

UNIVERSITÁ DI FERRARA

CORSO DI INFORMATICA

Reti di Telecomunicazioni

Autore:

Alessio CELENTANO



**Università
degli Studi
di Ferrara**

Contents

1	Introduzione	1
1.1	Premessa	1
1.2	4G vs. 5G	1
1.3	Evoluzione della moderna comunicazione	2
1.4	Reti	2
1.4.1	Multiservizi e localizzazione traffico	2
1.4.2	Classificazione delle reti	3
1.4.3	Topologia delle reti	3
1.4.4	Tecniche di commutazione	4
1.4.5	Tecniche di accesso	4
1.5	Internet	5
1.5.1	ISP	5
1.5.2	Internet per i servizi	5
1.5.3	Network edge	6
1.5.4	Rete di accesso	6
1.5.5	Rete di reti	8
1.6	Architettura di Rete	9
1.6.1	Caratteristiche delle moderne reti	9
1.6.2	Modello Open System Interconnection (OSI)	9
2	Elementi di teoria delle code	11
2.1	Sistemi a coda	11
2.2	Evoluzione dello stato	12
2.2.1	Equazione C-K in avanti	13
2.2.2	Equazione C-K all'indietro	14
2.2.3	Soluzione generale dell'equazione C-K	15
2.2.4	Stato singolo	15
2.2.5	Catene di Markov omogenee	15
2.3	Processi di nascita e morte	16
2.3.1	Asse di transizione	16
2.3.2	Processi di sole nascite (Poisson)	17
2.4	Steady state	18
2.4.1	Relazione di Little	19
2.5	Notazione di Kendall	19
2.6	Studio dei sistemi a coda	20
2.6.1	Sistema M/M/1	21
2.6.2	Sistema M/M/1 con arrivi rallentati	23
2.6.3	Sistema M/M/ ∞	25
2.6.4	Sistema M/M/C	26
2.6.5	Sistema M/M/1/Y	29
2.6.6	Sistema M/M/Y/Y	30

3 Data Link - Strato 2	31
3.1 FEC	31
3.2 ARQ	33
3.2.1 Elementi caratteristici strategia ARQ	34
3.2.2 Prestazioni strategia stop & wait	35
3.2.3 Prestazioni strategia Go-back-N	39
3.2.4 Prestazioni strategia Selective Repeat	41
3.3 Tecniche di accesso multiplo	42
3.3.1 Elementi caratteristici	43
3.3.2 Assunzioni	43
3.3.3 Accesso deterministico	43
Time Division Multiple Access (TDMA)	43
Frequency Division Multiple Access (FDMA)	44
Caratterizzazione sistemi TDMA e FDMA	44
3.3.4 Accesso controllato	45
3.3.5 Accesso casuale	47
Sistema Aloha puro	47
Caratterizzazione sistema Aloha	47
Sistema Slotted Aloha	49
Caratterizzazione Slotted Aloha	50
Effetto cattura	51
Carrier Sense Multiple Access (CSMA)	52
Caratterizzazione CSMA non persistente	54
Terminale nascosto e esposto	55
3.3.6 Conclusioni	56
4 Network - Strato 3	57
4.1 Reti di code	57
4.1.1 Assunzione di indipendenza di Kleinrock	58
4.1.2 Teorema di Burke	58
4.1.3 Teorema di Jackson	59
4.1.4 Parametri caratteristici di una rete di code	59
4.2 Algoritmi di instradamento (Routing)	60
4.2.1 Algoritmo di Dijkstra	61
4.2.2 Algoritmo Bellman-Ford	63
4.2.3 Algoritmo Distance Vector	64
4.2.4 Algoritmo Link-State	67
5 Controllo di flusso	69
5.1 Finestra mobile	69
5.2 Blocco sull'ingresso	72
6 Standard implementativi	75
6.1 Protocollo High-level Data Link Control (HDLC)	75
6.1.1 Normal Response Mode (NRM)	75
6.1.2 Asynchronous Response Mode (ARM)	75
6.1.3 Asynchronous Balance Mode (ARM)	76
6.1.4 Struttura pacchetto HDLC	76
6.1.5 Topologia delle reti	77
6.2 Comitato IEEE 802	78
6.2.1 IEEE 802.2 - Logical Link Control (LLC)	78

6.2.2	IEEE 802.3 - Ethernet II	79
6.2.3	IEEE 802.4 - Token bus	80
6.2.4	IEEE 802.5 - Token ring	81
6.2.5	IEEE 802.11 - WLAN	82
6.2.6	Altri protocolli	83
7	Internetworking	85
7.1	Livello 2	86
7.1.1	Bridge	86
7.1.2	Indirizzo MAC	86
7.1.3	Instradamento del bridge	86
7.1.4	Spanning Tree	88
7.2	Livello 3	88
7.2.1	Internet Protocol (IP)	88
7.2.2	Indirizzi IPv4	88
7.2.3	NETMASK	89
7.2.4	Router	90
7.2.5	Protocollo ARP	90
7.2.6	DHCP	91
7.2.7	ICMP	91
7.2.8	IPv6	92
7.3	Livello 4	92
7.3.1	TCP	92
7.3.2	UDP e RTP	93

Chapter 1

Introduzione

1.1 Premessa

Nessun sistema **intelligente** può operare senza comunicazione dell'informazione intra e inter nodo (dispositivo o sistema).

Le **reti di comunicazione** sono in continua evoluzione tecnologica e architettonica per soddisfare requisiti sempre più stringenti in diverse dimensioni.

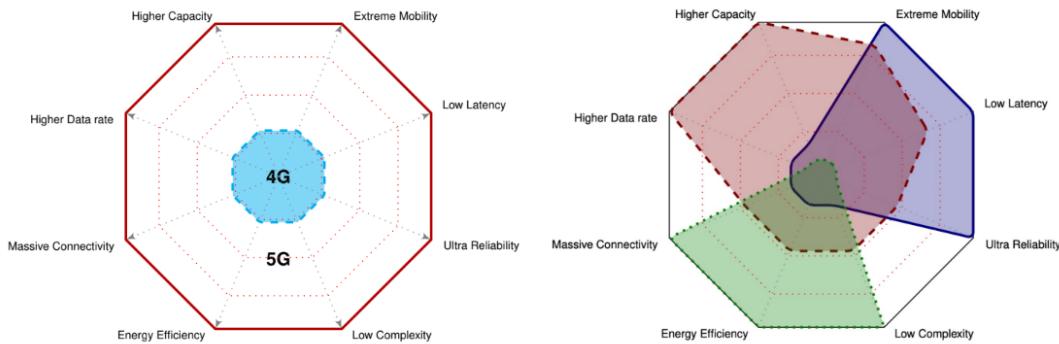
Internet: forse il più grande sistema ingegnerizzato creato dall'uomo con miliardi di dispositivi connessi/utenti e centinaia di milioni di infrastrutture al fine di trasmettere informazione da una o più sorgenti a una o più destinazioni (**traffico**). Internet è una **rete di reti** e sorgenti e destinazioni sono sia persone che cose (Internet-of-Things, IoT).

1.2 4G vs. 5G

Il 4G è stato realizzato per comunicazioni umano-centriche, per contenuti tolleranti ai delay e aveva un range di affidabilità tra il 95% e il 99%.

Per la tecnologia 5G ci sono vari scenari:

- **Banda larga mobile potenziata** (enhanced mobile broadband, eMBB) [rosso in immagine]
 - accesso ad internet
 - realtà aumentata
 - video streaming
- **Comunicazioni ultra-affidabili e a bassa latenza** (ultra-reliable and low latency communications, URLLC) [blu in immagine], applicazioni mission critical
 - sicurezza pubblica
 - macchine autonome
 - assistenza sanitaria
- **Comunicazioni tra macchine** (massive machine type communications, mMTC) [verde in immagine], applicazioni Internet-of-Things
 - smart cities
 - automazioni industriali
 - reti di sensori



1.3 Evoluzione della moderna comunicazione

- 1837. Morse inventa il telegrafo: accensione di un dispositivo a distanza, alfabeto di linee e punti che minimizza il numero di battute per trasmettere testo inglese
- 1876. Meucci inventa il telefono (brevettato da Bell) che trasmette voce umana tramite fili, garantendo l'immediatezza della comunicazione, un circuito diretto fra ogni coppia di utenti, centri di commutazione e linee di giunzione
- Inizio XX secolo. Strowger inventa il selettore elettromeccanico, poi sostituiti da quelli elettronici. Ogni utente ha un numero identificativo, nasce la commutazione automatica e con questa la segnalazione
- 1895. Marconi inventa la radio. La trasmissione dell'informazione diventa su un portante non fisso, il che garantisce mobilità e broadcasting
- 1932. Con la televisione inizia la multimedialità
- 1970. La fibra ottica permette collegamenti molto lunghi e ad elevata velocità

1.4 Reti

1.4.1 Multiservizi e localizzazione traffico

Le reti di telecomunicazioni supportano molteplici servizi in continua evoluzione. Sulle reti si appoggiano fornitori di servizi di vario tipo:

- **over the top (OTT)**: forniscono servizio finale all'utente (Google, Facebook, ...)
- **provider**: forniscono le risorse per garantire il servizio (Aruba, ...)
- **gestori**: forniscono i mezzi per le telecomunicazioni (Telecom, Vodafone, Wind, ...)

La maggior parte del traffico è indirizzata a utenti vicini, quindi la quantità di traffico è mediamente inversamente proporzionale alla distanza (localizzazione del traffico).

1.4.2 Classificazione delle reti

Una prima classificazione riguarda la **dimensione**.

- **Nanonetwork**: applicazioni biomedicali all'interno del corpo
- **Near-Field Communication Networks**: applicazioni che sfruttano l'accoppiamento induttivo fra antenne; comunicazioni wireless a qualche centimetro (e.g.: NFC)
- **Network On Chip (NoC)**: comunicazioni fra moduli interni a circuiti integrati, specie nei chip 3D; ottica wireless
- **PC Bus**: utilizzo all'interno di computer, veicoli e apparati industriali; da pochi centimetri al metro (velocità: 100 Mbps - 10 Gbps)
- **Body Area Network (BAN)**: collegano dispositivi indossabili e dispositivi interni al corpo (velocità: qualche Mbps)
- **Personal Area Network (PAN)**: estende BAN verso il mondo infrastrutturato e i servizi collegando dispositivi a qualche metro (e.g.: Bluetooth) (velocità: qualche Mbps)
- **Local Area Network (LAN)**: per interconnessioni trasparenti in campus, case, fabbriche. Sono sia cablate (Ethernet) che wireless (WLAN) (distanza: circa 80m) (velocità: 1 Mbps - 10 Gbps)
- **Metropolitan Area Network (MAN)**: interconnessione di LAN. Aggregano traffico a maggiore velocità delle LAN
- **Wide Area Network (WAN)**: interconnessione di MAN (livello regionale e nazionale) (distanza: circa 1000km) (velocità: circa 10 Tbps)

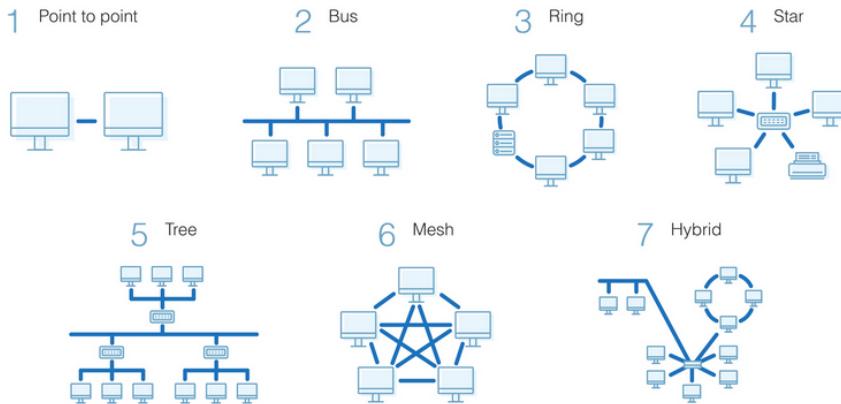
Tutte queste sono reti eterogenee che si interconnettono fra loro e con le reti seguenti:

- **Public Switched Telephone Network (PSTN)**: rete telefonica commutata (distanza: tutto il pianeta) (velocità: qualche Tbps)
- **Private Branch Exchange (PABCX)**: rete telefonica aziendale o enti locali (distanza: 100m - 10km) (velocità circa 10 Mbps)
- **Rete cellulare**: inizialmente era l'estensione mobile della PSTN, ora è la rete globale. Femto e picocelle come access point WLAN. Utilizzata nell'IoT, URLLC, 5G, ...
- **Reti satellitari**: a supporto di PSTN e rete cellulare

1.4.3 Topologia delle reti

I collegamenti sono di vario tipo, in base a prestazioni e velocità desiderate, e diversi mezzi fisici:

- cavi coax
- doppini in rame
- fibre ottiche
- onde elettromagnetiche



1.4.4 Tecniche di commutazione

Fare commutazione significa instradare gli elementi dell'informazione verso il collegamento giusto per permettere all'informazione che parte dal sorgente di raggiungere la destinazione. Per fare ciò esistono varie tecniche:

- **Commutazione di circuito:** viene costituito un circuito fisico dedicato fra sorgente e destinatario. Il circuito rimane instaurato per tutta la durata della comunicazione, anche quando non c'è trasmissione.

Non è una tecnica molto efficiente, quindi si è pensato di suddividere il flusso di bit in **pacchetti** di diversa lunghezza per i diversi servizi

- **Commutazione di pacchetto (datagram):** ogni pacchetto ha un proprio percorso fra sorgente e destinatario utilizzando le risorse in modo dinamico e permettendo di instaurare più comunicazioni con diverse destinazioni. Ogni pacchetto può seguire un percorso diverso e quindi deve contenere l'indirizzo della destinazione e un numero d'ordine affinché possa essere riordinato in ricezione. È la tecnica che utilizza Internet.

Con la commutazione di pacchetti si forma una coda in ogni nodo, la quale provoca aleatorietà del tempo di consegna quindi alla latenza, il che può essere un problema per servizi real-time.

- **Commutazione a circuito virtuale:** viene definito e identificato un percorso fra sorgente e destinatario. I pacchetti che contengono lo stesso identificativo vengono trasmessi in ordine su quel percorso. Anche altri flussi possono usare lo stesso percorso, anche in parte. Il circuito virtuale ha lo svantaggio di non adattarsi ai cambiamenti del traffico

1.4.5 Tecniche di accesso

Le tecniche di accesso al mezzo dipendono fortemente dal tipo di mezzo (fibre ottiche, wireless, ...) e di rete (BAN, MAN, ...)

- **Accesso deterministico:** diversi trasmettitori usano canali di comunicazione distinti (ortogonali) sullo stesso mezzo. Schema di allocazione fisso (TDMA, FDMA, TFDMA, CDMA). È la tecnica di accesso utilizzata con **traffico elevato**
- **Accesso controllato:** diversi trasmettitori usano lo stesso mezzo mediante un meccanismo di assegnazione di un turno basato su una fase di scansione (**polling**). Il passaggio del controllo avviene tramite la trasmissione di un pacchetto detto

testimone (**token**). Il controllo può essere mantenuto per un tempo non superiore ad un massimo prestabilito. La scansione e il passaggio del token richiedono un minimo tempo di attesa per i terminali

- **Accesso casuale:** diversi trasmettitori usano lo stesso mezzo e competono in modo indipendentemente per guadagnare l'accesso al mezzo. Vi é quindi una maggiore possibilità di collisione dei pacchetti, ma non c'è polling e non é richiesto coordinamento fra i trasmettitori. É la tecnica utilizzata con **traffico non elevato**

1.5 Internet

I dispositivi collegati si dividono in ospiti (**host**) e sistemi periferici (**end**), questi ultimi sono connessi tra loro mediante una rete di collegamenti (**communication link**) e commutatori di pacchetto (**packet switch**).

Quando un end vuole inviare informazioni ad un altro end le suddivide in parti e aggiunge un'intestazione: si parla di **pacchetti**. I pacchetti sono inviati attraverso la rete alla destinazione dove vengono riassemblati. Un commutatore di pacchetto prende un pacchetto che arriva da uno dei collegamenti in ingresso e lo ritrasmette verso un collegamento di uscita. Nel cuore della rete (**core net**) ci sono dei **router**, al suo ingresso (**access net**) abbiamo invece degli **switch**. La sequenza di collegamenti e commutatori da sorgente a destinazione é detto percorso (**path**).

1.5.1 ISP

Gli end accedono a Internet tramite **Internet Service Provider (ISP)**, anche detti **Mobile Service Provider (MSP)** nella versione mobile con accesso wireless. Gli ISP forniscono agli end diversi tipi di accesso alla rete tra cui residenziale a larga banda, locale wireless ad alta velocità e fornitori di contributi.

Gli ISP sono connessi fra loro formando diversi livelli gerarchici per coprire la totalità del territorio e dell'utenza: più alto é il livello gerarchico e maggiori sono le velocità di collegamento e commutazione necessarie (traffico aggregato).

Internet é un **sistema aperto**, pertanto sistemi periferici, commutatori di pacchetti e altre parti di Internet fanno uso di **protocolli** cioè di regole per invio e ricezione delle informazioni all'interno della rete (e.g.: convenzioni riguardo a nomi e indirizzi). Due sono i principali protocolli Internet che vedremo: il **Transmission Control Protocol (TCP)** e l'**Internet Protocol (IP)**.

1.5.2 Internet per i servizi

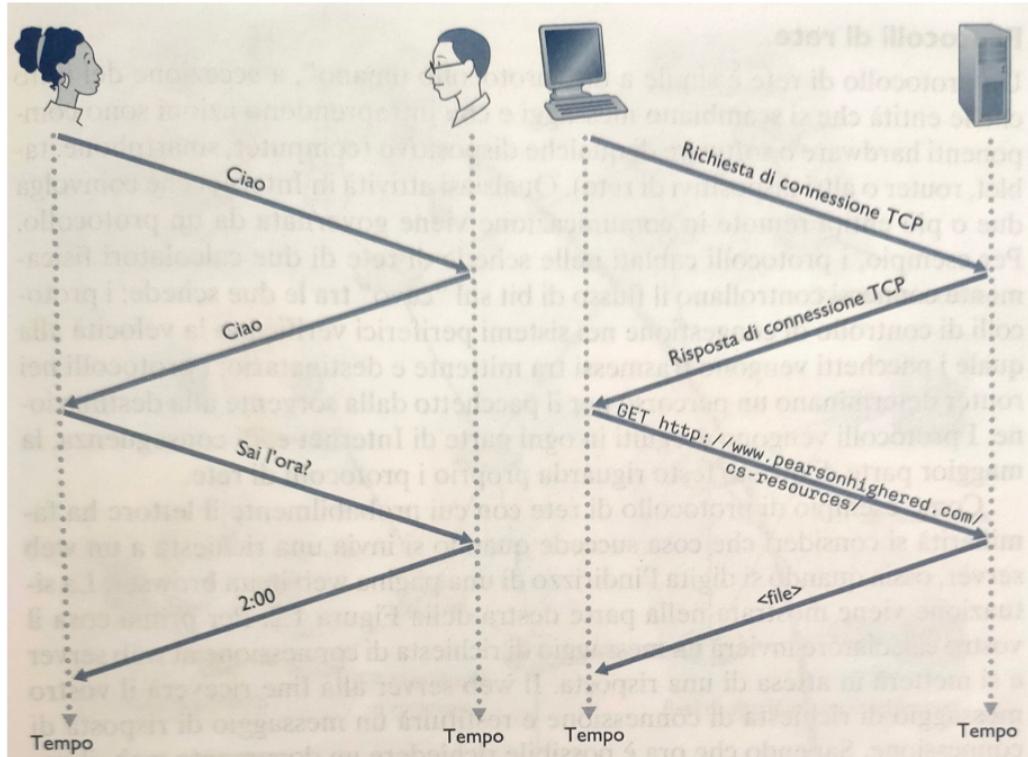
Dal punto di vista dei servizi, Internet é un'infrastruttura che fornisce servizi ad applicazioni distribuite (cioè applicazioni che coinvolgono più sistemi periferici). Le applicazioni Internet vengono eseguite sui sistemi periferici¹.

