sono visit ratio definiti a meno di un fattore di scala comune; nel caso simmetrico possiamo quindi sceglierli, per normalizzazione, pari a $\lambda_i = \mu_i = \mu$. Ne segue che tutti i rapporti λ_i / μ_i sono unitari. Questa è una scelta di normalizzazione del modello chiuso, non l'affermazione che ogni coda aperta lavori al limite di stabilità.

Quindi, il throughput di un generico nodo risulta:

$g(h, m)$	1	2	3	4	5	6
0	1	1	1	1	1	1
1	1	2	3	4	5	6
2	1	3	6	10	15	21
3	1	4	10	20	35	56
4	1	5	15	35	70	126

Tabella 5.2. Tabella della funzione g per $\lambda_i = \mu_i$, h in verticale, m in orizzontale.

$$S = \mu \frac{g(H-1, M)}{g(H, M)} \quad (5.20)$$

Dal teorema di Buzen, quando tutti i rapporti λ_i/μ_i sono unitari, $g(h, m)$ conta semplicemente il numero di vettori di interi non negativi $(n_1, \dots, n_m)$ la cui somma è h . Con l'argomento delle separazioni ("stars and bars"), si rappresentano i h clienti con h simboli e si inseriscono $m-1$ separatori per dividerli tra le m code. Le posizioni complessive sono $h+m-1$ e basta scegliere quelle dei $m-1$ separatori. Pertanto

$$g(h, m) = \binom{h+m-1}{m-1} = \frac{(h+m-1)!}{(m-1)! h!} \quad (5.21)$$

Applicando la (5.21) nella (5.20) si ha:

$$\frac{S}{\mu} = \frac{(H+M-2)!}{(M-1)!(H-1)!} \frac{(M-1)! H!}{(H+M-1)!} = \frac{H}{H+M-1} \quad (5.22)$$

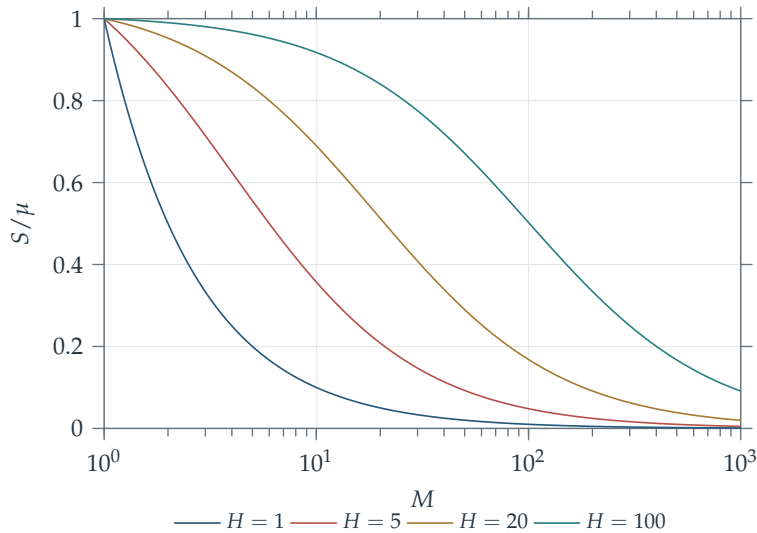


Figura 5.16. Throughput per sistema a finestra mobile.

Per $H \geq 1$, tagliamo idealmente l'anello dopo un nodo di riferimento e contiamo ogni passaggio come fine di un giro e inizio del successivo. Il tasso di questi passaggi è S , mentre nella rete rimangono H clienti: non si tratta di un tasso di arrivi esterni, che in una rete chiusa è nullo. Applicando Little ai giri si ottiene il tempo medio di un giro completo:

$$E[T] = \frac{E[k]}{S} = \frac{H}{S} = \frac{H + M - 1}{\mu}$$

Per $M > 1$ si può eliminare H dalla (5.22) e legare il tempo medio al throughput con $0 < S < \mu$. Le curve continue interpolano i valori realizzabili, che corrispondono a H intero. Per $M = 1$ si ha invece $S = \mu$ per ogni $H \geq 1$ ed $E[T] = H/\mu$; la divisione per $1 - S/\mu$ non è ammessa. Per $M > 1$:

$$H = \frac{M - 1}{\frac{\mu}{S} - 1} \Rightarrow \mu E[T] = \mu \frac{H}{S} = \frac{\mu}{S} \frac{M - 1}{\frac{\mu}{S} - 1} = \frac{M - 1}{1 - \frac{S}{\mu}} \quad (5.23)$$

In figura 5.16 è riportato il throughput sul generico nodo in funzione del numero dei nodi M al variare della dimensione della finestra H . Si noti come, una volta fissato H , S/μ cali con l'aumentare del numero dei nodi presenti nella rete. Questo avviene in quanto, a parità di numero di pacchetti circolanti nella rete chiusa, aumentando M diminuisce la probabilità che un dato nodo sia occupato. Inoltre, fissato M , S/μ cresce con H perché cresce la probabilità che il servitore trovi almeno un pacchetto da processare.

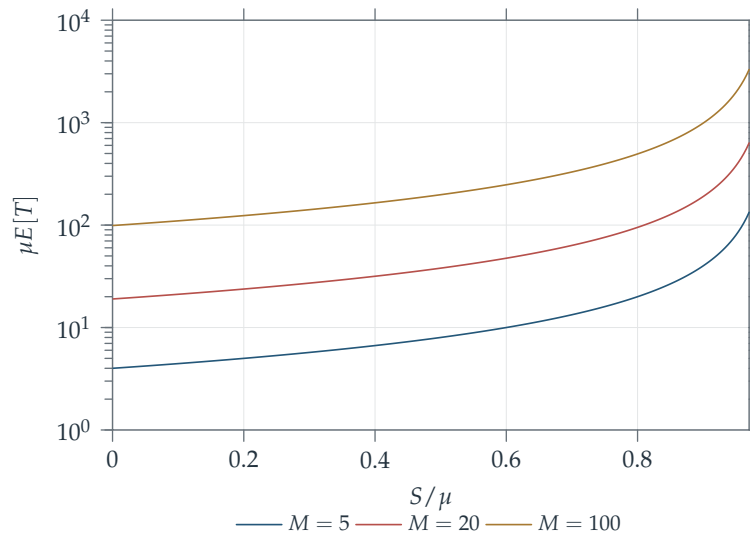


Figura 5.17. Ritardo in funzione del throughput al variare di M .

I valori ricavati dall'equazione (5.23) relativa al tempo medio, sono riportati in figura 5.17, per diverse scelte di M . Il ritardo medio aumenta sia con l'aumentare di H che con l'aumentare di M .

Dimensione ottimale della finestra

È interessante cercare di trovare la dimensione ottimale della finestra al variare del numero dei sistemi a coda che il pacchetto incontra nella rete. A questo scopo è possibile definire la figura di merito Γ , data dal rapporto tra il throughput ed il ritardo medio. Si cerca quindi il massimo di Γ , che rappresenta un compromesso fra throughput e ritardo, non la massimizzazione del primo e la minimizzazione del secondo separatamente. Nel caso $M > 1$:

$$\Gamma = \frac{\frac{S}{\mu}}{\mu E[T]} = \frac{S}{\mu} \frac{1 - \frac{S}{\mu}}{M - 1}$$

Si procede valutando il massimo del parametro di merito Γ rispetto al throughput:

$$\frac{d\Gamma}{d\frac{S}{\mu}} = \left[\frac{1 - \frac{S}{\mu}}{M-1} - \frac{\frac{S}{\mu}}{M-1} \right] = 0 \Rightarrow \frac{S}{\mu} = \frac{1}{2}$$

ed utilizzando la (5.22):

$$\frac{H}{H+M-1} = \frac{1}{2} \Rightarrow H = M-1$$

In conclusione, con la specifica figura di merito $\Gamma = (S/\mu)/(\mu E[T])$ scelta in questa sezione e nel modello simmetrico considerato, l'ottimo è $H = M-1$. Non si tratta quindi di una dimensione universalmente ottima della finestra. Per $M=1$, $\Gamma = 1/H$ e l'ottimo ammissibile è invece $H=1$. In generale, cambiando il criterio di costo, i ritardi di propagazione, le capacità o la struttura della rete cambia in generale anche il valore ottimale.

5.3.4 Blocco sull'ingresso

Un'altra classe di politiche limita l'ammissione direttamente presso una risorsa congestionata. Il modello seguente distingue il traffico nuovo proveniente dall'esterno da quello già ammesso e in transito, privilegiando quest'ultimo per evitare che lavoro già sostenuto dalla rete venga scartato inutilmente.

Sia n_S il numero di pacchetti nella coda dovuto ad arrivi dal mondo esterno e n_R il numero di pacchetti nella coda dovuto ad arrivi da altri sistemi a coda componenti la rete. Si suppone che la coda abbia un buffer limitato con Y posizioni.

Con queste ipotesi un possibile controllo di flusso consiste nel bloccare i pacchetti provenienti dal mondo esterno se questi superano una soglia prefissata $N_S < Y$, cioè il blocco avviene se c'è un arrivo con $n_S \geq N_S$ oppure quando il buffer è già pieno, $n_S + n_R = Y$. I pacchetti già presenti nella rete vengono privilegiati e quindi possono accedere a tutto il buffer e vengono bloccati quando $n_S + n_R \geq Y$.

L'analisi di questa metodologia di controllo di flusso è possibile per via analitica ma risulta piuttosto complessa. Occorre infatti notare che lo stato di ogni singolo sistema a coda è descritto dalla almeno dalla coppia (n_S, n_R) . Con FIFO può essere necessario conoscere anche l'ordine delle classi in coda: i soli conteggi non identificano quale classe partirà. Con una selezione casuale o altra disciplina che renda i conteggi sufficienti, le equazioni di bilancio tra tasso entrante ed uscente per ogni singolo stato dovrebbero essere riscritte utilizzando un modello bidimensionale. Si potrebbe procedere estendendo l'approccio presentato nella sezione 2.6, con le complicazioni dovute all'approccio bidimensionale. Partendo dalla scrittura delle equazioni bidimensionali si potrebbe poi ricavare la probabilità di blocco per i pacchetti presenti nella rete, P_{BR} , e per i pacchetti provenienti dal mondo esterno P_{BS} . Queste grandezze permettono di valutare il throughput e di ottimizzare la soglia N_S avendo fissato opportunamente le condizioni al contorno.

Una possibile politica alternativa consiste nel bloccare i pacchetti in ingresso dall'esterno se il numero di pacchetti complessivo supera la soglia N_S , cioè se $k = n_S + n_R \geq N_S$ ed è analizzabile con un approccio monodimensionale. Analogamente alla politica di controllo di flusso prima descritta, i pacchetti circolanti nella rete vengono privilegiati e possono occupare il buffer per tutta la sua capacità.

Per analizzare quantitativamente questa seconda politica si introduce un modello omogeneo e disaccoppiato. Tutti i nodi hanno lo stesso tasso di servizio μ e la stessa capacità Y ; gli arrivi esterni e quelli dovuti ai pacchetti già circolanti vengono approssimati come processi poissoniani indipendenti. Inoltre, la probabilità che un tentativo di inoltro verso il nodo successivo sia respinto viene trattata come indipendente dai tentativi precedenti e uguale alla probabilità stazionaria P_{BR} . Queste ipotesi trasformano la rete interagente in un problema a

punto fisso su un nodo rappresentativo: il risultato che segue è quindi esatto per il modello disaccoppiato, non per una rete arbitraria con correlazioni tra nodi.

Sia $\lambda_S = \lambda_s / M$ il tasso offerto al nodo dal mondo esterno e sia λ_R il tasso medio di tentativi di inoltro *in uscita* prodotti da ciascun nodo. Il tasso offerto in ingresso da altri nodi sarà invece $\lambda_r = (1 - I_d)\lambda_R$. Il modello effettua prima una prova di accettazione dell'inoltro, con probabilità $1 - P_{BR}$; soltanto dopo un'accettazione decide, con probabilità I_d , che il pacchetto abbia completato il percorso e lasci la rete. Un inoltro respinto lascia il pacchetto al nodo di partenza, dove richiede un nuovo servizio. Questa regola applica la stessa probabilità di blocco anche all'ultimo inoltro. Non descrive una rete in cui ogni uscita verso l'esterno è sempre libera: in quel caso il tasso efficace sarebbe diverso e le conclusioni di congestione andrebbero ricalcolate.

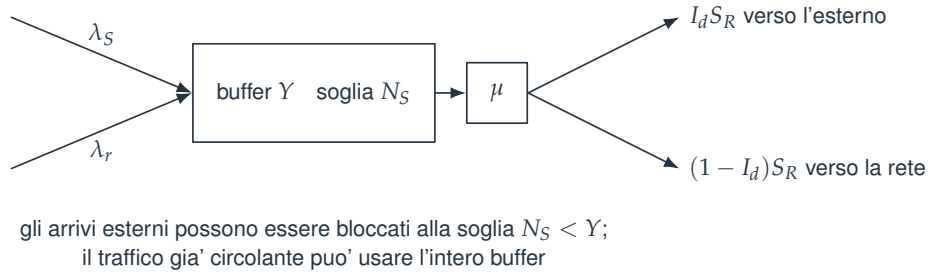


Figura 5.18. Nodo rappresentativo del modello con limitazione dell'ingresso.

Indichiamo con

$$S_S = \lambda_S(1 - P_{BS}), \quad S_R = \lambda_R(1 - P_{BR})$$

rispettivamente il flusso esterno ammesso e il flusso di inoltri completati per nodo. Il traffico interno ammesso in ingresso è $(1 - I_d)S_R$. Nell'aggregato omogeneo il traffico esterno ammesso deve compensare il traffico che lascia definitivamente la rete. La condizione di consistenza è pertanto

$$S_S = I_d S_R,$$

ossia

$$\lambda_S(1 - P_{BS}) = I_d \lambda_R(1 - P_{BR}). \quad (5.24)$$

Il traffico circolante che tenta di entrare nel nodo successivo è associato alla frazione che non lascia la rete:

$$\lambda_r = \lambda_R(1 - I_d).$$

Nel modello disaccoppiato ogni completamento di servizio richiede un tempo esponenziale di parametro μ e produce un inoltro accettato con probabilità $1 - P_{BR}$. La marcatura indipendente dei completamenti equivale quindi a un processo di completamenti efficaci con tasso

$$\hat{\mu} = \mu(1 - P_{BR}).$$

La stessa relazione si ottiene osservando che il numero medio di tentativi necessario per un inoltro accettato è $1/(1 - P_{BR})$. L'indipendenza fra tentativi è una delle ipotesi del modello a punto fisso.

Sia k il numero totale di pacchetti presenti nel nodo. Il traffico esterno è ammesso soltanto per $k < N_S$, mentre il traffico già circolante può usare l'intero buffer e viene respinto soltanto nello stato Y . Il nodo rappresentativo è quindi una catena di nascita e morte con

$$\lambda_k = \begin{cases} \lambda_x = \lambda_r + \lambda_S, & 0 \leq k < N_S, \\ \lambda_r, & N_S \leq k < Y, \\ 0, & k \geq Y, \end{cases} \quad \mu_k = \begin{cases} 0, & k = 0, \\ \hat{\mu}, & 1 \leq k \leq Y. \end{cases}$$

Definendo

$$\hat{A}_x = \frac{\lambda_x}{\hat{\mu}}, \quad \hat{A}_r = \frac{\lambda_r}{\hat{\mu}},$$

le equazioni locali di bilancio forniscono

$$P_k = \begin{cases} P_0 \hat{A}_x^k, & 0 \leq k \leq N_S, \\ P_0 \hat{A}_x^{N_S} \hat{A}_r^{k-N_S}, & N_S < k \leq Y, \\ 0, & k > Y. \end{cases}$$

Per evitare singolarità artificiali nelle somme geometriche, introduciamo

$$Q_n(x) = \sum_{j=0}^n x^j, \quad Q_{-1}(x) = 0.$$

In particolare $Q_n(1) = n + 1$. La normalizzazione diventa

$$P_0 = \left[Q_{N_S}(\hat{A}_x) + \hat{A}_x^{N_S} \hat{A}_r Q_{Y-N_S-1}(\hat{A}_r) \right]^{-1}.$$

Questa forma resta valida anche quando $\hat{A}_x = 1$ o $\hat{A}_r = 1$, senza dover introdurre casi limite separati.

Poiché nel modello gli arrivi sono poissoniani, vale la proprietà PASTA dimostrata nel capitolo 2. Un pacchetto esterno viene respinto quando trova $k \geq N_S$, mentre un pacchetto già circolante viene respinto soltanto quando trova il buffer pieno. Ne segue

$$P_{BS} = P_0 \hat{A}_x^{N_S} Q_{Y-N_S}(\hat{A}_r), \quad (5.25)$$

e

$$P_{BR} = P_Y = P_0 \hat{A}_x^{N_S} \hat{A}_r^{Y-N_S}. \quad (5.26)$$

La grandezza di interesse è il throughput del traffico esterno, normalizzato rispetto al tasso di servizio nominale:

$$\frac{S_S}{\mu} = A_S(1 - P_{BS}), \quad A_S = \frac{\lambda_S}{\mu}.$$

Dalla condizione di consistenza (5.24), ponendo $A_R = \lambda_R/\mu$, si ricava

$$A_R = \frac{A_S(1 - P_{BS})}{I_d(1 - P_{BR})}.$$

Di conseguenza

$$\begin{aligned} \hat{A}_r &= \frac{\lambda_r}{\hat{\mu}} = A_S \frac{(1 - I_d)(1 - P_{BS})}{I_d(1 - P_{BR})^2}, \\ \hat{A}_x &= \frac{\lambda_r + \lambda_S}{\hat{\mu}} \\ &= A_S \frac{1 - P_{BS} + I_d P_{BS} - I_d P_{BR}}{I_d(1 - P_{BR})^2}. \end{aligned} \quad (5.27)$$

Le equazioni (5.25), (5.26) e (5.27) costituiscono un sistema non lineare a punto fisso nelle probabilità di blocco. Non si tratta quindi di sostituire direttamente il carico in una formula chiusa: per ogni A_S , I_d , N_S e Y occorre trovare una soluzione autoconsistente. I grafici seguenti sono generati numericamente da questo sistema, usando continuazione sul carico, iterazione smorzata e verifica esplicita del residuo delle equazioni. Un'iterazione che raggiunge il limite di passi senza soddisfare le equazioni non viene accettata come soluzione.

Esiste inoltre un vincolo importante nel caso $N_S = Y$. Dal bilancio locale, il tasso ammesso è $S_S + (1 - I_d)S_R = S_R$, mentre il tasso di partenza è $\mu(1 - P_{BR})(1 - P_0)$. Per $P_{BR} < 1$ ne segue $\lambda_R = \mu(1 - P_0)$. Se le due classi hanno la stessa soglia, $P_{BS} = P_{BR}$ e la (5.24) implica

$$A_S = I_d(1 - P_0) < I_d.$$

Quindi per $A_S \geq I_d$ non esiste una soluzione interna con blocco strettamente minore di uno. Per $A_S \uparrow I_d$ il punto fisso tende al bordo $P_0 \rightarrow 0$, $P_{BR} \rightarrow 1$ e il throughput utile tende a zero. Il tratto $N_S = Y$ dei grafici viene mostrato soltanto per $A_S < I_d$; non si prolunga artificialmente il ramo sostituendo un denominatore nullo con un piccolo numero positivo.

La figura 5.19 mostra che una soglia intermedia può proteggere il traffico già ammesso e mantenere un throughput utile maggiore quando il carico cresce. Nel caso $N_S = Y$ il traffico nuovo non riceve alcuna protezione anticipata: nel modello a punto fisso l'aumento del blocco dei pacchetti circolanti riduce $\hat{\mu}$, genera ulteriori tentativi e può condurre a una regione di *congestion collapse*, nella quale il throughput utile tende quasi a zero. È il meccanismo qualitativo anticipato in figura 5.14; non va interpretato come una legge universale per ogni rete, ma come conseguenza delle ipotesi di retroazione del modello considerato.

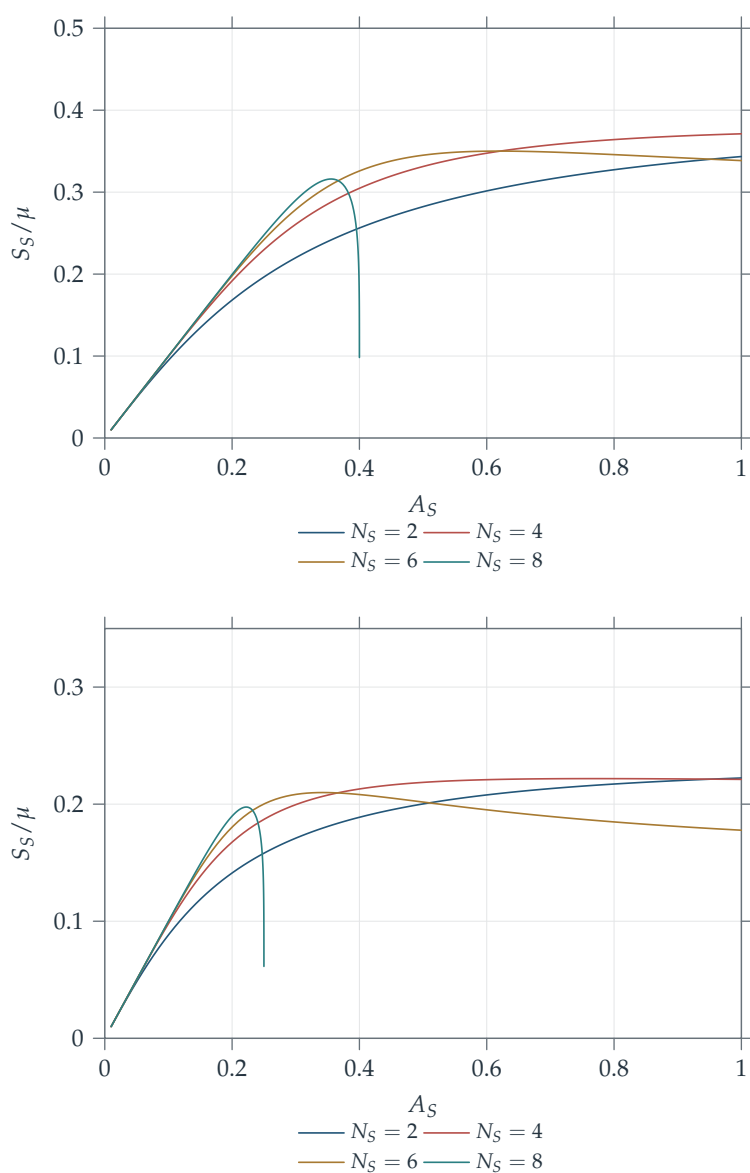


Figura 5.19. Throughput del modello a punto fisso con $Y = 8$ e diverse soglie N_S : $I_d = 0.4$ in alto, $I_d = 0.25$ in basso.

Reti locali e accesso al mezzo

Le reti locali mostrano in modo concreto come i principi studiati nei capitoli precedenti diventino meccanismi di collegamento: delimitazione delle trame, indirizzamento locale, controllo degli errori, accesso a un mezzo condiviso e commutazione. L'obiettivo non è catalogare tutte le varianti fisiche degli standard, ma isolare le scelte che hanno un valore generale e collegarle ai modelli quantitativi già introdotti.

Due casi sono particolarmente importanti. Ethernet permette di confrontare il modello a mezzo condiviso con la rete commutata full-duplex oggi normale; IEEE 802.11 mostra invece perché un mezzo radio richieda collision avoidance, riscontri e backoff. I sistemi a token vengono mantenuti come modelli di accesso controllato, perché consentono di derivare in modo trasparente il legame tra ciclo, propagazione, carico e attesa, senza conservare dettagli di tecnologie che non aggiungono valore concettuale.

6.1 Framing e controllo di collegamento

Un collegamento a livello 2 deve trasformare un flusso di simboli in unità riconoscibili, dette trame. Servono quindi almeno un criterio di delimitazione, un meccanismo per identificare sorgente o destinazione quando necessario, un controllo d'integrità e, nei servizi affidabili, numerazione e riscontri.

Una tecnica classica di delimitazione è il *bit stuffing*, usata anche dalla famiglia HDLC. Una sequenza riservata marca il confine della trama; per impedire che la stessa sequenza compaia nei dati, il trasmettitore inserisce un bit convenzionale dopo una particolare successione di bit, e il ricevitore esegue l'operazione inversa. Il principio generale è importante: la trasparenza del framing richiede che ogni possibile contenuto applicativo possa essere trasportato senza essere confuso con i delimitatori di protocollo.

La numerazione delle trame permette inoltre di applicare le finestre scorrevoli e le strategie ARQ studiate nel capitolo 3. Un riscontro può essere trasmesso in una trama di controllo oppure incorporato nel traffico nella direzione opposta (*piggybacking*); in quest'ultimo caso si riduce l'overhead, ma si introduce un vincolo temporale perché il riscontro non può essere ritardato indefinitamente nell'attesa di dati da inviare.

6.2 Ritardo di propagazione

In questo capitolo vengono analizzati alcuni protocolli che possono essere visti come applicazioni di quanto presentato nel capitolo 4 dove il ritardo di propagazione, τ_p , è stato assunto costante per tutte le stazioni. Una tale assunzione è realizzabile, per esempio, in una stella con rami uguali; non è però una proprietà di una topologia generale: il ritardo di propagazione dipende infatti dalla distanza tra le stazioni e quindi dalla topologia.

Quando un modello prestazionale usa un unico ritardo τ_p al posto dei diversi ritardi tra coppie di stazioni, la sostituzione con $E[\tau_p]$ è un'approssimazione di primo ordine. Le formule che seguono calcolano invece esattamente tale media nelle ipotesi topologiche dichiarate. La media viene espressa in funzione del massimo ritardo di propagazione τ_M . In tutte le medie topologiche le coppie ordinate di stazioni distinte sono equiprobabili; con traffico non uniforme occorre pesare ogni coppia con la sua probabilità di comunicazione. La media non

sostituisce mai il massimo nei vincoli di sicurezza, come la rivelazione della collisione durante la trama Ethernet.

6.2.1 Topologia a Bus

Si consideri una rete con topologia a bus, con N stazioni equispaziate, $N \geq 2$, e con comunicazioni bidirezionali. Il ritardo di propagazione tra una stazione e la successiva è $\tau_M / (N - 1)$ ed in generale il ritardo tra la stazione i e la stazione j è:

$$\tau_{ij} = \frac{\tau_M}{N-1} |i - j| \quad i, j = 1, \dots, N$$

Il valor medio del ritardo di propagazione tra una stazione i e tutte le altre è:

$$E[\tau_{p_i}] = \frac{1}{(N-1)} \sum_{j=1; j \neq i}^N \tau_{ij}$$

Quindi, il ritardo medio tra due stazioni qualunque può essere valutato come segue:

$$E[\tau_p] = \frac{1}{N} \sum_{i=1}^N E[\tau_{p_i}] = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{j=1; j \neq i}^N \tau_{ij}$$

Il termine $N - 1$ al denominatore e la condizione $j \neq i$ permettono di considerare il fatto che non ha senso trasmettere pacchetti a se stessi. Utilizzando la definizione di τ_{ij} si ha:

$$E[\tau_p] = \frac{\tau_M}{N(N-1)^2} \left[\sum_{i=1}^N \sum_{j=1}^{i-1} (i-j) + \sum_{i=1}^N \sum_{j=i+1}^N (j-i) \right]$$

La prima coppia di sommatorie risulta:

$$\begin{aligned} \sum_{i=1}^N \sum_{j=1}^{i-1} (i-j) &= \sum_{i=1}^N \left[i(i-1) - \frac{i(i-1)}{2} \right] = \frac{1}{2} \sum_{i=1}^N [i^2 - i] \\ &= \frac{1}{2} \left[\frac{N(N+1)(2N+1)}{6} - \frac{N(N+1)}{2} \right] = \frac{(N-1)N(N+1)}{6} \end{aligned}$$

Mentre per la seconda coppia di sommatorie, procedendo con due sostituzioni del tipo $v = j - i$ e $q = N - i + 1$, si ha:

$$\sum_{i=1}^N \sum_{j=i+1}^N (j-i) = \sum_{i=1}^N \sum_{v=1}^{N-i} v = \sum_{i=1}^N \frac{(N-i)(N-i+1)}{2} = \frac{1}{2} \sum_{q=1}^N q(q-1)$$

che quindi coincide con il risultato trovato per la prima coppia di sommatorie.

Complessivamente il ritardo medio di propagazione risulta:

$$E[\tau_p] = \frac{2\tau_M}{N(N-1)^2} \frac{(N-1)N(N+1)}{6} = \frac{\tau_M}{3} \frac{(N+1)}{(N-1)} \rightarrow \frac{\tau_M}{3}$$

Questo risultato mostra che, considerando una distribuzione uniforme delle stazioni con un numero di stazioni sufficientemente elevato, il ritardo medio di propagazione per un bus è un terzo del ritardo massimo.

Il ritardo di propagazione normalizzato, α , ampiamente utilizzato nel capitolo 4, può essere sostituito con $E[\alpha]$ come approssimazione del modello prestazionale, non come identità: in generale $E[f(\alpha)] \neq f(E[\alpha])$. Nel caso di una topologia a bus con stazioni equispaziate risulta quindi:

$$E[\alpha] = \frac{\alpha_M}{3} \frac{N+1}{N-1} \xrightarrow{N \rightarrow \infty} \frac{\alpha_M}{3}.$$

dove $\alpha_M = \tau_M / E[t_F]$ è il ritardo massimo di propagazione normalizzato.

6.2.2 Topologia ad anello

Consideriamo ora un anello *diretto*: ogni stazione riceve da un vicino e ritrasmette verso il successivo, così che l'informazione percorra una sola direzione. Una realizzazione bidirezionale può essere ottenuta con due direzioni logiche o fisiche controrotanti; per la derivazione seguente è sufficiente studiarne una. Assumiamo stazioni equispaziate e includiamo in τ_M il massimo ritardo di propagazione lungo il percorso diretto, trascurando per ora il tempo di elaborazione introdotto dalle stazioni.

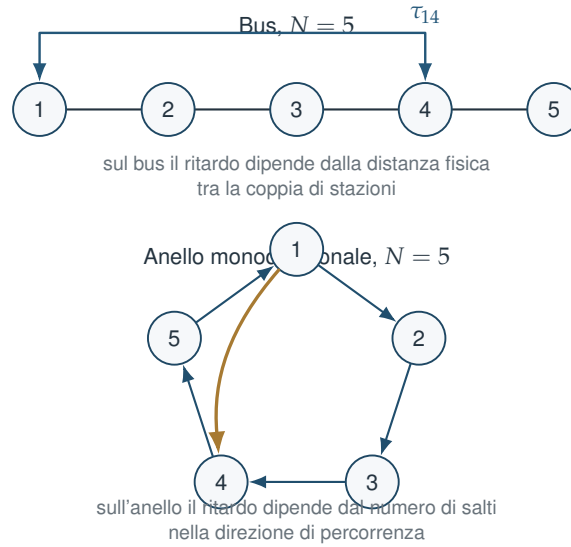


Figura 6.1. Confronto bus e anello, con $N = 5$.

La figura 6.1 riporta il confronto tra il caso a bus e la topologia ad anello, con 5 stazioni equamente distribuite in entrambi i casi. Il caso ad anello ha una topologia modulare che non è presente nel bus: nell'anello una traslazione degli indici modulo N conserva il numero di salti diretto fra le stazioni. Nel bus una traslazione che non attraversa un estremo conserva anch'essa la distanza, ma non esiste la stessa simmetria ciclica quando si supera il bordo. Sono gli estremi, non ogni spostamento della coppia, a rompere la simmetria.

Questo significa che nell'anello il ritardo medio di propagazione è investigabile prendendo come riferimento una particolare stazione i ; in particolare si prende a riferimento la stazione $i = 1$.

Il ritardo di propagazione tra la stazione 1 e la stazione j è:

$$\tau_{1j} = \frac{\tau_M}{N-1} (j-1) = \tau_j$$

Quindi:

$$\begin{aligned} E[\tau_{p_j}] &= \frac{1}{N-1} \sum_{j=2}^N \tau_j = \frac{\tau_M}{(N-1)^2} \sum_{j=2}^N (j-1) = \frac{\tau_M}{(N-1)^2} \sum_{j=1}^{N-1} j = \\ &= \frac{\tau_M}{(N-1)^2} \frac{(N-1)N}{2} = \frac{\tau_M}{2} \frac{N}{N-1} \rightarrow \frac{\tau_M}{2} \end{aligned}$$

Complessivamente il ritardo medio di propagazione risulta:

$$E[\tau_p] = \frac{\tau_M}{2} \frac{N}{N-1} \xrightarrow{N \rightarrow \infty} \frac{\tau_M}{2}. \quad (6.1)$$

Il ritardo medio di propagazione risulta superiore a quello dovuto ad una topologia a bus, a parità di ritardo massimo di propagazione. Questo risultato è spiegabile osservando che nell'anello una generica stazione i per comunicare con la stazione adiacente $i - 1$ deve effettuare tutto il giro dell'anello, mentre nel bus la comunicazione avviene direttamente.

È inoltre da osservare che il meccanismo di ricezione e ritrasmissione, presente ad ogni stazione, introduce un ulteriore ritardo, dovuto al processamento, che non è stato considerato in questa valutazione topologica.