Le app distribuite, ad esempio, sono eseguite sugli end e pertanto occorre progettare e scrivere moduli software che vengono eseguiti sugli end. Questi moduli software necessiteranno di scambio dati di quindi occorre istruire Internet affinché recapiti dati a un altro pezzo di software eseguito su un altro end. A tal fine gli end collegati a Internet forniscono un'**interfaccia socket** che specifica come un programma eseguito su un sistema periferico possa chiedere a Internet di recapitare

¹Non sui commutatori di pacchetto nel nucleo della rete; i commutatori di pacchetto consentono lo scambio dati tra i sistemi periferici ma non hanno a che fare con le applicazioni

dati a un programma eseguito su un altro sistema periferico. L'interfaccia socket é quindi un insieme di regole che il programma mittente deve eseguire affinché i suoi dati siano recapitati al programma di destinazione.

I protocolli vengono eseguiti in ogni parte di Internet e la maggior parte sono a livello di rete. Esistono protocolli diversi per diverse funzionalità, perciò vogliamo solo comprendere essenza, finalità e modalità operative dei protocolli.



1.5.3 Network edge

Gli **host** sono sistemi periferici che ospitano ed eseguono programmi applicativi. Un host può essere:

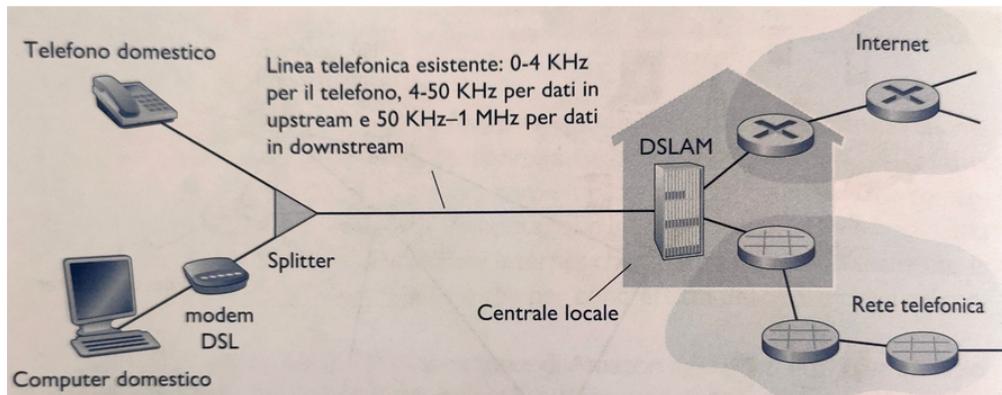
- **client:** host che richiede servizio
- **server:** host che eroga un servizio

La maggior parte dei server é collocata nei datacenter che vengono anche messi in cloud e utilizzati da aziende diverse.

1.5.4 Rete di accesso

La rete di accesso connette fisicamente un end a un router edge, che é il primo router sul percorso. Ce ne sono di vari tipi:

- **Digital Subscriber Line (DSL):** la compagnia di telecomunicazioni fornisce accesso e funge da ISP

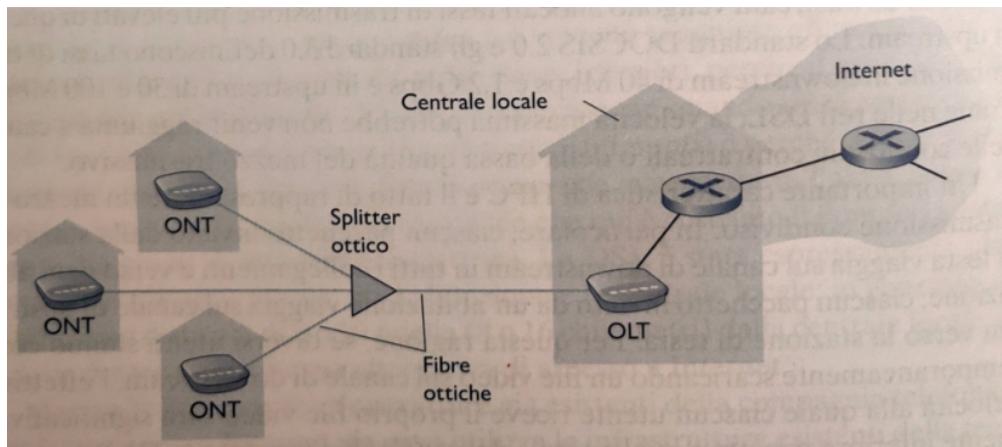


- **Fiber To The Home (FTTH)**: fornisce fibra ottica dalla centrale locale all'abitazione. La fibra che esce dalla centrale locale è condivisa da più abitazioni (splitter ottico). Esistono reti ottiche attive AON e **reti ottiche passive PON**, quest'ultime le più utilizzate.

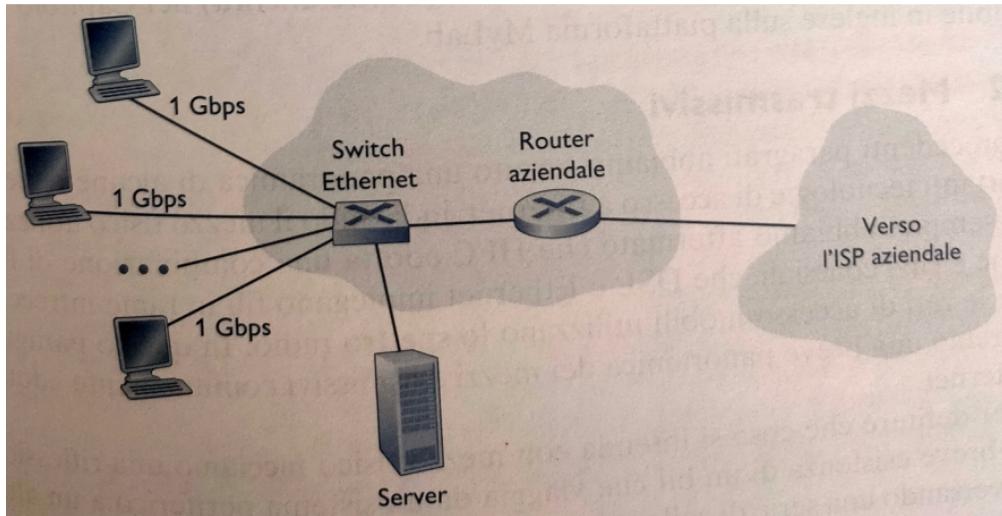
Ogni abitazione ha un terminatore ottico (**Optical Network Terminator ONT**) connesso a uno splitter ottico tramite fibra ottica dedicata.

Lo splitter combina più abitazioni in singola fibra ottica condivisa adatta a supportare traffico aggregato e che si collega al terminatore ottico di linea (**Optical Line Terminator OLT**) situato nella centrale locale della compagnia di telecomunicazioni.

Nell'abitazione i dispositivi si collegano a un router residenziale (tipicamente wireless) collegato all'ONT.



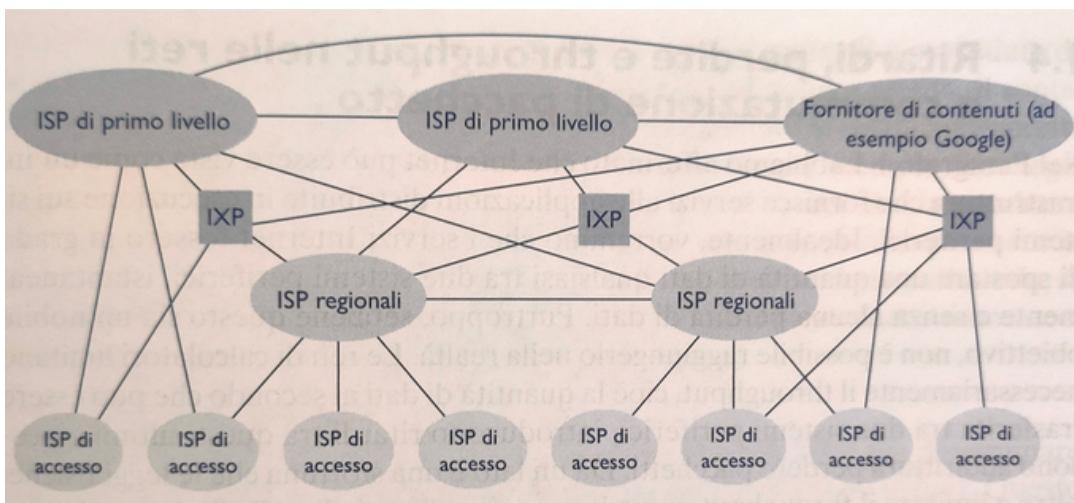
- **5G Fixed Wireless Access (FWA)**: accesso residenziale tramite 5G senza necessità di posa (costosa) di cavi e fibre. Evita guasti su cavi e fibre nel tratto verso la centrale locale. Utilizza beamforming dalla stazione radiobase al modem di casa.
- **Accesso wireless su scala geografica** (3G, LTE 4G, 5G)
- **Accesso aziendale Ethernet e Wi-Fi**: end collegati via doppino oppure onde elettromagnetiche a uno switch Ethernet/Access Point Wi-Fi



1.5.5 Rete di reti

Dato che una struttura a maglia completa (mesh) risulterebbe troppo complessa e costosa, si è deciso di creare una rete di reti (struttura incrementale e gerarchica) attraverso l'interconnessione fra ISP. Gli ISP forniscono:

- **Point of Presence (PoP)**: gruppo di router nella rete dei provider attraverso i quali ISP clienti si connettono a ISP fornitore. I PoP esistono a diversi livelli (eccetto accesso)
- **Multi-Homing (MH)**: qualunque ISP (eccetto quelli di primo livello) possono collegarsi a più di un ISP fornitore (e.g.: ISP accesso collegato a due ISP regionali) per ridondanza contro guasti



Gli ISP clienti pagano gli ISP fornitori con costo che dipende da quantità di traffico scambiato, velocità, ridondanza. Per ridurre tali costi una coppia di ISP vicini e di pari livello gerarchico può fare uso del **peering**, che consiste nel connettere direttamente le loro reti in modo che il traffico fra esse passi attraverso una connessione diretta piuttosto che per un intermediario. Anche gli ISP di primo livello fanno peering.

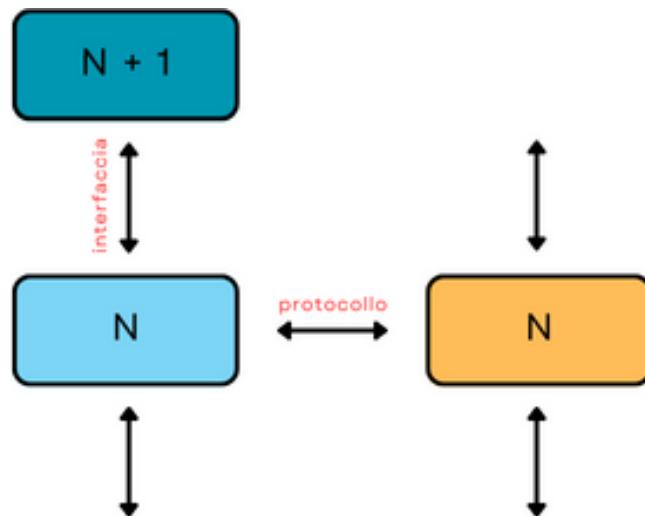
Mediante le connessioni fra ISP, un'azienda può creare un **Internet Exchange Point (IXP)** che è un punto d'incontro in cui diversi ISP possono fare peering. L'Internet attuale conta circa 600 IXP.

La Content Provider Network (CPN) fornisce i contenuti.

1.6 Architettura di Rete

1.6.1 Caratteristiche delle moderne reti

- **Organizzazione gerarchica:** al fine di permettere l'integrazione di reti eterogenee
- **Protocollo:** insieme di regole/convenzioni per la comunicazione con lo stesso livello di un'altra rete (protocollo di livello n)
- **Interfaccia:** insieme di regole/convenzioni per la comunicazione fra livelli contigui della stessa rete



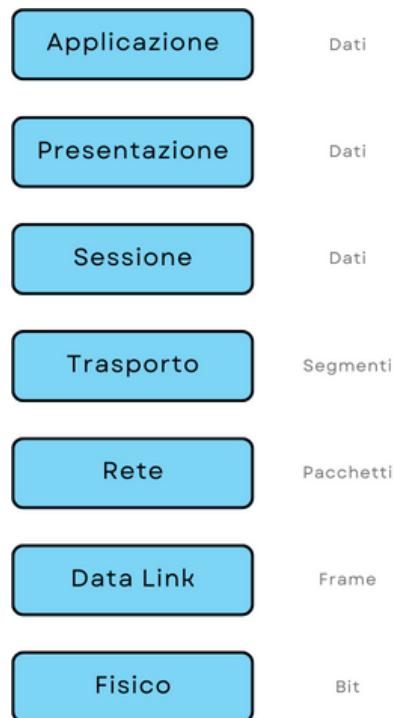
1.6.2 Modello Open System Interconnection (OSI)

L'International Standard Organization (ISO) ha proposto nel 1978 l'architettura OSI che è diventata un modello di riferimento per fare comunicare sistemi aperti con diverse finalità e/o tecnologie. Esso prevede **7 strati**, ognuno dei quali rappresenta un livello di astrazione con funzioni ben definite; i confini fra strati devono essere tali da minimizzare gli scambi di informazioni, in virtù di ciò il numero di strati è stato scelto sulla base di un compromesso funzioni/gestione. Essi sono:

- **Fisico** (Livello 1): si occupa della trasmissione dell'informazione sul canale, sia dal punto di vista dei segnali che delle meccaniche di connessione
Esempio: interfaccia seriale, porta USB
- **Data Link** (Livello 2): definisce il metodo di accesso al mezzo; ha al suo interno meccanismi di protezione (codifica per la correzione degli errori, ritrasmisione); gestisce il controllo di flusso
- **Rete** (Livello 3): si occupa dell'instradamento, ovvero di far arrivare a destinazione i pacchetti (routing), con percorsi fissi (circuiti virtuali) o dinamicamente modificabili (datagram).
Esempio: Internet Protocol (IP) [datagram], X25 (circuito virtuale)

- **Trasporto** (Livello 4): organizza i dati in segmenti/pacchetti crea i collegamenti; verifica l'integrità della comunicazione; definisce il livello di affidabilità delle classi di servizio
Esempio: Transmit Control Protocol (TCP)
- **Sessione** (Livello 5): struttura il colloquio; offre dei punti di sincronizzazione per la ritrasmissione in caso di errore; divide il dialogo in attività di trasferimento dati; identifica il destinatario sulla base dell'identificativo di macchina e del numero di processi
- **Presentazione** (Livello 6): gestisce la modalità di scambio dati soprattutto dal punto di vista della sintassi (e.g.: decide se i numeri sono in virgola fissa o mobile, stabilisce se i caratteri sono ASCII, ...); gestisce la compressione dell'informazione; se necessario aggiunge anche la crittografia
- **Applicazione** (Livello 7): stabilisce i meccanismi che permettono il funzionamento delle applicazioni (e.g.: terminale virtuale, filesystem, ...)

I protocolli dallo strato di trasporto in su vengono chiamati **end-to-end** mentre quelli tra i tre strati più bassi vengono detti **peer-to-peer**.

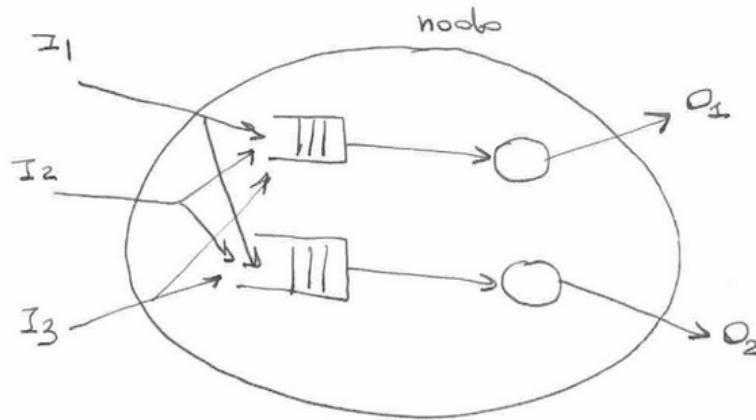


Chapter 2

Elementi di teoria delle code

2.1 Sistemi a coda

La teoria delle code trova applicazione in vari ambiti ed è fondamentale per il progetto e l'analisi delle reti di comunicazione.



I nodi di una rete sono un sistema con più ingressi e più uscite. Sono dotati di:

- un certo numero di **servitori**
- dei **buffer** in cui vengono immagazzinati in attesa di essere serviti e smaltiti

Un utente o un pacchetto *arriva* nel sistema per ottenere un servizio, eventualmente aspetta *in coda* e quando viene servito *parte*.

Arrivi e servizi sono aleatori (non deterministici) e seguono una certa distribuzione; sono quindi aleatori anche i tempi di interarrivo, i tempi di permanenza in coda/sistema e i tempi di latenza.

I sistemi a coda sono modellabili come **catene di Markov** con stato al tempo t $X(t) = j$ e la **probabilità di transizione** dallo stato i allo stato j si indica con:

$$p_{ij}(s, t) = \mathbb{P}\{X(t) = j | X(s) = i\}; \quad t > s$$

Lo **stato** rappresenta la situazione in cui si trova il sistema rispetto a variabili prese come riferimento (e.g.: numero di pacchetti nel sistema) e la sua evoluzione è descritta con una sequenza di salti fra gli stati stessi.

In una catena di Markov (tempo continuo) i tempi di permanenza nei singoli stati seguono una distribuzione esponenziale.

$$\text{PDF: } f_{\tau_i}(t) = f_{\tau_i}(0)e^{-f_{\tau_i}(0)t}$$

$$\text{CDF: } F_{\tau_i}(t) = 1 - e^{-f_{\tau_i}(0)t}$$

dove τ_i è il tempo di permanenza nello stato i .

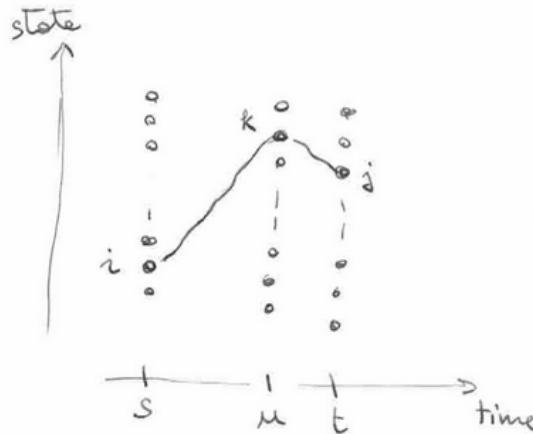
2.2 Evoluzione dello stato

Per comprendere l'evoluzione dello stato dal tempo s al tempo $t > s$ bisogna considerare la probabilità di transizione fra stati già definita:

$$p_{ij}(s, t) = \mathbb{P}\{X(t) = j \mid X(s) = i\}; \quad t > s$$

Consideriamo un istante intermedio u , compreso tra s e t , e valutiamo come può evolvere lo stato passando per u ; consideriamo anche lo stato k come lo stato all'istante u . La nuova probabilità di transizione è:

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



Da

$$\mathbb{P}\{A, B \mid C\} = \mathbb{P}\{A \mid B, C\} \mathbb{P}\{B \mid D\}$$

si ha che:

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

Dato che si parla di una catena di Markov, cioè di un sistema senza memoria, lo stato dipende dal tempo precedente, non da quelli ancora più antecedenti, perciò:

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

A questo punto si può esprimere questa equazione in termini di probabilità di transizione:

$$p_{ij}(s, t) = \sum_k p_{ik}(s, u)p_{kj}(u, t)$$

Questa equazione prende il nome di **equazione di Chapman-Kolmogorov (C-K)** e da essa si può dare una descrizione matriciale definendo per ogni coppia di istanti (s, t) la **matrice di probabilità di transizione**:

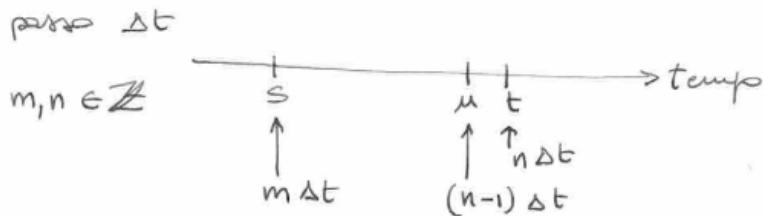
$$M(s, t) = [p_{ij}(s, t)]$$

Ricordando il prodotto tra matrici, l'equazione C-K può essere espressa in forma matriciale:

$$M(s, t) = M(s, u)M(u, t)$$

Nel caso tempo discreto, gli istanti temporali sono multipli di Δt . Al fine di valutare la velocità di transizione fra stati si considera il caso in cui u sia un Δt precedente a t (equazione C-K in avanti) e il caso in cui u sia un Δt successivo a s (equazione C-K all'indietro).

2.2.1 Equazione C-K in avanti



$$M(s, t) = M(m\Delta t, n\Delta t) = M(m\Delta t, (n-1)\Delta t)M((n-1)\Delta t, n\Delta t)$$

Definiamo:

$$V(t) = M(t, t + \Delta t) = [p_{ij}(t, \Delta t)]$$

Esprimiamo la matrice M in funzione di V :

$$M(s, t) = M(m\Delta t, (n-1)\Delta t)V((n-1)\Delta t)$$

La differenza fra due matrici di transizione in istanti distanziati di Δt è:

$$M(m\Delta t, n\Delta t) - M(m\Delta t, (n-1)\Delta t) = M(m\Delta t, (n-1)\Delta t)[V((n-1)\Delta t) - \underbrace{\mathbb{I}}_{\text{Identità}}]$$

Da questo si ottiene il **rappporto incrementale**:

$$\frac{M(m\Delta t, n\Delta t) - M(m\Delta t, (n-1)\Delta t)}{\Delta t} = M(m\Delta t, (n-1)\Delta t) \frac{V((n-1)\Delta t) - \mathbb{I}}{\Delta t}$$

Quindi al limite per $\Delta t \rightarrow 0$:

Equazione C-K in avanti

$$\frac{d}{dt} M(s, t) = M(s, t) Q(t)$$

Con $Q(t) = \lim_{\Delta t \rightarrow 0} \frac{V(t)-1}{\Delta t}$ detta **matrice delle frequenze di trasmissione**. La soluzione richiede di specificare le condizioni iniziali (a tempo s).

$$Q(t) = [q_{ij}(t)]$$

- $j = i$ (permanenza nello stato):

$$\begin{aligned} q_{ii}(t) &= \lim_{\Delta t \rightarrow 0} \frac{p_{ii}(t, t + \Delta t) - 1}{\Delta t} \\ \Rightarrow 1 - p_{ii}(t, t + \Delta t) &= \underbrace{-q_{ii}(t) \Delta t}_{\substack{\text{probabilità di} \\ \text{uscita dallo stato } i}} + o(\Delta t) \end{aligned}$$

velocità di
uscita dallo
stato i

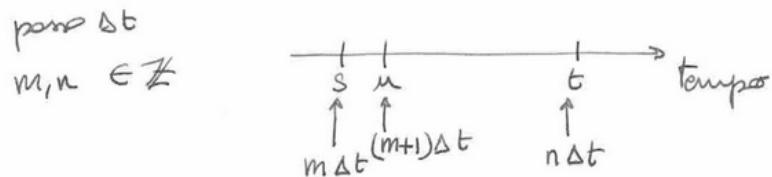
- $j \neq i$:

$$\begin{aligned} q_{ij} &= \lim_{\Delta t \rightarrow 0} \frac{p_{ij}(t, t + \Delta t)}{\Delta t} \\ \Rightarrow p_{ij}(t, t + \Delta t) &= \underbrace{q_{ij}}_{\substack{\text{probabilità} \\ i \rightarrow j}} \underbrace{\Delta t}_{\substack{\text{velocità} \\ i \rightarrow j}} \end{aligned}$$

La somma dei tassi di uscita da un nodo (incluso permanenza) è nulla.

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

2.2.2 Equazione C-K all'indietro



$$M(s, t) = M(m\Delta t, n\Delta t) = M(m\Delta t, (m-1)\Delta t)M((m-1)\Delta t, n\Delta t)$$

Rapporto incrementale:

$$\frac{M(m\Delta t, n\Delta t) - M((m+1)\Delta t, n\Delta t)}{\Delta t} = M((m+1)\Delta t, n\Delta t) \frac{V(m\Delta t) - 1}{\Delta t}$$

Quindi al limite per $\Delta t \rightarrow 0$:

Equazione C-K all'indietro

$$\frac{d}{ds} M(s, t) = -Q(s)M(s, t)$$

La soluzione richiede di specificare le condizioni iniziali (a tempo t).

2.2.3 Soluzione generale dell'equazione C-K

L'equazione C-K ammette una soluzione generale:

$$M(s, t) = e^{\int_s^t Q(u)du}$$

2.2.4 Stato singolo

Ora che abbiamo caratterizzato come si transita da uno stato all'altro, chiediamoci qual è la probabilità di essere in un certo stato i all'istante temporale t .

$$\begin{aligned} P_j(t) &= \mathbb{P}\{X(t) = j\} = \sum_i \mathbb{P}\{X(t) = j \mid X(0) = i\} \mathbb{P}\{X(0) = i\} \\ &= \sum_i P_i(0) p_{ij}(0, t) \\ &= P(0) \cdot M(0, t) \\ &= P(0) \cdot e^{\int_0^t Q(u)du} \end{aligned}$$

Con N stati la probabilità può essere espressa come:

$$P(t) = [P_1(t), P_2(t), \dots, P_N(t)]$$

2.2.5 Catene di Markov omogenee

Una catena di Markov si dice **omogenea** quando le probabilità di transizione da uno stato all'altro non dipendono dal tempo.

$$p_{ij}(s, s+t) = \check{p}_{ij}(t) \mapsto M(s, s+t) = \check{M}(t)$$

$$q_{ij}(t) = \check{q}_{ij} \mapsto Q(t) = \check{Q}$$

In questo caso le derivate parziali dell'equazione C-K si scrivono:

- **in avanti:** $\frac{d}{dt} \check{M}(t) = \check{M}(t) \check{Q}$

- **all'indietro:** $\frac{d}{dt} \check{M}(t) = \check{Q} \check{M}(t)$

La soluzione generale invece diventa:

$$\check{M}(0) = \mathbb{1}$$

$$\check{M}(t) = e^{\check{Q}t}$$

Infine, l'equazione differenziale sulla probabilità di stato diventa:

$$\frac{d}{dt} P(t) = P(t) \check{Q}$$

2.3 Processi di nascita e morte

Consideriamo un sistema a coda omogeneo in cui lo stato rappresenta, ad esempio, un numero di pacchetti. Vale la **condizione di adiacenza**, ovvero che non ci possono essere due nascite o due morti contemporaneamente ($q_{kj} = 0 \forall |k - j| > 1$). In particolare lo stato k può variare in questo modo:

- **nascita:** $k + 1$
- **permanenza:** k
- **morte:** $k - 1$

La matrice dei tassi di transizione permette di descrivere il:

- **tasso di nascita:** $\lambda_k = q_{k, k+1}$
- **tasso di morte:** $\mu_k = q_{k, k-1}$

Il principio di conservazione dell'energia

$$\sum_j q_{kj} = 0 \quad \forall k$$

riscritto in presenza delle condizioni di adiacenza ci porta a dimostrare che:

$$\underbrace{q_{k, k-1}}_{\mu_k} + \underbrace{q_{k, k} + q_{k, k+1}}_{\lambda_k} = 0$$

da cui:

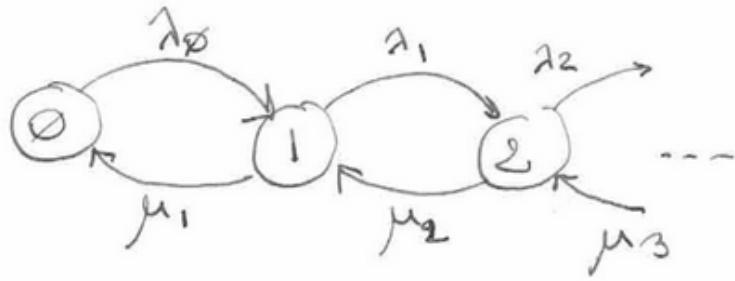
$$q_{k, k} = -(\lambda_k + \mu_k)$$

2.3.1 Asse di transizione

Dato $k = 0, 1, \dots, N-1$, $\mu_0 = 0$ e $\lambda_{N-1} = 0$, la matrice appena calcolata, detta **asse di transizione**, può essere visualizzata in questo modo: Può essere anche rappresentato

$$\mathbb{Q} = \begin{bmatrix} 0 & -\lambda_0 & \lambda_0 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ 1 & \mu_1 - (\lambda_1 + \mu_1) & \lambda_1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \mu_2 - (\lambda_2 + \mu_2) & \lambda_2 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & \mu_{N-2} - (\lambda_{N-2} + \mu_{N-2}) & \lambda_{N-2} & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 & \mu_{N-1} - \mu_{N-1} & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 & 0 & 0 \end{bmatrix}_{N \times N}$$

mediante il **diagramma delle frequenze**: Applicando quanto visto sulle catene di



Markov ai processi di nascita e morte:

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

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

2.3.2 Processi di sole nascite (Poisson)

Dati $\lambda_k = \lambda$ e $\mu_k = 0 \forall k$, si ottiene una **poissoniana**:

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

infatti:

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

con condizione iniziale (sistema inizialmente vuoto):

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

dà come soluzione proprio la poissoniana.

Di questa distribuzione poissoniana se ne conoscono i momenti statistici del I e del II ordine:

- $\mathbb{E}\{k\} = \lambda t$ (numero medio di pacchetti nel sistema)
- $\mathbb{E}\{k^2\} = (\lambda t)^2 + \lambda t$

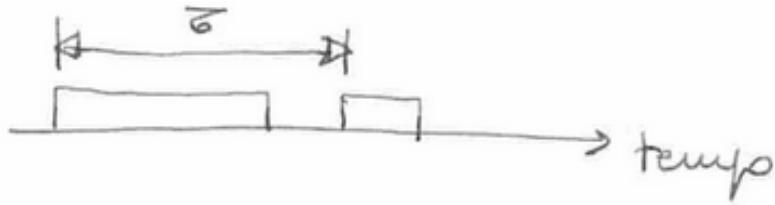
e la varianza:

$$\sigma_k^2 = \{k^2\} - \mathbb{E}\{k\}^2 - \lambda t$$

Il tempo di interarrivo τ , cioè il tempo che intercorre tra gli istanti iniziali di arrivo di due pacchetti contigui, segue una distribuzione esponenziale

$$F_\tau(t) = \mathbb{P}\{\tau \leq t\} = 1 - \mathbb{P}\{\tau > t\} = 1 - \mathbb{P}_0(t) = 1 - e^{-\lambda t}$$

$$f_\tau(t) = \frac{d}{dt}F_\tau(t) = \lambda e^{-\lambda t}$$



k arrivi si collezionano in un tempo Y , dato dalla somma di tutti i T_n tempi di arrivo dell' n -esimo pacchetto dopo l' $n-1$:

$$Y = \sum_{n=1}^k T_n$$

la cui PDF é la convoluzione delle PDF dei T_n ed é una **distribuzione di Erlang**:

$$f_Y(y) = \frac{\lambda^k y^{k-1}}{(k-1)!} e^{-\lambda y}$$

2.4 Steady state

Le equazioni C-K consentono di studiare i transitori, ma puó essere complicato risolverle. Se $P_k(t)$ tende a stabilizzarsi all'aumentare di t allora si individuano le **soluzioni in equilibrio** in regime asintotico ($t \rightarrow +\infty$)

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

se il limite esiste allora:

$$\frac{d}{dt} P_k(t) \xrightarrow[t \rightarrow +\infty]{} 0$$

da cui, in regime asintotico, le equazioni C-K diventano:

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

All'equilibrio, i processi di nascita e morte Markoviani soddisfano la **conservazione del flusso**

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

Se applichiamo la conservazione del flusso alle equazioni precedenti otteniamo:

$$\begin{aligned} P_k &= \frac{\lambda_{k-1}}{\mu_k} P_{k-1} & k \geq 1 \\ &= \frac{\lambda_{k-1} \lambda_{k-2} \dots \lambda_1 \lambda_0}{\mu_k \mu_{k-1} \dots \mu_2 \mu_1} P_0 \\ &= P_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \end{aligned}$$

Si può determinare P_0 ricordando che:

$$\sum_{k=0}^{+\infty} P_k = 1$$

da cui:

$$\begin{aligned} P_0 + \sum_{k=1}^{+\infty} P_k &= 1 \\ \implies P_0 + \sum_{k=1}^{+\infty} P_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} &= 1 \\ \implies P_0 \left[1 + \sum_{k=1}^{+\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \right] &= 1 \end{aligned}$$

$$\begin{aligned} P_0 &= \left[1 + \sum_{k=1}^{+\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \right]^{-1} \\ P_k &= P_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}; \quad k \geq 1 \end{aligned}$$

Queste due equazioni saranno fondamentali per lo studio dei sistemi a coda.

2.4.1 Relazione di Little

All'equilibrio, qualunque sia la disciplina di coda (e.g.: First-in-First-out):

$$\underbrace{\mathbb{E}\{k\}}_L = \underbrace{\mathbb{E}\{\lambda\}}_{\bar{\lambda}} \underbrace{\mathbb{E}\{T\}}_W$$

dove L è il numero medio di pacchetti nel sistema, $\bar{\lambda}$ è il tasso medio di arrivi e W il tempo medio di permanenza (attesa) nel sistema.

2.5 Notazione di Kendall

La notazione di Kendall denota le principali caratteristiche dei sistemi a coda. Formato (campi tra parentesi opzionali):

$$A / B / C [/ Y / N / Z]$$

A : distribuzione dei tempi di interarrivo

B : disciplina del servizio

C : numero di servitori

Y : capacità di sistema (i.e. numero massimo di utenti)

N : cardinalità potenziale della popolazione

Z : disciplina della coda

A e *B* vengono specificati dalle lettere:

- **M** (Markoviano): se la distribuzione è esponenziale
- **G** (generale): se si specifica la PDF
- **D** (deterministico): se il sistema non è aleatoria

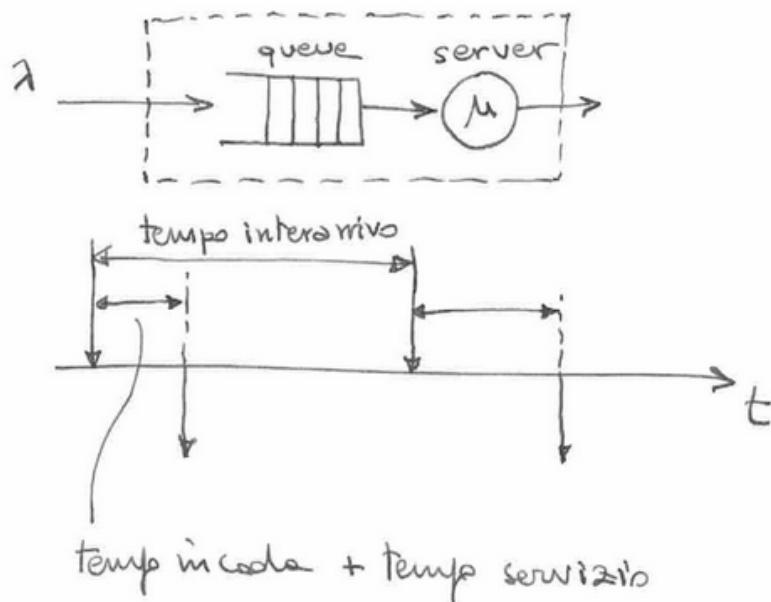
Se non specificati:

- **Y** → +∞
- **N** → +∞
- **Z** → FIFO

esempio: M/M/1 → Markoviano/Markoviano/1 servitore.

2.6 Studio dei sistemi a coda

Durante lo studio e la caratterizzazione dei sistemi a coda ci focalizzeremo su quelli con popolazione infinita in cui il servitore può essere singolo o multiplo e in cui la lunghezza della coda è finita o infinita.



Un parametro fondamentale per la progettazione dei sistemi a coda è il **fattore di utilizzo ρ**

$$\rho = \frac{\text{tasso ingresso}}{\text{massimo tasso smaltimento}}$$

A regime stazionario (steady state):

$$\rho = \underbrace{\mathbb{E}\{\lambda\}}_{\substack{\text{ritmo} \\ \text{arrivi}}} \underbrace{\mathbb{E}\{T_x\}}_{\substack{\text{tempo} \\ \text{servizio}}}$$

Ci dobbiamo aspettare che $\rho \leq 1$ perché in caso contrario il sistema a coda non riuscirebbe a smaltire tutte le richieste arrivate.