Riassumendo, per le topologia ad anello, monodirezionali, con stazioni equispaziate, il ritardo di propagazione normalizzato è:

$$E[\alpha] = \frac{\alpha_M}{2} \frac{N}{N-1} \xrightarrow{N \rightarrow \infty} \frac{\alpha_M}{2}.$$

6.2.3 Topologie ad albero

In un albero il ritardo tra due stazioni è la somma dei ritardi dei rami appartenenti all'unico cammino che le collega. Non esiste però una formula universale esprimibile soltanto mediante N e τ_M : distribuzione delle stazioni, profondità, grado dei nodi e lunghezza dei rami cambiano la distribuzione delle distanze a parità di diametro massimo.

Se $d(i, j)$ è il ritardo sul cammino tra i e j , il valore medio per coppie ordinate distinte è semplicemente

$$E[\tau_p] = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{j \neq i}^N d(i, j).$$

Questa espressione è esatta per la topologia specificata; eventuali approssimazioni richiedono quindi un modello esplicito della forma dell'albero o della distribuzione dei nodi.

6.3 Architettura delle LAN IEEE 802

La famiglia IEEE 802 separa concettualmente le funzioni di collegamento dalle funzioni specifiche di accesso al mezzo. Il sottolivello MAC (Medium Access Control) definisce indirizzamento locale, formato delle trame e regole con cui una stazione usa il mezzo; funzioni di collegamento logico possono essere collocate al di sopra del MAC. Questa separazione è utile perché lo stesso livello di rete può appoggiarsi a tecnologie MAC molto diverse, per esempio Ethernet e WLAN.

Nelle LAN moderne la commutazione a livello 2 è altrettanto importante dell'accesso al mezzo: uno switch apprende quali indirizzi MAC sono raggiungibili dalle proprie porte e inoltra selettivamente le trame. La separazione dei domini di collisione rende quindi profondamente diverso il comportamento di una Ethernet commutata full-duplex da quello di un unico mezzo condiviso.

6.4 Ethernet e IEEE 802.3

Ethernet è oggi soprattutto una tecnologia commutata full-duplex: ogni porta di uno switch è collegata punto-punto e può trasmettere e ricevere senza contendere il mezzo con le altre porte. In questa configurazione CSMA/CD non interviene. Il modello a mezzo condiviso rimane però importante perché mostra con chiarezza il legame fisico tra propagazione, collisioni, dimensione minima della trama e backoff.

6.4.1 Trama Ethernet

La trama MAC contiene gli indirizzi di destinazione e sorgente, un campo Length/Type, i dati e un controllo d'integrità FCS basato su CRC. Gli indirizzi MAC ordinari occupano 6 byte. Il campo dati ha normalmente una MTU di 1500 byte; quando il contenuto è troppo corto viene aggiunto padding affinché la trama raggiunga la dimensione minima prevista dal MAC. Il campo Length/Type consente di interpretare il valore come lunghezza oppure come identificatore del protocollo superiore, a seconda dell'intervallo numerico utilizzato.

Preambolo e delimitatore iniziale servono alla sincronizzazione e alla rivelazione dell'inizio della trama sul mezzo fisico. Questi elementi mostrano una separazione importante: la struttura logica della trama è stabile, mentre codifica di linea, velocità e mezzo fisico dipendono dalla specifica variante Ethernet.

6.4.2 CSMA/CD come modello di mezzo condiviso

Su un mezzo half-duplex condiviso una stazione ascolta il canale prima di trasmettere e continua a osservarlo durante la trasmissione. Se rileva una collisione interrompe la trama, rende la collisione riconoscibile alle altre stazioni e rimanda il tentativo dopo un tempo di backoff casuale. Il backoff riduce la probabilità che le stesse stazioni collidano nuovamente in modo sincronizzato.

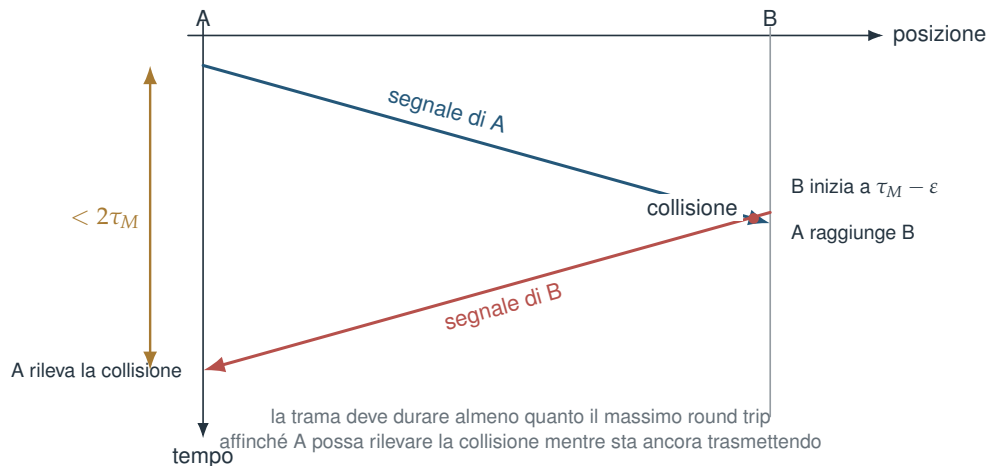


Figura 6.2. Caso limite per la rivelazione di una collisione su un mezzo condiviso.

Sia τ_M il massimo ritardo di propagazione unidirezionale tra due stazioni. Nel caso limite della figura 6.2, una stazione B inizia a trasmettere immediatamente prima che il segnale di A le arrivi. La perturbazione prodotta da B deve allora tornare fino ad A: la collisione può essere rivelata da A quasi $2\tau_M$ dopo l'inizio della sua trasmissione. Perché A sia ancora in trasmissione in quel momento deve quindi valere

$$t_F \geq 2\tau_M. \quad (6.2)$$

La relazione non è un'approssimazione: è la conseguenza geometrica del modello con ritardo massimo τ_M . Lo standard traduce questo vincolo in uno slot time e in una dimensione minima della trama.

Dopo una collisione il binary exponential backoff seleziona un numero casuale di slot prima di un nuovo tentativo. Dopo collisioni ripetute l'intervallo di estrazione cresce fino a un limite imposto dalla specifica. Nelle formule che seguono useremo, quando dichiarato, $t_B = 2\tau_M$ come durata dello slot del modello analitico. Questa identificazione serve a

collegare direttamente il backoff alla geometria della rete e non va confusa con una definizione universale di tutte le varianti Ethernet.

6.4.3 Prestazioni del protocollo CSMA/CD 1-persistente

Per isolare il ruolo di propagazione, collision detection e backoff adottiamo un modello analitico esplicito. Le trame hanno durata costante t_F , gli arrivi sono poissoniani e il canale non introduce errori indipendenti dalle collisioni. Il modello non vuole riprodurre ogni dettaglio di una particolare implementazione Ethernet: serve a quantificare, con ipotesi verificabili, l'effetto dei parametri fisici sulle prestazioni.

Nel CSMA non persistente studiato nel capitolo precedente un tentativo che trova il mezzo occupato viene rinviato; ciò rende naturale identificare cicli separati da intervalli idle. Nel caso 1-persistente, invece, una stazione in attesa trasmette non appena il mezzo appare libero. Più stazioni possono quindi accumularsi durante una trasmissione e tentare insieme al termine di essa. Per descrivere il processo adottiamo una chiusura semi-Markov a tre stati. Il numero esatto di stazioni in backoff e i loro contatori non sono conservati: dopo un tentativo fallito le richieste vengono riassorbite in un flusso poissoniano di tentativi effettivi di tasso G/t_F , comprendente nuovi pacchetti e ritrasmissioni. Le richieste sopraggiunte mentre il mezzo è percepito occupato vengono accumulate per l'opportunità successiva. Questa aggregazione non è una catena esatta del protocollo Ethernet; in particolare non deriva il traffico poissoniano dal binary exponential backoff reale.

Si procede definendo tre stati:

- $0 \rightarrow$ lo stato di idle,
- $1 \rightarrow$ lo stato in cui una sola stazione ha un pacchetto pendente all'atto del sensing del canale,
- $2 \rightarrow$ lo stato due o più stazioni hanno un pacchetto pendente all'atto del sensing del canale.

Siano:

- $E[T_i]$ il tempo medio di permanenza nello stato i ;
- P_i la probabilità stazionaria di visitare lo stato i nella catena incorporata agli istanti di cambiamento di stato. Poiché i tempi di permanenza T_i non sono uguali, P_i non coincide in generale con la frazione di tempo trascorsa nello stato i ;
- si consideri un ritardo di propagazione τ_p generico per semplicità, che può essere sostituito da $\tau_M/3$ qualora si consideri una rete a bus.

Ricordando dalla (4.20) l'espressione per il tempo utile, si ha la seguente espressione per il throughput:

$$S = \frac{P_1 t_F e^{-G\alpha}}{\sum_{i=0}^2 P_i E[T_i]}$$

I tre stati formano una catena di Markov come riportato in figura 6.3. Per determinare le probabilità di transizione ed i tempi di permanenza si consideri la figura 6.4 che riporta le temporizzazioni relative ai tre stati, dove nel caso di stato 1 si è supposto di avere una collisione. Si noti che nella figura è mostrata una possibile evoluzione temporale, ma di fatto i tre stati possono alternarsi in vario modo.

A commento della figura 6.4, si osserva che: t_j è la durata del segnale emesso dalle stazioni per indicare che c'è stata una collisione, segnale solitamente detto di JAMMING; t_{co} è il tempo di collisione relativo al primo pacchetto che entra in collisione, a differenza di t_{col} del CSMA

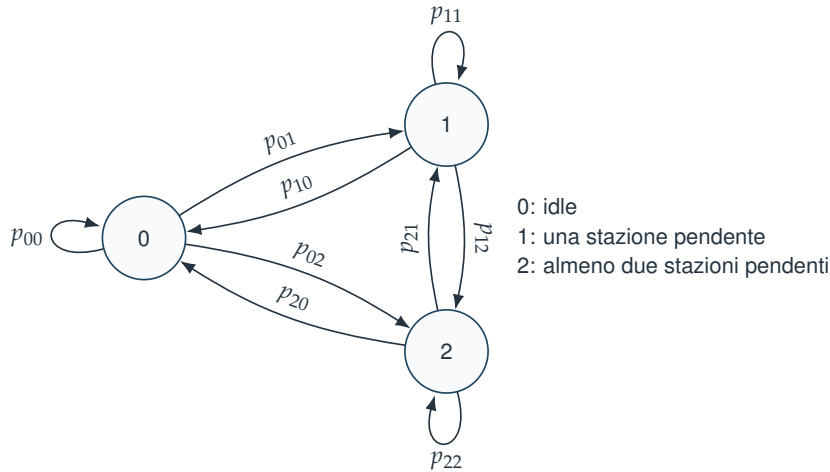


Figura 6.3. Catena di Markov per lo studio del throughput nel CSMA/CD 1-persistente.

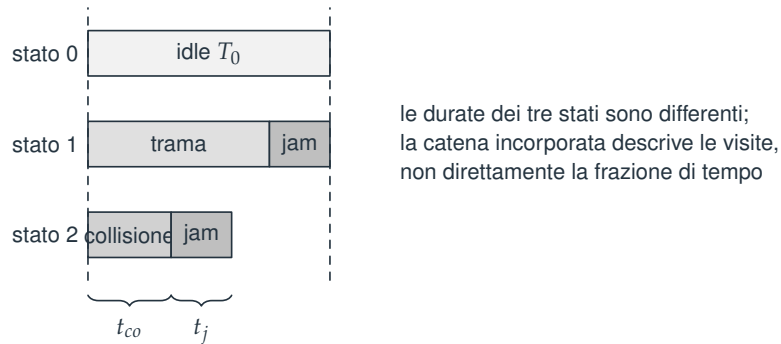


Figura 6.4. Tempistiche relative al protocollo di accesso CSMA/CD 1-persistente.

non-persistente classico che è riferito all'ultimo pacchetto che entra in collisione.

La differenza tra t_{co} e t_{col} è dovuta al fatto che nel non-persistente si valuta la massima occupazione del canale e quindi si deve tener conto dell'ultimo pacchetto colliso, nel CSMA/CD 1-persistente l'occupazione del canale è sempre uguale sia che ci sia un solo pacchetto colliso sia che ce ne sia più di uno dato che il meccanismo di emissione del segnale di JAMMING scatta sul primo pacchetto.

Si procede determinando i tempi di permanenza e le probabilità di transizione.

Stato 0

Il tempo medio di idle è uguale a quanto ricavato in (4.21), quindi:

$$E[T_0] = \frac{t_F}{G}$$

Dato che nella catena di Markov l'evoluzione è guidata dallo scatenarsi di un evento, e che nello stato di idle l'unico possibile evento è l'arrivo di un pacchetto, deve essere: $p_{00} = 0$. Questo risultato discende dal fatto che un cambiamento nello stato può avvenire solo in presenza di un evento nel sistema, evento che non può lasciare il sistema nello stato di idle.

È evidente che deve valere la seguente relazione generale di normalizzazione:

$$p_{i0} + p_{i1} + p_{i2} = 1 \quad i = 0, 1, 2$$

e quindi $p_{01} + p_{02} = 1$.

Inoltre non è possibile passare direttamente dallo stato 0 allo stato 2: in un processo di Poisson la probabilità di due o più arrivi nello stesso istante è nulla. Il primo evento che termina un intervallo idle è quindi un singolo arrivo; ne segue $p_{02} = 0$ e, di conseguenza, $p_{01} = 1$.

Stato 1

Per quanto concerne le valutazioni dello stato 1, l'analisi viene fatta attraverso i due possibili casi disgiunti: trasmissione con successo e collisione.

Nel caso di trasmissione con successo il tempo di permanenza nello stato 1 è pari al tempo per trasmettere il pacchetto più il ritardo di propagazione: $t_F + \tau_p$.

Nel caso di trasmissione con collisione, facendo riferimento alla figura 6.4: la stazione che ha il pacchetto pendente all'atto del sensing inizia la trasmissione; dopo $t_{co} \leq \tau_p$ una seconda stazione inizia a trasmettere (dato che non ha ancora avvertito la trasmissione della prima); dopo τ_p dall'inizio, la seconda stazione avverte la collisione ed emette il segnale di jamming di durata t_j ; dopo τ_p dall'inizio della trasmissione della seconda stazione anche la prima avverte la collisione e quindi emette il segnale di jamming, che impiega τ_p per propagarsi a tutta la rete. Quindi, in questo caso, il tempo di permanenza nello stato 1 è pari a $2\tau_p + t_j + t_{co}$.

La trasmissione con successo avviene con probabilità $e^{-\alpha G}$. Per le collisioni, $Ge^{-G\xi}$ dove $\xi \in [0, \alpha]$ e $\xi = t_{co}/t_F$, è una sottodensità di massa $1 - e^{-\alpha G}$; la densità condizionata alla collisione si ottiene dividendo per tale massa. La sottodensità è direttamente ottenibile da $\frac{d}{d\xi}(1 - e^{-\xi G})$ dato che il tempo interarrivo è esponenziale.

Quindi, definendo $\nu = \frac{t_j}{t_F}$ il tempo di jamming normalizzato, si ha:

$$\begin{aligned} E[T_1] &= t_F \left[(1 + \alpha)e^{-\alpha G} + \int_0^\alpha (2\alpha + \nu + \xi)Ge^{-\xi G}d\xi \right] = \\ &= t_F \left[(1 + \alpha)e^{-\alpha G} + (2\alpha + \nu)(1 - e^{-\alpha G}) + \frac{1 - e^{-\alpha G}}{G} - \alpha e^{-\alpha G} \right] = \\ &= t_F \left[e^{-\alpha G} + \left(2\alpha + \nu + \frac{1}{G} \right) (1 - e^{-\alpha G}) \right] \end{aligned}$$

Si osserva che lo stato visitato dopo lo stato 1 dipende da quello che succede dopo τ_p istanti dall'inizio dello stato 1, cioè dipende se ci sono arrivi o meno negli intervalli $[\tau_p, \tau_p + t_F]$ nel caso di trasmissione con successo e $[\tau_p, \tau_p + t_{co} + \tau_p + t_j]$ nel caso di collisione.

Si va nello stato 0 se non vi sono arrivi in $[\tau_p, \tau_p + t_F]$ nel caso con successo o se non vi sono arrivi in $[\tau_p, \tau_p + t_{co} + \tau_p + t_j]$ nel caso con collisione.

Poiché il processo di Poisson ha incrementi stazionari e indipendenti, traslare l'intervallo di osservazione di τ_p non modifica la distribuzione del numero di arrivi. Quindi:

$$p_{10} = e^{-G}e^{-\alpha G} + \int_0^\alpha e^{-(\xi + \alpha + \nu)G}Ge^{-G\xi}d\xi = e^{-(1+\alpha)G} + \frac{e^{-(\alpha + \nu)G}}{2}(1 - e^{-2\alpha G})$$

Si rimane nello stato 1 se in tali intervalli si verifica *esattamente un* nuovo arrivo; con due o più nuovi arrivi il processo passa invece nello stato 2. Pertanto:

$$\begin{aligned} p_{11} &= Ge^{-G}e^{-\alpha G} + \int_0^\alpha (\xi + \alpha + \nu)Ge^{-(\xi + \alpha + \nu)G}Ge^{-G\xi}d\xi = \\ &= Ge^{-(1+\alpha)G} + \frac{e^{-(\alpha + \nu)G}}{4} \left[(1 - e^{-2\alpha G})(1 + 2(\alpha + \nu)G) - 2\alpha Ge^{-2\alpha G} \right] \end{aligned}$$

Per verificare il termine finale basta usare $\int_0^\alpha \xi G^2 e^{-2G\xi} d\xi = [1 - (1 + 2\alpha G)e^{-2\alpha G}]/4$. Il fattore G accanto ad α è necessario anche per il corretto limite a carico nullo. Infine, dalla condizione di normalizzazione si ha $p_{12} = 1 - p_{10} - p_{11}$.

Stato 2

Lo stato 2 è caratterizzato dal fatto di avere 2 o più pacchetti pendenti in stazioni diverse all'atto del sensing, quindi dal fatto che due o più stazioni inizino a trasmettere contemporaneamente (in figura 6.4 è mostrato il caso di due sole stazioni), determinando una collisione.

Dopo τ_p dall'inizio della loro trasmissione, entrambe avvertono la collisione e quindi contemporaneamente emettono il segnale di jamming, che si propaga in tutta la rete dopo τ_p .

Quindi $E[T_2]$ ha la seguente espressione:

$$E[T_2] = t_F[2\alpha + \nu]$$

Le probabilità di transizione si ottengono con ragionamento analogo a quello fatto per il caso di stato 1. Se non c'è nessun arrivo in $\tau_p + t_j$ si va nello stato 0 e quindi:

$$p_{20} = e^{-(\alpha+\nu)G}$$

mentre se c'è un solo arrivo si va nello stato 1:

$$p_{21} = (\alpha + \nu)G e^{-(\alpha+\nu)G}$$

Infine, dalla condizione di normalizzazione si ha: $p_{22} = 1 - p_{20} - p_{21}$.

Insieme degli stati, throughput

Per valutare il throughput si calcola la distribuzione stazionaria della catena incorporata, con matrice $\underline{\underline{V}} = (p_{ij})$:

$$\underline{\underline{P}} = \underline{\underline{P}} \underline{\underline{V}}.$$

È il bilancio delle visite ai tre stati. Non si usa il generatore a tempo continuo della teoria delle code: le durate degli episodi non sono esponenziali e la distribuzione temporale sarà pesata con $E[T_i]$.

Inoltre deve essere $P_0 + P_1 + P_2 = 1$; partendo da questi presupposti si risolve il sistema dato in forma vettoriale dall'ultima equazione:

$$P_1 = \frac{p_{20} + p_{21}}{X}, \quad P_2 = \frac{1 - p_{10} - p_{11}}{X}, \quad P_0 = \frac{(1 - p_{11})p_{20} + p_{10}p_{21}}{X},$$

dove

$$X = (p_{20} + p_{21}) + (1 - p_{10} - p_{11}) + (1 - p_{11})p_{20} + p_{10}p_{21}.$$

La costante X normalizza le probabilità della catena incorporata. Nelle espressioni omogenee di grado zero, come il throughput scritto come rapporto tra reward medio e tempo medio di ciclo, essa si semplifica.

Tramite tutte le probabilità di transizione ricavate precedentemente per i vari stati si ottiene l'espressione finale del throughput in funzione del traffico offerto, che non viene riportata data la sua lunghezza, ma che si scrive immediatamente dalle equazioni ricavate.

Nelle figure 6.5 e 6.6 viene riportato il throughput in funzione del carico offerto al variare del tempo di jamming normalizzato ν e del ritardo di propagazione normalizzato α . Si nota che il throughput cala al crescere di ν , fissato $\alpha = 0.01$, dato che più è lungo il segnale di jamming e meno si avvertono i benefici del collision detect (al limite quando $\nu = 1 \rightarrow t_j = t_F$, il beneficio di interrompere la trasmissione non appena avvertita la collisione è nullo). Inoltre, fissato $\nu = 0.01$, il throughput cala al crescere di α dato che cresce l'intervallo di vulnerabilità.

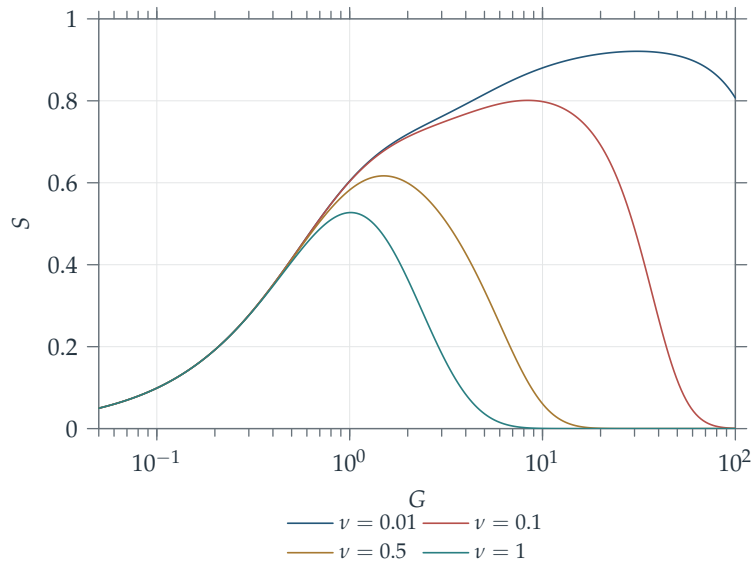


Figura 6.5. Throughput in funzione del carico offerto per $\alpha = 0.01$.

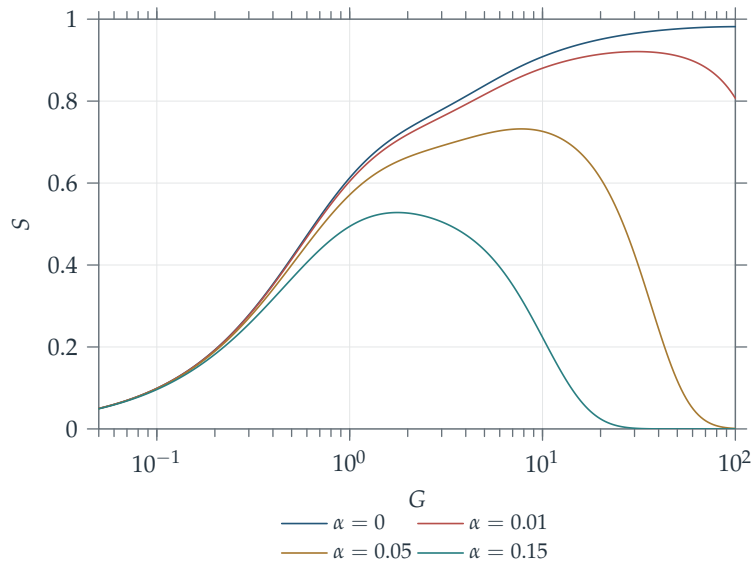


Figura 6.6. Throughput in funzione del carico offerto per $\nu = 0.01$.

Tempo di accesso

Per la valutazione del tempo di accesso si segue l'approccio proposto nel paragrafo 4.4.1 per l'Aloha, determinando quindi il tempo totale per effettuare le eventuali ritrasmissioni e il tempo per trasmettere il pacchetto corretto. Inoltre, come per tutti gli altri protocolli casuali studiati, per semplificare la trattazione si suppone che il tempo medio di accesso non sia da quando viene generato il pacchetto nella stazione alla sua consegna corretta, bensì sia il tempo da quando il pacchetto viene passato al livello MAC, per venire poi immesso sul canale, alla sua consegna finale. Questo tempo esclude la coda della sorgente. Non equivale a scartare gli altri arrivi e poi ritrasmetterli: tale politica cambierebbe il traffico. Il flusso poissoniano dei tentativi resta l'ipotesi di chiusura già dichiarata.

Nel CSMA/CD 1-persistente si accede al canale non appena questo risulta libero, quindi per la valutazione del tempo di attesa, t_{lib} , da quando arriva il pacchetto al MAC a quando si prova

a trasmetterlo, occorre considerare che: il pacchetto può essere trasmesso immediatamente se si è nello stato 0 o nei primi τ_p istanti degli stati 1 e 2, mentre occorre attendere la fine della operazione attuale in tutti gli altri casi.

Per arrivi di Poisson si può usare la proprietà PASTA: gli arrivi vedono le frazioni temporali di regime del processo. Poiché il processo descritto sopra è semi-Markov, la frazione di tempo trascorsa nello stato i è proporzionale a $P_i E[T_i]$, non a P_i da sola.

Definiamo

$$\bar{T} = \sum_{i=0}^2 P_i E[T_i].$$

Durante i primi τ_p secondi degli stati 1 e 2 una nuova stazione non ha ancora percepito l'occupazione del mezzo e può tentare subito la trasmissione; il tempo di attesa per trovare il mezzo disponibile è quindi associato soltanto alla parte residua dello stato successiva a τ_p .

Per lo stato 1 poniamo

$$M_1 = E[(T_1 - \tau_p)_+^2].$$

Usando i due casi già impiegati nella derivazione di $E[T_1]$ si ottiene

$$\frac{M_1}{t_F^2} = e^{-\alpha G} + \int_0^\alpha (\alpha + \nu + \xi)^2 G e^{-G\xi} d\xi.$$

Per lo stato 2, la permanenza è deterministica e quindi

$$M_2 = E[(T_2 - \tau_p)^2] = t_F^2 (\alpha + \nu)^2.$$

Il teorema del residuo temporale applicato ai tratti di mezzo non immediatamente accessibili fornisce allora

$$t_{lib} = \frac{P_1 M_1 + P_2 M_2}{2\bar{T}}. \quad (6.3)$$

Il fattore $1/2$ deriva dall'area triangolare del residuo: $\int_{\tau_p}^{T_i} (T_i - u) du = (T_i - \tau_p)_+^2 / 2$. Si tiene così conto del secondo momento della durata, non della metà della sola durata media. L'uso di questo ambiente per i tentativi della stazione marcata resta parte della chiusura disaccoppiata.

Nel modello adottato il canale è privo di errori indipendenti dalle collisioni. Non introduciamo quindi un ACK di livello MAC per il recupero delle collisioni: un tentativo fallito viene riconosciuto tramite collision detection e jamming. Questa è un'ipotesi del modello, non una proprietà generale di un collegamento reale.

Per modellare correttamente il binary exponential backoff occorre mantenere aleatorio il numero di collisioni. Indichiamo con p_c la probabilità che un singolo tentativo collida. Nel modello a tentativi indipendenti usato in questa analisi, la probabilità di successo per tentativo è S/G e quindi

$$p_c = 1 - \frac{S}{G}.$$

Se non si imponesse un limite al numero di tentativi, il numero R di collisioni prima del successo avrebbe distribuzione geometrica

$$P\{R = r\} = (1 - p_c) p_c^r, \quad r = 0, 1, \dots,$$

e $E[R] = p_c / (1 - p_c) = G/S - 1$.

Nel binary exponential backoff, dopo la j -esima collisione viene estratto un intero uniforme

$$K_j \in \{0, 1, \dots, 2^{\min(j, m)} - 1\},$$

dove m è il limite di crescita dell'esponente. Pertanto

$$E[K_j] = \frac{2^{\min(j,m)} - 1}{2}.$$

Lo standard Ethernet usa un backoff troncato: l'esponente non cresce oltre $m = 10$ e una trama viene abbandonata dopo un numero finito di tentativi.

Per mantenere anche il limite di tentativi nel modello analitico, indichiamo con L il numero massimo di tentativi. Condizionatamente al fatto che la trama venga consegnata entro L tentativi, la probabilità di aver già subito almeno j collisioni e di riuscire comunque entro il limite è

$$P\{R \geq j \mid R < L\} = \frac{p_c^j - p_c^L}{1 - p_c^L}, \quad j = 1, \dots, L-1.$$

Ne segue che il numero medio di collisioni di una trama consegnata è

$$\bar{R}_L = \sum_{j=1}^{L-1} \frac{p_c^j - p_c^L}{1 - p_c^L}, \quad (6.4)$$

mentre il tempo medio totale di backoff, con $t_B = 2\tau_M$, è

$$E[t_R \mid R < L] = \frac{t_B}{2} \sum_{j=1}^{L-1} \left(2^{\min(j,m)} - 1\right) \frac{p_c^j - p_c^L}{1 - p_c^L}. \quad (6.5)$$

Per Ethernet si possono porre $m = 10$ e $L = 16$. La probabilità di abbandono della trama nel modello indipendente è semplicemente

$$P_{drop} = p_c^L.$$

La (6.5) è una somma finita, direttamente valutabile e adatta alla generazione numerica dei grafici. Se si trascura il limite di tentativi, $L \rightarrow \infty$, si recupera $E[R] = G/S - 1$ e

$$\frac{E[t_R]}{t_F} = \alpha_M \sum_{j=1}^{\infty} p_c^j \left(2^{\min(j,m)} - 1\right),$$

che rimane finita per ogni $p_c < 1$ grazie al limite m .

Per completare il modello occorre distinguere la durata media di una collisione. Nello stato 1 la collisione avviene con probabilità $1 - e^{-\alpha G}$ e, condizionatamente alla collisione, la durata utile ai fini del nuovo tentativo è

$$\bar{t}_{c,1} = t_F \frac{\int_0^{\alpha} (\alpha + \nu + \xi) G e^{-G\xi} d\xi}{1 - e^{-\alpha G}}.$$

Nello stato 2 la collisione è certa e la corrispondente durata è

$$\bar{t}_{c,2} = t_F(\alpha + \nu).$$

Tra gli eventi di collisione, i pesi relativi dei due casi sono proporzionali rispettivamente a $P_1(1 - e^{-\alpha G})$ e P_2 . Usiamo questo campionamento per episodio come rappresentante del costo di una collisione del pacchetto marcato. Non è un'identità per tutti i tentativi della rete: un episodio con molte stazioni collise pesa più volte in un campionamento per tentativo. Con questa ulteriore approssimazione definiamo

$$\bar{t}_c = \frac{P_1(1 - e^{-\alpha G})\bar{t}_{c,1} + P_2\bar{t}_{c,2}}{P_1(1 - e^{-\alpha G}) + P_2}. \quad (6.6)$$

Il modello di accesso assume quindi che tentativi successivi vedano, in media, lo stesso ambiente stazionario: ogni collisione comporta un tempo medio $t_{lib} + \bar{t}_c$, il backoff complessivo è dato dalla (6.5) e il tentativo finale riuscito richiede $t_{lib} + t_F + \tau_p$. Ne segue

$$t_{acc} = (\bar{R}_L + 1)t_{lib} + \bar{R}_L \bar{t}_c + E[t_R | R < L] + t_F + \tau_p. \quad (6.7)$$

La relazione è interamente espressa mediante quantità già definite: t_{lib} dalla (6.3), $\bar{R}_L$ dalla (6.4), $\bar{t}_c$ dalla (6.6) e il backoff dalla (6.5). È il bilancio temporale di una chiusura approssimata: oltre all'indipendenza dei tentativi usa il costo rappresentativo per collisione appena definito. Il risultato è condizionato alla consegna entro L tentativi e va letto insieme a P_{drop} , non come ritardo incondizionato di tutte le trame generate.

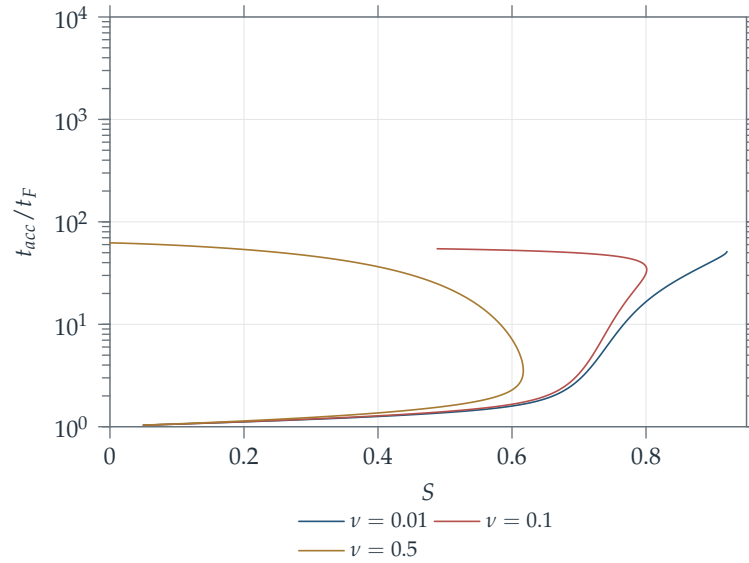


Figura 6.7. Tempo di accesso in funzione del throughput per $\alpha = 0.01$.