Per un qualsiasi sistema a coda con 1 servitore (G/G/1) si ha che il fattore di utilizzo è uguale alla probabilità che ci sia almeno un pacchetto nel sistema:

$$\rho = 1 - P_0$$

Se la disciplina di servizio è Markoviana significa che il tempo fra due servizi ha distribuzione esponenziale con parametro μ :

$$f_{T_x}(\xi) = \mu e^{-\mu\xi}$$

di cui si conoscono i momenti statistici:

$$\mathbb{E}\{T_x\} = \frac{1}{\mu}$$

$$\mathbb{E}\{T_x^2\} = \frac{2}{\mu^2}$$

$$\sigma_{T_x}^2 = \frac{1}{\mu^2}$$

I parametri caratteristici di un sistema a coda sono:

- **descrizione degli arrivi** (distribuzione, λ)
- **descrizione servizi** (distribuzione, μ)
- **numero di servitori** (c)

Da questi parametri si analizzano una serie di altri elementi caratteristici a regime:

- **probabilità di stato** P_k
- **numero medio di pacchetti nel sistema** $L_s = \mathbb{E}\{k\}$
- **numero medio di pacchetti in coda** $L_q = \mathbb{E}\{q\}$
- **tempo medio di attesa nel sistema** $W_s = \mathbb{E}\{T_s\}$
- **tempo medio attesa in coda** $W_q = \mathbb{E}\{T_q\}$

che a loro volta, in base al loro valore minimo/massimo determinato, serviranno per progettare i parametri caratteristici.

2.6.1 Sistema M/M/1

$$\lambda_k = \lambda$$

$$\mu_k = \mu$$

Dato che gli arrivi sono Markoviani (cioè non hanno memoria) lo stato del sistema non dipende dalla storia precedente.

- La distribuzione degli arrivi è poissoniana con parametro λ
- La distribuzione del tempo di servizio è esponenziale con parametro μ

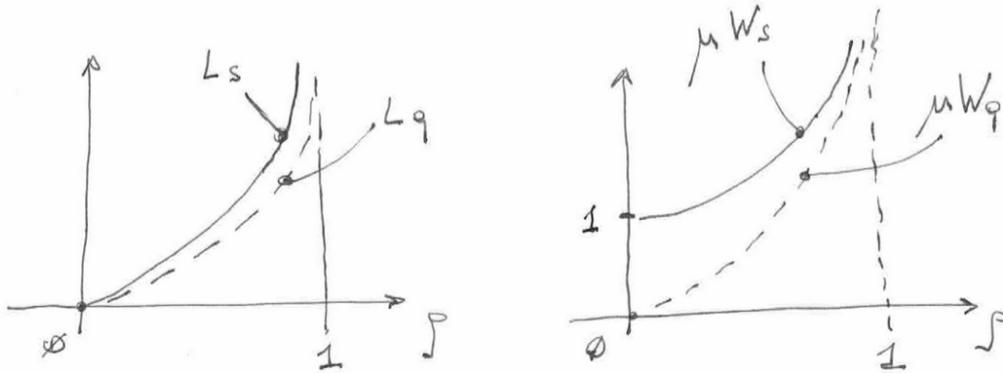
Probabilità di stato: sostituendo i valori P_k e P_0 delle equazioni fondamentali con quelli specificati nel sistema iniziale, otteniamo che:

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

$$\begin{aligned} P_0 &= 1 - \frac{\lambda}{\mu} \\ P_k &= \left(1 - \frac{\lambda}{\mu} \right) \left(\frac{\lambda}{\mu} \right)^k \quad k \geq 0 \end{aligned}$$

Parametri statistici

$$\begin{aligned} \mathbb{E}\{\lambda\} &= \sum_{k=0}^{+\infty} \lambda_k P_k = \lambda \sum_{k=0}^{+\infty} = \lambda \\ \mathbb{E}\{T_x\} &= \frac{1}{\mu} \quad \Rightarrow \quad \rho = \frac{\lambda}{\mu} \\ P_k &= (1 - \rho) \rho^k = P_0 (1 - P_0)^k \\ L_s &= \mathbb{E}\{k\} = \sum_{k=0}^{+\infty} k P_k = \frac{\rho}{1 - \rho} \\ W_s &= \frac{L_s}{\lambda} = \frac{\rho}{1 - \rho} \frac{1}{\lambda} = \frac{1/\mu}{1 - \rho} \\ W_q &= W_s - \mathbb{E}\{T_x\} = W_s - \frac{1}{\mu} = \frac{1}{\mu} \frac{\rho}{1 - \rho} \\ L_q &= \mathbb{E}\{q\} = \lambda W_x = \frac{\rho^2}{1 - \rho} \end{aligned}$$



★ **Esercizio:** consideriamo un sistema M/M/1 all'equilibrio (steady state) con $\lambda = 8$ pkt/s e $\mu = 9$ pkt/s.

$$\rho = \frac{\lambda}{\mu} = \frac{8}{9} \approx 0.89$$

$$P_0 = 1 - \rho = \frac{1}{9} \approx 0.11$$

$$\mathbb{P}\{\text{no queue}\} = P_0 + P_1 = \frac{1}{9} + (1 - \rho)\rho = \frac{1}{9} + \frac{8}{81} = \frac{17}{81} \approx 0.21$$

$$\mathbb{P}\{\text{queue}\} = P_2 + P_3 + \dots = 1 - (P_0 + P_1) \approx 0.79$$

$$L_s = \frac{\rho}{1 - \rho} = \frac{8/9}{1/9} = 8 \text{ pkt}$$

$$L_q = \frac{\rho^2}{1 - \rho} = \frac{(8/9)^2}{1/9} \approx 7.11 \text{ pkt}$$

$$W_s = \frac{L_s}{\lambda} = \frac{8}{8} = 1 \text{ s}$$

$$W_q = \frac{L_q}{\lambda} = \frac{7.11}{8} \approx 0.89 \text{ s}$$

Osservazione quando una risorsa Internet (M/M/1) è condivisa fra pochi utenti ($\rho \rightarrow 0$), allora:

$$W_s \rightarrow \frac{1}{\mu}, \quad W_q \rightarrow 0, \quad L_s \rightarrow 0, \quad L_1 \rightarrow 0$$

Si osserva che se la risorsa è condivisa fra pochi utenti l'arrivo dei pacchetti tende a essere deterministico con memoria. Al crescere del numero di utenti l'arrivo dei pacchetti tende ad essere senza memoria (Markoviano). Inoltre nei sistemi M/M/1 la probabilità di avere almeno N pacchetti/utente nel sistema (coda o server) è

$$\mathbb{P}\{k \geq N\} = \rho^N$$

2.6.2 Sistema M/M/1 con arrivi rallentati

$$\lambda_k = \frac{\alpha}{k+1}$$

$$\mu_k = \mu$$

Probabilità di stato

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

$$\begin{aligned}
 P_0 &= e^{-\frac{\alpha}{\mu}} \\
 P_k &= \frac{1}{k!} \left(\frac{\alpha}{\mu} \right)^k e^{-\frac{\alpha}{\mu}}; \quad k \geq 0
 \end{aligned}$$

Parametri statistici

$$\begin{aligned}
 \rho &= \mathbb{E}\{\lambda\} \mathbb{E}\{T_x\} = 1 - P_0 = 1 - e^{-\frac{\alpha}{\mu}} \\
 \mathbb{E}\{\lambda\} &= \frac{\rho}{\mathbb{E}\{T_x\}} = \frac{\rho}{1/\mu} = \rho\mu = \mu(1 - e^{-\frac{\alpha}{\mu}}) \\
 L_s &= \mathbb{E}\{k\} = \frac{\alpha}{\mu} \\
 W_s &= \frac{L_s}{\mathbb{E}\{\lambda\}} = \frac{\alpha}{\mu} \frac{1}{\mu(1 - e^{-\frac{\alpha}{\mu}})} = \frac{\alpha}{\mu^2 \rho} \\
 W_q &= W_s - \mathbb{E}\{T_x\} \\
 L_q &= W_q \mathbb{E}\{\lambda\} = \frac{\alpha - \rho\mu}{\mu}
 \end{aligned}$$

★ **Esercizio:** consideriamo un sistema M/M/1 con arrivi rallentati all'equilibrio (steady state) con $\lambda_k = \frac{\alpha}{k+1}$ e $\alpha = 10 \text{pkt/s}$. Si determini il minimo tasso di servizio μ^* affinché $W_s \leq W_s^* = 500 \text{ms}$.

$$\begin{aligned} W_s &= \frac{L_s}{\mathbb{E}\{\lambda\}} \\ W_s \leq W_s^* &\Leftrightarrow \frac{\alpha}{\mu^2(1 - e^{-\frac{\alpha}{\mu}})} \leq W_s^* \\ &\Leftrightarrow \mu^2(1 - e^{-\frac{\alpha}{\mu}}) \geq \frac{\alpha}{W_s^*} = 20 \text{ pkt} \\ g(\mu) &= \mu^2(1 - e^{-\frac{10}{\mu}}) \\ g(\mu) \geq 20 &\Leftrightarrow \mu \geq \mu^* \approx 4.8 \text{ pkt/s} \end{aligned}$$

2.6.3 Sistema M/M/ ∞

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

Probabilità di stato

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

$$\begin{aligned} P_0 &= e^{-\frac{\lambda}{\mu}} \\ P_k &= \frac{(\frac{\lambda}{\mu})^k}{k!} e^{-\frac{\lambda}{\mu}}; \quad k \geq 0 \end{aligned}$$

Parametri statistici

$$L_s = \mathbb{E}_s\{l\} = \frac{\lambda}{\mu}$$

$$L_q = 0$$

$$W_q = 0$$

$$W_s = \frac{L_s}{\mathbb{E}\{\lambda\}} = \frac{1}{\mu}$$

In questo caso non ha senso definire il fattore di utilizzo perché il servitore è in grado di reggere qualsiasi livello di utilizzo.

2.6.4 Sistema M/M/C

Il numero di servitori è c per cui il tasso di servizio può arrivare fino a $c\mu$.

$$\begin{aligned}\lambda_k &= \lambda \\ \mu_k &= \min\{k\mu, c\mu\}\end{aligned}$$

Probabilità di stato

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

$\mathbb{E}\{\lambda\}\mathbb{E}\{T_x\} = \frac{\lambda}{\mu}$ e il massimo tasso di smaltimento è c , quindi il fattore di utilizzo è $\rho = \frac{\lambda}{c\mu}$, dove $c\mu$ è il tasso complessivo di servizio μ_{tot} .

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

Imponiamo la condizione di normalizzazione $\sum_{k=0}^{+\infty}$ per determinare la P_0 :

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

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

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

Un pacchetto va in coda se ce ne sono già c nel sistema che stanno occupando tutti i servitori.

$$\begin{aligned} \mathbb{P}\{\text{queue}\} &= \sum_{k=c}^{+\infty} P_k \\ &= \sum_{k=c}^{+\infty} P_0 \frac{c^c}{c!} \rho^k \\ &= P_0 \frac{(c\rho)^c}{c!} \sum_{k=c}^{+\infty} \rho^{k-c} \\ &= P_0 \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} \\ &= \frac{\frac{(c\rho)^c}{c!} \frac{1}{1-\rho}}{\sum_{k=0}^{c-1} \frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!} \frac{1}{1-\rho}} \end{aligned}$$

La forma finale della probabilità di entrare in coda è una formula nota come **formula C-Erlang**:

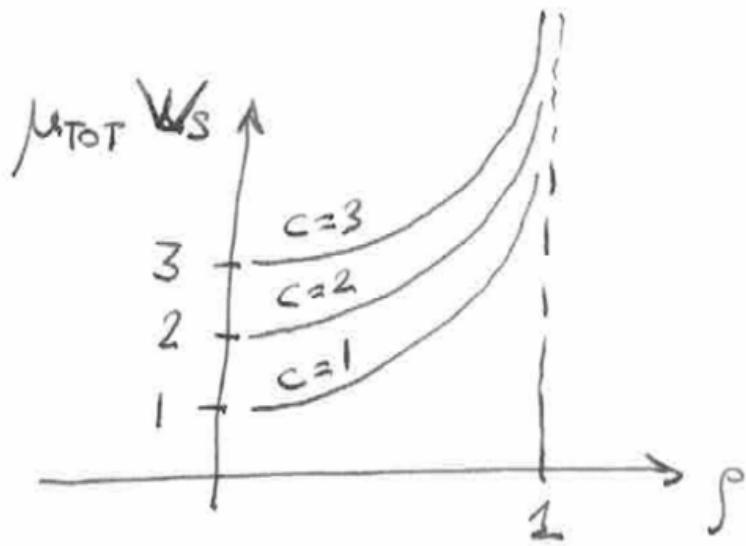
$$C(c, A) = \frac{\frac{(A)^c}{c!} \frac{1}{1-A/c}}{\sum_{k=0}^{c-1} \frac{A^k}{k!} + \frac{A^c}{c!} \frac{1}{1-A/c}}$$

quindi:

$$\mathbb{P}\{\text{queue}\} = C(c, A) \quad \text{con } c = \text{server}, A = c\rho$$

Parametri statistici

$$\begin{aligned} L_q &= \mathbb{E}\{q\} = \sum_{k=c}^{+\infty} (k-c) P_k = C(c, c\rho) \frac{\rho}{1-\rho} \\ L_x &= \mathbb{E}\{x\} = \sum_{k=0}^c k P_k + \sum_{k=c+1}^{+\infty} c P_k = c\rho \\ L_s &= \mathbb{E}\{k\} = L_q + L_x = \left[\frac{C(c, c\rho)}{1-\rho} + c \right] \rho \\ W_q &= \frac{L_q}{\mathbb{E}\{\lambda\}} = \frac{C(c, c\rho)}{\mu_{\text{tot}}(1-\rho)} \\ W_s &= \frac{L_s}{\mathbb{E}\{\lambda\}} = \frac{C(c, c\rho) + c(1-\rho)}{\mu_{\text{tot}}(1-\rho)} \end{aligned}$$



Osservazione: conviene un sistema con pochi servitori veloci rispetto ad uno con molti servitori lenti.

Esercizio: consideriamo un sistema M/M/C in steady state con $c = 2$ server e $\mu = 4$ pkt/s. Si determini il massimo ritmo degli arrivi affinché il tempo medio in coda non superi $W_q^* = 300\text{ms}$.

$$\begin{aligned}
 W_q &= \frac{C(c, c\rho)}{\mu_{\text{tot}}(1 - \rho)}; \quad \mu_{\text{tot}} = c\mu = 8 \text{ pkt/s} \\
 C(c, A) &= \frac{\frac{(A)^c}{c!} \frac{1}{1-A/c}}{\sum_{k=0}^{c-1} \frac{A^k}{k!} + \frac{A^c}{c!} \frac{1}{1-A/c}} \\
 A &= c\rho = c \frac{\lambda}{c\mu} = \frac{\lambda}{\mu} = \frac{\lambda}{4} \\
 \rho &= \frac{\lambda}{c\mu} = \frac{\lambda}{8} \\
 W_q \leq W_q^* &\Leftrightarrow \frac{C(c, c\rho)}{\mu_{\text{tot}}(1 - \rho)} \leq W_q^* \\
 &\Leftrightarrow \frac{C(2, \frac{\lambda}{4})}{1 - \frac{\lambda}{8}} \leq 2.4 \\
 &\Leftrightarrow \frac{\left(\frac{\lambda}{4}\right)^2 \frac{1}{2} \frac{1}{1-\lambda/8}}{1 + \frac{\lambda}{4} + \left(\frac{\lambda}{4}\right)^2 \frac{1}{2} \frac{1}{1-\frac{\lambda}{8}}} \frac{1}{1 - \frac{\lambda}{8}} \leq 2.4 \\
 &\Leftrightarrow \frac{\lambda^2}{32(1 - \frac{\lambda}{8})(1 + \frac{\lambda}{8})} \leq 2.4 \\
 &\Leftrightarrow \frac{\lambda^2}{32} \frac{1}{1 - \frac{\lambda^2}{64}} \leq 2.4 \\
 &\Leftrightarrow \lambda \leq \lambda^* \simeq 5.91
 \end{aligned}$$

2.6.5 Sistema M/M/1/Y

Al massimo Y pacchetti nel sistema, di cui 1 nel server. Se la coda è piena si scartano gli arrivi.

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

$$\mu_k = \mu$$

Probabilità di stato

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

Non si definisce un fattore di utilizzo che dipenderebbe dallo stato rispetto a Y. Definiamo $A = \frac{\lambda}{\mu}$.

$$P_0 = \left[1 + \sum_{k=1}^Y A^k \right]^{-1} = \left[\frac{1 - A^{Y+1}}{1 - A} \right]^{-1}$$

$$P_k = \begin{cases} \frac{1-A}{1-A^{Y+1}} A^k; & k \leq Y \\ 0; & k > Y \end{cases}$$

La probabilità di scartare un arrivo è la probabilità che ci siano già Y pacchetti nel sistema.

$$P_Y = \mathbb{P}\{\text{block}\} = \frac{(1-A)A^Y}{1 - A^{Y+1}}$$

Parametri statistici

$$L_s = \sum_{k=0}^{+\infty} k P_k = \frac{A}{1-A} - \frac{(Y+1)A^{Y+1}}{1-A^{Y+1}}$$

$$\mathbb{E}\{\lambda\} = \sum_{k=0}^{+\infty} \lambda_k P_k = \lambda \sum_{k=0}^{Y-1} P_k = \lambda(1 - P_Y) = \lambda \frac{1 - A^Y}{1 - A^{Y+1}}$$

$$W_s = \frac{L_s}{\mathbb{E}\{\lambda\}}$$

$$W_q = W_s - \frac{1}{\mu}$$

$$L_q = \mathbb{E}\{\lambda\} W_q$$

2.6.6 Sistema M/M/Y/Y

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

$$\mu_k = k\mu$$

Probabilità di stato

$$P_k = P_0 \sum_{i=0}^{+\infty} \frac{\lambda_i}{\mu_{i+1}}$$

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

$$= \begin{cases} P_0 \frac{A^k}{k!}; & k \leq Y \\ 0; & k > Y \end{cases}$$

$$P_0 = \left[1 + \sum_{i=0}^Y \frac{A^i}{i!} \right]^{-1} = \left[\sum_{i=0}^Y \frac{A^i}{i!} \right]^{-1}$$

$$P_k = \begin{cases} \frac{A^k}{k!}; & k \leq Y \\ 0; & k > Y \end{cases}$$

La probabilità di blocco degli arrivi è la probabilità che tutti i servitori risultino occupati.

$$\mathbb{P}\{\text{block}\} = P_Y = \frac{\frac{A^Y}{Y!}}{\sum_{i=0}^Y \frac{A^i}{i!}} = \underbrace{B(Y, A)}_{\text{Formula B-Erlang}}$$

Parametri caratteristici

$$L_s = \sum_{k=0}^{+\infty} k P_k = A [1 - B(Y, A)]$$

$$L_q = 0$$

$$W_s = \mathbb{E}\{T_x\} = \frac{1}{\mu}$$

$$\mathbb{E}\{\lambda\} = \sum_{k=0}^{+\infty} \lambda_k P_k = \lambda (1 - P_Y) = \lambda [1 - B(Y, A)]$$

$$L_s = \mathbb{E}\{k\} = W_s \mathbb{E}\{\lambda\} = A [1 - B(Y, A)]$$

Chapter 3

Data Link - Strato 2

Obiettivi principali:

- Protezione dell'informazione ai disturbi \rightarrow **forward error correction (FEC)**

La codifica FEC introduce una ridondanza controllata sul flusso di bit, organizzandoli in parole di codice, al fine di inserire delle regole per correggere fino ad un certo numero (capacità di correzione) di errori per parola causati dal canale di trasmissione o perlomeno per provarli ad identificarli. Maggiore è la ridondanza, maggiore saranno l'overhead e la capacità di correzione.

- ritrasmissione \rightarrow **automatic repeat request (ARQ)**

La ritrasmissione rileva se ci sono / permangono possibili errori e chiede agli stati superiori di ritrasmettere una parte dell'informazione.

- accesso al mezzo condiviso \rightarrow **multiple access**

- accesso deterministico
- accesso controllato
- accesso casuale

Le tecniche di accesso multiplo regolano la trasmissione sul mezzo di propagazione quando questo è utilizzato da più utenti.

3.1 FEC

In questa sezione ci occupiamo solo di una categoria di codici:

Codici lineari (n, k, t)

k bit di informazione $\rightarrow n$ bit codeword

t capacità di correzione, dipende dalla code rate (R_c)

$$R_c = \frac{k}{n} \quad n \geq k$$

Esempio: codice a ripetizione (3, 1, ...)

Codifica:

- 0 → 000
- 1 → 111

Spazio complessivo per una terna di bit: $2^3 = 8$ elementi.

Spazio delle parole di codice: 2 elementi (000 e 111).

Decodifica:

- ≥ 2 zeri → 0
- ≥ 2 uni → 1



$$0 \rightarrow 000 \xrightarrow{0 \text{ errori}} 000 \rightarrow 0$$

$$1 \rightarrow 111 \xrightarrow{0 \text{ errori}} 111 \rightarrow 1$$

$$0 \rightarrow 000 \xrightarrow{1 \text{ errore}} \begin{matrix} 001 \\ 010 \\ 100 \end{matrix}$$

$$1 \rightarrow 111 \xrightarrow{1 \text{ errore}} \begin{matrix} 110 \\ 101 \\ 011 \end{matrix}$$

$$0 \rightarrow 000 \xrightarrow{2 \text{ errori}} \begin{matrix} 011 \\ 101 \\ 110 \end{matrix}$$

$$1 \rightarrow 111 \xrightarrow{2 \text{ errori}} \begin{matrix} 100 \\ 010 \\ 001 \end{matrix}$$

$$0 \rightarrow 000 \xrightarrow{3 \text{ errori}} 111 \rightarrow 1$$

$$1 \rightarrow 111 \xrightarrow{3 \text{ errori}} 000 \rightarrow 0$$

Con 1 errore abbiamo una correzione, con 2 erroriabbiamo una rilevazione di un errore, con 3 errori non viene rilevato l'errore. Dunque la capacità di errore è uguale a 1.

In generale, le tipologie di codice che adottano una regola di decodifica basata sulla distanza rispetto alle parole di codice permesse definiscono:

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

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

È necessario dimensionare la parola di codice e il suo contenuto informativo per ottenere il desiderato trade-off fra efficienza nell'uso delle risorse e probabilità di errore.

In generale, la costruzione dei codici lineari presenta:

$$\begin{aligned} \underline{c} &= \underline{x} \underline{\underline{G}} \\ \underline{\underline{G}} &= \left[\underbrace{\underline{\underline{I}}}_k \mid \underbrace{\underline{\underline{P}}}_{n-k} \right] \} k \end{aligned}$$

Dove:

- \underline{x} $1 \times k$ info
- \underline{c} $1 \times n$ codeword
- $\underline{\underline{G}}$ $k \times n$ matrice generatrice
- $\underline{\underline{I}}$ $k \times k$ matrice identità
- $\underline{\underline{P}}$ $k \times (n - k)$ matrice parità

Si trasmette \underline{c} e si riceve $\underline{r} = \underline{c} + \underbrace{\underline{e}}_{\text{errore}}$.

In ricezione si verifica la correttezza dell'informazione e si correggono gli errori con tecniche di decodifica basate su sindromi di errore. Una particolare famiglia di codici lineari è quella dei codici ciclici che si realizzano mediante shift register (cyclic redundancy check, CRC).

3.2 ARQ

Qualora rimangano errori non correggibili si può richiedere una ritrasmissione: meccanismi/strategie ARQ (Automatic Repeat reQuest).

Rispetto a FEC, l'ARQ richiede una ridondanza inferiore:

$$\begin{aligned} \# \text{ errori rilevabili} &= d_{\min} - 1 \\ \# \text{ errori correggibili} &= \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \\ d_{\min} &\leq n - k + 1 \end{aligned}$$

Le strategie di ARQ prevedono che si trasmetta un acknowledgement (ACK) per confermare la corretta ricezione o un non acknowledgement (NACK) per indicare una ricezione non corretta. In una rete può capitare che un pacchetto venga distrutto o perso e quindi che si attenda invano un ACK/NACK, generando uno stallo

(deadlock). Per questo motivo si prevede un timeout, cioè un intervallo di tempo dalla trasmissione oltre il quale non si attende più un ACK/NACK e si forza la ri-trasmissione del pacchetto.

3.2.1 Elementi caratteristici strategia ARQ

- t_F : tempo di trasmissione singolo pacchetto
- t_T : tempo totale di trasmissione pacchetto

$$t_F + \text{ritrasmissioni}$$

- t_I : tempo di trasmissione singolo pacchetto (solo parte informativa)
- ρ : fattore utilizzo

$$\frac{t_F}{t_T}$$

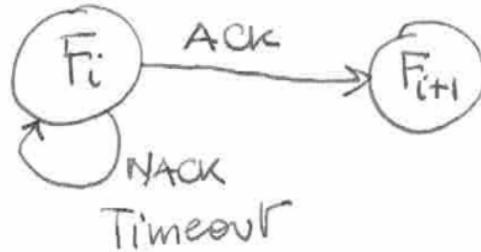
- η : efficienza

$$\frac{t_I}{t_T}$$

Chiamiamo F_i l' i -esimo frame (pacchetto). Esistono tre strategie di ARQ:

- **Stop & wait**: si trasmette F_i e si attende la risposta

- ACK \Rightarrow si trasmette F_{i+1}
- NACK \Rightarrow si ritrasmette F_i
- Timeout \Rightarrow si ritrasmette F_i

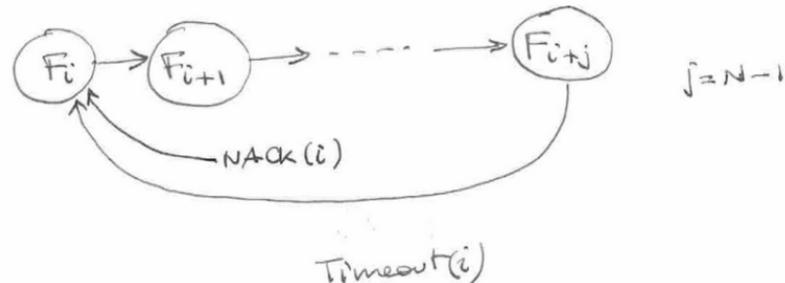


Il "wait" per ogni pacchetto può rendere il canale molto sottoutilizzato e per questo si è pensato alla prossima strategia

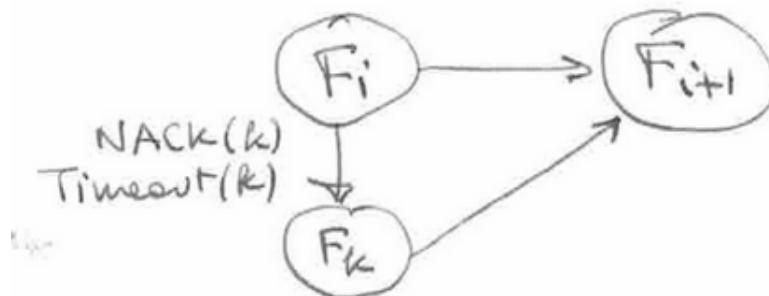
- **Go-back-N**: si trasmettono in continuazione pacchetti:

- se dopo altre $N - 1$ trasmissioni non è arrivato un ACK/NACK scatta il timeout e si ritrasmette dal primo degli N
- se arriva ACK sul pacchetto i allora la finestra trasla a partire dal successivo
- se arriva NACK sul primo si ritrasmette tutto dal primo

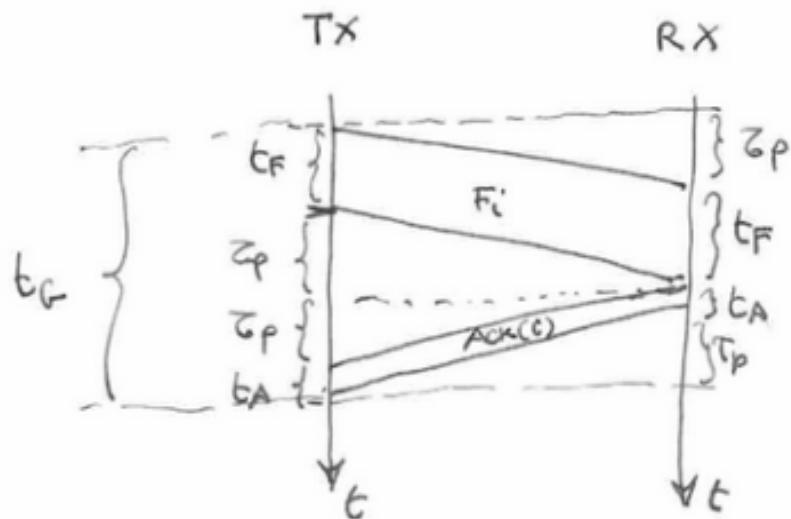
In questo modo evita l'attesa dell'ACK dopo ogni trasmissione. È fra le tecniche più usate.



- **Selective repeat:** rispetto al Go-back-N si ritrasmette solo il pacchetto che risulta errato o per cui scatta il timeout. È più efficiente ma si perde l'ordine dei pacchetti che va ripristinato.



3.2.2 Prestazioni strategia stop & wait



dove:

- T_x : trasmissione
- R_x : ricezione

- t_F : durata singolo pacchetto
- t_A : durata ACK/NACK pacchetto
- τ_p : ritardo di propagazione
- t_G : tempo complessivo per la trasmissione di un pacchetto indipendentemente da errori o ritrasmissioni

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

Da cui:

- t_0 : timeout
- $$t_0 \geq t_A + 2\tau_p$$
- t_T : tempo medio trasmissione pacchetto

$$t_T = t_G + \mathbb{E}\{t_{\text{ritrasmissioni}}\}$$

- $p = \mathbb{P}\{\text{pacchetto errato}\}$

$$p = 1 - (1 - P_b)^F \approx FP_b$$

dove:

- P_b : probabilità di errore sul singolo bit
- $(1 - P_b)^F$: probabilità ricostruzione corretta del pacchetto
- $P_i = \mathbb{P}\{\text{pacchetto corretto alla } i+1 \text{ trasmissione}\} \Rightarrow i \text{ ritrasmissioni}$

$$P_i = p^i(1-p)$$

- t_i : tempo speso in ritrasmissione con numero pacchetti errati o persi pari a i

$$t_i = it_G$$

- Tempo medio di ritrasmissione:

$$\begin{aligned} \mathbb{E}\{t_{\text{ritrasmissioni}}\} &= \sum_{i=0}^{+\infty} t_i P_i \\ &= t_G(1-p) \sum_{i=0}^{+\infty} ip^i \\ &= t_G \frac{p}{1-p} \end{aligned}$$

t_T può essere riscritto come

$$\begin{aligned} t_T &= t_G \left[1 + \frac{p}{1-p} \right] \\ &= \frac{t_G}{1-p} \end{aligned}$$

Facendo le opportune ipotesi:

$$\begin{cases} t_0 = t_A + 2\tau_p \\ \text{ACK/NACK privi di errore} \end{cases} \Rightarrow t_G = t_F + t_0$$

allora t_T può essere ridotto ulteriormente a:

$$\begin{aligned} t_T &= \frac{t_F + t_0}{1-p} \\ &= \frac{t_F + t_A + 2\tau_p}{1-p} \end{aligned}$$

Volendo mettere in evidenza le lunghezze dei pacchetti

$$\begin{aligned} F &= ct_F \\ A &= ct_A \\ \tau_p &= \alpha t_F \\ t_T &= \frac{t_F}{1-p} \left[1 + \frac{A}{F} + 2\alpha \right] \end{aligned}$$

dove

- c : velocità di trasmissione [bit/s]
- $\alpha = \frac{\tau_p}{t_F}$: ritardo di propagazione normalizzato alla durata del pacchetto

Per quanto riguarda il fattore di utilizzo:

$$\begin{aligned} \rho &= \frac{t_F}{t_T} \\ &= \frac{t_F(1-p)}{t_F + t_0} \\ &= \frac{\frac{F}{c}(1-p)}{\frac{F}{c} + \frac{A}{c} + 2\tau_p} \\ &= \frac{F(1-p)}{F + A + 2c\tau_p} \end{aligned}$$

- se $p \rightarrow 0$: $\rho = \frac{t_F}{t_F + t_0} \Rightarrow$ attesa interminabile di ACK/NACK o timeout
- se $t_0 \rightarrow 0$: $\rho = 1 - p \Rightarrow$ limitato dagli errori sul canale di trasmissione

Mettiamo in evidenza come altri parametri entrano all'interno del fattore di utilizzo:

- $I = ct_I$: numero di bit di informazione all'interno di un pacchetto
- $O = ct_o$: numero di bit dedicati all'overhead all'interno di un pacchetto

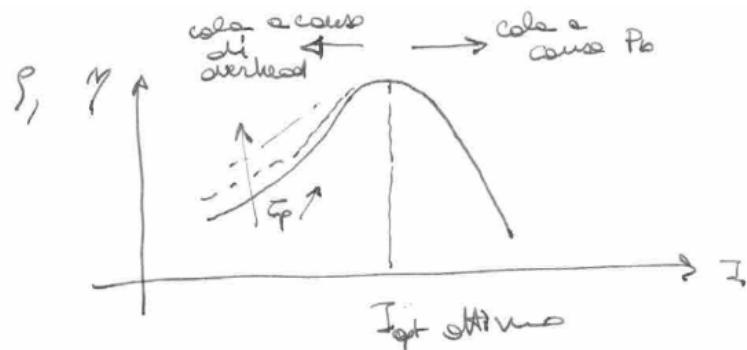
Utilizzo

$$\begin{aligned}\rho &\approx \frac{F(1 - FP_b)}{F + A + 2c\tau_p} \\ &\approx \frac{(I + O)(1 - (I + O)P_b)}{I + 2O + 2c\tau_p}\end{aligned}$$

dato che $F = I + O$ e $A = O$

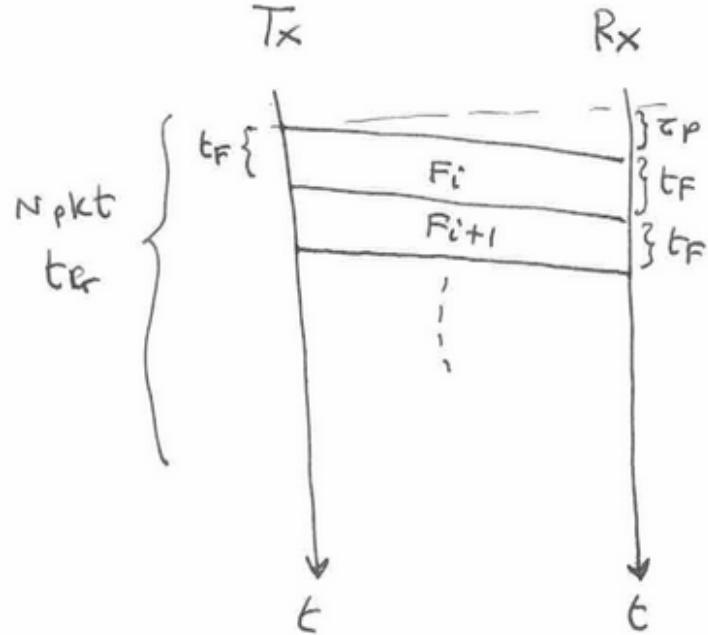
Efficienza

$$\begin{aligned}\eta &= \frac{t_I}{t_T} \\ &= \frac{t_I}{t_F} \frac{t_F}{t_T} \\ &= \frac{I}{I + O} \rho \\ &= \frac{I(1 - (I + O)P_b)}{I + 2O + 2c\tau_p}\end{aligned}$$



$$O \ll I, c\tau_p \ll I, P_b \ll I \Rightarrow I_{\text{opt}} \approx \sqrt{\frac{2(O + c\tau_p)}{P_b}}$$

3.2.3 Prestazioni strategia Go-back-N



- timeout t_0

$$\begin{aligned}
 t_0 &= (N - 1)t_F \\
 t_0 &\geq t_A + 2\tau_p \\
 (N - 1)t_F &= t_A + 2\tau_p \quad \text{timeout stringente}
 \end{aligned}$$

- tempo totale trasmissione di N pacchetti t_G

$$t_G = Nt_F = t_F + t_0 = t_F + t_A + 2\tau_p$$

- tempo di i ritrasmissioni t_i

$$t_i = iNt_F = it_G$$

- probabilitá di avere i ritrasmissioni P_i

$$P_i = p^i(1-p)$$

- tempo medio di ritrasmissione

$$\begin{aligned}
 \mathbb{E}\{t_{\text{ritrasmissione}}\} &= \sum_{i=1}^{+\infty} t_i P_i \\
 &= \sum_{i=1}^{+\infty} t_G (1-p) i p^i \\
 &= t_G (1-p) \sum_{i=1}^{+\infty} i p^i \\
 &= t_G (1-p) \frac{p}{(1-p)^2} \\
 &= \frac{t_G p}{1-p}
 \end{aligned}$$

- tempo totale medio di trasmissione t_T

$$\begin{aligned}
 t_T &= t_F + \mathbb{E}\{t_{\text{ritrasmissioni}}\} \\
 &= t_F + t_G \frac{p}{1-p} \\
 &= t_F + (t_F + t_A + 2\tau_p) \frac{p}{1-p} \\
 &= t_F + (t_F + t_0) \frac{p}{1-p} \\
 &= t_F \left(1 + \frac{p}{1-p}\right) + t_0 \frac{p}{1-p} \\
 &= t_F \frac{1}{1-p} + t_0 \frac{p}{1-p} \\
 &= \frac{t_F + t_0 p}{1-p} \\
 &= \frac{t_F}{1-p} \left[1 + \frac{(t_A + 2\tau_p)}{t_F} p\right] \\
 &= \frac{t_F}{1-p} \left[1 + \frac{A}{F} p + 2\alpha p\right] \\
 t_T &\xrightarrow{p \rightarrow 0} t_F \Rightarrow \rho \xrightarrow{p \rightarrow 0} 1
 \end{aligned}$$