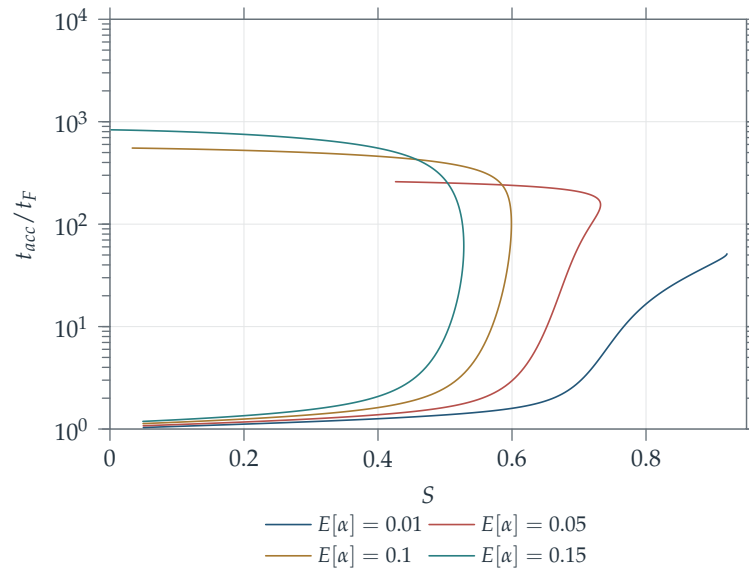


Figura 6.8. Tempo di accesso in funzione del throughput per $\nu = 0.01$.

Nelle figure 6.7 e 6.8 vengono riportati i tempi di accesso in funzione del throughput al variare del tempo di jamming normalizzato ν e del ritardo di propagazione normalizzato α . Si

nota che il tempo di accesso cresce con ν e con α a causa di considerazioni analoghe a quelle fatte nel caso dei grafici di S in funzione di G .

Nei grafici di questo modello si adotta l'approssimazione del bus con molte stazioni, $\alpha_M \simeq 3\alpha$. I parametri di propagazione sono scelti in modo da rispettare $2\alpha_M \leq 1$, coerentemente con la (6.2); il caso $\alpha = 0$ resta un limite formale della chiusura. Non si usa il ritardo medio per giustificare la dimensione minima della trama.

6.5 Accesso a token

Un accesso a token sostituisce la contesa con un'autorizzazione esplicita: soltanto la stazione che possiede il token può iniziare una trasmissione. Se il token visita ciclicamente le stazioni, il sistema può essere modellato come un polling ciclico. La proprietà essenziale non è la particolare codifica di una tecnologia storica, ma il fatto che il diritto di accesso abbia un ordine controllato e che il passaggio del token introduca un tempo di ciclo non nullo.

Consideriamo un anello logico con N stazioni. Una stazione che riceve il token e ha dati in coda può trasmettere secondo una disciplina prefissata; terminato il servizio, il token passa alla stazione successiva. Discipline diverse producono prestazioni diverse. Nel seguito si sceglie esplicitamente una disciplina *1-limited*, nella quale viene servita al massimo una trama per visita. Questa ipotesi rende il modello confrontabile con i sistemi di polling studiati nel capitolo 4 senza attribuire a una specifica tecnologia proprietà che dipendono invece dalla disciplina di servizio.

L'accesso a token evita le collisioni tra stazioni che rispettano il protocollo, ma non elimina i costi di coordinamento: propagazione, passaggio del token, eventuale rigenerazione e tempo trascorso servendo altre code contribuiscono tutti al ritardo di accesso. Il modello quantitativo seguente rende espliciti questi contributi.

6.5.1 Prestazioni

Per rendere coerente il modello con l'ipotesi adottata in questa sezione, supponiamo che ogni stazione trasmetta al più una trama per visita del token. Il sistema è quindi modellato come un polling system ciclico simmetrico con disciplina *1-limited*. Questa scelta è diversa dal servizio *exhaustive* analizzato nel capitolo 4: le due discipline non hanno, in generale, lo stesso tempo medio di attesa.

La relazione di conservazione del lavoro per polling *exhaustive* è stata derivata nel capitolo 4, equazione (4.10). Nel caso *1-limited* compare un contributo ulteriore, perché al termine di una visita la coda può non essere vuota. Deriviamo esplicitamente tale contributo prima di usare la relazione.

Siano:

- N il numero di stazioni;
- λ il tasso di arrivo poissoniano di trame in ciascuna stazione, con flussi indipendenti;
- X il tempo di servizio effettivo associato a una trama, indipendente dagli arrivi e dalle altre durate, con primi due momenti finiti;
- w il tempo di passaggio del token da una stazione alla successiva;
- $s = Nw$ il tempo totale di switch-over in un giro completo del token;
- all'interno di ogni stazione le trame sono servite in ordine FCFS.

Indichiamo inoltre con Z_{ii} il lavoro che rimane nella coda i immediatamente dopo una sua visita. In una disciplina *1-limited* una visita produce al più una partenza. In regime stazionario il numero medio di partenze della coda i per ciclo deve uguagliare il numero medio di arrivi nello stesso ciclo, cioè $\lambda_i E[C]$; questa è quindi anche la probabilità che una

visita trovi almeno una trama da servire.

Condizionatamente a una partenza, per arrivi di Poisson il numero medio di trame che restano dietro alla trama appena servita è uguale al numero medio di arrivi durante il suo tempo di permanenza nel sistema:

$$\lambda_i(E[W_i] + E[X_i]).$$

Ogni trama lasciata in coda porta in media $E[X_i]$ unità di lavoro. Pertanto

$$\begin{aligned} E[Z_{ii}] &= \lambda_i E[C] \lambda_i (E[W_i] + E[X_i]) E[X_i] \\ &= E[C] \rho_i (\lambda_i E[W_i] + \rho_i), \end{aligned} \quad (6.8)$$

dove $\rho_i = \lambda_i E[X_i]$.

La conservazione del tempo dà

$$E[C] = \frac{s}{1-\rho}, \quad \rho = \sum_i \rho_i.$$

La relazione di workload già dimostrata per il polling exhaustive si estende quindi aggiungendo $\sum_i E[Z_{ii}]$ al membro destro. Con tempo totale di trasferimento $T_s = s$ deterministico e usando la (6.8) si ottiene

$$\begin{aligned} \sum_i \rho_i E[W_i] &= \frac{\rho}{2(1-\rho)} \sum_i \lambda_i E[X_i^2] + \frac{\rho s}{2} + \frac{s}{2(1-\rho)} \left(\rho^2 - \sum_i \rho_i^2 \right) \\ &\quad + \frac{s}{1-\rho} \sum_i \rho_i (\lambda_i E[W_i] + \rho_i). \end{aligned}$$

Portando a sinistra i termini che contengono $E[W_i]$ e raccogliendo i termini quadratici si ottiene la relazione di conservazione 1-limited che verrà specializzata dopo aver definito il tempo di servizio X .

Nel modello semplificato dell'anello poniamo

$$X = \max\{\tau_R, t_F\}, \quad \tau_R = \tau_M \frac{N}{N-1}, \quad (6.9)$$

dove τ_R è il tempo di propagazione necessario per un giro completo dell'anello e t_F il tempo di trasmissione della trama. La relazione specifica un'ulteriore regola del modello: si rilascia il token soltanto dopo aver concluso l'emissione e dopo il ritorno della testa della trama alla sorgente. Non segue dalla sola disciplina 1-limited. Un rilascio immediato al termine dell'emissione o un'attesa del ritorno dell'intera trama darebbero tempi di servizio diversi. Il passaggio del token alla stazione seguente viene poi contabilizzato separatamente nel tempo w .

Il carico complessivo del server è

$$\rho = N\lambda E[X]. \quad (6.10)$$

Per una disciplina 1-limited non basta richiedere $\rho < 1$: ogni coda deve poter smaltire, in media, almeno una trama per ciclo. Nel caso simmetrico la condizione di stabilità è

$$\rho + \lambda s < 1. \quad (6.11)$$

Infatti il tempo medio di ciclo è $E[C] = s/(1-\rho)$ e la condizione $\lambda E[C] < 1$ equivale esattamente alla (6.11). Per la sufficienza nel caso simmetrico si può confrontare il sistema con

uno più lento che, a ogni visita, impiega un servizio fittizio X anche quando la coda è vuota. I suoi cicli hanno durata indipendente $s + \sum_{i=1}^N X_i$ e ogni coda può smaltire un cliente per ciclo. Il numero medio di arrivi per ciclo è $\lambda(s + NE[X]) = \rho + \lambda s < 1$. La ricorrenza della coda è quindi quella di un accumulo con incrementi medi negativi quando non vuoto; le somme di tali incrementi tendono a $-\infty$ e il loro supremo a ritroso è finito quasi certamente. Esiste così un regime stazionario per ciascuna coda del sistema rallentato. Eliminare i servizi fittizi non può aumentare il numero di arrivi fra due opportunità di servizio, e dà la stabilità del sistema originale.

Derivazione del tempo medio di attesa

La derivazione precedente fornisce, per switch-over deterministico s , la relazione

$$\sum_{i=1}^N \rho_i \left(1 - \frac{\lambda_i s}{1 - \rho}\right) E[W_i] = \frac{\rho}{2(1 - \rho)} \sum_{i=1}^N \lambda_i E[X_i^2] + \frac{\rho s}{2} + \frac{s}{2(1 - \rho)} \left(\rho^2 + \sum_{i=1}^N \rho_i^2\right). \quad (6.12)$$

Nel caso simmetrico

$$\lambda_i = \lambda, \quad \rho_i = \frac{\rho}{N}, \quad E[W_i] = E[W], \quad X_i = X.$$

Il membro sinistro della (6.12) diventa

$$\rho \left(1 - \frac{\lambda s}{1 - \rho}\right) E[W] = \frac{\rho(1 - \rho - \lambda s)}{1 - \rho} E[W].$$

Il membro destro diventa invece

$$\frac{\rho N \lambda E[X^2]}{2(1 - \rho)} + \frac{\rho s}{2} + \frac{s \rho^2 (1 + 1/N)}{2(1 - \rho)}.$$

Moltiplicando entrambi i membri per $(1 - \rho)/\rho$ e raccogliendo i termini in s si ottiene

$$(1 - \rho - \lambda s) E[W] = \frac{N \lambda E[X^2]}{2} + \frac{s}{2} \left(1 + \frac{\rho}{N}\right).$$

Pertanto il tempo medio di attesa prima dell'inizio del servizio è

$$E[W] = \frac{N \lambda E[X^2] + s(1 + \rho/N)}{2(1 - \rho - \lambda s)}. \quad (6.13)$$

La divergenza del denominatore quando $\rho + \lambda s \rightarrow 1$ è coerente con la condizione di stabilità (6.11).

Il tempo medio di accesso fino alla consegna della trama viene quindi modellato come

$$E[T_{acc}] = E[W] + E[t_F] + E[\tau_p], \quad (6.14)$$

dove, per un anello monodirezionale con stazioni equispaziate,

$$E[\tau_p] = \frac{\tau_M}{2} \frac{N}{N - 1}.$$

Per il passaggio del token assumiamo

$$w = t_A + \frac{\tau_M}{N - 1}, \quad s = N t_A + \tau_M \frac{N}{N - 1}. \quad (6.15)$$

Questa forma mantiene il fattore esatto in N ; per N grande si recupera l'approssimazione $s \simeq N t_A + \tau_M$.

Forma normalizzata

Definiamo

$$T = E[t_F], \quad a = \frac{\tau_R}{T}, \quad \beta = \frac{t_A}{T}, \quad y = \frac{X}{T}.$$

Dalla (6.9) segue

$$y = \max \left\{ a, \frac{t_F}{T} \right\}. \quad (6.16)$$

Introduciamo inoltre il throughput utile normalizzato rispetto al tempo di trasmissione della trama,

$$S = N\lambda T.$$

Poiché $\rho = N\lambda E[X]$, si ha

$$\rho = SE[y]. \quad (6.17)$$

Il tempo di switch-over normalizzato è

$$\frac{s}{T} = N\beta + a. \quad (6.18)$$

Sostituendo nella (6.13) si ottiene la forma direttamente utilizzabile per i grafici:

$$\frac{E[W]}{T} = \frac{SE[y^2] + (N\beta + a) \left(1 + \frac{SE[y]}{N} \right)}{2 [1 - SE[y] - \frac{s}{N}(N\beta + a)]}. \quad (6.19)$$

La condizione di stabilità diventa

$$SE[y] + \frac{s}{N}(N\beta + a) < 1. \quad (6.20)$$

Trame di lunghezza costante

Se $t_F = T$ è deterministico, ponendo

$$\varphi = \max\{a, 1\},$$

si ha immediatamente

$$E[y] = \varphi, \quad E[y^2] = \varphi^2.$$

La (6.19) diventa quindi

$$\frac{E[W]}{T} = \frac{S\varphi^2 + (N\beta + a)(1 + S\varphi/N)}{2 [1 - S\varphi - \frac{s}{N}(N\beta + a)]}. \quad (6.21)$$

Trame di lunghezza esponenziale

Se $Z = t_F/T$ è esponenziale di media unitaria, allora $y = \max\{a, Z\}$. La distribuzione di y possiede una massa $1 - e^{-a}$ nel punto a e densità e^{-y} per $y > a$. Il primo momento è

$$\begin{aligned} E[y] &= aP(Z \leq a) + \int_a^\infty ye^{-y} dy \\ &= a(1 - e^{-a}) + (a + 1)e^{-a} \\ &= a + e^{-a}, \end{aligned}$$

mentre il secondo momento è

$$\begin{aligned} E[y^2] &= a^2 P(Z \leq a) + \int_a^\infty y^2 e^{-y} dy \\ &= a^2(1 - e^{-a}) + (a^2 + 2a + 2)e^{-a} \\ &= a^2 + 2(a + 1)e^{-a}. \end{aligned}$$

Quindi

$$E[y] = a + e^{-a}, \quad E[y^2] = a^2 + 2(a + 1)e^{-a}. \quad (6.22)$$

La sostituzione della (6.22) nella (6.19) fornisce il tempo medio di attesa per trame con lunghezza esponenziale senza ulteriori approssimazioni.

Le prestazioni del modello a token sono determinate dalle (6.21) e (6.22), insieme alla (6.14). La figura 6.9 mostra il caso a lunghezza costante per alcuni valori del ritardo normalizzato del giro.

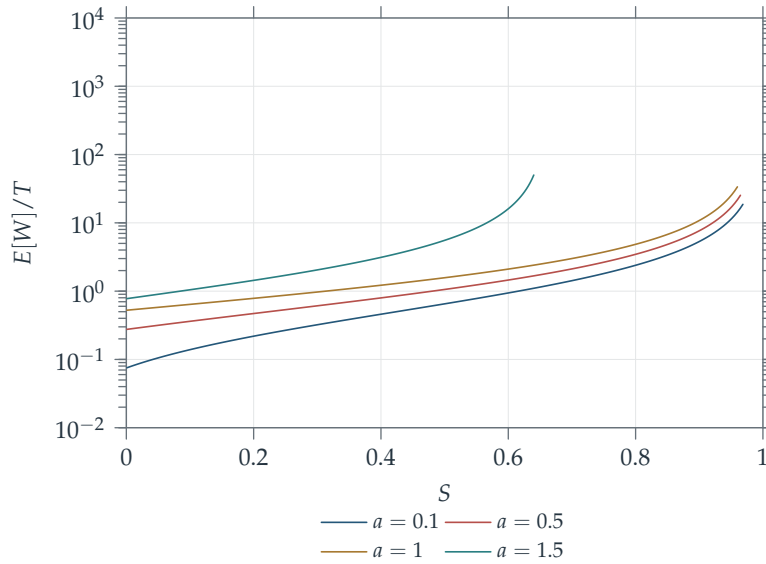


Figura 6.9. Accesso a token 1-limited: tempo medio di attesa normalizzato per trame di lunghezza costante, $N = 100$ e $\beta = 5 \cdot 10^{-4}$.

6.6 Disciplina timed-token

Una variante più generale dell'accesso a token non assegna a ogni stazione un numero prefissato di trame, ma limita il tempo di trasmissione in funzione della durata del giro precedente. L'obiettivo è mantenere il tempo di rotazione del token intorno a un valore di riferimento T . Questo principio è interessante indipendentemente dalla tecnologia che lo implementa perché realizza un controllo ad anello chiuso sul tempo di accesso.

Nel modello seguente si considera dapprima la regola di servizio e poi un caso simmetrico deterministico risolvibile in forma chiusa. La separazione è importante: la regola timed-token è una definizione della disciplina, mentre le formule di utilizzazione ricavate dopo valgono soltanto sotto le ipotesi del modello deterministico.

Regola timed-token

Sia T il Target Token Rotation Time (TTRT) e sia R_i il tempo di rotazione del token misurato dalla stazione i al suo arrivo. Nel caso in cui si consideri soltanto traffico asincrono e una

singola classe di priorità, il tempo massimo che la stazione può dedicare al traffico asincrono è

$$H_i = (T - R_i)^+ = \max\{T - R_i, 0\}. \quad (6.23)$$

La (6.23) esprime direttamente il principio timed-token: se il token è in anticipo rispetto al target, la differenza può essere usata per traffico asincrono; se il token è in ritardo, il traffico asincrono non riceve ulteriore tempo in quella visita.

La relazione è importante perché mostra anche perché un modello puramente exhaustive non è esatto: il tempo di servizio disponibile in una visita dipende dal tempo di rotazione osservato nella visita stessa.

Modello simmetrico deterministico a punto fisso

Consideriamo ora un modello volutamente semplice, adatto a essere risolto in forma chiusa. Assumiamo:

- N stazioni identiche e sempre sature di traffico asincrono;
- assenza di traffico sincrono e di priorità asincrone;
- tempo complessivo deterministico s per propagazione del token, latenze di stazione e trasmissione dei token in un giro completo;
- ricerca di un regime simmetrico costante, in cui il tempo di rotazione vale C e ogni stazione usa lo stesso tempo asincrono h . L'esistenza algebrica di questo punto fisso non ne implica l'attrattività.

In tale regime la (6.23) diventa

$$h = (T - C)^+. \quad (6.24)$$

Il tempo di un ciclo è la somma del tempo non dati s e dei tempi di trasmissione delle N stazioni:

$$C = s + Nh. \quad (6.25)$$

Se $s < T$, nel punto fisso risulta $C < T$ e quindi dalla (6.24) $h = T - C$. Sostituendo nella (6.25) si ottiene

$$C = s + N(T - C),$$

da cui

$$C = \frac{s + NT}{N + 1} \quad (6.26)$$

e, di conseguenza,

$$h = \frac{T - s}{N + 1}. \quad (6.27)$$

Le due espressioni verificano automaticamente $C < T$ e $h > 0$ se e solo se $s < T$.

Il tempo complessivo dedicato ai dati asincroni durante un ciclo è Nh . Pertanto la frazione del tempo di canale usata per dati asincroni, cioè il throughput normalizzato rispetto alla capacità fisica nel modello saturo, è

$$\eta_A = \frac{Nh}{C}. \quad (6.28)$$

Sostituendo le (6.26) e (6.27) si ricava

$$\eta_A = \frac{N(T - s)}{s + NT}. \quad (6.29)$$

La formula ha i limiti corretti:

$$\eta_A \rightarrow 1 \quad \text{per } s/T \rightarrow 0, \quad \eta_A \rightarrow 0 \quad \text{per } s/T \rightarrow 1^-.$$

Definendo il rapporto adimensionale

$$\theta = \frac{s}{T},$$

le relazioni precedenti assumono la forma

$$\frac{C}{T} = \frac{N + \theta}{N + 1}, \quad \frac{h}{T} = \frac{1 - \theta}{N + 1}, \quad \eta_A = \frac{N(1 - \theta)}{N + \theta}. \quad (6.30)$$

Queste tre grandezze costituiscono il modello base per la valutazione prestazionale della disciplina timed-token. La figura 6.10 mostra la frazione di capacità disponibile per traffico asincrono al variare del rapporto $\theta = s/T$.

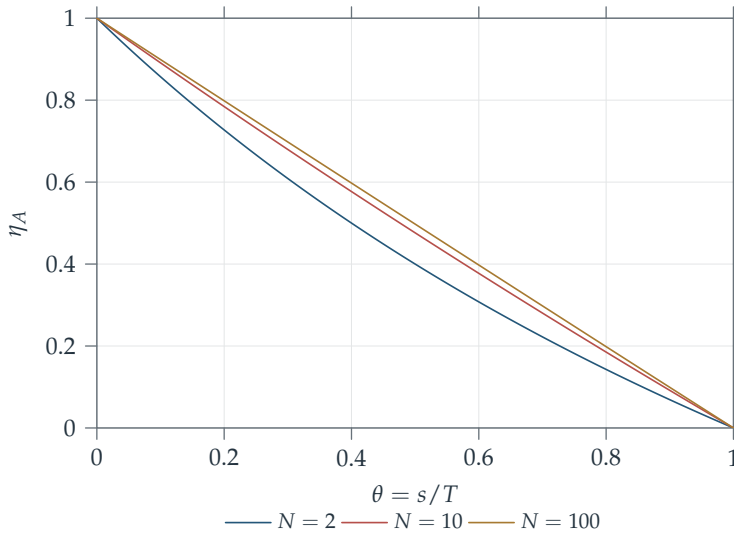


Figura 6.10. Timed-token nel modello simmetrico: utilizzazione asincrona η_A al variare di $\theta = s/T$.

La distinzione dalla convergenza è concreta. Già per $N = 1$ una realizzazione ricorsiva della regola produce $h_{n+1} = (T - s - h_n)^+$. Se $0 < h_0 < T - s$, si alternano h_0 e $T - s - h_0$ senza convergere, tranne nel caso $h_0 = (T - s)/2$. Il grafico descrive quindi il regime costante ipotizzato, non una prova di stabilizzazione di qualunque dinamica timed-token.

Tempo residuo alla prossima opportunità

Nel modello deterministico il token visita ciascuna stazione una volta ogni C secondi. Se un osservatore indipendente dal ciclo arriva in un istante uniformemente distribuito nel ciclo, il tempo residuo R alla prossima visita è uniforme in $[0, C]$ e quindi

$$E[R] = \frac{C}{2}. \quad (6.31)$$

Questa quantità non è il ritardo completo di coda del sistema timed-token: misura soltanto il residuo medio alla prossima opportunità nel modello simmetrico deterministico. Nel modello generale, dove il tempo di rotazione è aleatorio e il servizio è limitato dal timed-token, il ritardo medio richiede i momenti del ciclo e la dinamica delle code.

La relazione (6.23) definisce esattamente la disciplina timed-token qui considerata; le (6.26)–(6.30) appartengono invece al modello simmetrico deterministico e non vanno estese a sistemi con rotazioni aleatorie senza una nuova analisi.

6.7 IEEE 802.11 e WLAN

IEEE 802.11 definisce la famiglia di reti locali radio comunemente associate al Wi-Fi. Il livello fisico si è evoluto attraverso molte generazioni, con modulazioni, larghezze di banda, tecniche multiantenna e meccanismi multiutente differenti. Per comprendere il funzionamento della rete è però più utile separare questi dettagli dal problema MAC: più stazioni condividono un mezzo radio sul quale, durante la propria trasmissione, non possono assumere di rivelare in modo affidabile una collisione.

L'architettura di base distingue reti infrastrutturate e reti ad hoc. In un Basic Service Set (BSS) infrastrutturato le stazioni si associano a un Access Point (AP), che collega il BSS al sistema di distribuzione. Più BSS possono cooperare nello stesso Extended Service Set (ESS), come illustrato in figura 6.11. Una configurazione indipendente può invece operare senza Access Point.

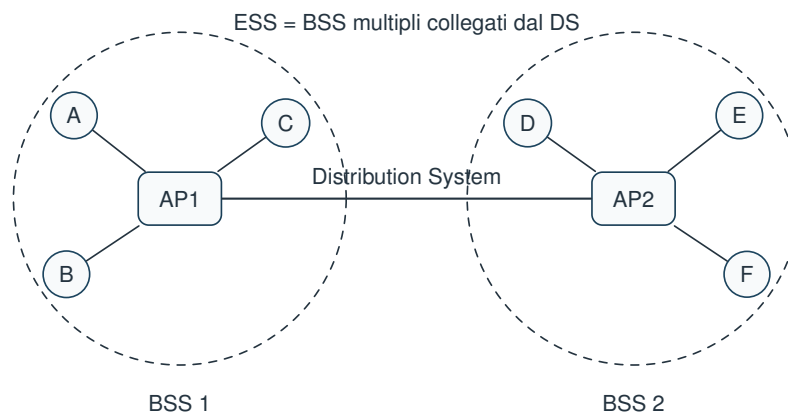


Figura 6.11. Più BSS collegati attraverso il sistema di distribuzione formano un ESS.

Per l'accesso distribuito il riferimento concettuale è CSMA/CA: sensing del canale, backoff casuale e conferma esplicita della ricezione. Le estensioni di qualità del servizio modificano parametri e priorità di accesso, ma non cambiano il principio fondamentale che analizziamo nel seguito.

6.7.1 Accesso

Nel DCF l'accesso al mezzo è basato su Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). A differenza di Ethernet half-duplex, una stazione radio non può in generale rilevare in modo affidabile una collisione mentre trasmette; per questo 802.11 combina carrier sense, backoff casuale e acknowledgment espliciti.

Il carrier sense ha una componente fisica e una componente virtuale. La componente fisica valuta lo stato del mezzo radio; quella virtuale usa le informazioni di durata presenti nelle trame per aggiornare il Network Allocation Vector (NAV), cioè l'intervallo durante il quale la stazione considera il mezzo prenotato da altri.

Lo scambio Request To Send (RTS) / Clear To Send (CTS), illustrato in figura 6.12, è opzionale e può essere impiegato prima di una trama dati. Il mittente, dopo aver osservato il mezzo libero per l'intervallo richiesto, invia RTS; il destinatario risponde dopo SIFS con CTS. Le stazioni che ricevono RTS o CTS possono aggiornare il NAV e rinviare l'accesso. Lo schema riduce l'impatto del problema del terminale nascosto, ma non lo elimina in ogni topologia e introduce overhead; per trame corte può essere preferibile non usarlo. Nel caso dati unicast qui considerato segue un ACK dopo SIFS. Le trasmissioni broadcast e multicast ordinarie non usano lo stesso riscontro per destinatario; aggregazione e Block ACK richiedono un modello

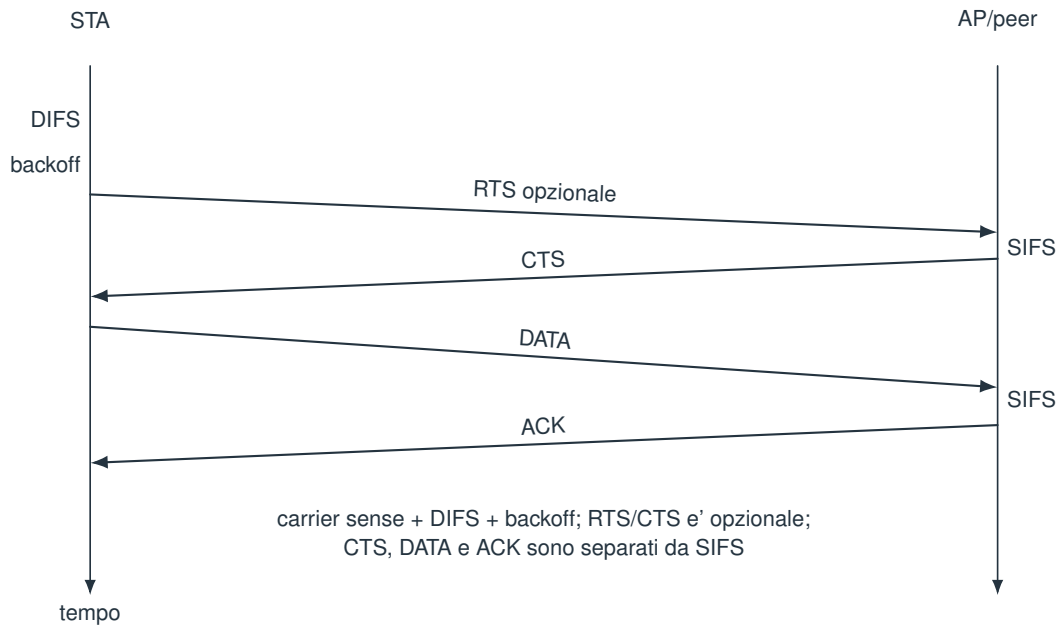


Figura 6.12. Accesso CSMA/CA con scambio opzionale RTS/CTS.

distinto.

Il backoff viene espresso in un numero intero di slot. A ogni nuovo tentativo la stazione estrae un valore casuale nell'intervallo $[0, CW]$, dove la Contention Window è compresa tra CW_{min} e CW_{max} . Il contatore viene decrementato di uno per ogni slot in cui il mezzo rimane libero; quando il mezzo diventa occupato il contatore viene congelato e riprende dopo che il mezzo torna libero per l'intervallo previsto. Dopo una trasmissione non riuscita la finestra di contesa cresce, secondo il binary exponential backoff, fino al massimo previsto; dopo un successo viene riportata al valore minimo. I valori numerici di CW_{min} , CW_{max} , slot time e degli inter-frame space dipendono dal livello fisico e, nelle estensioni QoS, anche dalla categoria di accesso: non esiste quindi un unico valore universale come $CW_{min} = 7$ valido per tutta la famiglia 802.11.

Questa descrizione è il modello di riferimento usato nelle analisi seguenti. Le implementazioni moderne aggiungono molte funzioni ulteriori, ma il principio fondamentale resta l'accesso distribuito con sensing, backoff e conferma esplicita della ricezione.

6.7.2 Struttura della trama

Le trame 802.11 possono essere di gestione, controllo o dati e non hanno tutte la stessa struttura. Il campo Frame Control identifica il tipo e il sottotipo della trama e contiene i flag che descrivono, tra l'altro, direzione rispetto al sistema di distribuzione, ritrasmissione, gestione energetica e protezione del contenuto. Il campo Duration/ID può contribuire alla prenotazione virtuale del mezzo mediante il NAV.

Una trama può contenere più indirizzi MAC perché, in una rete infrastrutturata o distribuita, trasmettitore e ricevitore radio non coincidono necessariamente con sorgente e destinazione logiche. A seconda dei bit To DS e From DS gli indirizzi assumono quindi i ruoli di Receiver Address, Transmitter Address, Source Address, Destination Address e BSSID. Sequence Control permette il riordinamento e la gestione della frammentazione; il Frame Check Sequence protegge la trama con un CRC a 32 bit.

Le trame RTS, CTS e ACK sono intenzionalmente più corte delle trame dati: quando una collisione coinvolge una trama di controllo, il tempo di canale sprecato è inferiore. È

proprio questo compromesso tra overhead aggiuntivo e riduzione del costo delle collisioni che rende RTS/CTS conveniente soltanto in alcune condizioni di carico, dimensione delle trame e topologia radio.

6.7.3 Prestazioni: modello di Bianchi

Per l'analisi prestazionale della DCF adottiamo il modello saturo classico basato sulla catena bidimensionale di backoff, comunemente indicato come modello di Bianchi. La derivazione è riportata nel testo e rende esplicite le ipotesi di disaccoppiamento fra stazioni. Il modello descrive in modo compatto il binary exponential backoff e permette di ricavare throughput e tempo medio di servizio sia per Basic Access sia per RTS/CTS.

Le ipotesi del modello base sono:

- n stazioni identiche e sempre sature, cioè sempre con un pacchetto pronto da trasmettere;
- canale ideale, senza errori indipendenti dalle collisioni;
- tutte le stazioni sono mutuamente visibili, quindi non vi sono terminali nascosti nel modello base;
- la probabilità condizionata di collisione p di un tentativo è costante e indipendente dallo stadio di backoff;
- la finestra minima contiene $W = CW_{min} + 1 \geq 2$ valori di backoff e raddoppia fino allo stadio massimo m , dopo il quale rimane costante;
- non si impone un limite di ritrasmissioni: una trama continua a ritentare fino al successo; gli scarti per retry limit non sono quindi presenti in questo modello;
- non si includono aggregazione, Block ACK, differenziazione QoS o trasmissioni multiutente programmate dall'access point.

Dimostrazione delle equazioni di punto fisso

Indichiamo con $i \in \{0, \dots, m\}$ lo stadio di backoff e con $k \in \{0, \dots, W_i - 1\}$ il valore corrente del contatore, con

$$W_i = 2^i W, \quad i = 0, \dots, m.$$

La dimostrazione usa $m \geq 1$; per $m = 0$ esiste un solo stadio e l'estrazione media di $(W - 1)/2$ slot dà direttamente $\tau = 1/[1 + (W - 1)/2] = 2/(W + 1)$. Lo stato della stazione è quindi la coppia (i, k) . Indichiamo con $b_{i,k}$ la probabilità stazionaria di trovarsi nello stato (i, k) . Quando $k > 0$ il contatore viene decrementato di uno a ogni slot di backoff osservato dal modello. Si tratta di opportunità virtuali: non tutti gli intervalli hanno durata fisica σ , perché il contatore reale si congela durante l'occupazione del mezzo. Il tempo fisico viene ricostruito con le durate di successo, collisione e slot vuoto. Quando $k = 0$ la stazione trasmette. Un successo riporta allo stadio 0; una collisione porta allo stadio successivo e, raggiunto lo stadio m , ulteriori collisioni mantengono la stazione nello stesso stadio. A ogni ingresso in uno stadio il nuovo contatore è scelto uniformemente tra i W_i valori possibili.