- fattore di utilizzo ρ

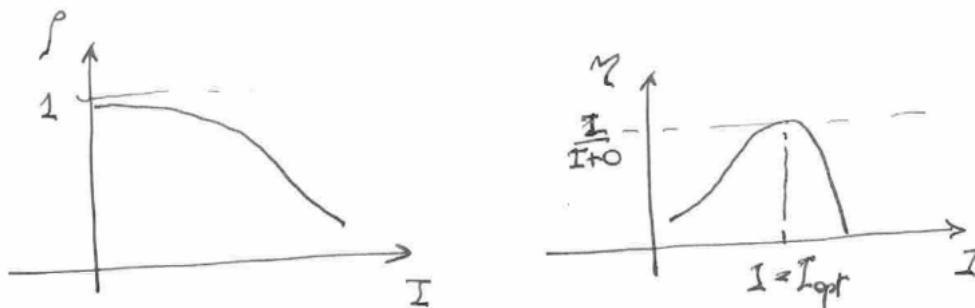
$$\begin{aligned}
 \rho &= \frac{t_F}{t_T} \\
 &= \frac{t_F(1-p)}{t_F + t_0 p} \\
 &\approx \frac{\frac{F}{c}(1-FP_b)}{\frac{F}{c} + FP_b(\frac{A}{c} + 2\tau_p)} \\
 &\approx \frac{F(1-FP_b)}{F + FP_b(A + 2c\tau_p)} \\
 &\approx \frac{1 - FP_b}{1 + P_b(O + 2c\tau_p)} \quad \text{ponendo } A=O \\
 &\approx \frac{1 - (I+O)P_b}{1 + (O + 2c\tau_p)P_b} \quad \text{ricordandoci che } F = I + O \\
 \rho &\xrightarrow{p \rightarrow 0} 1 \quad \text{non è limitato da ACK/NACK} \\
 \rho &\xrightarrow{t_0 \rightarrow 0} 1 - p \quad \text{limitato da errore sulla ricostruzione del pacchetto}
 \end{aligned}$$

- probabilità di errore sul pacchetto p (per P_b molto bassi)

$$p \approx FP_b$$

- velocità di trasmissione c [bit/s]
- efficienza η

$$\begin{aligned}
 \frac{t_I}{t_T} &= \frac{t_F}{t_T} \frac{t_I}{t_F} \\
 &= \rho \frac{I}{I+O}
 \end{aligned}$$



$$O \ll I \Rightarrow I_{opt} \approx \sqrt{\frac{O}{P_b}}$$

3.2.4 Prestazioni strategia Selective Repeat

Come Go-Back-N ma $t_i = i t_F$

- tempo totale medio di trasmissione t_T

$$\begin{aligned}
 t_T &= t_F + \sum_{i=0}^{+\infty} t_i P_i \\
 &= t_F + t_F (1-p) \sum_{i=0}^{+\infty} i p^i \\
 &= \frac{t_F}{1-p}
 \end{aligned}$$

- fattore di utilizzo ρ

$$\begin{aligned}
 \rho &= \frac{t_F}{t_T} = 1-p \\
 &\simeq 1 - FP_b \\
 &\simeq 1 - (I+O)P_b
 \end{aligned}$$

- efficienza η

$$\begin{aligned}
 \eta &= \frac{I}{I+O} \rho \\
 &= \frac{I}{I+O} [1 - (I+O)P_b] \\
 &= \frac{I}{I+O} - IP_b
 \end{aligned}$$



$$I_{\text{opt}} \simeq \sqrt{\frac{O}{P_b}}$$

3.3 Tecniche di accesso multiplo

Tecniche per condividere nel modo più efficiente possibile una risorsa comune (canale di trasmissione).

L'accesso può essere:

- deterministico:** l'utilizzo della risorsa è determinata a priori e ciascuna stazione sa quando accedervi
- controllato:** la stazione chiede il permesso di utilizzare la risorsa, attende il proprio turno e usa la risorsa in modo esclusivo

- **casuale**: ogni stazione accede alla risorsa indipendentemente da quello che fanno le altre stazioni

3.3.1 Elementi caratteristici

- **Throughput** S : in una sezione della rete, S è il rapporto fra la velocità di trasmissione riferita ai soli bit corretti e la velocità di trasmissione. Nei casi che considereremo $S = \rho$
- **Traffico offerto** G : in una sezione della rete, G è il rapporto fra il numero medio di bit/s che un terminale prova a trasmettere e la velocità di trasmissione
- **Tempo di accesso** t_{acc} : tempo medio fra l'arrivo di un pacchetto ad una stazione e la corretta ricezione al destinatario

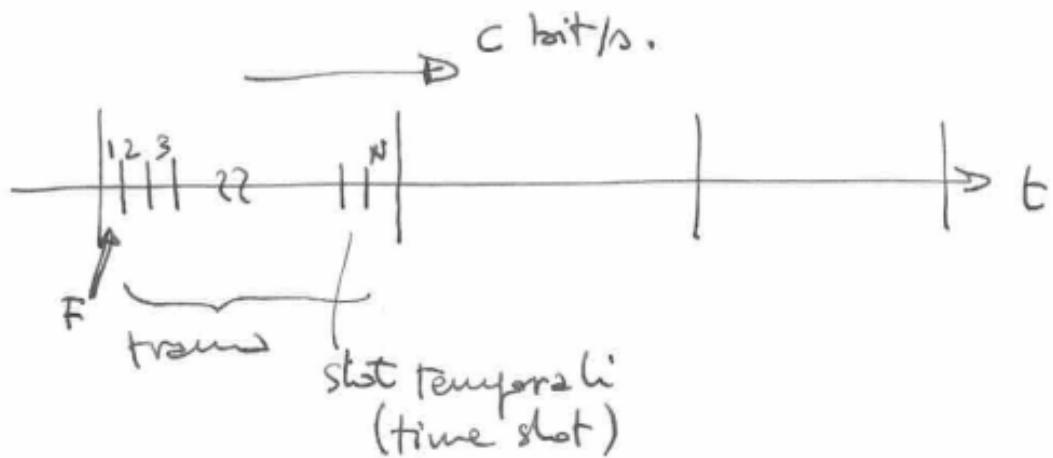
3.3.2 Assunzioni

- la probabilità di errore sul canale è trascurabile
- N stazioni che vogliono accedere alla risorsa condivisa
- C [bit/s]: velocità di trasmissione (incluse eventuali ritrasmissioni)
- la distribuzione dei tempi di interarrivo dei pacchetti è poissoniana con arrival rate λ pkt/s su singola stazione
- F : lunghezza dei pacchetti (determinata o aleatoria)

3.3.3 Accesso deterministico

Affinché ogni stazione possa utilizzare la risorsa senza andare a collidere con i segnali delle altre stazioni, si utilizzano **canali ortogonali**.

Time Division Multiple Access (TDMA)

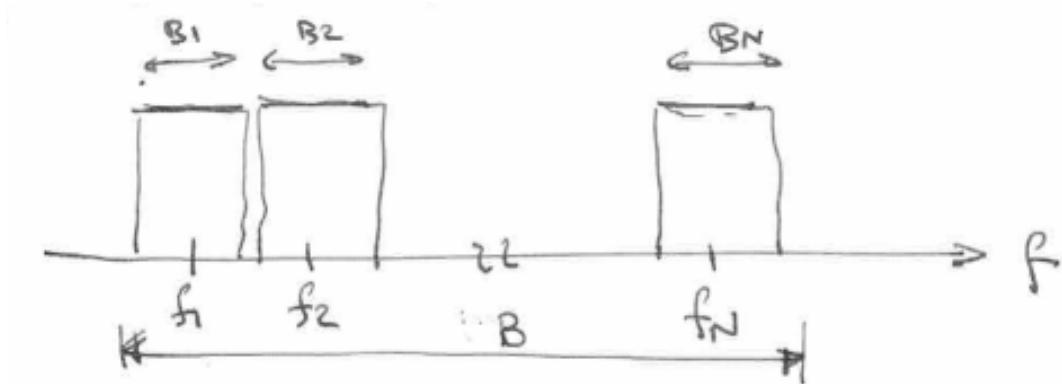


Con questa tecnica il tempo viene diviso in **trame** al cui interno si allocano periodicamente la trasmissione di ciascuna delle N stazioni, suddividendo ciascuna trama in N **time slot**, ognuno dei quali affida il canale ad una stazione permettendogli di

trasmettere un pacchetto lungo F bit. Il canale sostiene una velocità di trasmissione pari a c bit/s.

- velocità di trasmissione di una stazione: $\frac{C}{N}$ bit/s
- F bit/pkt
- tasso di servizio $\mu = \frac{C}{NF}$ [pkt/s]

Frequency Division Multiple Access (FDMA)



Con questa tecnica ogni stazione può utilizzare il canale per tutto l'arco temporale perché gli viene dedicata una particolare sottobanda di frequenza. La velocità di trasmissione è proporzionale alla larghezza della banda.

- larghezza della sottobanda $B_i = \frac{B}{N}$
- velocità di trasmissione di ogni stazione $\alpha \frac{C}{N}$ bit/s
- $\mu = \frac{C}{NF}$ pkt/s

Caratterizzazione sistemi TDMA e FDMA

- Throughput (TDMA e FDMA)

$$S = \rho = \frac{\lambda}{\mu} = \lambda \frac{NF}{C}$$

- Tempo di accesso (TDMA)

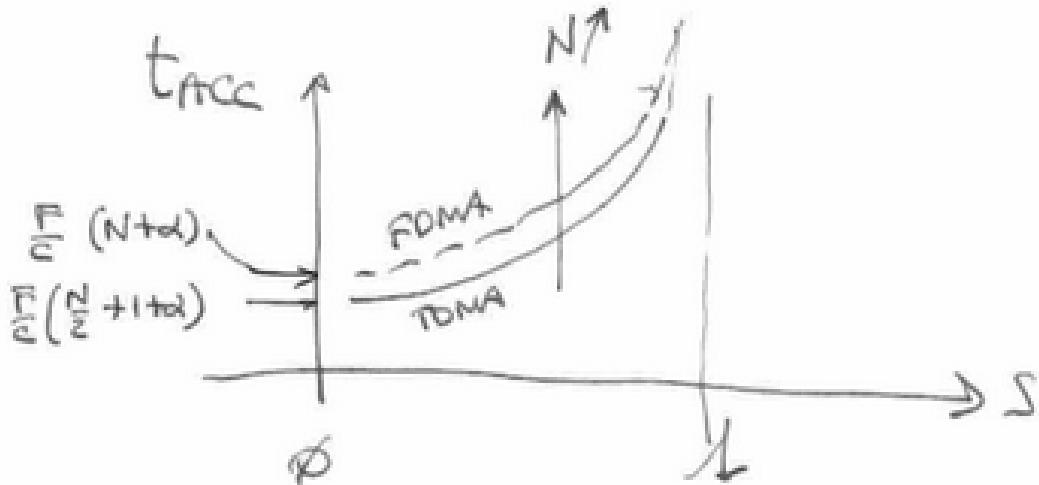
$$t_{\text{acc}} = W_q + W_{\text{slot}} + t_{x|c} + \tau_p$$

dove W_q è l'attesa in coda, W_{slot} è l'attesa dello slot, $t_{x|c}$ è il tempo di servizio condizionato alla piena disponibilità del canale e τ_p è il ritardo di propagazione. Svolgendo i calcoli:

$$t_{\text{acc}} = \frac{F}{C} \left[\frac{S}{1-S} \frac{N}{2} + \frac{N}{2} + 1 + \alpha \right]$$

- Tempo di accesso (FDMA). Dato che $t_x = \frac{N F}{C}$:

$$\begin{aligned} t_{\text{acc}} &= W_q + t_x + \tau_p \\ &= \frac{F}{C} \left[\frac{S}{1-S} \frac{N}{2} + N + \alpha \right] \end{aligned}$$



Sia per TDMA che per FDMA, il tempo di accesso è proporzionale ad N . Inoltre, il TDMA ha sempre prestazioni migliori del FDMA per le ipotesi fatte (traffico markoviano).

3.3.4 Accesso controllato

Il permesso di accedere alla risorsa comune viene concesso alle stazioni tramite il passaggio di un testimone (token).

La distribuzione del token può essere centralizzata (con entità centrale che assegna il token) o distribuita (le stesse stazioni si assegnano il token).

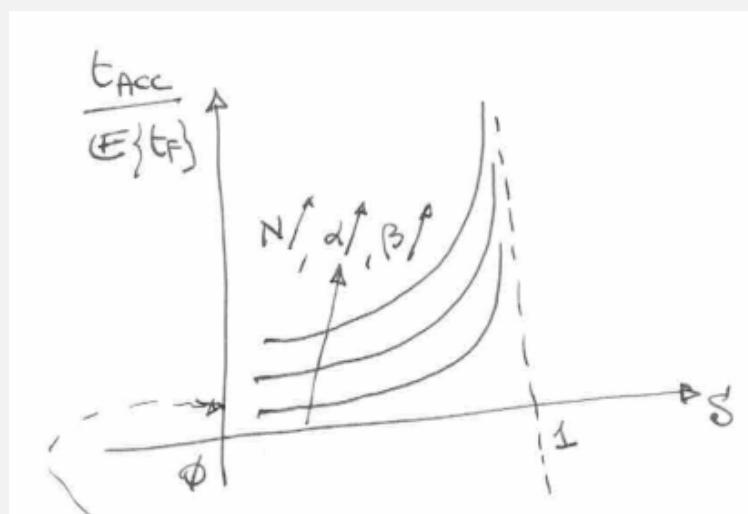
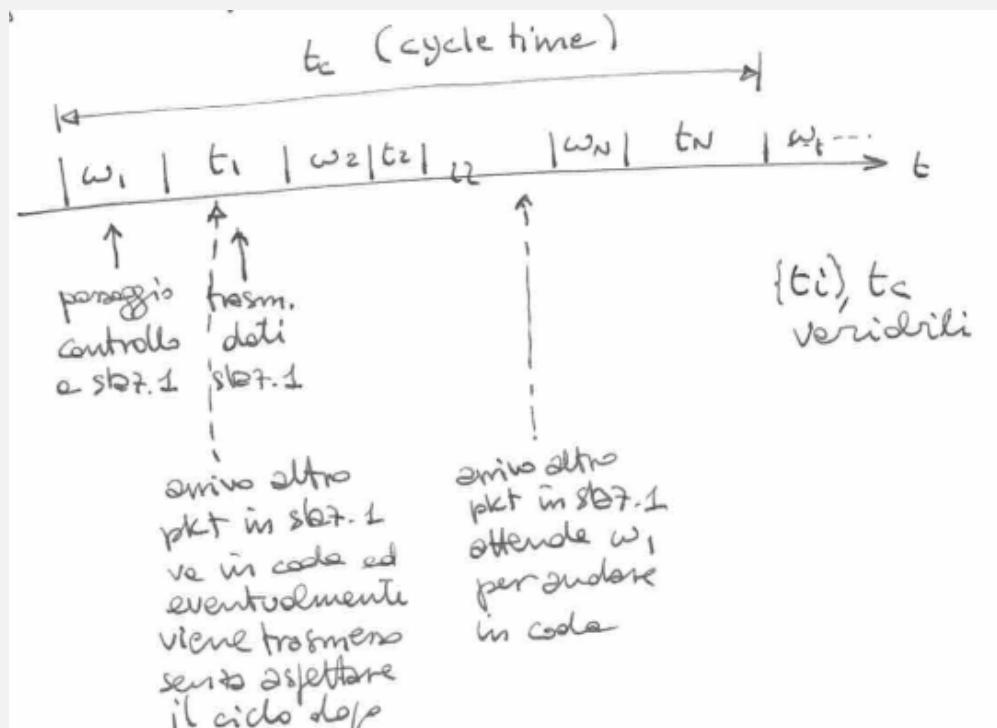
La sequenza dei vari accessi è detta **polling** e può variare dinamicamente sulla base del traffico alle stazioni.

Esempio (polling bilanciato fra le stazioni) L'occupazione della risorsa è dinamica e, al contrario del TDMA, ogni stazione mantiene il controllo per il tempo necessario, ciò significa che il tempo atteso da una stazione per prendere il token è variabile.

Con traffico uniforme:

$$S = \frac{N \lambda \mathbb{E}\{F\}}{C}$$

$$t_{\text{acc}} = W_q + \mathbb{E}\{t_F\} + \tau_p$$



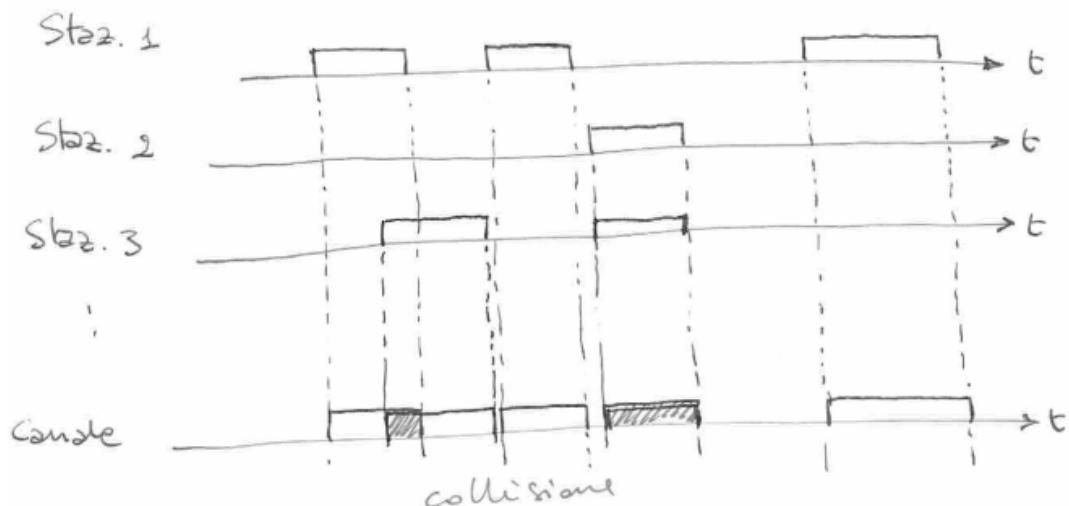
3.3.5 Accesso casuale

Le stazioni operano indipendentemente l'una dall'altra e trasmettono immediatamente i pacchetti. Utilizzato spesso come metodo per le applicazioni IoT. Questo semplifica molto accesso e gestione, se più utenti trasmettono contemporaneamente c'è collisione.

Il capostipite dei sistemi ad accesso casuale è **Aloha** che aveva lo scopo di collegare tramite satellite le università delle Hawaii.

Sistema Aloha puro

Quando una stazione ha un pacchetto da trasmettere, lo fa indipendentemente da quello che stanno facendo le altre stazioni (stato del canale).



Anche con un canale libero da errori, si possono comunque verificare errori dovuti a collisione (sovraposizione temporale di pacchetti trasmessi). A quel punto è opportuno prevedere un meccanismo di trasmissione (ACK, timeout) che introduca un ritardo diverso per ciascuna stazione, altrimenti si ripresenterebbe la stessa stazione shiftata nel tempo. Lo stato di sospensione prima della ritrasmissione è detto **back off**. Ci può essere comunque ancora collisione (instabilità).