Sia p la probabilità condizionata di collisione di un tentativo. Per gli stati con contatore nullo, il bilancio stazionario fornisce

$$b_{i,0} = p^i b_{0,0}, \quad 0 \leq i < m, \quad (6.32)$$

mentre, nello stadio massimo, la permanenza dovuta alle collisioni successive produce la serie geometrica

$$b_{m,0} = p^m b_{0,0} \sum_{j=0}^{\infty} p^j = \frac{p^m}{1-p} b_{0,0}. \quad (6.33)$$

Per un dato stadio i , uno stato (i, k) con $k > 0$ viene attraversato se il valore estratto all'ingresso nello stadio è almeno k . Poiché l'estrazione è uniforme,

$$b_{i,k} = \frac{W_i - k}{W_i} b_{i,0}, \quad k = 0, \dots, W_i - 1. \quad (6.34)$$

La somma delle probabilità di tutti gli stati dello stadio i vale quindi

$$\sum_{k=0}^{W_i-1} b_{i,k} = b_{i,0} \frac{W_i + 1}{2}. \quad (6.35)$$

Infatti

$$\sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} = \frac{1}{W_i} \sum_{h=1}^{W_i} h = \frac{W_i + 1}{2}.$$

Imponendo la normalizzazione della distribuzione stazionaria e usando le (6.32)–(6.35) si ottiene

$$\begin{aligned} 1 &= \frac{1}{2} \sum_{i=0}^m b_{i,0} (W_i + 1) \\ &= \frac{b_{0,0}}{2} \left[\sum_{i=0}^{m-1} p^i (2^i W + 1) + \frac{p^m}{1-p} (2^m W + 1) \right]. \end{aligned}$$

Separando i termini con e senza W ,

$$1 = \frac{b_{0,0}}{2} \left[\frac{1}{1-p} + W \left(\frac{1 - (2p)^m}{1-2p} + \frac{(2p)^m}{1-p} \right) \right].$$

Una stazione trasmette esattamente quando il suo contatore vale zero. La probabilità di trasmissione in uno slot è pertanto

$$\tau = \sum_{i=0}^m b_{i,0}. \quad (6.36)$$

Dalle (6.32) e (6.33),

$$\tau = b_{0,0} \left(\sum_{i=0}^{m-1} p^i + \frac{p^m}{1-p} \right) = \frac{b_{0,0}}{1-p},$$

cioè

$$b_{0,0} = \tau(1-p).$$

Sostituendo questa relazione nella normalizzazione e semplificando le serie geometriche si ha

$$\begin{aligned} 1 &= \frac{\tau}{2} \left[1 + W \left(\frac{(1-p)[1 - (2p)^m]}{1-2p} + (2p)^m \right) \right] \\ &= \frac{\tau}{2} \left[1 + W \frac{1-p-p(2p)^m}{1-2p} \right]. \end{aligned}$$

Risolvendo rispetto a τ si ottiene quindi

$$\tau = \frac{2(1-2p)}{(1-2p)(W+1) + pW[1 - (2p)^m]} \quad (6.37)$$

che è la prima equazione di punto fisso del modello. Una forma algebricamente equivalente e numericamente più robusta è

$$\tau(p) = \frac{2}{W+1 + Wp \sum_{j=0}^{m-1} (2p)^j}.$$

È ottenuta sostituendo la somma geometrica finita a $[1 - (2p)^m]/(1 - 2p)$ ed elimina la singolarità removibile.

Nel punto $p = 1/2$ numeratore e denominatore della (6.37) si annullano. Applicando il limite, oppure usando direttamente la forma precedente alla semplificazione, si ottiene

$$\tau = \frac{4}{2(W+1) + mW}.$$

Resta da legare p a τ . Condizionatamente al fatto che una stazione trasmetta, non si verifica collisione soltanto se nessuna delle altre $n - 1$ stazioni trasmette nello stesso slot. Nell'ipotesi di indipendenza del modello questa probabilità è $(1 - \tau)^{n-1}$. Pertanto

$$p = 1 - (1 - \tau)^{n-1} \quad (6.38)$$

che è la seconda equazione di punto fisso. Le (6.37) e (6.38) determinano quindi numericamente la coppia (τ, p) . Per $W \geq 2$, $\tau(p)$ è continua e non crescente, quindi $1 - [1 - \tau(p)]^{n-1}$ è non crescente. La differenza fra p e tale espressione è strettamente crescente: per $n > 1$ è negativa in 0 e positiva in 1. Il punto fisso è dunque unico in $(0, 1)$ e si trova per bisezione. Con una sola stazione $p = 0$.

Derivazione del throughput

Determinati τ e p , consideriamo ora uno slot generico della rete. Lo slot è vuoto se nessuna delle n stazioni trasmette; per indipendenza

$$P_0 = (1 - \tau)^n. \quad (6.39)$$

Uno slot contiene una trasmissione riuscita se trasmette esattamente una stazione. Vi sono n possibili stazioni trasmettenti; per ciascuna la probabilità che essa trasmetta e tutte le altre tacciano è $\tau(1 - \tau)^{n-1}$. Gli eventi sono disgiunti, quindi

$$P_s = n\tau(1 - \tau)^{n-1}. \quad (6.40)$$

Tutti i casi restanti corrispondono a collisione:

$$P_c = 1 - P_0 - P_s. \quad (6.41)$$

Le tre probabilità sommano evidentemente a uno.

Sia σ la durata di uno slot vuoto, T_s la durata di uno slot con successo e T_c la durata di uno slot con collisione. Per la legge dell'attesa totale la durata media dello slot è

$$\bar{T} = P_0\sigma + P_sT_s + P_cT_c. \quad (6.42)$$

Se ogni successo consegna in media L bit utili, il reward utile di uno slot è L con probabilità P_s e zero negli altri casi. Il reward medio per slot è pertanto P_sL . Applicando il rapporto tra reward medio e durata media del ciclo si ottiene il throughput utile

$$S = \frac{P_sL}{\bar{T}}. \quad (6.43)$$

Indicando con R il bit rate fisico e con $T_P = L/R$ il tempo di trasmissione del solo payload,

$$\frac{S}{R} = \frac{P_sT_P}{\bar{T}}. \quad (6.44)$$

Questa forma normalizzata discende semplicemente dalla sostituzione $L = RT_P$.

Basic Access

Nel Basic Access una trasmissione corretta comprende la trama dati, l'ACK e gli intervalli temporali necessari. Indicando con H il tempo di trasmissione degli header PHY e MAC e con δ il ritardo di propagazione, nel modello di Bianchi si può scrivere

$$T_s^{(B)} = H + T_p + SIFS + \delta + T_{ACK} + DIFS + \delta, \quad (6.45)$$

mentre una collisione occupa il mezzo per

$$T_c^{(B)} = H + T_p + DIFS + \delta. \quad (6.46)$$

Queste durate rappresentano il modello temporale scelto; una implementazione può spendere tempi diversi per ACK timeout, attesa estesa dopo errori e ripresa del backoff. Tali tempi vanno sostituiti coerentemente in T_s e T_c prima di ricavare una previsione per uno specifico PHY/MAC. La (6.43) con $T_s = T_s^{(B)}$ e $T_c = T_c^{(B)}$ fornisce direttamente il throughput sotto Basic Access.

RTS/CTS

Con RTS/CTS il successo comprende RTS, CTS, dati e ACK:

$$\begin{aligned} T_s^{(R)} = & T_{RTS} + SIFS + \delta + T_{CTS} + SIFS + \delta \\ & + H + T_p + SIFS + \delta + T_{ACK} + DIFS + \delta, \end{aligned} \quad (6.47)$$

mentre una collisione interessa essenzialmente l'RTS e quindi

$$T_c^{(R)} = T_{RTS} + DIFS + \delta. \quad (6.48)$$

Il vantaggio di RTS/CTS emerge quindi quando il costo di una collisione sulla trama dati sarebbe elevato rispetto all'overhead aggiuntivo del quattro-vie.

Tempo medio di servizio in saturazione

Il modello permette di ricavare anche una grandezza temporale senza introdurre un'ulteriore approssimazione sul backoff. In regime di saturazione il numero medio di successi per unità di tempo dell'intera rete è $P_s/\bar{T}$; per simmetria ogni stazione ne ottiene $1/n$. Il tempo medio tra due propri successi consecutivi, che nel modello saturo coincide con il tempo medio di servizio MAC di una trama posta in testa alla coda immediatamente dopo il successo precedente, è quindi

$$E[T_{serv}] = \frac{n\bar{T}}{P_s}. \quad (6.49)$$

Normalizzando rispetto al tempo di payload e usando la (6.44) si ottiene la forma particolarmente semplice

$$\frac{E[T_{serv}]}{T_p} = \frac{n}{S/R}. \quad (6.50)$$

Questa non è il ritardo di accodamento di una rete non satura: è il tempo medio di servizio per stazione nel modello saturo di Bianchi.

Terminali nascosti

Il modello base presuppone che tutte le stazioni si sentano tra loro e non incorpora il terminale nascosto mediante un parametro empirico. Il problema dei terminali nascosti resta importante, ma va trattato come estensione del modello o mediante un modello topologico separato. In

questo modo la derivazione base rimane chiara e verificabile e l'effetto di RTS/CTS può essere studiato senza confondere collisioni da contesa e mancata visibilità reciproca.

Le figure seguenti sono generate direttamente dalle (6.37)–(6.50) usando tempi normalizzati rispetto a $T_p = 1$ con

$$\sigma = 0.02, \quad H = 0.08, \quad SIFS = 0.02, \quad DIFS = 0.05, \\ T_{ACK} = T_{RTS} = T_{CTS} = 0.04, \quad \delta = 0.005, \quad m = 5.$$

Salvo le curve che fanno variare W , si pone $W = 16$. Sono parametri didattici adimensionali, non una configurazione prescritta per una particolare generazione Wi-Fi.

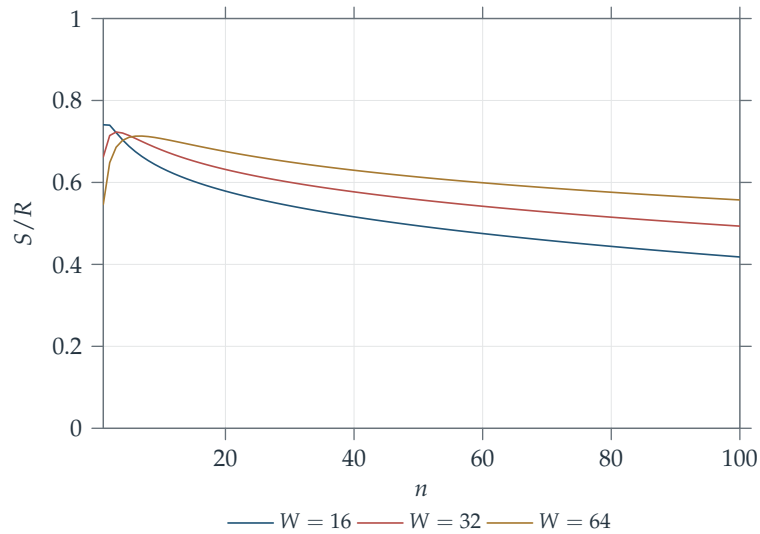


Figura 6.13. Modello di Bianchi: throughput normalizzato in funzione del numero di stazioni per diversi valori di W .

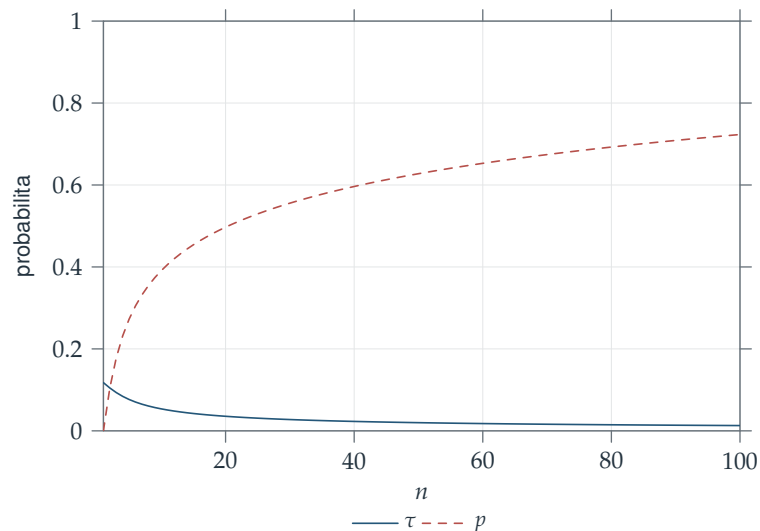


Figura 6.14. Modello di Bianchi: probabilità di trasmissione τ e probabilità di collisione p in funzione del numero di stazioni.

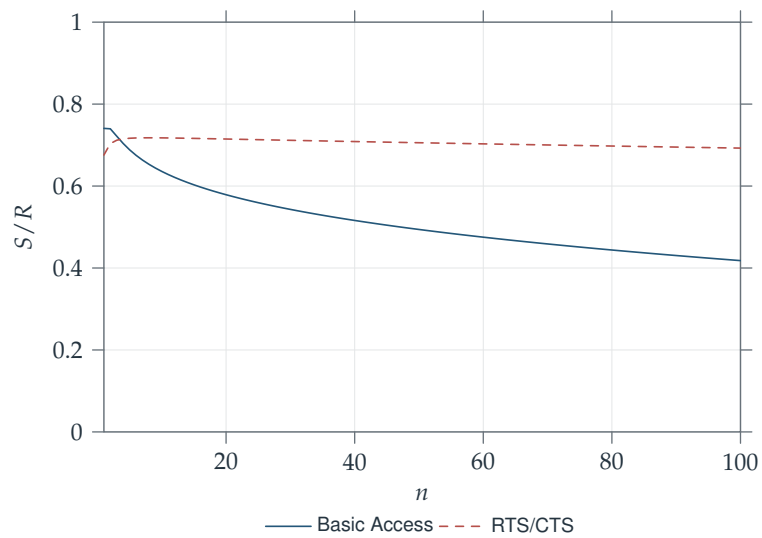


Figura 6.15. Modello di Bianchi: confronto del throughput normalizzato tra Basic Access e RTS/CTS.

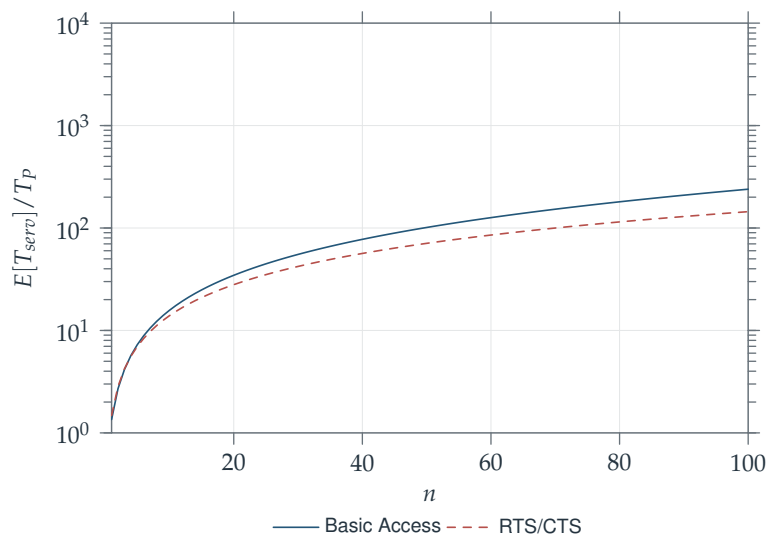


Figura 6.16. Modello di Bianchi: tempo medio di servizio normalizzato per stazione in saturazione.

Commutazione di circuito

La commutazione di circuito riserva, prima del trasferimento, un insieme di risorse che forma un percorso tra sorgente e destinazione. Per tutta la durata della comunicazione quelle risorse restano associate al circuito secondo le regole del sistema. Il modello si distingue quindi dalla commutazione di pacchetto, nella quale le risorse di trasmissione vengono normalmente condivise nel tempo tra flussi differenti e il traffico può formare code.

Il capitolo studia tre aspetti che conservano valore generale. Il primo è la struttura delle reti di commutazione, con particolare attenzione alle reti multistadio e alle condizioni di non bloccaggio. Il secondo è il dimensionamento statistico: anche se una singola connessione usa risorse dedicate, il numero di connessioni contemporaneamente attive è aleatorio e determina la probabilità di blocco. Il terzo è la segnalazione necessaria a instaurare, mantenere e rilasciare la connessione.

Le derivazioni sulle reti di Clos e sui modelli di traffico sono risultati matematici ottenuti sotto le ipotesi dichiarate. Le procedure di segnalazione, invece, sono descritte come meccanismi di protocollo: non sono teoremi e non vanno confuse con i modelli prestazionali.

7.1 Sistemi di commutazione

Si definisce rete di commutazione **strettamente non bloccante** un sistema in cui è sempre possibile instaurare una nuova connessione tra un ingresso libero e un'uscita libera senza modificare le connessioni già attive. Questo elemento è il mattone fondamentale per costruire un sistema a commutazione di circuito, in quanto consente alle varie centrali, intermedie tra la sorgente ed il destinatario, di costruire il circuito utilizzando le risorse di giunzione disponibili. La proprietà riguarda il blocco *interno*, condizionatamente a un ingresso e a un'uscita liberi. Non elimina il rifiuto di una richiesta quando la destinazione o tutte le risorse esterne richieste sono occupate: quest'ultimo è il problema statistico studiato più avanti con Erlang B.

Di seguito si dà una descrizione dei più comuni schemi di realizzazione di una rete di commutazione: a divisione di spazio (tipo S), a divisione di tempo (tipo T), ibridi e multistadio, fornendo qualche dettaglio sulla scelta dei parametri per renderle non bloccanti.

7.1.1 Reti a divisioni di spazio, tipo S

In figura 7.1 è riportata la più semplice rete di commutazione non bloccante, che permette di connettere direttamente ogni ingresso ad ogni possibile uscita. Questa soluzione appartiene alla classe delle reti dette a DIVISIONE DI SPAZIO (o reti S) in quanto tutte le connessioni presenti tra ingresso ed uscita sono localizzate in uno spazio fisico.

È interessante definire la complessità della rete, U , intesa come numero di punti di commutazione, che risulta:

$$U = N^2 \quad (7.1)$$

per il caso riportato in figura 7.1.

Una soluzione alternativa, concettualmente più complessa, è composta da tre stadi posti in cascata come mostrato in figura 7.2, dove il primo stadio è composto da N/x matrici di

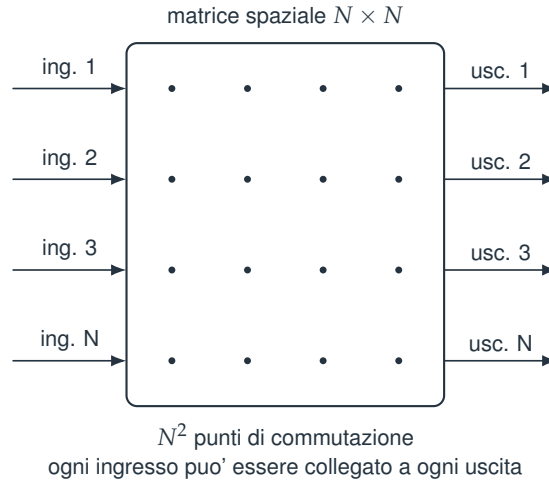


Figura 7.1. Matrice di commutazione con tutti i collegamenti possibili.

dimensione $x \cdot k$, il secondo stadio da k matrici $N/x \cdot N/x$ e il terzo stadio da N/x matrici $k \cdot x$. Si assume qui che x sia un divisore intero di N , in modo che N/x sia intero. I collegamenti uscenti da una singola matrice del primo stadio finiscono tutti su matrici diverse del secondo stadio. Analogamente accade, in modo speculare, per i collegamenti tra il secondo ed il terzo stadio. Per quanto riguarda la complessità si ha:

$$U = \frac{N}{x}xk + k \left(\frac{N}{x} \right)^2 + \frac{N}{x}xk = 2Nk + k \left(\frac{N}{x} \right)^2 \quad (7.2)$$

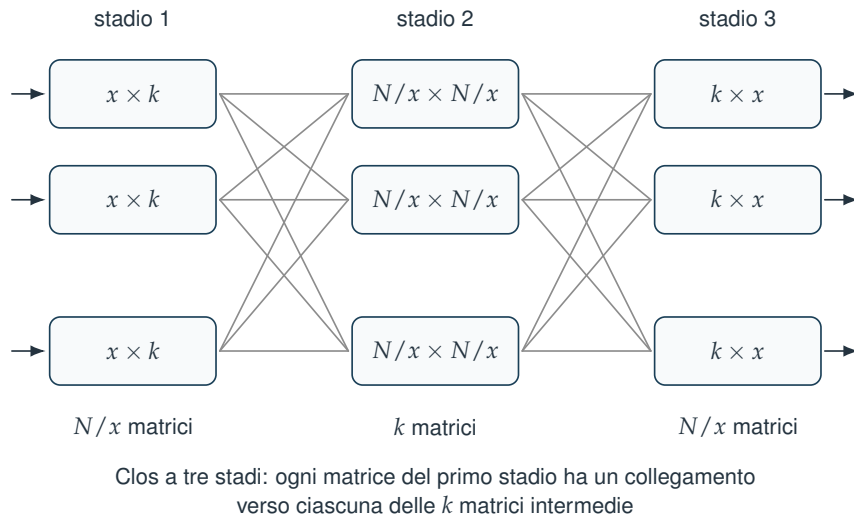


Figura 7.2. Matrice di commutazione a 3 stadi.

Condizione necessaria per avere una rete non bloccante è $k \geq x$, come si verifica immediatamente ragionando sul primo stadio: infatti si può osservare che, se si vogliono attivare tutti gli x ingressi di una matrice del primo stadio, è necessario avere almeno x uscite attive. La condizione sufficiente è data dal seguente teorema.

Teorema di CLOS

Teorema 11. *Una rete di Clos a tre stadi è strettamente non bloccante se $k \geq 2x - 1$. Il valore $k = 2x - 1$ è quindi il minimo numero di matrici intermedie garantito da questa condizione sufficiente.*

Sia I una matrice di ingresso presente nel primo stadio ed O una matrice di uscita del terzo stadio. Siano I ed O interessate ad una nuova connessione che le coinvolge entrambe. Il caso peggiore si ha quando I possiede già $x - 1$ connessioni e O è già raggiunta da $x - 1$ connessioni. Le prime possono occupare al più $x - 1$ matrici intermedie e le seconde al più altre $x - 1$. Se i due insiemi sono disgiunti, che è la configurazione peggiore, risultano indisponibili $2(x - 1)$ matrici intermedie. Con $k \geq 2(x - 1) + 1 = 2x - 1$ ne rimane quindi almeno una libera sia sul lato di I sia sul lato di O e la nuova connessione può essere instaurata senza riordinare quelle esistenti. Questo dimostra la sufficienza della condizione.

Per minimizzare la complessità si considera il valore minimo $k = 2x - 1$ e lo si sostituisce nella (7.2):

$$U = 2N(2x - 1) + (2x - 1) \left(\frac{N}{x} \right)^2 = (2x - 1) \left(2N + \frac{N^2}{x^2} \right)$$

Per ottenere una stima analitica del valore di x che minimizza la complessità, rilassiamo temporaneamente il vincolo discreto e consideriamo $1 \leq x \leq N$ come variabile reale; consentire $x \downarrow 0$ renderebbe negativo $k = 2x - 1$ e non rappresenterebbe più una rete. Il progetto effettivo sceglierà poi fra i divisori interi di N . Per i punti stazionari interni si calcola:

$$\begin{aligned} \frac{dU}{dx} &= 2 \left(2N + \frac{N^2}{x^2} \right) - (2x - 1) \frac{2N^2}{x^3} = 0, \\ 4N + \frac{2N^2}{x^2} - \frac{4N^2}{x^2} + \frac{2N^2}{x^3} &= 0, \\ 4x^3 - 2xN + 2N &= 0. \end{aligned}$$

L'equazione esatta dei punti stazionari è

$$2x^3 - Nx + N = 0.$$

Non ogni sua radice ammissibile è un minimo. Infatti

$$\frac{d^2U}{dx^2} = \frac{2N^2(2x - 3)}{x^4} :$$

un punto stazionario con $x > 3/2$ è un minimo locale, mentre con $x < 3/2$ è un massimo locale. Il minimo globale sul dominio fisico si ottiene confrontando gli eventuali minimi interni con gli estremi $x = 1$ e $x = N$. Per N grande, il termine N è di ordine inferiore rispetto a Nx nell'intorno dell'ottimo, per cui

$$2x^3 - Nx \simeq 0.$$

Escludendo la soluzione non fisica $x = 0$ si ottiene

$$x \simeq \sqrt{\frac{N}{2}}.$$

Il valore usato nel progetto va infine scelto tra i divisori di N che minimizzano realmente la (7.2), includendo gli estremi: un arrotondamento arbitrario dell'ottimo continuo può non essere ammissibile o non essere ottimale.

Con la scelta approssimata $x \simeq \sqrt{N/2}$ e mantenendo $k = 2x - 1$, la complessità vale

$$U \simeq (2\sqrt{N/2} - 1)4N = 4\sqrt{2}N^{3/2} - 4N.$$

Per N grande il termine dominante è quindi

$$U \sim 4\sqrt{2}N^{3/2},$$

che è notevolmente ridotto rispetto al caso composto da un singolo stadio, espresso dalla (7.1). Queste considerazioni portano a concludere che può essere conveniente realizzare sistemi di commutazione a più stadi anche se risultano più complessi da progettare e da gestire; infatti, nel caso a tre stadi occorre programmare tre matrici per ottenere la commutazione, invece di una nel caso ad un solo stadio.

7.1.2 Reti a divisioni di tempo, tipo T

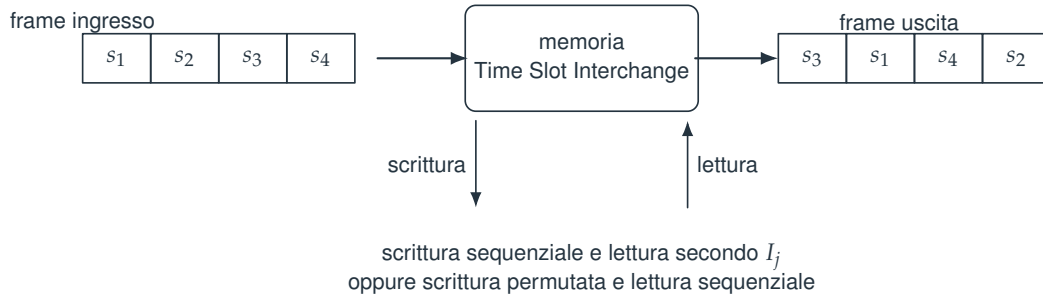


Figura 7.3. Matrice di commutazione a divisione di tempo.

Si consideri lo schema di figura 7.3 dove i diversi flussi informativi sono posti sullo stesso canale a divisione di tempo secondo la tecnica TDM e la commutazione avviene memorizzando sequenzialmente i vari slot in un buffer e rileggendoli secondo una funzione di commutazione I_j , che dichiara da quale slot di ingresso memorizzato deve essere prelevato il contenuto che deve occupare lo slot j in uscita.

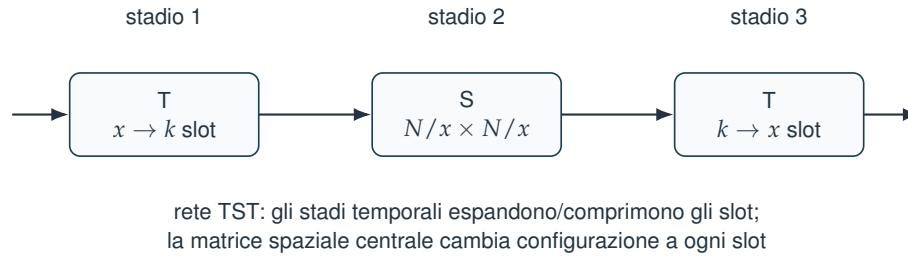
In modo complementare è possibile memorizzare i vari slot secondo una funzione $\tilde{I}_j$ e rileggerli sequenzialmente; in tal caso $\tilde{I}_j$ dichiara in quale posizione deve essere memorizzato il contenuto che si trova nello slot j di ingresso.

Questo tipo di sistema funziona a DIVISIONE DI TEMPO (o rete T) e spesso viene chiamato Time Slot Interchange (TSI). Per realizzare una permutazione arbitraria, il controllo deve anche impedire che un campione venga letto prima di essere stato scritto o sovrascritto prima della lettura. Si può, per esempio, memorizzare una trama completa e leggerla nel periodo successivo, separando le aree di memoria delle due trame. La permutazione comporta quindi anche un vincolo di buffering e latenza, non soltanto una selezione di indirizzi.

Per una rete T il parametro di interesse è il massimo numero di slot commutabili, n . Nel modello più semplice ogni slot richiede una scrittura e una lettura della memoria, eseguite sequenzialmente. Se l'intero insieme degli slot deve essere trattato entro un intervallo T_{max} e ogni accesso alla memoria richiede t_{ac} , allora:

$$n = \left\lfloor \frac{T_{max}}{2t_{ac}} \right\rfloor$$

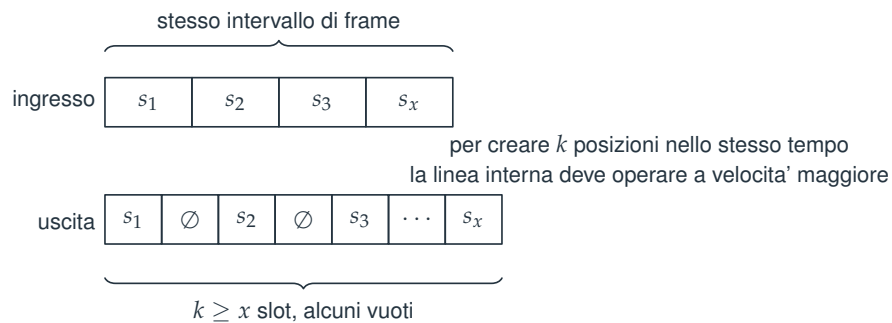
dove il termine 2 è dovuto alla cascata della procedura di scrittura e lettura. Le prestazioni di questo sistema sono quindi limitate dalla velocità della memoria.

**Figura 7.4.** Rete a tre stadi TST.

7.1.3 Reti ibride

Si tratta di combinazioni di reti sia a divisione di spazio che di tempo. Un esempio, di tipo TST, è riportato in figura 7.4, dove le reti T svolgono le funzioni che nella rete di figura 7.2 erano effettuate dalle matrici S di dimensione $x \cdot k$. Lo stadio S è invece composto da una singola matrice di dimensione $N/x \cdot N/x$ che però deve cambiare configurazione per ogni nuovo slot in ingresso, cioè deve cambiare k volte per ogni frame composto da x slot.

Riguardo agli stadi T, se $k > x$ il sistema deve mappare gli x slot di ingresso in k slot interni nello stesso intervallo temporale. A parità di bit per slot occorre allora un bit rate interno maggiore, come in figura 7.5, e rimangono posizioni non usate dai soli x ingressi. Con $k = x$, invece, non è richiesto un aumento di bit rate e non segue necessariamente la presenza di slot vuoti.

**Figura 7.5.** Rete T con cambio di velocità.

Dal punto di vista della scelta di un cammino interno, lo schema TST e lo schema SSS di figura 7.2 hanno quindi una struttura analoga; non sono però equivalenti sul piano fisico, perché nel TST la selezione degli slot richiede memoria e riconfigurazione temporale. È interessante osservare che rispetto al caso SSS, dove il secondo stadio ha complessità $k \cdot N/x \cdot N/x$, ora il secondo stadio ha complessità $N/x \cdot N/x$; però commuta k volte per ogni trama di x slot.

Anche per questo tipo di sistema sussiste un teorema di interesse, analogo a quello di Clos:

Teorema 12. *Nel modello descritto una rete TST è strettamente non bloccante se $k \geq 2x - 1$; il valore $k = 2x - 1$ è il minimo che garantisce la condizione con l'argomento seguente.*

Con la stessa simbologia del teorema di Clos, una nuova connessione può trovare indisponibili al più $x - 1$ posizioni temporali per effetto delle connessioni già attive sul lato di ingresso e al più altre $x - 1$ sul lato di uscita. Nel caso peggiore i due insiemi sono disgiunti. Servono quindi almeno $k = (x - 1) + (x - 1) + 1 = 2x - 1$ posizioni affinché ne resti sempre una comune libera. La figura 7.6 rappresenta questa configurazione limite.



nel caso peggiore i due insiemi di slot occupati sono disgiunti;
serve una posizione ulteriore comune per la nuova connessione

$$k \geq (x - 1) + (x - 1) + 1 = 2x - 1$$

Figura 7.6. Ingressi ed uscite nodo S.

7.1.4 Sistemi multistadio

Aumentare il numero degli stadi permette di ottenere molte realizzazioni diverse e di scambiare complessità hardware, capacità di riconfigurazione e probabilità di blocco. Per il principio che ci interessa non è necessario catalogare tutte le sequenze possibili di stadi S e T: è più utile distinguere due problemi diversi.

Il primo è *strutturale*: si cercano condizioni che garantiscano l'assenza di blocco per qualunque configurazione ammissibile delle connessioni già attive. Il teorema di Clos appartiene a questa classe e fornisce una garanzia deterministica. Il secondo è *prestazionale*: se il sistema viene progettato con meno risorse di quelle richieste dalla garanzia stretta, si stima con quale probabilità una richiesta trovi tutti i cammini interni occupati. Questa seconda valutazione richiede un modello statistico del traffico e, in generale, non coincide con una proprietà esatta della rete.

Consideriamo nuovamente la rete a tre stadi SSS; la stessa approssimazione può essere applicata alla struttura TST quando le posizioni temporali alternative svolgono il ruolo dei cammini intermedi. Sia

$$a = P\{\text{un ingresso è occupato}\},$$

e supponiamo sistema omogeneo e traffico uniformemente distribuito sui k collegamenti interstadio. Il carico medio prodotto dai x ingressi di una matrice su uno specifico collegamento è

$$p = \frac{x}{k}a. \quad (7.3)$$

Nel modello di indipendenza comunemente detto modello di Lee, p viene identificato con la probabilità che un collegamento interstadio sia occupato e le occupazioni dei diversi collegamenti vengono considerate indipendenti. Sono precisamente queste le ipotesi approssimate del modello.

Per un dato stadio intermedio, il cammino tra la matrice di ingresso e quella di uscita è disponibile soltanto se sono liberi entrambi i collegamenti che lo compongono. La probabilità di disponibilità del cammino è quindi, nel modello indipendente,

$$(1 - p)^2,$$

mentre la probabilità che quel cammino non sia disponibile è $1 - (1 - p)^2$. Assumendo indipendenti anche le k alternative, il blocco richiede che tutte siano indisponibili e si ottiene

$$P_B^{(\text{Lee})} = [1 - (1 - p)^2]^k = \left[1 - \left(1 - \frac{xa}{k}\right)^2\right]^k. \quad (7.4)$$

La formula è quindi una *stima* del blocco interno per una richiesta con estremi liberi, non una probabilità esatta della rete di Clos. La sostituzione dell'occupazione condizionata con

il valore medio xa/k appartiene anch'essa all'approssimazione. Si usano $0 \leq a \leq 1$ e $k \geq x$, così che $0 \leq p \leq 1$; le curve continue interpolano valori fisicamente realizzabili soltanto per k intero.

Questo punto è importante vicino alla soglia di non blocco. Per $k \geq 2x - 1$ il teorema di Clos garantisce esattamente

$$P_B = 0$$

per ogni configurazione delle connessioni ammissibile, mentre la (7.4) può restituire un valore positivo perché ignora le dipendenze strutturali che producono quella garanzia. La stima di Lee non deve quindi essere usata per contraddire il risultato combinatorio. In figura 7.7, con $x = 100$, la curva viene mostrata soltanto nella regione $x \leq k < 2x - 1$, cioè prima della soglia strettamente non bloccante.

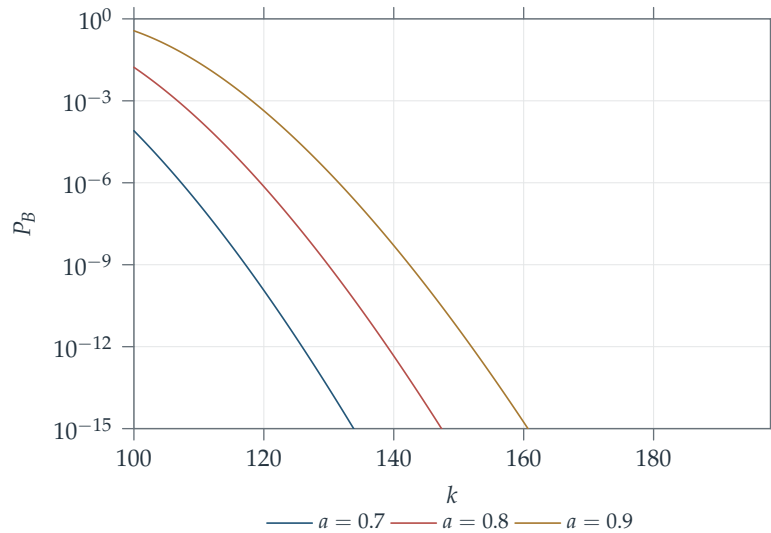


Figura 7.7. Stima di Lee della probabilità di blocco per una rete a tre stadi con $x = 100$, nella regione precedente la soglia esatta $k = 2x - 1 = 199$.

Al crescere di a aumenta l'occupazione interna e quindi cresce la stima di blocco; al crescere di k aumentano invece i cammini alternativi e la stima diminuisce. Il modello rende dunque esplicito il compromesso fra risorse interne e probabilità di rifiuto, mentre la condizione di Clos resta il riferimento quando si richiede una garanzia deterministica di non blocco.

7.2 Prestazioni dei sistemi a commutazione di circuito

Un sistema a commutazione di circuito può essere modellato come un sistema a coda con un certo numero Y di servitori, a cui i vari clienti accedono per poter instaurare un collegamento verso il destinatario. Il numero dei servitori rappresenta anche il numero massimo di connessioni che si possono instaurare. Se gli arrivi formano un processo di Poisson e i tempi di occupazione sono esponenziali, il sistema di commutazione è assimilabile a un sistema a perdita di tipo $M/M/Y/Y$, discusso nel capitolo 2, purché ogni richiesta occupi una sola fra Y risorse equivalenti e tutte le risorse siano accessibili a ogni richiesta. Si assumono durate indipendenti fra loro e dagli arrivi. Una connessione che richieda simultaneamente più risorse specifiche in nodi diversi non è descritta, in generale, da un unico gruppo Erlang B. Se una richiesta arriva quando tutte le Y risorse sono occupate, viene bloccata e cancellata.

Nel dimensionamento di un sistema a perdita conviene separare le seguenti grandezze:

- **Traffico offerto A :** quantità media di lavoro offerta nell'intervallo pari al tempo medio di

occupazione. Con arrivi di tasso λ e tempo medio di occupazione $1/\mu$,

$$A = \frac{\lambda}{\mu}.$$

È una grandezza adimensionale, espressa convenzionalmente in Erlang (Erl).

- **Traffico smaltito** A_s : parte del traffico offerto che viene effettivamente accettata. In un sistema a circuiti coincide con il numero medio di risorse contemporaneamente occupate. Un traffico smaltito di un Erlang corrisponde quindi, per esempio, a una singola risorsa occupata in media continuamente oppure a più risorse con somma delle utilizzazioni pari a uno.
- **Traffico perduto** A_p : parte del traffico offerto che non può essere accettata perché tutte le risorse compatibili sono occupate. Per definizione

$$A = A_s + A_p.$$

- **Rendimento** η : traffico smaltito medio per risorsa,

$$\eta = \frac{A_s}{Y}, \quad 0 \leq \eta \leq 1.$$

Nel modello omogeneo misura direttamente la frazione media di capacità utilizzata; non va confuso con la probabilità che una singola richiesta venga accettata.

Definendo con P_B la probabilità di blocco del sistema, si ha:

$$A_p = P_B A$$

$$A_s = A - A_p = (1 - P_B) A$$

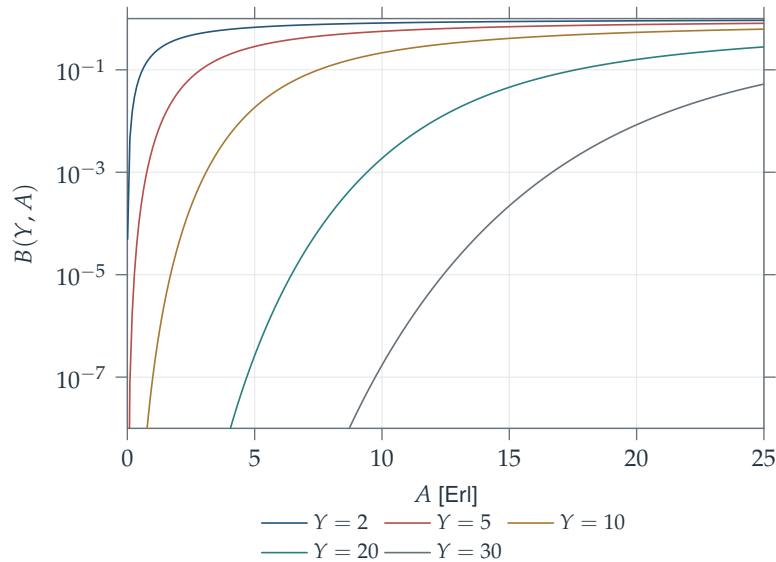


Figura 7.8. Probabilità di blocco secondo la B di Erlang al variare del numero dei servitori Y .

Considerando inizialmente un sistema $M/M/Y/Y$ si ha:

$$A_s = A[1 - B(Y, A)] = E[k]$$

$$\eta = \frac{A_s}{Y} = \frac{A[1 - B(Y, A)]}{Y}$$

$$P_B = B(Y, A)$$

Nel seguito manteniamo il modello $M/M/Y/Y$ già derivato nel capitolo 2; non è necessario introdurre qui estensioni a distribuzioni generiche dei tempi di occupazione.

La funzione $B(Y, A)$ di Erlang è stata ricavata nel capitolo 2, equazione (2.34). Per il calcolo numerico è preferibile usare una ricorsione che evita fattoriali e potenze elevate. Per $A > 0$, dalla definizione,

$$B(Y, A) = \frac{A^Y/Y!}{\sum_{k=0}^Y A^k/k!}.$$

Separando l'ultimo termine della somma al denominatore e usando

$$\sum_{k=0}^{Y-1} \frac{A^k}{k!} = \frac{A^{Y-1}/(Y-1)!}{B(Y-1, A)},$$

si ottiene

$$\begin{aligned} B(Y, A) &= \frac{A^Y/Y!}{A^{Y-1}/[(Y-1)!B(Y-1, A)] + A^Y/Y!} \\ &= \frac{AB(Y-1, A)}{Y + AB(Y-1, A)}. \end{aligned}$$

Quindi

$$B(Y, A) = \frac{AB(Y-1, A)}{Y + AB(Y-1, A)} \quad (7.5)$$

con condizione iniziale $B(0, A) = 1$. Per $A = 0$ la ricorsione restituisce $B(Y, 0) = 0$ per ogni $Y \geq 1$, come richiede l'assenza di traffico; $B(0, 0) = 1$ è soltanto una convenzione per inizializzarla. Questa forma è numericamente stabile e viene usata per il dimensionamento e per rigenerare i grafici.

In figura 7.8 viene presentata la probabilità di blocco P_B in funzione del traffico offerto al variare del numero dei servitori Y . Si noti come la probabilità di blocco cresca con il crescere del traffico offerto e cali all'aumentare del numero dei servitori.

Il numero minimo dei servitori necessario a garantire una determinata probabilità di blocco viene riportato in figura 7.9 al variare del traffico offerto. Fissata P_B , Y cresce al crescere di A , dato che si deve smaltire maggiore traffico. Inoltre, fissato A , Y cala al crescere di P_B , dato che è necessario avere un minor numero di servitori per soddisfare lo stesso traffico offerto con una maggiore probabilità di blocco. L'andamento a gradini è una conseguenza diretta del fatto che Y deve essere intero: il dimensionamento cambia soltanto quando il vincolo di blocco impone l'aggiunta di una nuova risorsa.

Ai fini della stima del rendimento

$$\eta = \frac{A_s}{Y} = \frac{A[1 - B(Y, A)]}{Y},$$

è importante distinguere due confronti diversi. Se A è fissato e si aumenta soltanto Y , il traffico smaltito tende ad A mentre il denominatore cresce, quindi non è corretto affermare che η aumenti semplicemente con Y .

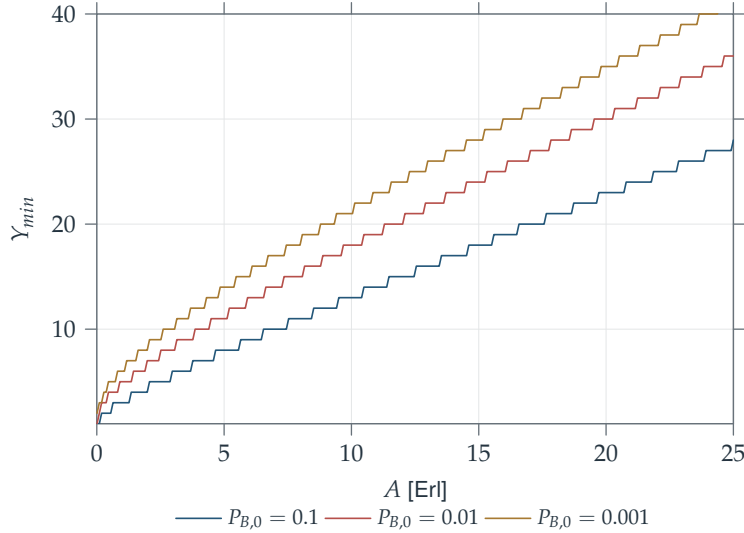


Figura 7.9. Numero dei server necessari a garantire una determinata probabilità di blocco.

Nel dimensionamento di un sistema a perdita interessa invece fissare un grado di servizio, cioè una probabilità di blocco obiettivo $P_{B,0}$, e confrontare sistemi di scala diversa. Per $A > 0$ e $0 < P_{B,0} < 1$ definiamo

$$Y_{min}(A, P_{B,0}) = \min\{Y \in \mathbb{N}, Y \geq 1 : B(Y, A) \leq P_{B,0}\}.$$

Il rendimento del sistema dimensionato è allora

$$\eta_{dim}(A, P_{B,0}) = \frac{A[1 - B(Y_{min}, A)]}{Y_{min}}. \quad (7.6)$$

È questo il confronto corretto per valutare l'effetto di aggregazione: A e Y devono variare insieme sotto lo stesso vincolo di blocco. La figura 7.10 calcola direttamente la (7.6) e mostra, per i valori considerati, come il rendimento cambi al crescere del traffico offerto. Le piccole cadute periodiche sono dovute all'incremento discreto di Y_{min} quando serve aggiungere una nuova risorsa. L'esempio seguente quantifica lo stesso fenomeno confrontando una risorsa aggregata con più sottogruppi.

7.2.1 Esempio di dimensionamento

Consideriamo un sistema che riceve in media 300 richieste l'ora, con durata media di occupazione pari a 120 s, e imponiamo una probabilità di blocco non superiore a 10^{-3} . Il traffico offerto è

$$A = \frac{300 \cdot 120}{3600} = 10 \text{ Erl.}$$

Dalla ricorsione (7.5), coerentemente con la figura 7.9, si ottiene $B(20, 10) \simeq 1.87 \cdot 10^{-3}$ mentre $B(21, 10) \simeq 8.89 \cdot 10^{-4}$: il minimo è quindi $Y = 21$.

Se invece il traffico viene suddiviso in sottogruppi indipendenti, ciascuno servito da $Y = 5$ risorse, la ricorsione (7.5) mostra che il massimo traffico per cui $B(5, A) \leq 0.001$ è circa $A = 0.762$ Erl. Per un traffico complessivamente offerto di 10 Erl, suddiviso uniformemente fra sottogruppi indipendenti, con lo stesso grado di perdita servono quindi

$$\left\lceil \frac{10}{0.762} \right\rceil = 14$$

sottogruppi, per un totale di $14 \cdot 5 = 70$ risorse, contro le 21 richieste dal sistema completamente aggregato. L'esempio quantifica direttamente il vantaggio di trunking.

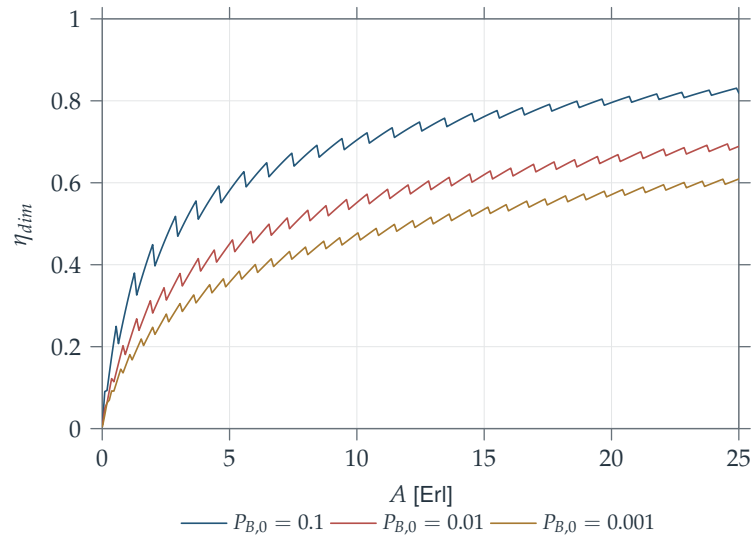


Figura 7.10. Rendimento per una data probabilità di blocco.

7.3 Procedure di segnalazione

La riserva delle risorse non nasce automaticamente dal trasferimento dei dati: serve un piano di controllo che costruisca lo stato della connessione nei nodi attraversati. La segnalazione comprende quindi le informazioni necessarie a richiedere il circuito, scegliere o confermare il percorso, verificare la disponibilità delle risorse, attivarle e infine rilasciarle. Queste sono funzioni di protocollo, non risultati matematici.

Una sequenza astratta di instaurazione può essere descritta in quattro passi:

1. la sorgente genera una richiesta di connessione specificando la destinazione e i parametri di servizio necessari;
2. ciascun nodo attraversato individua il passo successivo e verifica se la risorsa richiesta è disponibile; se il controllo di ammissione fallisce, la richiesta viene rifiutata e le eventuali risorse già prenotate devono essere liberate;
3. raggiunta la destinazione, una conferma torna verso la sorgente e rende operativo lo stato installato lungo il percorso;
4. al termine della comunicazione un messaggio di rilascio elimina lo stato della connessione e rende nuovamente disponibili le risorse.

La figura 7.11 rappresenta questa logica con nomi di messaggio volutamente generici. Non descrive una particolare specifica di segnalazione: mostra la dipendenza causale fra instaurazione, trasferimento e rilascio.

È utile distinguere il *piano di controllo*, che scambia la segnalazione e modifica lo stato dei nodi, dal *piano dati*, che usa il circuito dopo che le risorse sono state predisposte. La separazione logica non impone che i due piani usino necessariamente mezzi fisici distinti.

La segnalazione è detta *in banda* quando viaggia sulla stessa risorsa usata dall'informazione utente, ed è detta *fuori banda* quando usa una risorsa di controllo separata, eventualmente condivisa fra molte connessioni. La seconda soluzione consente al controllo di procedere anche quando i circuiti dati sono occupati e facilita una gestione centralizzata o distribuita dello stato; richiede però che la rete di controllo sia a sua volta disponibile e coerente con quella dei dati.

Il tempo di instaurazione di una connessione comprende quindi almeno i ritardi di propagazione dei messaggi di controllo, i tempi di elaborazione nei nodi e gli eventuali

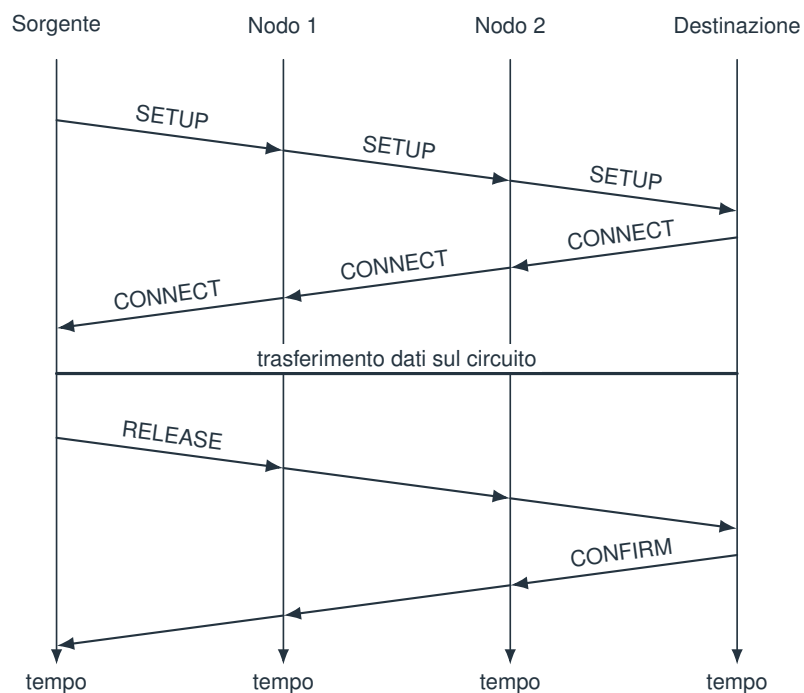


Figura 7.11. Sequenza astratta di instaurazione, uso e rilascio di un circuito attraverso nodi intermedi.

tempi necessari alla selezione o prenotazione delle risorse. È distinto dal tempo di trasferimento sul circuito già costituito. Analogamente, il rilascio non è soltanto un messaggio finale: deve portare il sistema in uno stato consistente, evitando che risorse non più utilizzate rimangano allocate.

Internetworking

Una rete reale è quasi sempre composta da più domini di collegamento e di instradamento. L'*internetworking* è l'insieme dei meccanismi che permette a sistemi appartenenti a reti differenti di comunicare mantenendo separate, quando opportuno, le funzioni dei diversi livelli. Il problema non consiste soltanto nel trasferire un segnale da un mezzo all'altro: occorre decidere dove inoltrare i pacchetti, come identificare gli endpoint, come gestire percorsi alternativi, errori, congestione e vincoli di sicurezza.

È utile distinguere le funzioni degli apparati senza legarle a una particolare realizzazione fisica:

- **Repeater.** Opera al livello fisico e rigenera o ritrasmette il segnale. Non interpreta gli indirizzi di livello superiore e non separa il dominio di traffico.
- **Bridge e switch.** Operano al livello di collegamento e inoltrano trame in base agli indirizzi MAC. Uno switch Ethernet può essere visto, dal punto di vista funzionale, come un bridge multiporta: apprende la posizione delle sorgenti, filtra il traffico che non deve attraversare una porta e inoltra quello necessario verso il segmento appropriato.
- **Router.** Opera al livello di rete e inoltra i pacchetti tra reti IP differenti usando una tabella di forwarding. La decisione dipende dal prefisso di destinazione e dalle informazioni prodotte dalla configurazione o dai protocolli di instradamento.
- **Gateway.** Il termine è usato in senso più generale per indicare un sistema che termina o traduce funzioni di livelli superiori, oppure che realizza un confine applicativo tra architetture diverse. Nel linguaggio operativo "default gateway" indica invece normalmente il router usato da un host per raggiungere reti non direttamente connesse.

Il capitolo procede dal collegamento alla rete e poi al trasporto. Si parte dagli indirizzi MAC e dalla commutazione di livello 2; si passa a IPv4 e IPv6, ai protocolli di instradamento e alle tecniche moderne di forwarding; si collegano quindi indirizzamento applicativo, TCP, UDP e RTP. Le sezioni finali richiamano i principi di sicurezza necessari a comprendere la protezione del traffico di rete.

Quando viene descritta la struttura di un protocollo, i valori dei campi e le procedure sono *specifiche*, non risultati da dimostrare. Le formule introdotte per code, scheduling, controllo di congestione o dimensionamento sono invece modelli analitici e vengono presentate con le relative ipotesi. Questa distinzione evita di confondere ciò che un protocollo prescrive con ciò che un modello matematico predice.

8.1 Indirizzi MAC

Nel caso Ethernet l'indirizzo di livello 2 è normalmente un indirizzo MAC di 48 bit, rappresentato come sei ottetti esadecimali separati da ":". Un esempio è 00:01:02:94:df:20. L'indirizzo compare nell'header della trama e permette agli apparati di livello 2 di selezionare il destinatario all'interno dello stesso dominio di collegamento.

È utile distinguere tre forme di destinazione:

- **unicast**, quando la trama è destinata a una singola interfaccia;

- **multicast**, quando l'indirizzo identifica un gruppo di ricevitori;
- **broadcast**, con indirizzo `ff:ff:ff:ff:ff:ff`, quando la trama deve essere ricevuta da tutte le interfacce del dominio di broadcast.

Il broadcast non è un indirizzo sorgente valido.

La struttura dell'indirizzo consente sia assegnazioni universalmente amministrate sia indirizzi localmente amministrati. Non è quindi corretto assumere che ogni MAC osservato in una rete identifichi per sempre e a livello mondiale un particolare dispositivo fisico: virtualizzazione, configurazioni locali e meccanismi di privacy possono produrre indirizzi non derivati da un identificatore hardware stabile. Ciò che conta per il forwarding di livello 2 è che l'indirizzo sia coerente nel dominio in cui viene utilizzato.

Una scheda di rete filtra normalmente le trame in base agli indirizzi che deve accettare, evitando di consegnare al sistema operativo tutto il traffico presente sul mezzo. In modalità promiscua il filtro può essere disabilitato per osservare le trame che raggiungono realmente l'interfaccia. Su una rete commutata questo non rende visibile il traffico che lo switch inoltra soltanto verso altre porte: per osservarlo occorre un punto di cattura appropriato.

8.2 Bridge e switch

Un bridge inoltra trame tra segmenti di livello 2 senza modificare la semantica degli indirizzi degli endpoint. Uno switch Ethernet è, dal punto di vista funzionale, un bridge multiporta. Il principio centrale è l'*apprendimento all'indietro*: osservando il MAC sorgente di ogni trama ricevuta, lo switch associa quel MAC alla porta di ingresso. Una voce della tabella può quindi essere schematizzata come

(VLAN, MAC) $\longrightarrow$ porta.

Le associazioni dinamiche sono soggette ad aging, così che cambiamenti di topologia o spostamenti degli endpoint non lascino indefinitamente informazioni obsolete.

Quando arriva una trama, la decisione di forwarding segue regole semplici:

- se il MAC destinatario è noto e associato a una porta diversa da quella di ingresso, la trama viene inoltrata soltanto su quella porta;
- se il MAC destinatario è noto sulla stessa porta di ingresso, la trama viene filtrata;
- se il MAC destinatario è sconosciuto, la trama viene replicata sulle altre porte appartenenti allo stesso dominio di broadcast;
- broadcast e, secondo la configurazione, multicast vengono replicati nel dominio pertinente.

L'apprendimento riguarda il MAC *sorgente*; la decisione di forwarding usa invece il MAC *destinatario*. Questa separazione è ciò che consente allo switch di costruire la tabella senza un protocollo di registrazione degli host.

8.2.1 Loop di livello 2 e spanning tree

La ridondanza fisica è utile, ma una topologia Ethernet con cicli crea un problema particolare: una trama broadcast o una trama unicast a destinazione sconosciuta può essere replicata lungo più rami e tornare sugli stessi switch. Poiché l'header Ethernet non contiene un contatore di hop analogo al TTL di IPv4, le copie possono continuare a circolare, consumando capacità e rendendo instabile anche l'apprendimento dei MAC. La figura 8.1 mostra il caso minimo con due percorsi paralleli fra gli stessi domini.

Lo spanning tree risolve il problema selezionando, tra i collegamenti fisicamente disponibili, un sottoinsieme privo di cicli. In forma concettuale la procedura richiede:

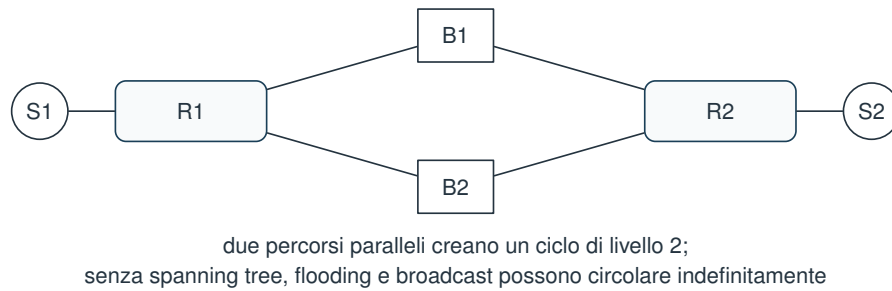
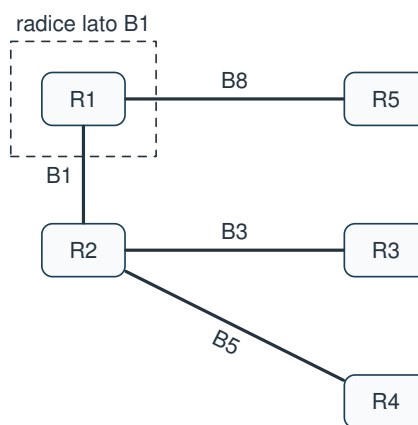


Figura 8.1. Due bridge paralleli creano un ciclo di livello 2 se entrambi i percorsi rimangono simultaneamente attivi.

1. l'elezione di un bridge radice;
2. per ogni switch non radice, la scelta del percorso a costo minimo verso la radice;
3. per ogni segmento, la scelta della porta che deve rappresentare il cammino attivo verso la radice;
4. il blocco logico delle porte ridondanti che chiuderebbero un ciclo.

Gli switch scambiano informazioni di controllo mediante BPDU (Bridge Protocol Data Unit). Se la topologia cambia, il sistema ricalcola l'albero e può attivare un collegamento precedentemente ridondante. Varianti moderne accelerano la convergenza, ma il principio resta quello illustrato in figura 8.2: mantenere la ridondanza fisica senza consentire cicli nel piano di forwarding.



spanning tree: un solo sottoinsieme di bridge resta in forwarding;
i collegamenti ridondanti rimangono disponibili ma bloccati

Figura 8.2. Esempio concettuale di topologia ridondante ridotta a un albero attivo per il forwarding di livello 2.

8.2.2 Switching full-duplex e aggregazione dei link

Con collegamenti punto-punto full-duplex tra host e switch non esiste un mezzo condiviso sul quale più stazioni competono tramite CSMA/CD. Ogni porta costituisce un collegamento indipendente e lo switch può trasferire simultaneamente più trame, entro i limiti della propria capacità interna e delle porte di uscita. Il throughput aggregato non coincide quindi con la velocità di una singola porta.

Più collegamenti fisici tra due apparati possono essere aggregati in un unico collegamento

logico. Per evitare di riordinare i pacchetti di uno stesso flusso, la scelta del membro fisico viene normalmente mantenuta coerente per il flusso, ad esempio mediante una funzione hash sui campi di intestazione. L'aggregazione aumenta capacità e ridondanza, ma non implica che un singolo flusso possa necessariamente usare la somma delle velocità di tutti i membri.

8.2.3 Virtual LAN, VLAN

Una Virtual LAN separa logicamente più domini di livello 2 pur usando la stessa infrastruttura di switch. Le trame appartenenti a VLAN differenti non vengono commutate tra loro come se appartenessero allo stesso dominio di broadcast: per comunicare fra VLAN è necessaria una funzione di livello 3.

Una porta collegata a un endpoint può essere configurata come porta di accesso e associata a una singola VLAN. Un collegamento tra switch può invece trasportare più VLAN, identificandole mediante un tag inserito nella trama. Il tag 802.1Q contiene, tra gli altri campi, un VLAN Identifier di 12 bit; alcuni valori sono riservati, quindi non tutti i 2^{12} valori rappresentano VLAN utilizzabili.

Nel modello di apprendimento indipendente per VLAN, la tabella è indicizzata dalla coppia VLAN-MAC: lo stesso MAC può quindi essere associato a porte diverse in VLAN differenti. Questa è una scelta di modello, non una descrizione obbligatoria della memoria di ogni switch; alcune configurazioni condividono un database di apprendimento fra più VLAN. Restano distinti l'apprendimento degli indirizzi e l'appartenenza delle porte ai domini di inoltro.

8.3 Router

Un router interconnette reti di livello 3 e inoltra i pacchetti in base all'indirizzo di rete di destinazione. La distinzione essenziale rispetto a uno switch di livello 2 non è una differenza di velocità o di tecnologia realizzativa, ma il tipo di informazione su cui viene presa la decisione: lo switch usa principalmente il dominio di collegamento e gli indirizzi MAC, il router separa domini IP e applica una decisione di forwarding a ogni pacchetto.

È utile separare ancora una volta *piano di controllo* e *piano dati*. Il piano di controllo apprende o calcola le rotte, applica le politiche e costruisce una tabella di instradamento. Da essa viene derivata una struttura ottimizzata per il forwarding. Il piano dati, per ogni pacchetto ricevuto, esegue invece operazioni che devono essere molto rapide:

1. verifica la validità minima dell'header di rete;
2. determina la migliore voce di forwarding compatibile con la destinazione;
3. individua interfaccia e next hop;
4. aggiorna i campi che devono cambiare durante l'attraversamento del router, come TTL o Hop Limit;
5. costruisce il nuovo incapsulamento di livello 2 per il collegamento di uscita;
6. accoda il pacchetto sulla relativa interfaccia, dove scheduling e gestione della congestione determinano quando e se sarà trasmesso.

Le implementazioni moderne possono realizzare gran parte di queste operazioni in hardware specializzato; la distinzione hardware/software non coincide quindi con quella fra livello 2 e livello 3.

La decisione ordinaria di forwarding IP usa il *longest prefix match*: fra tutte le rotte il cui prefisso contiene l'indirizzo di destinazione viene scelta quella con il prefisso più lungo, cioè la più specifica. Una rotta di default, con prefisso di lunghezza zero, può essere usata quando nessuna rotta più specifica è disponibile. Politiche aggiuntive possono utilizzare anche altre

informazioni, ma vanno distinte dal comportamento fondamentale basato sulla destinazione.

Un router svolge inoltre funzioni che dipendono dall'architettura e dalla configurazione: filtraggio tramite access control list o firewall, traduzione degli indirizzi, classificazione del traffico, misurazione, tunneling e gestione di più tabelle o domini virtuali. Queste funzioni non definiscono il router in quanto tale e non devono essere confuse con l'algoritmo di forwarding.