Caratterizzazione sistema Aloha

Ipotesi:

- tutti i pacchetti della stessa lunghezza F
- lunghezza ACK << F (canale segnalazione ausiliario)
- N stazioni, C bit/s, λ arrival rate

Elementi caratteristici:

- Throughput:

$$S = \lambda \frac{NF}{C} = \lambda N t_F$$

- Traffico offerto:

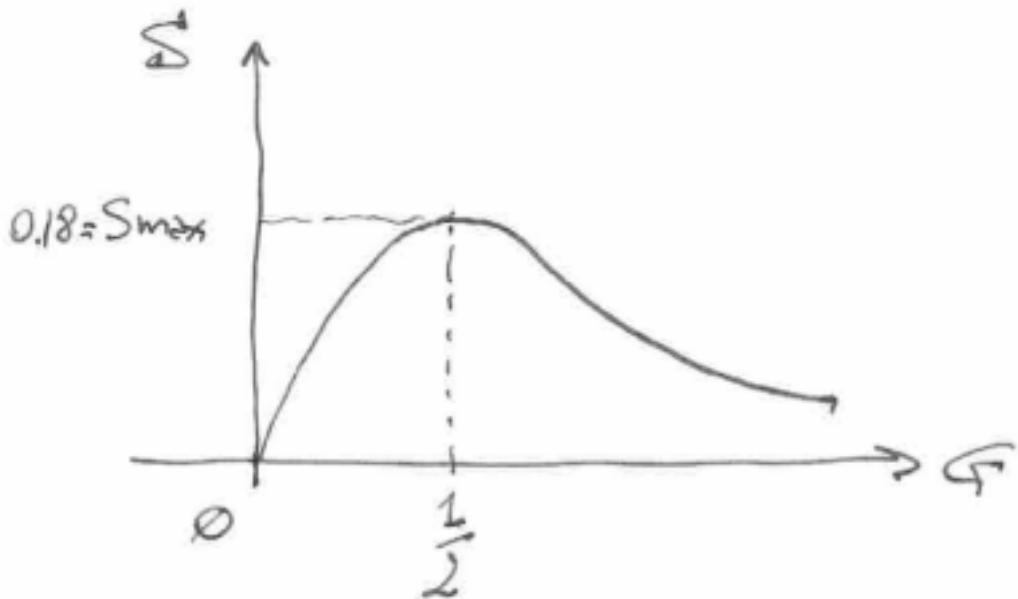
$$G = \lambda' \frac{NF}{C} = \lambda' N t_F$$

dove $\lambda' = \lambda + \underbrace{\lambda_R}_{\text{ritrasm.}} \geq \lambda$

A causa delle ritrasmissioni, il processo degli arrivi non è più poissoniano

Analisi:

- G può essere > 1
- $S \leq G$ e $S \leq 1$
- $\frac{S}{G} = \mathbb{P}\{\text{no collisioni}\} = \mathbb{P}\{\text{no arrivi in } 2t_F\} = P_0(2t_F) = e^{-\lambda' 2t_F} = e^{-2G}$
 - Dato che gli arrivi sono poissoniani: $P_k(t) = \frac{(\bar{\lambda}t)^k}{k!} e^{-\bar{\lambda}t}$
 - $\bar{\lambda} = N \lambda'$
 - e $G = \lambda' N t_F$
- $S = Ge^{-2G} \implies$ Espressione che governa l'Aloha puro (legame tra throughput e traffico offerto)

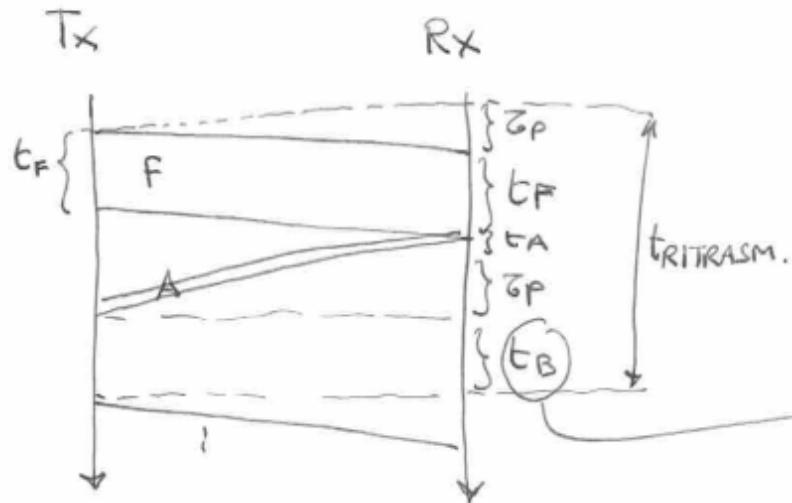


$$\frac{d}{dG} S(G) = 0 \implies G = \frac{1}{2}$$

$$S_{\max} = \frac{1}{2} e^{-2 \frac{1}{2}} \simeq 0.18$$

Al più si utilizza il canale al 18% della sua capacità.

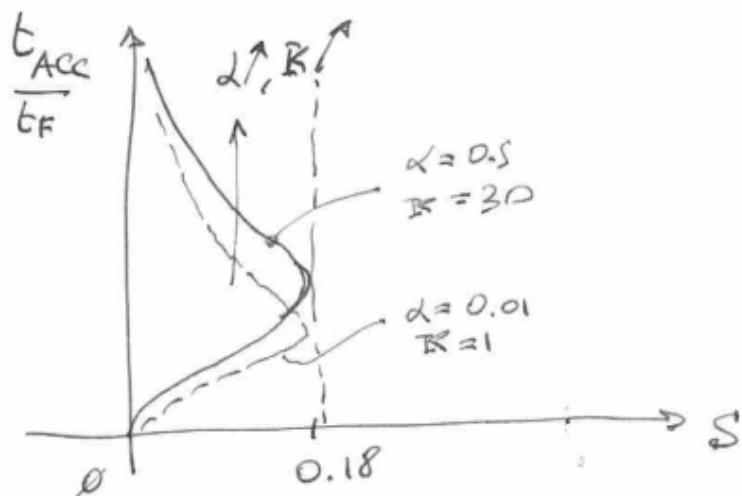
Effettuiamo di seguito l'analisi del tempo di accesso medio. L'analisi è semplificata perché trascuriamo la coda e consideriamo il traffico poissoniano nonostante la presenza di ritrasmissioni.



- $\frac{G}{S}$: numero medio di trasmissioni per consegnare correttamente un pacchetto al destinatario
- $\frac{G}{S} - 1$: numero medio di ritrasmissioni
- t_{acc} : numero di ritramissioni \cdot tempo medio di ritrasmissione + tempo ultima trasmissione corretta

$$\begin{aligned} t_{acc} &= \left(\frac{G}{S} - 1\right)(t_F + \tau_p + t_A + \tau_p + \mathbb{E}\{t_B\}) + t_F + \tau_p \\ &= t_F \left[(e^{2G} - 1) \left(1 + 2\alpha + \frac{K}{2} + 1 + \alpha \right) \right] \end{aligned}$$

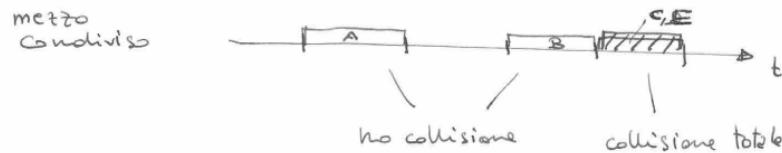
dove t_B é aleatorio e $t_A \ll t_F$



Sistema Slotted Aloha

Le prestazioni del sistema Aloha puro sono limitate dalla durata dell'intervallo di vulnerabilità $2t_F$. Minore é tale durata e migliori sono le prestazioni. Questo ha portato alla versione con slot di Aloha.

Il sistema slotted Aloha consiste nel suddividere l'asse dei tempi in slot di durata pari a t_F (tempo di trasmissione singola per pacchetto). Ogni stazione inizia a trasmettere solo all'inizio di uno slot \Rightarrow no collisioni parziali (cioè nessuna collisione o collisione totale). Si dimezza pertanto il tempo di vulnerabilità che diventa t_F (in Aloha puro era $2t_F$).



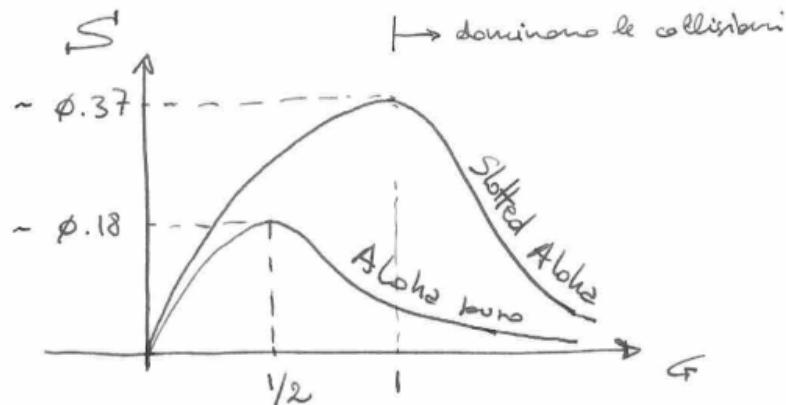
Caratterizzazione Slotted Aloha

Consideriamo le stesse ipotesi dell'Aloha puro:

- tutti i pacchetti della stessa lunghezza F
- lunghezza ACK $\ll F$
- N stazioni, C bit/s, λ arrival rate

Elementi caratteristici

- ritmo degli arrivi (poissoniani): $\lambda' = \lambda + \lambda_R$
- $\frac{S}{G} = \mathbb{P}\{\text{no collisioni}\} = \mathbb{P}\{\text{no arrivi in } t_F\} = P_0(t_F) = e^{-\lambda' N t_F} = e^{-G}$
- $S = G e^{-G} \Rightarrow$ Espressione che governa lo Slotted Aloha (legame tra throughput e traffico offerto)



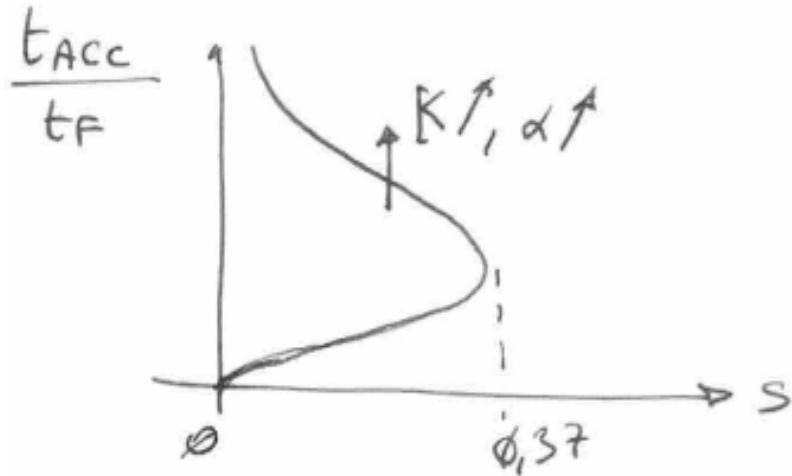
$$\frac{d}{dG} S = 0 \iff G = 1$$

$$S_{\max} = \frac{1}{e} \approx 0.37$$

Per quanto riguarda il tempo medio di accesso, risulta:

$$\begin{aligned} t_{\text{acc}} &= \left(\frac{G}{S} - 1\right)(t_F + 2\tau_p + \mathbb{E}\{t_B\}) + t_F + \frac{t_F}{2} + \tau_p \\ &= t_F \left[(e^G - 1) \left(1 + 2\alpha + \frac{K}{2} \right) + \frac{3}{2} + \alpha \right] \end{aligned}$$

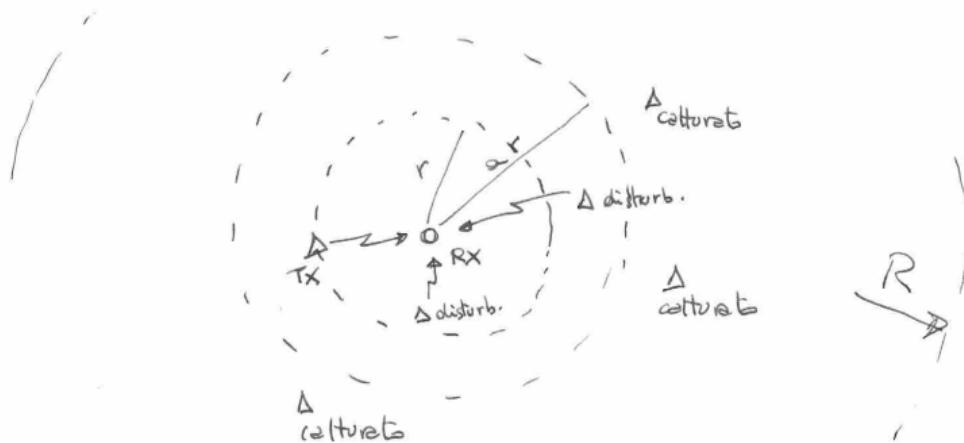
dove dove $\frac{t_F}{2}$ è l'attesa media dello slot successivo.



Considerazione: stazioni che emettono da lontano disturbano poco e non necessariamente distruggono un pacchetto con cui collidono.

Effetto cattura

I pacchetti trasmessi da una stazione a distanza r vengono ricevuti correttamente se in un tempo di vulnerabilità (t_F o $2t_F$) non trasmette un'altra stazione a distanza non superiore di ar (con a parametro di cattura).



Se il traffico offerto non dipende dalla distanza $G(r) = G_0$, allora il throughput medio per stazioni in una circonferenza di raggio R è:

$$S_R = \frac{1}{2a^2} \left(1 - e^{-2\pi G_0 a^2 R^2} \right) \quad \text{Aloha puro con cattura}$$

$$S_\infty = \frac{1}{2a^2}$$

Esiste un caso detto **cattura perfetta** dove $a = 1$, cioè tutti i trasmettitori più distanti di un certo trasmettitore non creano al ricevitore collisioni disastrose. Il suo throughput $S_\infty = 0.5$ è maggiore del throughput dell'Aloha puro (0.18). Il

parametro di cattura a è legato alla prestazione di livello fisico (Bit Error Probability, BEP).

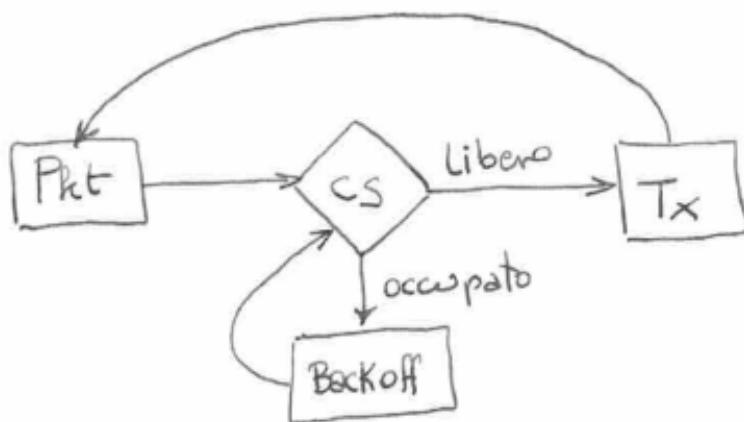
Carrier Sense Multiple Access (CSMA)

Rispetto al sistema Aloha, c'è un coordinamento fra le stazioni nel momento in cui decidano di trasmettere, pur rimanendo casuale l'accesso; tale coordinamento avviene mediante un ascolto del canale (**carrier sensing**) prima di trasmettere. Se durante l'ascolto il canale risulta già occupato, non si trasmette. A causa dei ritardi di propagazione è comunque possibile che si verifichino collisioni.

Esistono diversi tipi di CSMA:

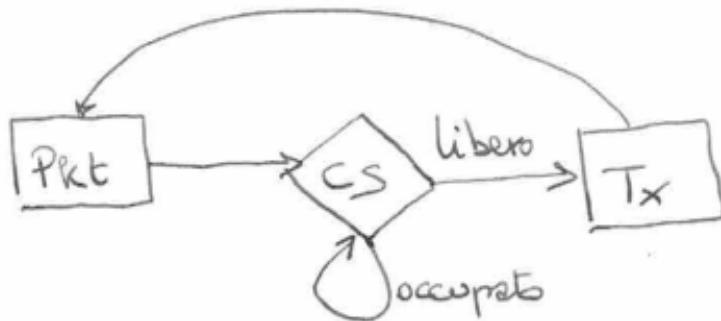
- **CSMA non persistente:**

- Arriva un pacchetto
- Si fa il carrier sensing del canale
 - * canale libero: si trasmette subito
 - * canale occupato: si mette in backoff (si attende un certo tempo aleatorio) e si rifà il carrier sensing
- Ci si rimette in attesa di un altro pacchetto



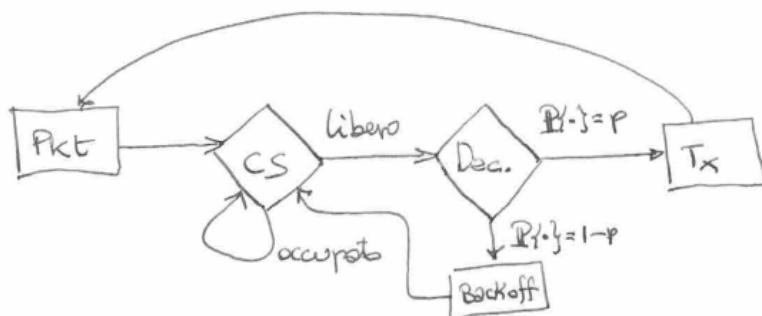
- **CSMA 1-persistente**

- Arriva un pacchetto
- Si fa il carrier sensing del canale
 - * canale libero: si trasmette subito
 - * canale occupato: si persiste nel fare il carrier sensing del canale (no backoff)
- Ci si rimette in attesa di un altro pacchetto

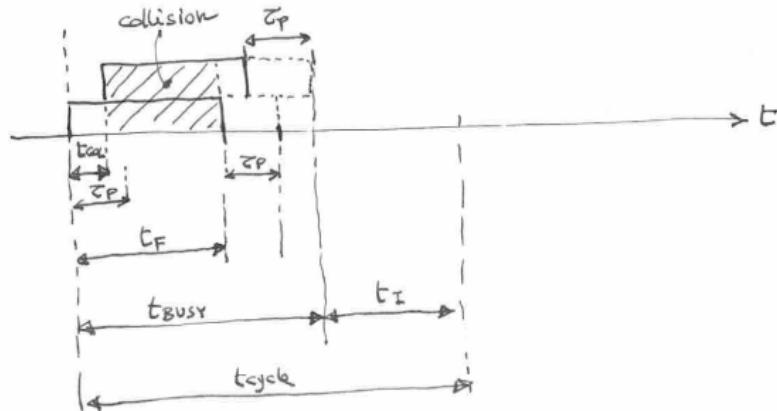


- CSMA p -persistente

- Arriva un pacchetto
- Si fa il carrier sensing del canale
 - * canale libero: si passa in decodifica
 - * canale occupato: si persiste nel fare il carrier sensing del canale (no backoff)
- blocco decodifica
 - * con probabilitá p si trasmette
 - * con probabilitá $1 - p$ ci si mette in backoff e si rifá il carrier sensing
- Ci si rimette in attesa di un altro pacchetto



Caratterizzazione CSMA non persistente



Elementi caratteristici

- t_{col} : tempo di inizio collisione (in caso di collisioni multiple si fa riferimento all'ultima)
- $t_{vul} = \tau_p$: tempo di vulnerabilità (intervallo temporale in cui una stazione ha emesso un pacchetto ma non viene ancora sentito da un'altra stazione che vuole trasmettere)
- t_{busy} : tempo medio in cui il sistema è occupato (trasmissione corretta o non collisioni)

$$t_{busy} = \mathbb{E}\{t_{col}\} + t_F + \tau_p$$

- t_u : tempo utile in cui il canale è usato senza conflitto

$$\begin{aligned} t_u &= t_F \mathbb{P}\{\text{no transmissions in } t_{vul}\} \\ &= t_F P_0(\tau_p) \\ &= t_F e^{-N\lambda'\tau_p} \\ &= t_F e^{-N\lambda' t_F \frac{\tau_p}{t_F}} \\ &= t_F e^{-G\alpha} \end{aligned}$$