La dimensione massima trasportabile sul collegamento di uscita è un altro vincolo importante. Con IPv4 la frammentazione lungo il percorso è possibile se il datagramma la consente; con IPv6 i router non frammentano. In entrambi i casi le architetture moderne cercano di adattare la dimensione trasmessa alla Path MTU e di evitare la frammentazione quando possibile. La gestione dettagliata viene ripresa nelle sezioni dedicate a IPv4 e IPv6.

Infine, il router è un punto naturale di accodamento. Quando la somma dei rate in ingresso diretti alla stessa uscita supera temporaneamente la capacità disponibile, si forma una coda. Dimensione dei buffer, politica di scheduling, marcatura ECN e Active Queue Management influenzano allora perdita, ritardo e ripartizione della capacità. Per questo le funzioni di qualità del servizio vengono trattate separatamente nel seguito.

8.4 Qualità del servizio, scheduling e gestione delle code

Un router non deve soltanto scegliere la porta di uscita: quando più pacchetti attendono sullo stesso collegamento deve anche stabilire l'ordine di servizio e decidere come reagire alla crescita della coda. Queste funzioni collegano direttamente la teoria delle code ai meccanismi di qualità del servizio e controllo della congestione.

8.4.1 Scheduling delle code

La disciplina più semplice è FIFO, nella quale i pacchetti vengono serviti nell'ordine di arrivo. Se esistono classi con requisiti diversi si possono usare code separate. Con priorità stretta il server serve sempre la classe a priorità maggiore quando questa non è vuota; il meccanismo riduce il ritardo della classe privilegiata ma, senza ulteriori vincoli, può affamare le altre classi.

Un riferimento teorico utile è il Generalized Processor Sharing (GPS). Se il collegamento ha capacità C , l'insieme delle classi backlogged nell'istante t è $\mathcal{B}(t)$ e alla classe i è assegnato il peso $w_i > 0$, per $i \in \mathcal{B}(t)$ e $\mathcal{B}(t) \neq \emptyset$ il rate istantaneo ideale è

$$C_i(t) = C \frac{w_i}{\sum_{j \in \mathcal{B}(t)} w_j}. \quad (8.1)$$

La somma dei rate assegnati alle classi attive è infatti

$$\sum_{i \in \mathcal{B}(t)} C_i(t) = C,$$

quindi l'intera capacità viene utilizzata quando almeno una classe ha lavoro. Le classi non backlogged ricevono rate nullo; quando $\mathcal{B}(t) = \emptyset$ tutti i rate sono zero e il server è idle. Algoritmi packet-based come Weighted Fair Queuing approssimano questo comportamento; algoritmi come Deficit Round Robin ottengono una ripartizione pesata con complessità più ridotta.

8.4.2 Regolazione del traffico

Una sorgente può essere vincolata mediante un token bucket con rate r espresso in bit/s e capacità $\sigma \geq 0$ espressa in bit. Indichiamo con $A(t)$ il numero cumulativo di bit ammessi fino

a t ; il vincolo, per ogni intervallo di durata $T \geq 0$, è

$$A(t+T) - A(t) \leq \sigma + rT. \quad (8.2)$$

Dividendo per $T > 0$ e passando al limite superiore per $T \rightarrow \infty$, il rate medio non supera r . La capacità σ permette invece un accumulo iniziale di token spendibile in un burst. Il simbolo r è un rate dimensionato, non il carico adimensionale ρ dei modelli di coda.

Per ricavare un limite di backlog e ritardo assumiamo un server *fluid*, FIFO e work-conserving, inizialmente vuoto, dedicato a questo traffico, con capacità costante $C > r$, senza interruzioni né ritardo aggiuntivo. Sia u l'inizio del periodo di occupazione che contiene t ; se un burst è istantaneo si usa il limite subito prima del suo arrivo. Durante $T = t - u$ il server esegue CT bit e quindi

$$Q(t) = A(t) - A(u) - CT \leq \sigma - (C - r)T \leq \sigma.$$

Se il server è vuoto, la stessa disuguaglianza vale perché $Q(t) = 0$. Il limite uniforme è dunque $Q(t) \leq \sigma$. Il valore σ può essere avvicinato o raggiunto da un burst iniziale nel modello fluido; non si ricava ponendo $T = 0$ nell'incremento $A(t+T) - A(t)$, che a istanti coincidenti è zero.

Per FIFO i bit che arrivano dopo un bit marcato non lo superano. Il lavoro che lo precede è al più σ , quindi

$$D_{max} \leq \frac{\sigma}{C}. \quad (8.3)$$

Questo è un limite del modello dichiarato. Non vale automaticamente per uno scheduler arbitrario, per un server condiviso con traffico non contabilizzato o per un nodo con latenza aggiuntiva. Con pacchetti occorre inoltre specificare quali byte entrino nel conteggio e come venga contabilizzata la serializzazione.

Shaping e policing usano il vincolo con scopi diversi: lo shaper ritarda il traffico eccedente, mentre il policer può marcarlo o scartarlo.

8.4.3 DiffServ

Nell'architettura Differentiated Services i pacchetti sono classificati in un numero limitato di classi e il campo DS dell'header IP trasporta il DSCP usato per selezionare il trattamento per-hop. La rete non mantiene necessariamente uno stato per ogni flusso: il comportamento è aggregato per classe. Lo scheduling e la regolazione del traffico costituiscono quindi i meccanismi con cui un router realizza il trattamento associato alla classe.

8.4.4 Active Queue Management ed ECN

Con una politica tail-drop un pacchetto viene scartato soltanto quando il buffer è pieno. Buffer molto grandi possono evitare perdite per lunghi periodi ma produrre ritardi elevati. Per collegare quantità in bit e ritardi bisogna specificare la popolazione mediata. Consideriamo una coda stazionaria senza abbandoni dopo l'ammissione; Q conta i bit contenuti nei pacchetti ancora in attesa, escludendo quello in servizio. Se λ_p è il tasso dei pacchetti ammessi, F la loro lunghezza e $W_q^{(p)}$ l'attesa di un pacchetto, il bilancio dell'area bit-tempo dà

$$E[Q] = \lambda_p E[FW_q^{(p)}], \quad \lambda_b = \lambda_p E[F].$$

Definiamo il ritardo medio pesato per bit come $E[W_q^{(b)}] = E[FW_q^{(p)}] / E[F]$. Si ottiene

$$E[W_q^{(b)}] = \frac{E[Q]}{\lambda_b}. \quad (8.4)$$

Il rapporto bit/(bit/s) è un tempo, ma non è automaticamente la media per pacchetto: le due coincidono per lunghezze costanti oppure quando lunghezza e attesa sono indipendenti. A parità di rate, un backlog medio maggiore implica quindi un maggiore ritardo medio per bit. Se si scartano pacchetti già accodati, anche la loro permanenza contribuisce all'area e non può essere ignorata dividendo soltanto per il rate dei pacchetti infine trasmessi.

L'Active Queue Management (AQM) genera un segnale di congestione prima che il buffer sia necessariamente pieno, controllando la lunghezza o il tempo di permanenza nella coda. Lo scopo è mantenere il sistema lontano dalla saturazione persistente e rendere il segnale di congestione più tempestivo della semplice perdita per overflow.

Un modello elementare, utile solo per comprendere il principio, è una probabilità di segnalazione lineare in funzione del backlog $q \geq 0$, con $0 \leq q_{min} < q_{max}$ e $0 \leq p_{max} \leq 1$:

$$p(q) = \begin{cases} 0, & q < q_{min}, \\ p_{max} \frac{q - q_{min}}{q_{max} - q_{min}}, & q_{min} \leq q < q_{max}, \\ 1, & q \geq q_{max}. \end{cases} \quad (8.5)$$

Algoritmi reali possono usare grandezze differenti, ad esempio il tempo di permanenza dei pacchetti, e non devono necessariamente seguire la (8.5). La figura 8.3 visualizza il semplice modello lineare usato qui a scopo didattico.

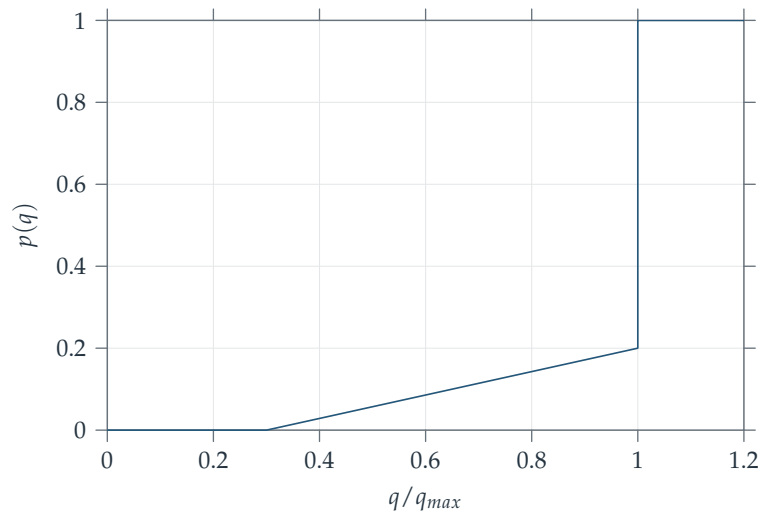


Figura 8.3. Esempio di probabilità di segnalazione AQM lineare con $q_{min} = 0.3q_{max}$ e $p_{max} = 0.2$.

Explicit Congestion Notification (ECN) consente a un router di segnalare la congestione marcando il pacchetto invece di scartarlo, quando gli estremi supportano ECN. Il ricevitore riporta il segnale al trasmettitore, che deve reagire riducendo il carico secondo il proprio algoritmo di controllo della congestione. ECN separa quindi, quando possibile, il segnale di congestione dalla perdita del pacchetto.

8.5 IP versione 4

IPv4 (Internet Protocol version 4) realizza un servizio di rete a datagram e best effort. Ogni datagramma viene inoltrato in modo indipendente: IP non garantisce consegna, ordinamento, assenza di duplicati, banda o ritardo massimo. Queste proprietà, quando necessarie, devono essere costruite ai livelli superiori o mediante meccanismi aggiuntivi della rete.

Il router decide il next hop in base all'indirizzo di destinazione e alla propria tabella di forwarding. Durante l'attraversamento della rete il datagramma può essere scartato per

congestione, errori di header, assenza di una rotta valida o esaurimento del TTL. ICMP permette di segnalare molte di queste condizioni alla sorgente.

8.5.1 Struttura del pacchetto

L'header IPv4 ha una parte base di 20 byte e può contenere opzioni, fino a una lunghezza massima di 60 byte. I campi principali sono:

- **Version, 4 bit:** vale 4.
- **IHL, 4 bit:** lunghezza dell'header in parole da 32 bit; il valore minimo è 5.
- **DS field, 8 bit:** contiene DSCP per la classificazione del traffico ed ECN per la segnalazione esplicita della congestione.
- **Total Length, 16 bit:** lunghezza complessiva di header e payload; il massimo rappresentabile è 65535 byte.
- **Identification, Flags, Fragment Offset:** campi usati per la frammentazione e il riasssemblaggio. Il bit DF vieta la frammentazione; MF indica che seguiranno altri frammenti; il Fragment Offset è espresso in unità di 8 byte.
- **TTL, 8 bit:** viene decrementato a ogni router; se arriva a zero il pacchetto viene scartato. Impedisce a un datagramma coinvolto in un loop di circolare indefinitamente.
- **Protocol, 8 bit:** identifica il protocollo cui il payload deve essere consegnato, ad esempio TCP, UDP o ICMP.
- **Header Checksum, 16 bit:** protegge il solo header IPv4. Poiché il TTL cambia a ogni hop, il valore deve essere aggiornato dai router che inoltrano il pacchetto.
- **Source Address e Destination Address, 32 bit ciascuno.**
- **Options e Padding:** campi opzionali; il padding porta l'header a un multiplo di 32 bit.

Se un datagramma supera la MTU del collegamento di uscita e DF non è posto, un router IPv4 può frammentarlo. Il riasssemblaggio avviene soltanto alla destinazione. Tutti i frammenti, salvo eventualmente l'ultimo, trasportano una quantità di dati multipla di 8 byte, così che l'offset possa essere rappresentato nelle unità previste dal campo. Nelle reti moderne è preferibile evitare la frammentazione lungo il percorso e scegliere alla sorgente una dimensione compatibile con la Path MTU.

8.5.2 Indirizzi IPv4

Un indirizzo IPv4 contiene 32 bit e viene normalmente scritto in notazione decimale puntata. L'instradamento usa prefissi CIDR: una scrittura come 192.0.2.0/24 indica che i primi 24 bit sono il prefisso e i restanti 8 distinguono gli indirizzi all'interno di quel prefisso. La maschera 255.255.255.0 è una rappresentazione equivalente della lunghezza /24.

Dato un prefisso di lunghezza p , il numero di combinazioni per la parte non prefissata è

$$N_{addr} = 2^{32-p}. \quad (8.6)$$

La relazione conta le combinazioni binarie del blocco; quali indirizzi siano effettivamente assegnabili dipende poi dal tipo di prefisso e dalle regole applicabili a quel dominio.

Tra gli spazi di uso comune vi sono i tre blocchi privati

$$10.0.0.0/8, \quad 172.16.0.0/12, \quad 192.168.0.0/16,$$

che non vengono instradati globalmente su Internet. Il blocco 127.0.0.0/8 è riservato al loop-back. Negli esempi del libro si usano, quando possibile, prefissi destinati alla documentazione, come 192.0.2.0/24, 198.51.100.0/24 e 203.0.113.0/24.

Per un normale prefisso unicast, l'indirizzo con la parte host tutta a zero rappresenta il prefisso stesso. L'indirizzo con la parte host tutta a uno può rappresentare il directed broadcast del prefisso. Questa nozione va distinta dal broadcast limitato 255.255.255.255, valido sul collegamento locale. Non si deve applicare indiscriminatamente la sottrazione di due indirizzi a ogni prefisso: su un collegamento punto-punto /31 i due valori identificano i due estremi, mentre /32 identifica un singolo indirizzo.

8.5.3 Forwarding locale, next hop e ARP

Sia D l'indirizzo IPv4 di destinazione. Un host o un router confronta D con le rotte disponibili e sceglie la voce compatibile con il prefisso più lungo. Non conta quindi l'ordine con cui le righe sono memorizzate: conta la specificità del prefisso. Una rotta 0.0.0.0/0 costituisce la default route e viene scelta solo se nessuna voce più specifica è applicabile.

Una tabella semplificata per un host nella rete 192.0.2.0/24 può essere rappresentata così:

Prefisso	Next hop	Uscita
192.0.2.0/24	diretto	interfaccia locale
198.51.100.128/26	192.0.2.194	interfaccia locale
0.0.0.0/0	192.0.2.1	interfaccia locale

Tabella 8.1. Esempio di tabella IPv4 semplificata.

Se la rotta scelta è direttamente connessa, il next hop di livello 3 è la destinazione stessa. Se invece la rotta indica un router, il next hop è l'indirizzo di quel router sul collegamento locale. In entrambi i casi, prima di trasmettere una trama Ethernet occorre conoscere il MAC associato all'indirizzo IPv4 del next hop.

ARP risolve proprio questa associazione sul collegamento locale. In forma essenziale:

1. il nodo consulta la propria cache ARP;
2. se l'associazione non è disponibile, trasmette una ARP Request in broadcast chiedendo quale interfaccia possieda l'indirizzo IPv4 del next hop;
3. il nodo che possiede quell'indirizzo risponde comunicando il proprio MAC, normalmente con una risposta unicast;
4. l'associazione viene memorizzata temporaneamente e usata per costruire l'header Ethernet della trama.

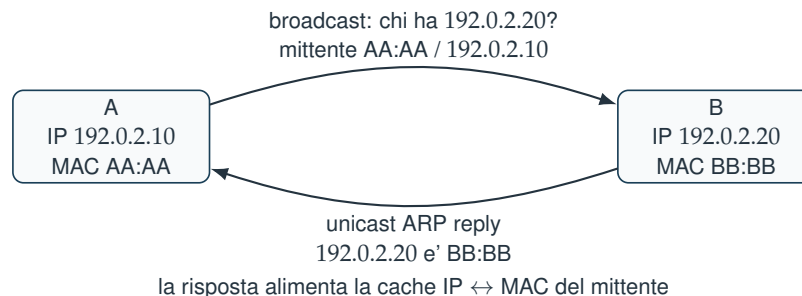


Figura 8.4. Risoluzione dell'indirizzo MAC del next hop mediante ARP.

ARP risolve indirizzi soltanto sul dominio di livello 2 direttamente raggiungibile. Un host non cerca quindi il MAC della destinazione remota: cerca il MAC del router scelto come

next hop. Le associazioni hanno una durata finita perché configurazione, mobilità o failover possono modificare la corrispondenza fra indirizzi di livello 3 e 2.

8.5.4 Internet Control Message Protocol, ICMP

ICMP trasporta messaggi di controllo associati al funzionamento di IPv4. Non rende IP affidabile e un messaggio ICMP può a sua volta essere perso; fornisce però informazioni essenziali per diagnosi e adattamento del comportamento degli endpoint.

Tra i messaggi concettualmente più importanti vi sono:

- **Destination Unreachable**, quando il datagramma non può essere consegnato per una delle cause previste;
- **Time Exceeded**, quando il TTL si esaurisce; questo meccanismo rende possibile anche l'osservazione progressiva dei router attraversati da strumenti di tipo traceroute;
- **Parameter Problem**, per errori in campi dell'header;
- **Echo Request/Echo Reply**, usati per verificare raggiungibilità e misurare un round-trip time;
- **Redirect**, con cui un router può indicare a un host un next hop migliore sullo stesso collegamento locale, quando la politica di rete ne consente l'uso.

ICMP partecipa anche alla Path MTU Discovery. Se un router deve inoltrare un datagramma IPv4 più grande della MTU di uscita e il bit DF vieta la frammentazione, il pacchetto viene scartato e la sorgente può essere informata della limitazione. Riducendo la dimensione dei pacchetti, la sorgente converge verso un valore compatibile con il percorso senza richiedere frammentazione intermedia, purché il feedback necessario arrivi. La perdita o il filtraggio dei messaggi ICMP può invece produrre un black hole per i pacchetti grandi. La Packetization Layer Path MTU Discovery usa sonde e riscontri del livello superiore per verificare le dimensioni effettivamente trasportabili, senza dipendere esclusivamente dai messaggi ICMP.

8.5.5 Multicast

L'unicast consegna una copia distinta a ogni destinatario; il broadcast raggiunge tutti i nodi del dominio previsto; il multicast identifica invece un gruppo di ricevitori. La rete può replicare il pacchetto nei punti in cui i percorsi verso i membri del gruppo divergono, evitando che la sorgente debba inviare necessariamente una copia separata per ogni ricevitore.

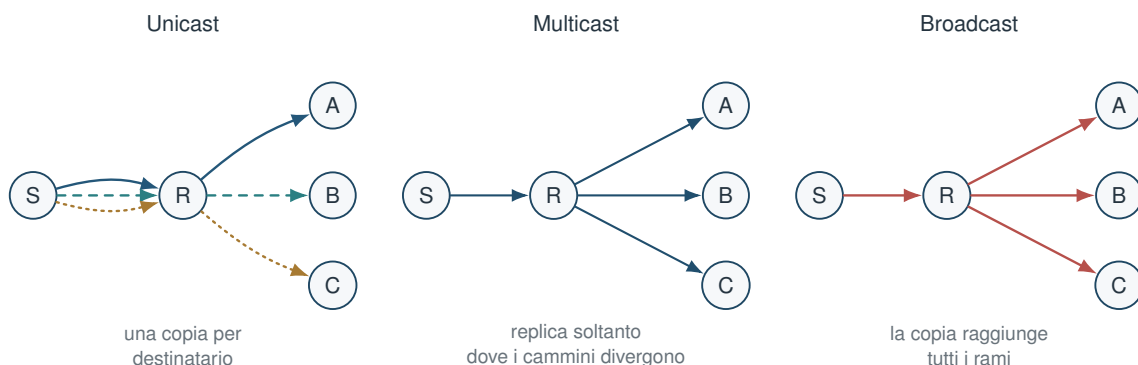


Figura 8.5. Confronto concettuale tra consegna unicast, multicast e broadcast.

Gli indirizzi multicast IPv4 appartengono al prefisso 224.0.0.0/4. Un indirizzo identifica un gruppo, non un particolare host. Sul collegamento locale, IGMP permette agli host di segnalare ai router l'interesse per i gruppi multicast; il router deve conoscere se sul link

esistono ricevitori interessati, non necessariamente il loro numero esatto.

Su Ethernet un indirizzo multicast IPv4 viene mappato nello spazio MAC che inizia con 01:00:5e. I 23 bit meno significativi dell'indirizzo IPv4 multicast vengono riportati nel MAC; poiché il prefisso multicast IPv4 lascia 28 bit variabili, $2^{28-23} = 32$ indirizzi IPv4 distinti possono produrre lo stesso MAC multicast. Il filtro di livello 2 non è quindi sufficiente, da solo, a distinguere tutti i gruppi: il livello IP deve verificare l'indirizzo multicast completo.

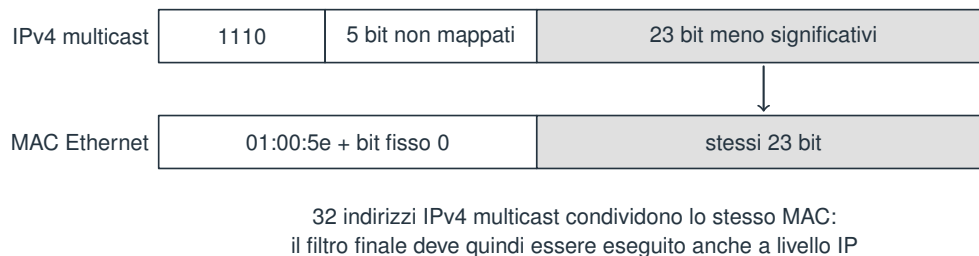


Figura 8.6. Mappatura di un indirizzo multicast IPv4 nel corrispondente spazio MAC Ethernet.

La costruzione dell'albero multicast fra router è un problema distinto dall'appartenenza locale al gruppo. Più avanti verranno descritti il controllo RPF e il principio di PIM, che utilizzano le informazioni di routing unicast per costruire o verificare i rami dell'albero.

8.5.6 Dynamic Host Configuration Protocol, DHCP

DHCP permette a un host IPv4 di ottenere automaticamente parametri di configurazione come indirizzo, prefisso, router di default, server DNS e durata della concessione. L'assegnazione è normalmente una *lease*: il client può usare i parametri per un intervallo finito e deve rinnovarli se intende conservarli.

Una sequenza iniziale tipica è riassunta da quattro messaggi:

1. **Discover:** il client, ancora privo di configurazione, cerca server disponibili;
2. **Offer:** uno o più server propongono una configurazione;
3. **Request:** il client indica l'offerta che intende usare;
4. **Acknowledgement:** il server conferma la lease e i relativi parametri.

La sequenza descrive la specifica del protocollo, non un risultato da dimostrare.

Poiché il client iniziale non sa ancora necessariamente comunicare in unicast sulla rete IP, parte della procedura usa broadcast sul link. Un DHCP relay può inoltrare le richieste a un server posto su un'altra rete, evitando di richiedere un server in ogni dominio di broadcast. Un amministratore può inoltre associare stabilmente una particolare configurazione a un client oppure allocare dinamicamente gli indirizzi da un pool.

8.6 IP versione 6

IPv6 mantiene il modello datagram best effort di IP ma amplia lo spazio di indirizzamento a 128 bit e semplifica l'header elaborato a ogni hop. L'header base ha lunghezza fissa di 40 byte; le funzioni opzionali sono trasportate mediante extension header concatenati. IPv4 e IPv6 sono due protocolli di rete distinti e possono coesistere nella stessa infrastruttura mediante apparati e applicazioni dual stack o altri meccanismi di transizione.

IPv6 non introduce, da solo, garanzie di consegna, banda o ritardo. I campi Traffic Class e Flow Label possono aiutare classificazione e trattamento coerente dei flussi, ma non equivalgono a una prenotazione di risorse.

8.6.1 Struttura del pacchetto

L'header base contiene:

- **Version, 4 bit:** vale 6.
- **Traffic Class, 8 bit:** trasporta informazioni di classificazione e congestione, con uso coerente con DSCP ed ECN.
- **Flow Label, 20 bit:** permette alla sorgente di associare pacchetti allo stesso flusso affinché la rete possa trattarli in modo coerente senza dover dipendere necessariamente dai livelli superiori.
- **Payload Length, 16 bit:** lunghezza, in byte, di ciò che segue l'header base. Funzioni specifiche consentono payload di lunghezza superiore allo spazio ordinario del campo, ma non sono il caso normale.
- **Next Header, 8 bit:** identifica l'estension header successivo oppure, se la catena termina, il protocollo di livello superiore.
- **Hop Limit, 8 bit:** viene decrementato da ogni router e svolge il ruolo del TTL di IPv4.
- **Source Address e Destination Address, 128 bit ciascuno.**

L'header base non contiene una checksum. L'integrità dei dati viene controllata dai livelli superiori o dai meccanismi di protezione applicabili; eliminare una checksum da ricalcolare a ogni hop riduce il lavoro obbligatorio del router.

8.6.2 Indirizzi IPv6

Un indirizzo IPv6 è rappresentato come otto gruppi esadecimali da 16 bit separati da ":". Gli zeri iniziali di ogni gruppo possono essere omessi e una sola sequenza contigua di gruppi nulli può essere compressa con "::". Per esempio

$$2001:db8:49:0:0:0:0:1 = 2001:db8:49::1.$$

Il prefisso 2001:db8::/32 è riservato agli esempi e alla documentazione.

Anche IPv6 usa prefissi CIDR. Un prefisso di lunghezza p contiene

$$N_{addr} = 2^{128-p} \quad (8.7)$$

combinazioni nella parte restante. Uno stesso host possiede normalmente più indirizzi contemporaneamente, con scopi e ambiti differenti.

Tra le categorie principali vi sono:

- **Global Unicast:** indirizzi instradabili su scala globale; una parte ampia delle assegnazioni ricade in 2000::/3;
- **Link Local:** fe80::/10, validi sul singolo link e fondamentali per Neighbor Discovery;
- **Unique Local:** fc00::/7, destinati a uso locale e non normalmente instradati globalmente;
- **Multicast:** ff00::/8; IPv6 non usa broadcast;
- **Loopback:** ::1; l'indirizzo tutto zero :: rappresenta l'indirizzo non specificato in contesti in cui la specifica ne consente l'uso.

La configurazione può essere statica oppure dinamica. Con Stateless Address Autoconfiguration (SLAAC) un host apprende dai Router Advertisement le informazioni di prefisso e costruisce gli indirizzi secondo le regole applicabili. L'Interface Identifier non deve essere necessariamente derivato dal MAC: possono essere usati identificatori stabili non legati direttamente all'hardware e indirizzi temporanei orientati alla privacy. DHCPv6 può fornire

configurazione stateful o altre informazioni e può coesistere con SLAAC. La scelta automatica dei router di default è invece una funzione di Router Discovery, tramite i Router Advertisement: ottenere un indirizzo da DHCPv6 non sostituisce questa funzione.

8.6.3 Extension header e frammentazione

Il campo Next Header concatena zero o più extension header prima del protocollo superiore. Fra quelli concettualmente più importanti vi sono Hop-by-Hop Options, Routing, Fragment, Destination Options, Authentication Header ed Encapsulating Security Payload. Non tutti i router devono elaborare tutti gli extension header: la semantica di ciascuno stabilisce quali nodi devono interpretarlo.

IPv6 non permette ai router di frammentare un pacchetto in transito. Se la dimensione supera la MTU del collegamento successivo, il router scarta il pacchetto e invia alla sorgente un messaggio ICMPv6 *Packet Too Big* contenente la MTU ammissibile. La sorgente può quindi ridurre la dimensione trasmessa. Se è necessaria frammentazione IPv6, essa viene eseguita dalla sorgente mediante il Fragment Header e il riassemblaggio avviene alla destinazione.

Questa separazione mantiene semplice il forwarding: la decisione di adattare la dimensione non richiede al router di creare nuovi frammenti. Il comportamento è quindi diverso da IPv4, dove la frammentazione intermedia resta possibile quando DF non la vieta.

8.6.4 ICMPv6 e Neighbor Discovery

ICMPv6 svolge sia funzioni di segnalazione degli errori sia funzioni necessarie al funzionamento locale di IPv6. Tra i messaggi di errore rientrano Destination Unreachable, Packet Too Big, Time Exceeded e Parameter Problem; Echo Request ed Echo Reply forniscono la consueta funzione di test di raggiungibilità.

Neighbor Discovery usa messaggi ICMPv6 per sostituire e ampliare varie funzioni che in IPv4 sono distribuite tra ARP e altri meccanismi:

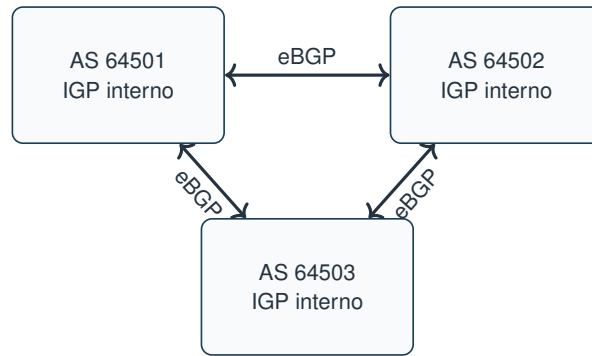
- **Router Solicitation** e **Router Advertisement** per scoprire router e parametri del link;
- **Neighbor Solicitation** e **Neighbor Advertisement** per risoluzione degli indirizzi di livello 2, verifica di raggiungibilità e Duplicate Address Detection;
- **Redirect** per suggerire, quando ammesso, un next hop migliore sullo stesso link.

La risoluzione di un indirizzo IPv6 nel corrispondente indirizzo di livello 2 usa multicast, non broadcast. Le associazioni vengono mantenute in una neighbor cache il cui stato riflette anche la raggiungibilità recente del vicino. Un Neighbor Advertisement può essere una risposta a una richiesta oppure, in casi previsti, essere inviato senza una richiesta precedente; non va interpretato come un annuncio periodico generale di tutti gli host.

8.7 Protocolli di instradamento

Le rotte possono essere configurate staticamente oppure apprese da un protocollo dinamico. Una configurazione statica è semplice e prevedibile, ma non reagisce da sola a un guasto o a una modifica della topologia. Un protocollo dinamico scambia invece informazioni di raggiungibilità e aggiorna lo stato da cui vengono ricavate le rotte. La decisione finale installata nel piano dati resta distinta dal protocollo che l'ha prodotta.

Internet è suddivisa in sistemi autonomi (AS), cioè domini sotto una politica amministrativa comune. All'interno di un AS si usano protocolli IGP; fra AS il protocollo dominante è BGP, nel quale la scelta della rotta riflette anche politiche e relazioni amministrative, non soltanto una distanza numerica.



l'IGP costruisce i cammini interni; BGP scambia raggiungibilità e policy tra AS

Figura 8.7. Sistemi autonomi con instradamento interno e scambio di raggiungibilità fra domini.

8.7.1 Routing Information Protocol, RIP

RIP è un esempio diretto di protocollo distance-vector e rende concreto l'algoritmo discusso nel capitolo 5. Ogni router annuncia ai vicini le destinazioni conosciute e la relativa metrica; ricevendo un annuncio dal vicino v , un router può aggiornare la distanza verso d secondo la forma Bellman–Ford

$$D_x(d) = \min_v \{c(x, v) + D_v(d)\}.$$

Nel caso RIP la metrica fondamentale è il numero di hop. Il valore massimo utile è deliberatamente piccolo, così da limitare il fenomeno di count-to-infinity; ciò rende RIP adatto soprattutto a domini semplici e ne fa un buon esempio didattico, non un modello generale di routing su larga scala.

8.7.2 Open Shortest Path First, OSPF

OSPF è un protocollo link-state. I router descrivono lo stato dei propri collegamenti e diffondono queste informazioni nel dominio di pertinenza; ciascun router costruisce così un database topologico coerente e calcola localmente l'albero dei cammini minimi. Il calcolo è quello di Dijkstra già derivato nel capitolo 5.

A ogni collegamento è associato un costo positivo configurato. Il costo di un percorso è la somma dei costi attraversati e più cammini a costo uguale possono essere installati simultaneamente. L'uso di aree consente di introdurre gerarchia e limitare la quantità di stato che deve essere distribuita e ricalcolata all'interno di un grande dominio.

8.7.3 Border Gateway Protocol, BGP

BGP scambia raggiungibilità fra sistemi autonomi e viene classificato come protocollo path-vector. Un annuncio associa a un prefisso un insieme di attributi; fra questi, AS_PATH elenca i sistemi autonomi attraversati dall'annuncio e consente anche di riconoscere un loop quando il proprio AS compare già nel percorso ricevuto.

La selezione non equivale semplicemente a minimizzare il numero di AS. Le politiche possono preferire o rifiutare rotte in base ad attributi, relazioni amministrative e criteri locali. BGP usa una sessione di trasporto affidabile fra peer e propaga variazioni incrementali di raggiungibilità. Il risultato del processo di selezione alimenta poi la tabella di forwarding, eventualmente insieme a più next hop compatibili con ECMP.

8.7.4 Reverse Path Forwarding, RPF

RPF non è, da solo, un protocollo multicast completo: è un criterio di verifica che sfrutta la conoscenza del routing unicast. Per una sorgente S , un router determina quale interfaccia userebbe per raggiungere S . Un pacchetto multicast proveniente da S supera il controllo RPF soltanto se arriva dall'interfaccia coerente con quel cammino inverso; in caso contrario viene scartato.

Questa è la verifica per un albero orientato alla sorgente. Su un albero condiviso PIM radicato in un Rendezvous Point, il vicino upstream viene invece determinato verso il RP; i dettagli dipendono dallo stato multicast installato. In una topologia di routing convergente e priva di cicli, il criterio evita che copie dello stesso pacchetto vengano accettate indefinitamente attraverso rami diversi e fornisce una base naturale per costruire alberi orientati alla sorgente. Non stabilisce però, da solo, su quali interfacce esistano ricevitori interessati: questa informazione deve provenire dai meccanismi di appartenenza locale e dal protocollo multicast.

8.7.5 Protocol Independent Multicast, PIM

PIM usa le informazioni di instradamento unicast per i controlli RPF ma non richiede uno specifico IGP; da qui il termine *protocol independent*. Il caso più importante concettualmente è il funzionamento sparse-mode, nel quale non si presume che quasi ogni ramo della rete contenga ricevitori.

Per un gruppo G può essere costruito inizialmente un albero condiviso radicato in un Rendezvous Point (RP). I router che hanno ricevitori locali per G inviano messaggi di join verso il RP, creando stato $(*, G)$ lungo il percorso. Quando compare una sorgente S , il router adiacente alla sorgente rende il traffico disponibile al RP; da qui il flusso può seguire l'albero condiviso verso i ricevitori.

Se conviene evitare il passaggio permanente attraverso il RP, i router possono costruire un ramo specifico per la sorgente, con stato (S, G) , e passare a un percorso più diretto verificato tramite RPF. La distinzione fondamentale è quindi fra *scoperta/appartenenza al gruppo*, gestita sul link locale, e *costruzione dell'albero fra router*, gestita da PIM e dalle informazioni di routing sottostanti.

8.8 MPLS e Segment Routing

Multiprotocol Label Switching introduce un piano di forwarding basato su label corte anziché sulla ricerca completa dell'indirizzo IP a ogni hop.

All'ingresso di un dominio MPLS i pacchetti vengono associati a una Forwarding Equivalence Class (FEC). Pacchetti appartenenti alla stessa FEC ricevono lo stesso trattamento di forwarding. Una label MPLS ha significato locale al tratto di rete in cui viene utilizzata; un Label Switching Router può sostituire la label presente con quella prevista per il collegamento successivo.

Le operazioni fondamentali sono:

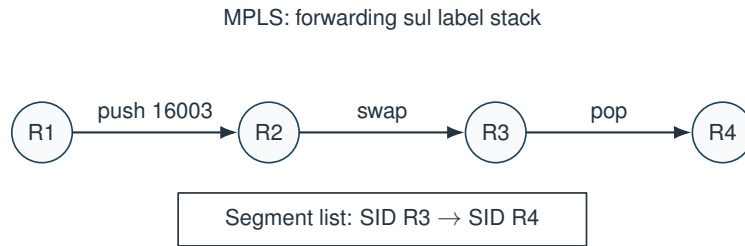
- *push*, inserimento di una label nello stack;
- *swap*, sostituzione della label in testa;
- *pop*, rimozione della label in testa.

Indicando con

$$\mathcal{L} = [L_1, L_2, \dots, L_m]$$

lo stack, il forwarding ordinario opera su L_1 ; una pop trasforma lo stack in $[L_2, \dots, L_m]$. Lo stack permette quindi di comporre più livelli di instradamento o tunneling mantenendo

semplice il forwarding sui nodi intermedi.



Segment Routing: il nodo d'ingresso codifica nel pacchetto la sequenza di segmenti da seguire

Figura 8.8. Forwarding MPLS e principio di una lista di segmenti in Segment Routing.

8.8.1 Segment Routing

Segment Routing generalizza l'idea di guidare il pacchetto attraverso una sequenza ordinata di istruzioni dette segmenti. Una SR Policy può essere rappresentata come

$$\mathcal{S} = [S_1, S_2, \dots, S_m],$$

dove ogni S_i identifica un'istruzione topologica o di servizio. Il nodo di ingresso sceglie la sequenza; i nodi interni non devono mantenere necessariamente uno stato per ogni flusso.

Nel data plane MPLS un segmento è rappresentato da una label e la sequenza da uno stack di label. In SRv6 un segmento è identificato da un SID IPv6 e la sequenza può essere trasportata mediante il Segment Routing Header.

La differenza concettuale rispetto al solo shortest-path routing è importante: la destinazione specifica dove deve arrivare il pacchetto, mentre una lista di segmenti può specificare anche come deve attraversare il dominio. Questo rende naturale l'uso di Segment Routing per traffic engineering, protezione e concatenazione di funzioni di rete.

8.9 Reti di data center: Clos, ECMP e BGP

Le reti di data center mostrano come strutture nate nella teoria della commutazione riappaiano nelle reti a pacchetto. Una topologia leaf-spine è una realizzazione a due livelli riconducibile a una rete di Clos: ogni switch leaf è connesso a tutti gli switch spine e i server sono collegati ai leaf. Il capitolo 7 introduce le proprietà generali delle reti di Clos.

Supponiamo che vi siano n spine e che ogni leaf sia collegato a tutti gli spine. Tra due leaf distinti esistono allora esattamente n cammini minimi di uguale lunghezza, uno per ogni spine. Se ogni collegamento ha lo stesso costo,

$$N_{ECMP} = n. \quad (8.8)$$

Equal Cost Multipath (ECMP) permette di installare simultaneamente questi next-hop e ripartire i flussi tra essi. Una realizzazione comune usa un hash dei campi che identificano il flusso:

$$j = h(\text{flow}) \bmod n, \quad (8.9)$$

così che i pacchetti dello stesso flusso tendano a seguire lo stesso cammino, evitando riordinamenti introdotti dalla sola ripartizione per-packet.

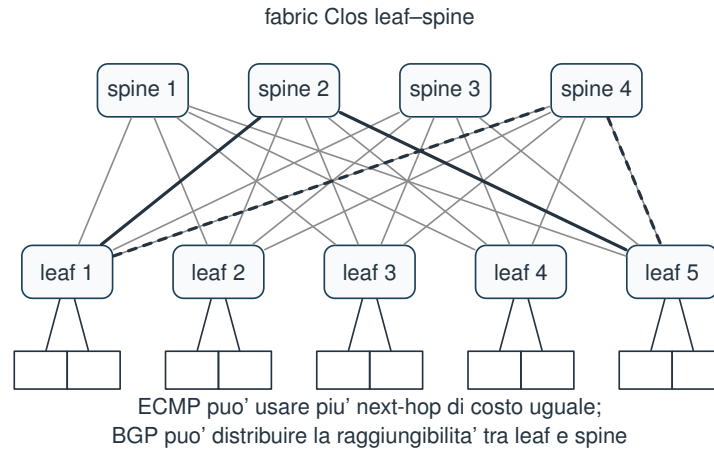


Figura 8.9. Fabric leaf-spine di tipo Clos con cammini ECMP alternativi.

8.9.1 Oversubscription

Sia un leaf con k porte verso server, ciascuna di capacità C_d , e n uplink verso gli spine, ciascuno di capacità C_u . La capacità aggregata lato server è kC_d , mentre quella verso la fabric è nC_u . Definiamo il rapporto di oversubscription

$$O = \frac{kC_d}{nC_u}. \quad (8.10)$$

Con $O \leq 1$ la capacità aggregata di uplink è almeno pari a quella dei downlink; con $O > 1$ la fabric è dimensionata assumendo che non tutti i server trasmettano contemporaneamente al massimo rate verso altri leaf. Il parametro O collega quindi direttamente topologia e modello di traffico. Non costituisce però una prova di non bloccaggio: con $O \leq 1$ due flussi possono essere assegnati dallo stesso hash allo stesso uplink e saturarlo mentre un altro uplink resta libero. La piena capacità aggregata è una condizione di dimensionamento; la distribuzione effettiva dipende dai flussi e dall'instradamento. La garanzia combinatoria di Clos del capitolo 7 riguarda un modello di scelta dei cammini diverso da una ripartizione ECMP statistica già fissata per flusso.

8.9.2 BGP nella fabric

BGP può essere utilizzato anche all'interno di data center Clos di grandi dimensioni, spesso con sessioni eBGP tra leaf e spine. In questa architettura ECMP è essenziale per sfruttare in parallelo i numerosi cammini equivalenti offerti dalla fabric.

Il punto concettuale non è che BGP sia l'unica scelta possibile, ma che una topologia altamente regolare e multipath consente di separare bene due problemi: il protocollo distribuisce la raggiungibilità e i next-hop equivalenti, mentre ECMP realizza nel data plane la ripartizione del traffico. La ridondanza della topologia permette inoltre di continuare a utilizzare la fabric quando un singolo collegamento o uno spine non è disponibile, riducendo il numero di cammini ma non necessariamente interrompendo la raggiungibilità.

8.10 Demultiplazione e porte di trasporto

La consegna di un pacchetto a un'applicazione richiede più decisioni di demultiplazione, una per ciascun livello interessato. Queste decisioni usano identificatori diversi e non vanno confusi fra loro. In una comunicazione Ethernet/IP tipica:

1. l'interfaccia di rete accetta la trama in base alla destinazione MAC e il campo di tipo identifica il protocollo di livello superiore;
2. IP verifica che la destinazione sia locale e usa Protocol in IPv4 o la catena Next Header in IPv6 per individuare il protocollo di trasporto;
3. TCP o UDP usano le porte e lo stato locale per consegnare i dati al corretto endpoint di trasporto.

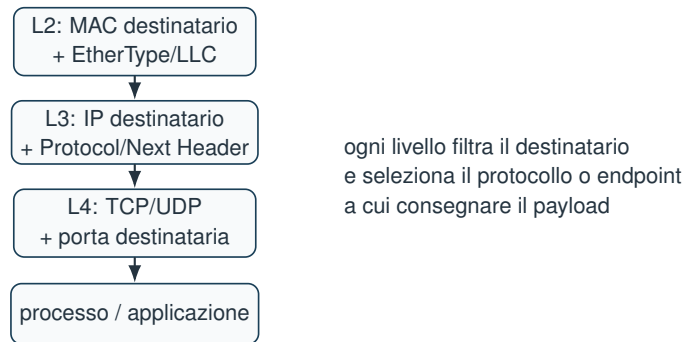


Figura 8.10. Demultiplazione progressiva dal collegamento al trasporto.

TCP e UDP possiedono porte sorgente e destinazione di 16 bit. Una porta non identifica in modo assoluto “un’applicazione”: è un numero nello spazio del particolare protocollo di trasporto e viene associato dal sistema operativo a uno o più endpoint locali secondo le regole della sua API. La stessa porta numerica può quindi esistere contemporaneamente in TCP e UDP e più connessioni TCP possono condividere la stessa porta locale di ascolto.

Una connessione TCP viene normalmente identificata dalla quadrupla

(IP sorgente, porta sorgente, IP destinazione, porta destinazione),

insieme al protocollo di trasporto. Questa informazione permette a un server di mantenere molte connessioni simultanee sulla stessa porta locale, purché le quadruple siano differenti. Per UDP, che non crea una connessione TCP-like, il sistema operativo usa comunque indirizzi, porte e binding locali per decidere a quale socket consegnare ogni datagramma.

Alcuni numeri di porta sono convenzionalmente associati a servizi noti, ad esempio 53 per DNS, 80 per HTTP e 443 per HTTPS; questa associazione facilita l’individuazione iniziale del servizio ma non impone che un programma possa usare soltanto quella porta. I client usano normalmente porte locali scelte dinamicamente e possono aprire più comunicazioni contemporanee.

8.10.1 Network Address Translation, NAT

Gli indirizzi IPv4 privati non sono instradati sulla Internet globale. Un dispositivo di confine può tradurre gli indirizzi dei pacchetti che attraversano tale confine. Con una traduzione statica una relazione preconfigurata associa indirizzi interni ed esterni; con NAT/PAT più host possono condividere uno o pochi indirizzi pubblici perché la traduzione usa anche gli identificatori di trasporto.

Per una nuova comunicazione TCP o UDP in uscita, una rappresentazione semplificata dello stato è

$$(\text{IP interno}, p_i, \text{protocollo}) \longleftrightarrow (\text{IP pubblico}, p_e, \text{protocollo}),$$

dove p_i e p_e sono rispettivamente porta interna ed esterna. Le implementazioni reali possono includere anche indirizzo e porta del peer remoto nella chiave di classificazione. Il pacchetto in

uscita viene riscritto coerentemente con la mappatura; il traffico di ritorno viene riconosciuto mediante lo stato e tradotto nella direzione opposta.

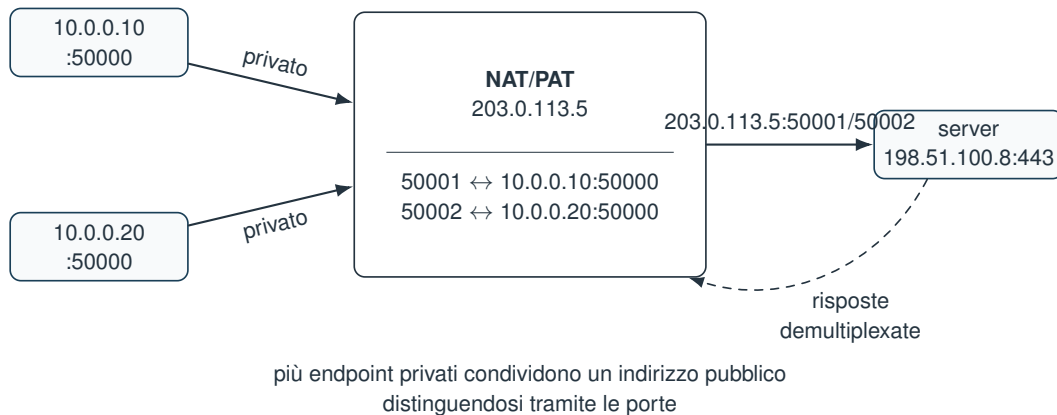


Figura 8.11. Esempio di più flussi interni distinti mediante traduzione di indirizzi e porte sul confine IPv4.

Lo stato dinamico ha una durata finita. Per UDP l'apparato non può osservare un'apertura o una chiusura di connessione e usa quindi principalmente timer di inattività; per TCP può anche seguire, entro i limiti dell'osservazione disponibile, l'evoluzione della connessione. La capacità di traduzione non è semplicemente "65536 connessioni per indirizzo": dipende da porte utilizzabili, protocolli, criteri di mappatura, peer remoti, timeout e numero di indirizzi pubblici.

Una comunicazione iniziata dall'esterno non deve necessariamente fallire. Può essere resa possibile da una mappatura statica, da un port forwarding o da altro stato predisposto esplicitamente. Ciò mostra anche perché NAT e firewall siano concetti distinti: il NAT modifica identificatori e mantiene associazioni; il firewall decide quale traffico è ammesso secondo una politica. Un singolo apparato può realizzare entrambe le funzioni, ma l'una non implica l'altra.

8.11 Transmission Control Protocol, TCP

TCP fornisce un servizio di trasporto affidabile, bidirezionale e orientato alla connessione sopra un servizio IP che, da solo, è best effort. L'unità logica presentata all'applicazione non è una sequenza di messaggi ma un *flusso ordinato di byte*: i confini fra le scritture eseguite dall'applicazione non vengono preservati necessariamente nel modo in cui i dati sono segmentati sulla rete.

L'affidabilità nasce dalla combinazione di numeri di sequenza, acknowledgement, checksum, ritrasmissione e controllo della finestra. TCP realizza inoltre due controlli distinti:

- il **controllo di flusso**, con cui il ricevitore limita i dati che il trasmettitore può mantenere in volo in funzione del proprio buffer;
- il **controllo di congestione**, con cui il trasmettitore adatta il carico allo stato stimato della rete.

Una connessione è identificata dagli indirizzi IP e dalle porte dei due endpoint insieme al protocollo TCP; più connessioni possono quindi terminare sulla stessa porta di ascolto di un server.

8.11.1 Struttura del segmento

L'header TCP base occupa 20 byte e può essere esteso mediante opzioni. I campi principali sono:

- **Source Port** e **Destination Port**, 16 bit ciascuno;
- **Sequence Number**, 32 bit: identifica, nel caso ordinario, il numero di sequenza del primo byte di dati contenuto nel segmento;
- **Acknowledgement Number**, 32 bit: quando il flag ACK è valido, indica il prossimo numero di sequenza atteso e realizza quindi un acknowledgement cumulativo;
- **Data Offset**: specifica la lunghezza dell'header TCP in parole da 32 bit;
- **Control Flags**: comprendono, fra gli altri, SYN e FIN per la gestione della connessione, RST per il reset, ACK per la validità dell'acknowledgement, PSH e URG per particolari semantiche del flusso, e i bit usati con ECN;
- **Window**, 16 bit: pubblicizza la quantità di spazio che il ricevitore rende disponibile a partire dal byte indicato dall'acknowledgement; opportune opzioni consentono di rappresentare finestre effettive più grandi del valore base a 16 bit;
- **Checksum**, 16 bit: protegge header e dati TCP e include nel calcolo uno pseudo-header derivato da IP, così da legare il segmento anche agli endpoint e al protocollo di trasporto;
- **Urgent Pointer**: usato quando è attiva la relativa semantica di dati urgenti;
- **Options**: permettono, tra l'altro, di negoziare MSS, window scaling, timestamp e Selective Acknowledgement;
- **Data**: porzione del flusso di byte trasportata dal segmento.

Questi campi sono parte della specifica del protocollo; non sono risultati matematici da dimostrare.

La Maximum Segment Size (MSS) esprime la massima quantità di dati TCP che un endpoint dichiara di poter ricevere in un singolo segmento. La dimensione effettivamente trasmessa dipende anche dalla Path MTU, dalle opzioni presenti, dalle finestre di flusso e congestione e dalle scelte dell'implementazione. TCP non attende necessariamente di raggiungere la MSS prima di trasmettere.

8.11.2 Instaurazione e chiusura della connessione

L'instaurazione ordinaria usa un three-way handshake. Indicando con x e y gli Initial Sequence Number scelti dai due endpoint:

1. il client invia SYN con numero di sequenza x ;
2. il server risponde con SYN+ACK, numero di sequenza y e acknowledgement $x + 1$;
3. il client conferma con ACK pari a $y + 1$.

Il SYN occupa una posizione nello spazio dei numeri di sequenza. La procedura conferma che entrambi i lati sono raggiungibili, sincronizza i due spazi di sequenza e permette di negoziare opzioni valide per la connessione.

Le due direzioni del flusso possono essere chiuse indipendentemente. Un endpoint segnala con FIN di non avere altri byte da inviare; il peer ne conferma la ricezione e può chiudere la propria direzione in un momento successivo. Per questo la chiusura completa è spesso mostrata come una sequenza a quattro segmenti, anche se ACK e FIN possono essere combinati quando le condizioni lo consentono. Anche FIN consuma una posizione nello spazio di sequenza.

Dopo la chiusura attiva, uno stato di attesa temporaneo evita che segmenti ritardati appartenenti alla vecchia connessione vengano confusi con una nuova connessione che

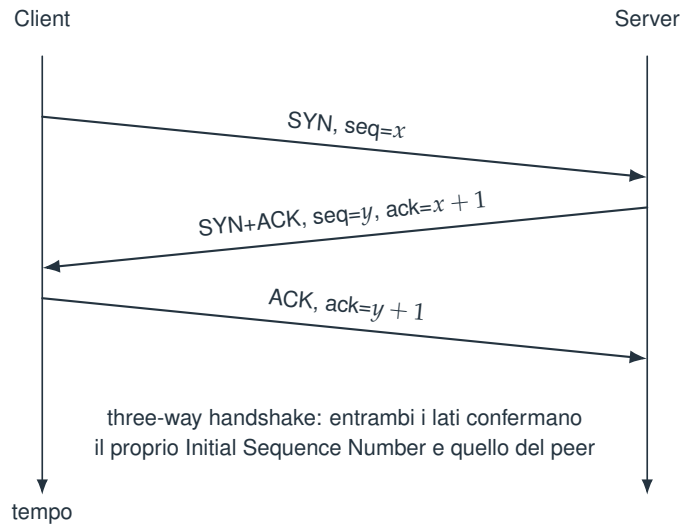


Figura 8.12. Handshake a tre vie per instaurare una connessione TCP.

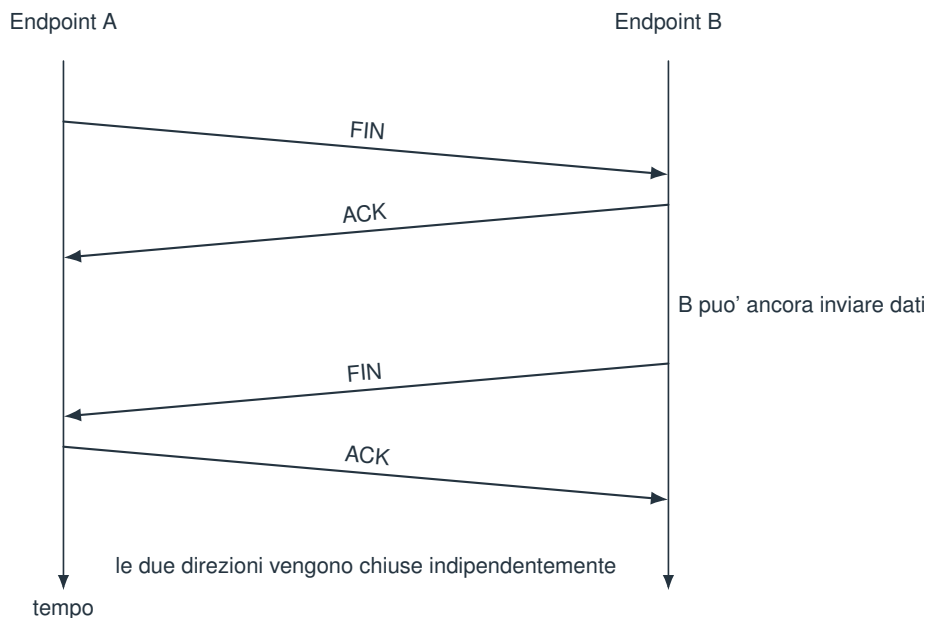


Figura 8.13. Chiusura indipendente delle due direzioni di una connessione TCP.

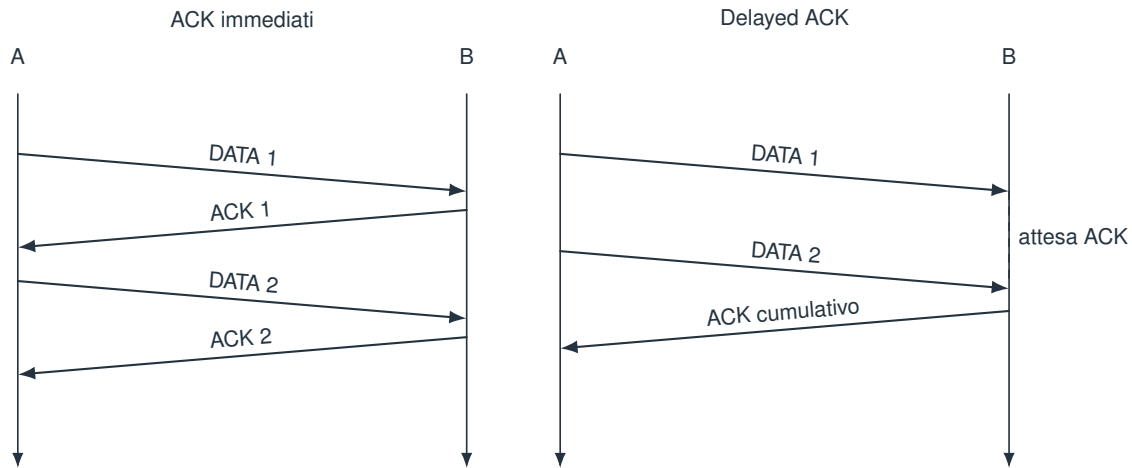
riutilizzi troppo presto gli stessi identificatori. Il dettaglio dei timer è parte della macchina a stati TCP, mentre il principio generale è impedire ambiguità fra incarnazioni successive della stessa quadrupla di endpoint.

8.11.3 Segmentazione, acknowledgement e piccoli trasferimenti

L'applicazione consegna byte a TCP; il trasmettitore decide come raggrupparli in segmenti rispettando MSS, finestre disponibili e politiche temporali. Il ricevitore riconosce cumulativamente i byte contigui ricevuti: un ACK con valore a dichiara che tutti i byte precedenti a sono stati ricevuti nell'ordine necessario al flusso e che il prossimo byte atteso è a . È una conferma del trasporto: non dimostra che l'applicazione abbia già letto, elaborato o salvato stabilmente quei dati.

Gli ACK possono essere ritardati per ridurre il numero di segmenti di puro controllo

e favorire il piggyback su dati trasmessi nella direzione opposta. Il ritardo non è un valore universale fissato dal modello del libro: è regolato dalle regole del protocollo e dalla politica dell'implementazione. Un ritardo eccessivo peggiorerebbe latenza e dinamica del trasmettitore, per cui il delayed ACK è un compromesso fra overhead e tempestività del feedback.



L'ACK ritardato può riconoscere più dati ed essere combinato con dati nella direzione opposta

Figura 8.14. Esempio concettuale di acknowledgement immediato e ritardato.

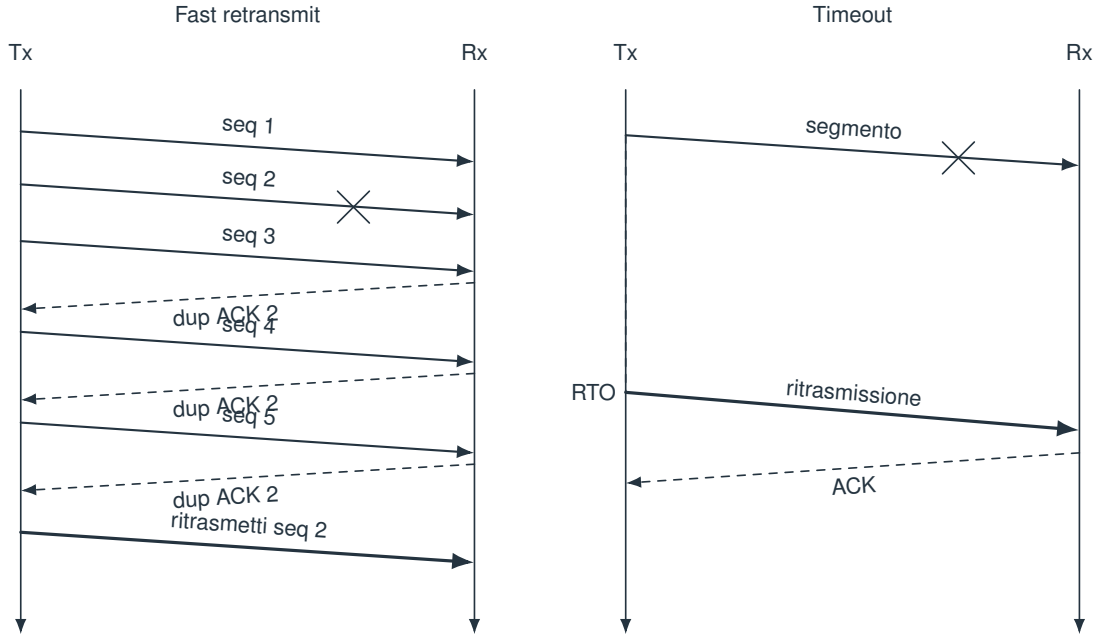
Le applicazioni interattive possono produrre scritture molto piccole. L'algoritmo di Nagle rinvia i nuovi segmenti piccoli quando esistono dati non ancora confermati, anche se quei dati erano stati inviati in segmenti grandi. Attende che i dati pendenti siano confermati oppure che si possa inviare un segmento pieno. Le implementazioni possono adottare varianti; l'applicazione deve poter disabilitare questa aggregazione per una connessione. Essa può interagire con il delayed ACK e aumentare la latenza delle richieste brevi.

Il flag PSH non va interpretato come un comando che obbliga la rete a trasmettere immediatamente un particolare pacchetto. Segnala piuttosto una condizione di push nella semantica del flusso e influenza il modo in cui i dati vengono consegnati all'estremo ricevente; segmentazione e scheduling rimangono soggetti agli altri vincoli TCP.

8.11.4 Rilevazione delle perdite e ritrasmissione

La checksum permette di rifiutare molti segmenti corrotti, ma non rileva ogni possibile alterazione e non autentica il mittente contro un attaccante, mentre i numeri di sequenza consentono di riconoscere duplicati, dati mancanti e arrivi fuori ordine. Un ricevitore può conservare dati fuori ordine in attesa del tratto mancante; gli acknowledgement cumulativi continuano nel frattempo a indicare il primo byte non ancora ricevuto in sequenza. Quando è stata negoziata l'opzione SACK, il ricevitore può inoltre comunicare blocchi di dati ricevuti oltre il buco, permettendo al trasmettitore di individuare con maggiore precisione ciò che deve ritrasmettere.

Nel comportamento Reno classico, la ricezione di tre ACK duplicati è usata come evidenza sufficiente di una perdita e innesca il fast retransmit prima della scadenza del timer. Il numero tre appartiene a quel comportamento classico e non deve essere interpretato come l'unico possibile meccanismo di rilevazione delle perdite in tutte le implementazioni. Anche un riordinamento può produrre ACK duplicati: si tratta di un criterio operativo di inferenza, non di una prova logica che il segmento sia perduto.



tre ACK duplicati possono anticipare la ritrasmissione rispetto alla scadenza del timer

Figura 8.15. Esempio concettuale di ritrasmissione anticipata e di ritrasmissione dopo timeout.

Il secondo meccanismo fondamentale è il Retransmission Timeout (RTO). Il trasmettitore mantiene una stima smussata del round-trip time e una misura smussata della sua variabilità. Sia R la prima misura valida dell'RTT e G la granularità dell'orologio. Si inizializza

$$SRTT = R, \quad RTTVAR = \frac{R}{2}, \quad RTO = SRTT + \max\{G, 4RTTVAR\}. \quad (8.11)$$

Per una misura successiva R' si aggiorna prima

$$RTTVAR \leftarrow (1 - \beta)RTTVAR + \beta|SRTT - R'|, \quad \beta = \frac{1}{4}, \quad (8.12)$$

usando nella differenza il valore precedente di $SRTT$, poi

$$SRTT \leftarrow (1 - \alpha)SRTT + \alpha R', \quad \alpha = \frac{1}{8}, \quad (8.13)$$

e infine

$$RTO \leftarrow SRTT + \max\{G, 4RTTVAR\}. \quad (8.14)$$

La (8.13) è un filtro esponenziale del primo ordine. Per k campioni,

$$SRTT_k = (1 - \alpha)^k SRTT_0 + \alpha \sum_{j=1}^k (1 - \alpha)^{k-j} R_j,$$

come si verifica per sostituzione iterativa. Il peso della condizione iniziale decade geometricamente e i campioni recenti ricevono peso maggiore. $RTTVAR$ non è la varianza statistica dell'RTT: è una deviazione assoluta smussata impiegata per aumentare il margine del timer quando l'RTT è instabile.

Le espressioni precedenti danno la stima del timer prima dei limiti operativi. La regola conservativa standard raccomanda un RTO iniziale di 1 s e di arrotondare a 1 s i valori

calcolati inferiori; non è una costante da inserire al posto dell'RTT misurato. Un eventuale limite massimo del timer deve essere almeno 60 s. Dopo ogni scadenza si raddoppia l'RTO, fino all'eventuale limite. Se il timer scade in attesa dell'ACK di un SYN mentre l'RTO è inferiore a 3 s, la fase dati riparte con RTO almeno 3 s. I coefficienti e queste regole sono scelte di protocollo, non conseguenze matematiche del filtro esponenziale. Una misura RTT associata a un segmento ritrasmesso non deve essere usata quando non è possibile stabilire quale trasmissione abbia generato l'ACK: è il principio di Karn. Operativamente TCP mantiene il timer di ritrasmissione in relazione al più vecchio tratto di dati rilevante, non un timer indipendente per ogni segmento trasmesso.

8.11.5 Controllo di flusso

Il ricevitore pubblicizza una receiver window, $rwnd$, che rappresenta lo spazio disponibile per accettare nuovi byte. Se A è il primo byte non ancora acknowledged e la finestra pubblicizzata vale W , il trasmettitore non deve, per il solo vincolo di flusso, trasmettere nuovi byte con numero di sequenza uguale o successivo al bordo destro $A + W$: il bordo è escluso. Per $W > 0$ l'ultimo byte consentito è $A + W - 1$. Si ragiona nello spazio seriale della connessione; i numeri nel campo TCP sono rappresentati modulo 2^{32} . La finestra cambia nel tempo mentre l'applicazione ricevente consuma i dati dal buffer.

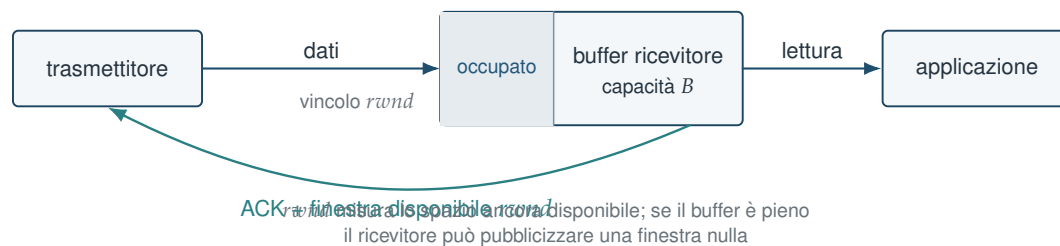


Figura 8.16. Esempio di controllo di flusso mediante finestra pubblicizzata dal ricevitore.

Se il buffer disponibile si esaurisce, il ricevitore può pubblicizzare una finestra nulla. Il trasmettitore sospende allora l'invio di nuovi dati, ma mantiene un meccanismo di persistenza per sondare periodicamente la finestra ed evitare un deadlock nel caso in cui l'aggiornamento che la riapre venga perso.

Il controllo di flusso non è il controllo di congestione. Il primo protegge il ricevitore; il secondo protegge il percorso di rete. Se $cwnd$ è la congestion window, il limite effettivo ai dati in volo è, in prima approssimazione,

$$W_{eff} = \min\{rwnd, cwnd\}. \quad (8.15)$$

È questa separazione che consente a TCP di reagire indipendentemente a un'applicazione ricevente lenta e a una rete congestionata.

8.11.6 Controllo di congestione

TCP distingue il controllo di flusso, imposto dalla finestra pubblicizzata dal ricevitore, dal controllo di congestione, mantenuto dal trasmettitore. Indichiamo con $rwnd$ la receiver window e con $cwnd$ la congestion window. La quantità di dati che può rimanere in volo è limitata, in prima approssimazione, dal minimo tra queste due grandezze.

Per introdurre i principi si considera anzitutto il TCP Reno classico. All'inizio di una connessione il trasmettitore non conosce la capacità disponibile lungo il cammino e usa slow start. Il valore iniziale della congestion window non è fissato dal modello elementare a un solo MSS e può essere maggiore. La scelta di un solo MSS nella figura iniziale di questa sezione è

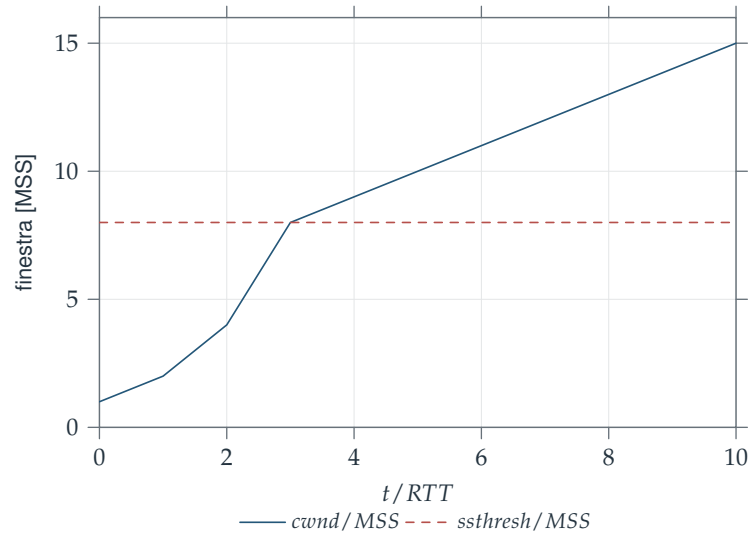


Figura 8.17. Esempio del modello classico Reno: slow start e congestion avoidance con una soglia prefissata.

quindi un'ipotesi didattica utile a rendere visibile la dinamica, non un valore obbligatorio del protocollo.

Nel modello elementare assumiamo un mittente sempre alimentato, non limitato da $rwnd$, segmenti pieni e un ACK per segmento. Se $cwnd$ cresce di un MSS per ogni ACK che riconosce nuovi dati, durante slow start in un RTT vengono riconosciuti circa $cwnd / MSS$ segmenti. Ne segue

$$cwnd^+ = cwnd + MSS \frac{cwnd}{MSS} = 2 cwnd,$$

cioè una crescita approssimativamente esponenziale per RTT. La crescita prosegue fino a quando viene rilevata congestione o viene raggiunta la slow-start threshold, $ssthresh$. La soglia iniziale non è una costante universale di 64 kbyte: può essere inizializzata a un valore elevato e viene poi aggiornata in seguito agli eventi di congestione.

Quando $cwnd$ supera $ssthresh$ il Reno classico entra in congestion avoidance. L'incremento per ACK può essere approssimato da $MSS^2 / cwnd$, per cui nell'arco di un RTT

$$cwnd^+ = cwnd + \frac{MSS^2 cwnd}{cwnd MSS} = cwnd + MSS.$$

Si ottiene così una crescita additiva di circa un segmento per RTT. Se invece un ACK riconosce mediamente b segmenti e si mantengono le stesse regole di incremento per ACK, gli ACK per RTT sono circa $cwnd / (b MSS)$: il fattore slow start diventa circa $1 + 1/b$ e l'incremento in congestion avoidance circa MSS/b . Politiche basate sui byte confermati possono compensare questa dipendenza. Le leggi per RTT non sono quindi indipendenti dalla politica dei riscontri. Non esiste una seconda soglia standard oltre la quale la congestion window viene semplicemente congelata: la sua evoluzione continua in funzione degli ACK, degli eventi di congestione e degli altri limiti della connessione.

La figura 8.17 è calcolata dalla ricorrenza didattica $w_0 = 1$, $w_{n+1} = \min\{2w_n, 8\}$ per $w_n < 8$ e $w_{n+1} = w_n + 1$ per $w_n \geq 8$, con $w_n = cwnd_n / MSS$. L'interpolazione fra campioni ai confini degli RTT visualizza il modello; non è una traccia misurata né una specifica di aggiornamento per ogni singolo ACK.

Nel Reno classico tre ACK duplicati possono innescare fast retransmit e fast recovery. L'idea fondamentale è trattare la perdita rilevata dagli ACK come un segnale di congestione senza

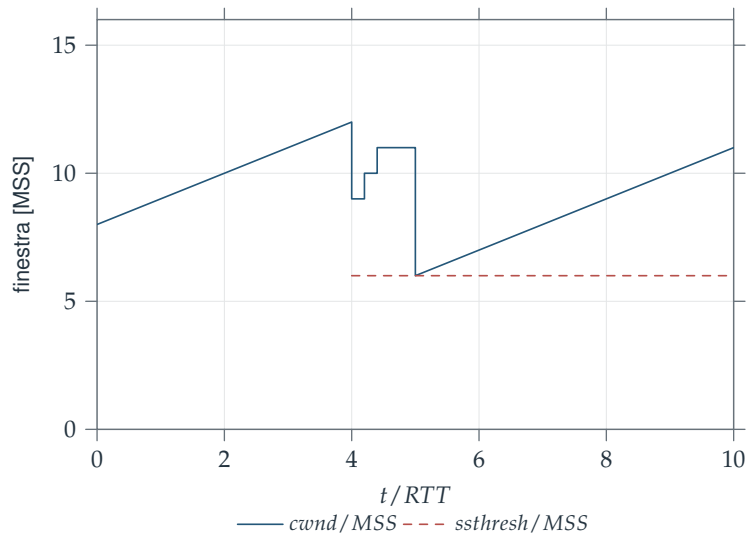


Figura 8.18. Esempio del modello Reno in presenza di eventi rilevati tramite ACK duplicati.

azzerare necessariamente tutta la dinamica della finestra. In forma semplificata, *ssthresh* viene posto a circa metà dei dati in volo; la gestione puntuale di *cwnd* durante fast recovery dipende dalla variante considerata e non è correttamente descritta dalla sola regola $cwnd^+ = cwnd/2$. Nello scenario numerico della figura 8.18, la finestra parte da 8 MSS e cresce di un MSS per RTT fino a 12. All'istante $t/RTT = 4$ tre ACK duplicati trovano 12 MSS in volo: si pone $ssthresh = 6$ MSS e, nel Reno classico, $cwnd = ssthresh + 3$ MSS. Due ulteriori ACK duplicati, prescritti agli istanti 4.2 e 4.4, aggiungono ciascuno un MSS. Un ACK che completa il recupero all'istante 5 riporta la finestra a 6 MSS; riprende poi la crescita additiva. Gli istanti degli eventi sono dati dello scenario, non predizioni del modello di congestione.

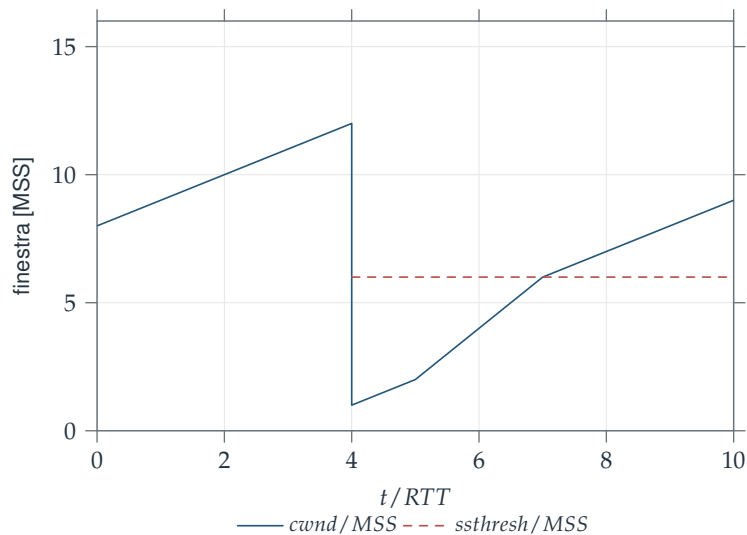


Figura 8.19. Esempio del modello Reno in presenza di scadenze del retransmission timeout.

La scadenza del retransmission timeout è un segnale più severo. Nel modello standard Reno la loss window viene riportata a un segmento SMSS e *ssthresh* viene ridotta in funzione dei dati in volo; la trasmissione riparte quindi con slow start. La figura 8.19 usa lo stesso tratto iniziale da 8 a 12 MSS, ma impone un timeout all'istante 4. La soglia diventa 6 MSS e la finestra riparte da 1 MSS. I campioni successivi sono calcolati con $w^+ = \min\{2w, 6\}$ sotto

soglia e $w^+ = w + 1$ dopo la soglia. Anche qui si specifica l'istante del timeout: non lo si ricava arbitrariamente dalla sola finestra.

Le regole Reno restano fondamentali per comprendere l'additive increase/multiplicative decrease, ma non esauriscono il TCP moderno. Algoritmi standardizzati e largamente utilizzati, come CUBIC, possono adottare una legge di crescita diversa durante congestion avoidance; inoltre SACK, ECN e meccanismi di rilevazione delle perdite più recenti modificano il comportamento operativo. Il principio generale rimane che *cwnd* è controllata dal trasmettitore in base allo stato stimato della rete, mentre *rwnd* è controllata dal ricevitore; il limite effettivo sui dati in volo è determinato dal più restrittivo dei due vincoli.

8.12 User Datagram Protocol, UDP

UDP fornisce un servizio datagram minimale e senza connessione. A IP aggiunge principalmente il multiplexing tramite porte e un controllo d'integrità del datagramma. UDP non garantisce consegna, ordine, unicità o ritrasmissione: un'applicazione che richieda una o più di queste proprietà deve realizzarle esplicitamente oppure scegliere un protocollo di trasporto che le offra. Analogamente, un'applicazione che usa UDP deve adottare un comportamento di controllo della congestione appropriato al proprio caso d'uso.

8.12.1 Struttura del pacchetto

Il pacchetto UDP contiene:

- **Source port, 16 bit**, porta dell'applicazione sorgente;
- **Destination port, 16 bit**, porta dell'applicazione destinataria;
- **Length, 16 bit**, lunghezza in byte dell'header UDP più i dati;
- **Checksum, 16 bit**, complemento a uno della somma in complemento a uno calcolata su uno pseudo-header IP, sull'header UDP e sui dati. La procedura è affine a quella di TCP, non al checksum dell'header IPv4;
- **Data**, dati dell'applicazione.

Nell'UDP su IPv4 il valore zero del campo checksum può indicare che il checksum non è stato generato; con IPv6 il checksum UDP è in via generale richiesto, salvo eccezioni definite per casi specifici.

8.13 Real-time Transport Protocol, RTP

RTP è un protocollo di trasporto per dati con caratteristiche real time, quali audio e video interattivi. È comunemente usato sopra UDP, ma la specifica non lo vincola a UDP. RTP non riserva risorse, non garantisce un ritardo massimo e non rende affidabile la consegna: il suo compito è fornire alle applicazioni informazioni utili per ricostruire temporalmente e identificare un flusso multimediale.

L'header RTP comprende, tra gli altri, payload type, sequence number, timestamp e identificatore SSRC della sorgente di sincronizzazione. Il sequence number permette al ricevitore di rilevare perdite e riordinamenti; il timestamp rappresenta la temporizzazione del contenuto secondo il clock del formato multimediale. Un'applicazione può usare queste informazioni per realizzare un jitter buffer e per presentare i campioni con la corretta cadenza, ma il riordinamento e la riproduzione temporale non sono eseguiti magicamente da RTP stesso. Un mixer può inoltre indicare, mediante gli identificatori CSRC, le sorgenti che hanno contribuito a un flusso composto.

A RTP si affianca Real-time Transport Control Protocol (RTCP). RTCP trasporta informazioni di controllo e report di ricezione/trasmissione. In particolare può fornire statistiche

su perdite, numero di sequenza esteso e jitter inter-arrival, informazioni temporali utili anche alla sincronizzazione tra flussi e identificatori persistenti come CNAME. Questi dati consentono alle applicazioni di valutare la qualità della sessione e adattare, quando previsto, il proprio comportamento. RTP/RTCP non forniscono comunque una garanzia di qualità del servizio: tale garanzia richiederebbe meccanismi aggiuntivi nella rete o nell'applicazione.

8.14 Servizi Internet e risoluzione dei nomi

L'architettura Internet separa il trasporto offerto dalla rete dalle funzioni applicative. I servizi evolvono rapidamente, mentre restano stabili alcuni principi: identificazione delle risorse, risoluzione dei nomi, comunicazione client-server o peer-to-peer, controllo degli accessi e osservazione del comportamento della rete.

Tra i protocolli applicativi di uso generale rientrano, ad esempio, HTTP per il trasferimento di contenuti e API, SMTP per la posta elettronica, SSH per l'accesso remoto sicuro e DNS per la risoluzione dei nomi. Le applicazioni possono usare TCP, UDP o trasporti costruiti sopra UDP, come QUIC.

8.14.1 Domain Name System, DNS

Il Domain Name System associa nomi gerarchici a informazioni necessarie alle applicazioni e alla rete. La funzione più nota è la risoluzione di un nome in uno o più indirizzi IP, ma DNS trasporta anche altre informazioni, ad esempio server di posta, alias e dati relativi alla delega dei domini.

Lo spazio dei nomi è gerarchico. Il resolver locale di un'applicazione può affidare una richiesta ricorsiva a un resolver che ne cerca la risposta. In assenza di informazioni già note, quest'ultimo segue le deleghe dalla radice fino ai server autoritativi della zona pertinente. Una zona è la porzione amministrata da quei server; non coincide necessariamente con tutto il sottoalbero del dominio, perché può delegarne parti a zone figlie. Nella pratica i resolver usano cache, che riducono il numero di interrogazioni e il tempo medio di risoluzione. La validità temporale delle informazioni memorizzate è controllata mediante il Time To Live dei record.

La risoluzione inversa di IPv4 usa il dominio `in-addr.arpa`; quella di IPv6 usa `ip6.arpa`, con i simboli esadecimali dell'indirizzo riportati in ordine inverso a livello di nibble. I server autoritativi possono essere replicati per aumentare disponibilità e robustezza. La sicurezza delle risposte non segue dalla sola gerarchia: DNSSEC permette di verificarne origine e integrità nelle zone firmate, attraverso una catena di fiducia. Non cifra le interrogazioni. Trasporti come DNS over TLS o DNS over HTTPS proteggono invece il canale verso il resolver utilizzato; non sostituiscono, da soli, l'autenticazione dei dati DNS.

8.15 Sicurezza in Internet

Una comunicazione sicura deve essere valutata rispetto a proprietà precise. Le principali sono:

- **confidenzialità**: i dati non devono essere comprensibili a soggetti non autorizzati;
- **integrità**: una modifica non autorizzata deve essere rilevabile;
- **autenticità**: un'entità o un messaggio devono poter essere ricondotti all'origine che si intende autenticare;
- **protezione dal replay**: un messaggio valido già osservato non deve poter essere riutilizzato fuori dal contesto previsto;
- **disponibilità**: il servizio deve continuare a operare entro il modello di guasti e attacchi

per cui è stato progettato.

Nessun singolo algoritmo garantisce tutte queste proprietà. La sicurezza nasce dalla composizione di primitive crittografiche, gestione delle chiavi, identificatori di sessione, numeri di sequenza, politiche di autorizzazione e corretta implementazione.

Un attaccante passivo osserva il traffico senza modificarlo; un attaccante attivo può invece inserire, eliminare, riordinare, modificare o riprodurre messaggi, impersonare un peer o tentare di rendere indisponibile il servizio. Questa distinzione spiega perché la sola cifratura non sia sufficiente: nascondere il contenuto non implica necessariamente rilevare una modifica o autenticare il mittente.

8.15.1 Crittografia simmetrica e autenticazione dei dati

Nella crittografia simmetrica i due estremi condividono materiale segreto. Indicando con K la chiave, una trasformazione di cifratura produce un ciphertext C a partire dal plaintext P e la decifratura ricostruisce P usando la stessa chiave o materiale direttamente derivato da essa.

Nelle comunicazioni moderne è particolarmente utile una primitiva AEAD (Authenticated Encryption with Associated Data). In forma astratta,

$$(C, T) = \text{Enc}_K(N, P, A), \quad (8.16)$$

dove N è un nonce, A è informazione associata che deve essere autenticata ma non cifrata, e T è il tag di autenticazione. Il ricevitore calcola

$$P = \text{Dec}_K(N, C, A, T) \quad (8.17)$$

soltanto se il tag è valido; in caso contrario il messaggio deve essere rifiutato. Le (8.16)–(8.17) descrivono l'interfaccia della primitiva, non dimostrano la sicurezza di un particolare algoritmo.

Il nonce non è necessariamente segreto, ma deve rispettare le condizioni di unicità o non ripetizione richieste dalla primitiva usata. Riutilizzare in modo improprio lo stesso nonce con la stessa chiave può compromettere confidenzialità o integrità. AES-GCM e ChaCha20-Poly1305 sono esempi di costruzioni AEAD diffuse; il punto architetturale importante è l'uso congiunto di cifratura e verifica d'integrità. Un tag valido non impedisce però il replay: un messaggio già osservato conserva il proprio tag valido. Il protocollo deve legare il messaggio alla sessione e verificare numeri di sequenza o altri identificatori di freschezza, mantenendo lo stato necessario.

Quando non serve confidenzialità ma occorre autenticare il messaggio si può usare un Message Authentication Code (MAC):

$$T = \text{MAC}_K(M).$$

La sicurezza richiesta è che chi non conosce la chiave non possa produrre tag validi per messaggi nuovi con probabilità non trascurabile e risorse compatibili con il modello di attacco. Ogni titolare della chiave può invece produrli: un MAC simmetrico non attribuisce una firma pubblicamente verificabile a uno solo dei partecipanti. HMAC è una costruzione di MAC basata su funzioni hash e mostra perché un hash semplice e un MAC abbiano ruoli differenti.

8.15.2 Funzioni hash crittografiche

Una funzione hash crittografica trasforma un messaggio di lunghezza arbitraria in un digest di lunghezza fissa:

$$h = H(M).$$

Per l'uso crittografico si richiedono proprietà diverse:

- **resistenza alla preimmagine:** dato h , deve essere computazionalmente impraticabile trovare un messaggio M con $H(M) = h$;
- **resistenza alla seconda preimmagine:** dato M , deve essere impraticabile trovare $M' \neq M$ con lo stesso digest;
- **resistenza alle collisioni:** deve essere impraticabile trovare una qualunque coppia distinta M, M' con $H(M) = H(M')$.

Queste proprietà non sono equivalenti e non seguono dalla sola non-invertibilità intuitiva della funzione.

Un digest non autenticato non impedisce a un attaccante attivo di modificare sia il messaggio sia il digest. Per autenticare dati con una chiave condivisa serve quindi un MAC; per attribuire pubblicamente una firma a un soggetto serve invece una primitiva di firma digitale. Le famiglie SHA-2 e SHA-3 forniscono esempi di hash moderni; non è utile qui conservare il dettaglio interno di algoritmi superati, perché non aggiunge informazione ai meccanismi di rete trattati nel libro.

8.15.3 Crittografia a chiave pubblica

Nei sistemi asimmetrici una chiave pubblica e una chiave privata sono legate matematicamente ma svolgono ruoli diversi. A seconda della primitiva, la coppia può essere usata per cifratura, firma digitale o per contribuire a un accordo di chiavi. Non è corretto riassumere tutta la crittografia asimmetrica con la sola frase “si cifra con la chiave pubblica e si decifra con la privata”: firme e key agreement hanno una semantica differente.

La disponibilità di una chiave pubblica non ne dimostra da sola l'appartenenza a un'identità. Occorre un meccanismo di autenticazione della chiave, ad esempio una configurazione preventiva o una catena di certificati verificata rispetto ad ancore di fiducia. Separare *possessione della chiave privata* e *identità attribuita alla chiave pubblica* è essenziale per comprendere i protocolli sicuri.

RSA

RSA offre un esempio particolarmente chiaro del legame fra aritmetica modulare e crittografia asimmetrica. Nel modello matematico elementare si considerano

$$C = P^e \bmod n, \quad P = C^d \bmod n,$$

con chiave pubblica (e, n) e informazione privata che permette di conoscere d .

Si scelgono due primi distinti p e q e si pone

$$n = pq, \quad \varphi(n) = (p-1)(q-1).$$

Il valore di $\varphi(n)$ conta i residui coprimi con n : fra i pq residui vi sono q multipli di p e p multipli di q , con il solo zero contato due volte, per cui ne restano $pq - q - p + 1 = (p-1)(q-1)$. Si sceglie e positivo e coprimo con $\varphi(n)$ e si determina d come l'inverso moltiplicativo di e modulo $\varphi(n)$:

$$ed \equiv 1 \pmod{\varphi(n)}. \quad (8.18)$$

L'inverso esiste perché i residui $0, e, 2e, \dots, (\varphi(n)-1)e$ modulo $\varphi(n)$ sono tutti distinti: l'uguaglianza di due residui implicherebbe, per coprimarietà, l'uguaglianza dei due indici modulo $\varphi(n)$. Essi coprono dunque tutte le classi, inclusa 1. Si prende per d un rappresentante positivo dell'inverso. Esiste quindi un intero $h \geq 0$ tale che

$$ed = 1 + h(p-1)(q-1).$$

Correttezza di RSA. Vogliamo dimostrare che, per ogni intero $0 \leq P < n$,

$$P^{ed} \equiv P \pmod{n}. \quad (8.19)$$

Consideriamo prima il risultato modulo p . Se $p \mid P$, allora $P \equiv 0 \pmod{p}$ e quindi $P^{ed} \equiv P \equiv 0 \pmod{p}$. Se invece $p \nmid P$, la moltiplicazione per P permuta i residui non nulli modulo p : $Pa \equiv Pb$ implica $a \equiv b$, perché p è primo e non divide P . Moltiplicando tutti i residui della permutazione,

$$P^{p-1}(p-1)! \equiv (p-1)! \pmod{p}.$$

Nessun fattore di $(p-1)!$ è divisibile per p , perciò si può cancellare il prodotto e ottenere

$$P^{p-1} \equiv 1 \pmod{p}.$$

È il piccolo teorema di Fermat nella forma necessaria alla prova, qui derivato direttamente. Usando $ed = 1 + h(p-1)(q-1)$,

$$P^{ed} = P \left(P^{p-1} \right)^{h(q-1)} \equiv P \pmod{p}.$$

Lo stesso argomento, scambiando p e q , dà

$$P^{ed} \equiv P \pmod{q}.$$

La differenza $D = P^{ed} - P$ è divisibile sia per p sia per q . Scrivendo $D = pa$, dalla divisibilità per q e da $q \nmid p$ segue $q \mid a$; quindi $pq \mid D$. Questo prova direttamente la (8.19), senza dover assumere un ulteriore teorema di scomposizione delle congruenze. Pertanto

$$C = P^e \bmod n \implies C^d \bmod n = P.$$

La correttezza algebrica non è una prova di sicurezza del protocollo: RSA reale richiede schemi di encoding e padding progettati per lo specifico uso, come cifratura o firma, e parametri di sicurezza scelti in modo appropriato.

8.15.4 Gestione e derivazione delle chiavi

La protezione di una sessione richiede chiavi che devono essere generate con sufficiente entropia, associate al peer corretto, rinnovate e infine eliminate quando non servono più. Un protocollo di key establishment combina normalmente autenticazione e accordo di chiavi, così che un osservatore della rete non possa determinare il segreto di sessione e un attaccante attivo non possa sostituirsi a uno degli estremi senza essere rilevato.

Le chiavi di traffico non devono necessariamente coincidere con il segreto prodotto direttamente dall'accordo. Una key derivation function può trasformare il segreto condiviso e il contesto della sessione in chiavi distinte per cifratura, autenticazione e direzioni diverse. La separazione delle chiavi limita l'interdipendenza fra usi crittografici differenti.

8.15.5 IPsec

IPsec protegge traffico a livello IP mediante Security Association monodirezionali. Una SA identifica algoritmi, chiavi, parametri anti-replay e altre informazioni necessarie alla protezione. Numeri di sequenza e finestra anti-replay permettono al ricevitore di rifiutare pacchetti protetti già osservati senza richiedere affidabilità al livello IP.

Encapsulating Security Payload (ESP) può fornire confidenzialità e integrità/autenticazione del contenuto protetto. Authentication Header (AH) fornisce integrità e autenticazione per le

parti coperte ma non confidenzialità. Nelle architetture che richiedono cifratura e integrità, ESP è il meccanismo centrale.

In **transport mode** viene protetto principalmente il payload del pacchetto IP originario e la comunicazione è tipicamente end-to-end. In **tunnel mode** l'intero pacchetto IP originario viene incapsulato in un nuovo pacchetto: i router intermedi vedono l'header esterno necessario al forwarding, mentre il pacchetto interno è protetto secondo le proprietà della SA. Questo modello è naturale per VPN fra gateway o fra un endpoint e un gateway.

Le SA e il materiale di chiave possono essere configurati manualmente, ma la gestione automatica evita di dover predisporre staticamente ogni associazione. IKEv2 autentica i peer, negozia i parametri e stabilisce il materiale necessario alle SA; non si tratta quindi di un semplice scambio statico di chiavi pubbliche.

8.15.6 Sicurezza di trasporto: TLS e QUIC

TLS protegge un flusso applicativo sopra un trasporto affidabile. Nel modello moderno il handshake autentica almeno il peer previsto dalla politica della sessione, negozia i parametri crittografici e deriva chiavi di traffico distinte. I record applicativi vengono poi protetti con primitive autenticate, così che confidenzialità e integrità siano legate alla stessa sessione crittografica. Il certificato del server, quando usato, collega la chiave pubblica all'identità verificata dal client; l'autenticazione del client può essere realizzata con meccanismi differenti a seconda dell'applicazione.

QUIC costruisce un trasporto sicuro sopra UDP e usa il handshake TLS 1.3 per autenticazione e derivazione delle chiavi. Non trasporta per questo un normale flusso di record TLS: la protezione dei pacchetti è applicata da QUIC con le chiavi derivate dal handshake. Una connessione QUIC può contenere più stream indipendenti: l'ordinamento è mantenuto entro il singolo stream senza imporre che una perdita in uno stream blocchi la consegna ordinata dei dati già disponibili negli altri. QUIC integra inoltre acknowledgement, rilevazione delle perdite, controllo di flusso e controllo della congestione. L'indipendenza degli stream riguarda l'ordinamento del trasporto, non ogni risorsa: la finestra di congestione e i limiti complessivi della connessione sono condivisi, e l'applicazione può introdurre ulteriori dipendenze fra stream.

L'invio anticipato di dati 0-RTT, quando abilitato nella ripresa di una sessione TLS o QUIC, riduce l'attesa iniziale ma non offre la stessa garanzia anti-replay dei dati inviati dopo il completamento del handshake. Le operazioni ammesse in 0-RTT devono essere sicure anche rispetto a duplicazioni; la sola cifratura non fornisce una semantica applicativa "esegui una volta sola".

Gli identificatori di connessione separano in parte l'identità logica della connessione dalla semplice coppia indirizzo/porta e permettono, quando le condizioni e la validazione del percorso lo consentono, di sopportare cambiamenti dell'indirizzo di rete del peer. HTTP/3 usa QUIC come trasporto. Il confronto con TCP+TLS mostra quindi due modi diversi di comporre affidabilità, multiplexing e sicurezza senza modificare il servizio best effort fornito da IP.

Indice analitico

- $D/M/1$, 52
- $G/M/1$, 45
- $M/D/1$, 44
- $M/G/1$, 38
- $M/M/1$, 25
- $M/M/1$ con arrivi rallentati, 27
- $M/M/1//N$, 35
- $M/M/1/Y$, 32
- $M/M/Y/Y$, 34
- $M/M/\infty$, 28
- $M/M/\infty//N$, 37
- $M/M/c$, 29

- Accesso casuale, 3, 81
- Accesso controllato, 3, 76
- Accesso deterministico, 3, 73
- Aloha, 81
- AODV, 114
- AQM, 175
- ARP, 179
- ARQ, 61

- B di Erlang, 35, 167
- Back pressure, 118
- Backoff, 81
- Bellman Ford, 111
- Bellman-Ford distribuito, 112
- BGP, 184
- Bifurcating routing, 115
- Blocco sull'ingresso, 126
- Bridge, 171, 172
- Burke, 104

- C di Erlang, 30
- Catene di Markov, 11
- Catene di Markov omogenee, 15
- Chapman kolmogorov, 13
- Chiamate bloccate e cancellate, 165
- Chiave pubblica, 200
- Circuito virtuale, 3
- Clos, 186
- Codice ciclico duale, 58
- Codice duale, 56
- Codici a blocchi, 55
- Codici ciclici, 56

- Commutazione di circuito, 2
- Commutazione di pacchetto, 2
- Condizioni di adiacenza, 16
- Confronto TDMA-FDMA, 75
- Congestion avoidance, 195
- Correzione errori, 60
- CRC, 60
- Crittografia simmetrica, 199
- CSMA, 94
- CSMA/CA, 151
- CSMA/CD, 135
- CTS, 152
- Cut through, 117

- Data center, 186
- Datagram, 3
- Deadlock, 62
- Deflection routing, 115
- DHCP, 181
- Dijkstra, 109
- Distance vector, 112
- Distribuzione di Poisson, 19
- Distribuzione di Erlang, 21
- Divisione di tempo, 162
- Divisione di spazio, 159
- DNS, 198
- DV, 112

- ECMP, 186
- ECN, 175
- Effetto cattura, 90
- Equazione in avanti, 14, 16
- Equazione in indietro, 14, 16
- Ethernet, 134

- Fast recovery, 195
- Fattore di utilizzazione, 24
- FDMA, 73, 75
- FEC, 61
- FIFO, 39
- Finestra mobile, 123
- Forma sistematica, 55
- Full duplex, 62
- Funzioni di Hash, 199

- Gateway, 171

- Go back N, 66
- HDLC, 131
- Hot potato, 115
- ICMP, 180
- ICMPv6, 183
- IEEE 802.11, 150
- IEEE 802.3, 134
- Indirizzi IPv4, 178
- Indirizzi IPv6, 182
- Indirizzi MAC, 171
- Instradamento, 108
- Instradamento ottimo, 116
- IPsec, 201
- IPv4, 177
- IPv6, 181
- Jackson, 106
- Jamming, 136
- Kleinrock, 104
- Lee, 164
- Link state, 113
- Localizzazione traffico, 2
- LS, 113
- Matrice frequenze di transizione, 14
- Matrice tassi di transizione, 14
- MPLS, 185
- Multicast, 180
- Multipath routing, 115
- Nagle, 192
- NAT, 188
- Notazione di Kendall, 24
- OSI, 4
- OSPF, 184
- PIM, 185
- Pollaczek Khinchine, 42, 43
- Porte, 187
- Proactive, 109
- Probabilità di stato, 15
- Processi di Markov, 11
- Processi di nascita e morte, 11, 16
- Processi indipendenti, 11
- Processi stazionari, 11
- Processi stazionari in senso lato, 11
- Protocolli di Instradamento, 183
- Punto-punto, 3
- Qualità del servizio, 175
- QUIC, 202
- Reactive, 109
- Repeater, 171
- Rete di code chiuse, 119
- Rete non bloccante, 159
- Reti di code, 103
- RIP, 184
- Risultato di Little, 23
- Rivelazione errori, 60
- Router, 171, 174
- RPF, 185
- RREP, 114
- RREQ, 114
- RSA, 200
- RTCP, 197
- RTO, 193
- RTP, 197
- RTS, 152
- SAP, 7
- Scheduling, 175
- Segment Routing, 185
- Segnalazione fuori banda, 169
- Segnalazione in banda, 169
- Selective Repeat, 69
- Servizi Internet, 198
- Sindrome di errore, 56
- Sistemi multistadio, 164
- Slotted Aloha, 84
- Slow start, 194
- Source routing, 109
- Spanning tree, 172
- SST, 195
- Stop and wait, 62
- Switch, 171, 172
- TCP, 189
- TDMA, 73, 74
- Tempi di permanenza, 11
- Tempo di accesso, 73, 80, 83, 86, 99
- Tempo di ciclo, 76
- Terminale esposto, 98
- Terminale nascosto, 98
- Throughput, 73, 74, 77, 82, 85, 96
- Timed-token, 148
- Timeout, 62
- TLS, 202

Token, 76
Token ring, 144
Traffico offerto, 73

UDP, 197

VLAN, 174

WLAN, 150