- t_I : tempo di interarrivo in cui il canale è libero

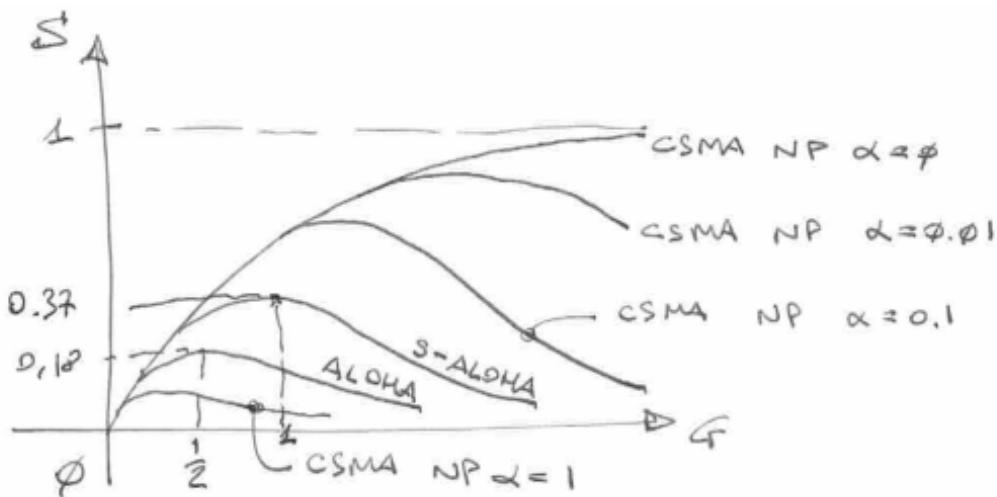
$$t_I = \frac{1}{N\lambda'} = \frac{t_F}{G}$$

- Throughput

$$S = \frac{t_u}{t_{\text{cycle}}} = \frac{t_u}{t_{\text{busy}} + t_I}$$

$$\frac{Ge^{-G\alpha}}{G(1 + 2\alpha) + e^{-G\alpha}}$$

$$S \xrightarrow{\alpha \rightarrow 0} \frac{G}{1+G} \xrightarrow{G \rightarrow +\infty} 1$$

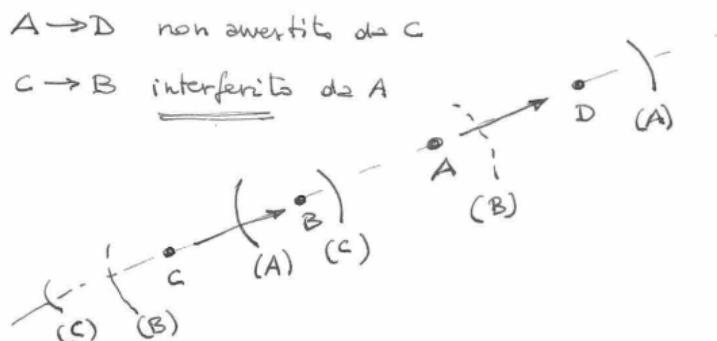


Anche il CSMA prevede una versione slotted che migliora le prestazioni, ma richiede la sincronizzazione delle stazioni.

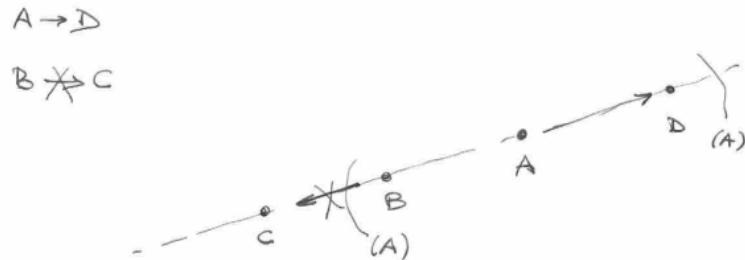
Laddove le potenze trasmesse variano da stazione a stazione oppure il canale di trasmissione ha comportamenti aleatori (es.: mezzo radio, reti wireless) sussiste il tema del **terminale nascosto** e quello del **terminale esposto**

Terminale nascosto e esposto

Nell'esempio in figura, C non sente A perché è al di fuori della sua area di copertura, perciò trasmette a B, che però è nella area di copertura di A, quindi la trasmissione da C a B è interferita da A. A è in questo esempio il **terminale nascosto**.



Nell'esempio in figura, La trasmissione da B a C non é interferita da A, peró B sente A dato che é all'interno della sua area di copertura, perciò non trasmette. In questo esempio A é il **terminale esposto**.



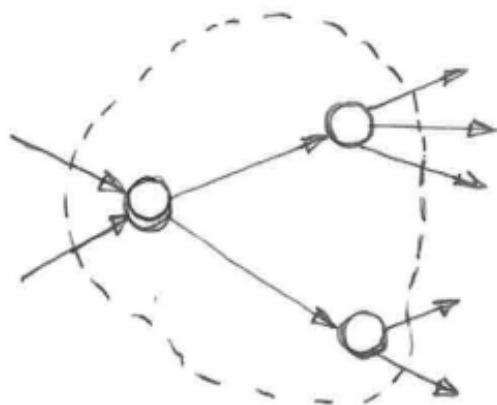
3.3.6 Conclusioni

- I protocolli deterministici di accesso multiplo dipendono poco dal ritardo di propagazione. Di contro, i protocolli e il CSMA dipendono molto da ritardo di propagazione
- I protocolli casuali sono molto pronti per basso carico della rete
- Fattori decisivi: topologia rete, traffico, tecniche trasmissive, ritardo propagazione
- Non esiste un sistema migliore in tutte le situazioni: la scelta del protocollo dipende dal scenario medio, dal scenario critico e dalla complessità sostenibile

Chapter 4

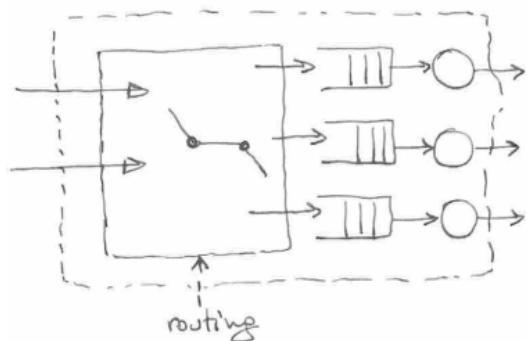
Network - Strato 3

L'obiettivo dello strato 3 è quello di consegnare i pacchetti attraverso una rete costituita da una serie di nodi che vengono opportunamente connessi tra di loro (**routing**). Un nodo di una rete è un insieme di sistemi a coda con più ingressi e più uscite: il compito dello strato 3 è quello di prendere i pacchetti che arrivano da un certo ingresso, direzionarli verso l'uscita giusta, eventualmente facendolo attendere in coda.



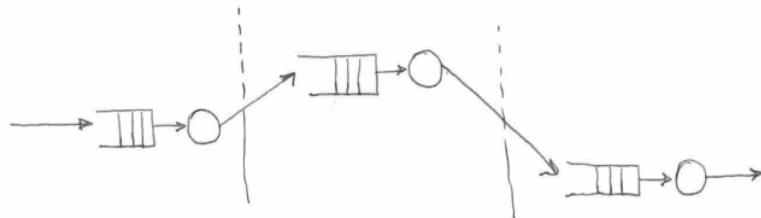
4.1 Reti di code

Una rete a commutazione di pacchetto è formata da molteplici store & forward configurati dinamicamente sulla base delle richieste degli utenti. Per ogni linea di uscita del nodo c'è un servitore e per ogni linea di entrata c'è la selezione dalla coda.



Stiamo trattando una cascata di sistemi a coda con:

- N nodi
- o_i uscite del nodo i
- $M = \sum_{i=1}^N o_i$ sistemi coda-servitore

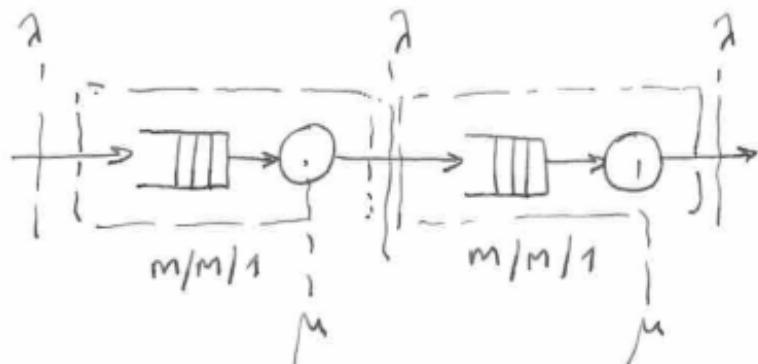


Un sistema composto da più sistemi coda-servitore risulta particolarmente complesso a causa della forte correlazione fra tempi di interarrivo e lunghezza dei pacchetti dopo il passaggio per un servitore. Per un dimensionamento di massima si fa un'assunzione forte.

4.1.1 Assunzione di indipendenza di Kleinrock

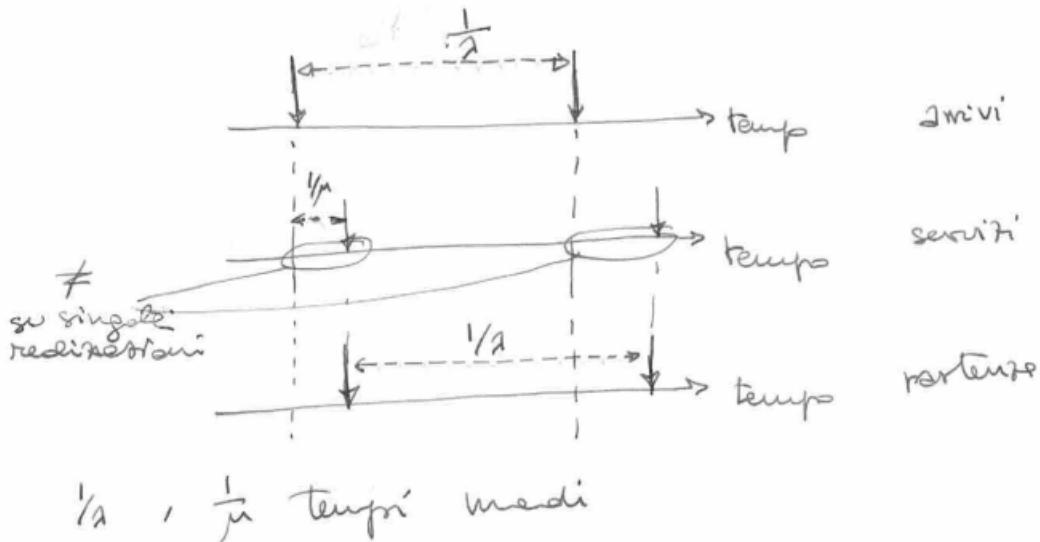
Se ad un nodo arriva un elevato numero di flussi da altri nodi, si considera l'indipendenza fra i tempi di interarrivo e la lunghezza dei pacchetti. Inoltre, l'elevato numero di flussi in ingresso introduce l'indipendenza e l'assenza di memoria fra arrivi successivi \Rightarrow distribuzione dei tempi di interarrivo poissoniana.

Il generico nodo i è composto da o_i sistemi a coda di tipo M/M/1.



4.1.2 Teorema di Burke

In un sistema M/M/1 con tasso di arrivo λ il processo di partenza è ancora di Poisson con tasso λ .



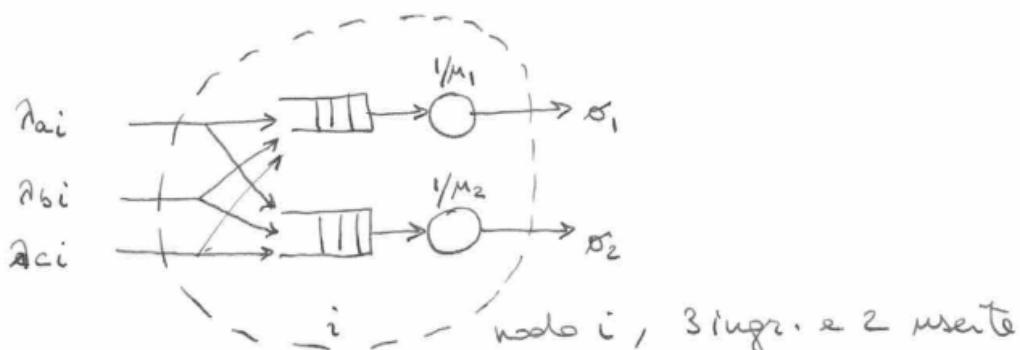
4.1.3 Teorema di Jackson

All'equilibrio, la probabilità che il sistema (rete di code) sia nello stato n è data dal prodotto delle probabilità di stato di ogni sistema a coda che lo compone.

$$\underline{n} = (n_1, n_2, \dots, n_M)$$

$$\mathbb{P}\{\underline{n}\} = P_1(n_1) P_2(n_2) \dots P_n(n_M)$$

dove n_i rappresenta il numero di pacchetti nell' i -esimo sistema a coda e $P_i(n_i)$ rappresenta la probabilità che all'equilibrio l' i -esimo sistema a coda abbia n_i pacchetti.



4.1.4 Parametri caratteristici di una rete di code

- Numero medio di pacchetti nella rete di code

$$\begin{aligned}
 L_s &= \sum_{i=1}^M \mathbb{E}\{n_i\} \\
 &= \sum_{i=1}^M \frac{\rho_i}{1 - \rho_i} \quad \text{dove } \rho_i = \frac{\lambda_i}{\mu_i} \\
 &= \sum_{i=1}^M \frac{\lambda_i}{\mu_i - \lambda_i}
 \end{aligned}$$

- Tempo medio speso da un pacchetto nella rete (non considerando i ritardi di propagazione τ_i)

$$W_s = \frac{1}{\lambda_s} \sum_{i=1}^M \frac{\lambda_i}{\mu_i - \lambda_i}$$

4.2 Algoritmi di instradamento (Routing)

Ogni nodo della rete ha un meccanismo in grado di decidere su quale sistema a coda deve essere inviato il pacchetto in transito al fine di raggiungere efficientemente il destinatario.

Il modo con cui questa decisione è presa si chiama **politica di instradamento** che è realizzata tramite un **algoritmo di instradamento** che può essere:

- centralizzato o distribuito
 - centralizzato: un'unica entità decide per tutti i nodi \Rightarrow questa deve conoscere lo stato dell'intera rete
 - distribuito: ogni nodo è paritario e decide come instradare verso i vicini sulla base di informazioni locali
- statico o dinamico
 - statico: ogni nodo ha tabelle ingresso-uscita definite a priori
 - l'instradamento si adatta al traffico \Rightarrow si cerca di evitare la congestione di parte della rete
- proattivo o reattivo
 - proattivo: l'instradamento viene creato non appena si instaura il collegamento
 - reattivo: l'instradamento viene creato al momento del bisogno
- uniforme o non uniforme: i nodi potrebbero avere ruoli diversi (es.: WSN con cluster di nodi e cluster head) e la segnalazione per il routing dipende dal ruolo

A seconda del tipo di instradamento c'è un diverso compromesso fra segnalazione e tempo di consegna dei pacchetti.

Gli algoritmi di instradamento risolvono problemi di ottimizzazione con funzioni di costo legate a:

- numero di collegamenti
- costo fisico dei percorsi
- ritardo complessivo o locale
- throughput complessivo o locale

4.2.1 Algoritmo di Dijkstra

Risolve il problema della ricerca del percorso minimo (shortest path). Ne vedremo una versione centralizzata (1959). Definiamo:

- $D(v)$: distanza / costo di v dalla sorgente s
- $l(i, j)$: distanza / costo del percorso $i \rightarrow j$ (∞ se non esiste un percorso)
- F : insieme di nodi selezionati ad una iterazione

Step dell'algoritmo:

i) inizializzazione

$$\begin{aligned} F &= \{S\}, \quad \forall v \notin F \\ D(v) &= l(S, v) \end{aligned}$$

ii) fra i nodi che non siano già stati selezionati, scelta di quello con distanza minore

$$\begin{aligned} \omega &\notin F \mid \forall z \notin F, z \neq \omega \\ D(\omega) &\leq D(z) \end{aligned}$$

iii) selezione del nodo scelto

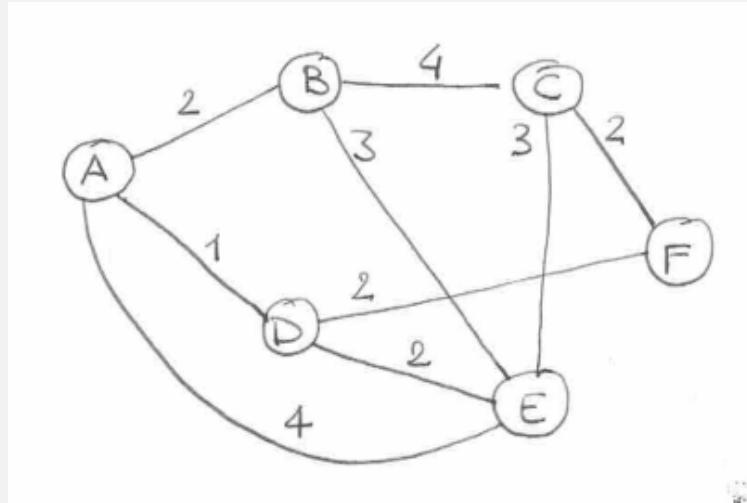
$$F = F \cup \{\omega\}$$

iv) calcolo nuove distanze considerato ω

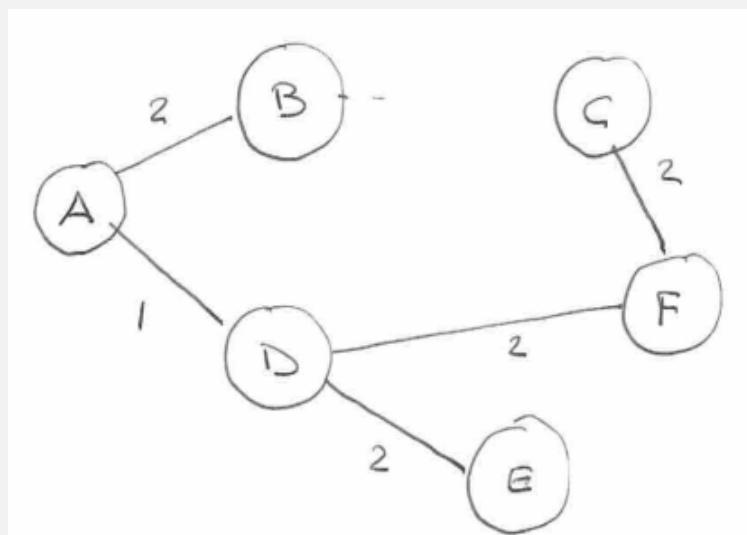
$$D(z) = \min\{D(z), d(\omega) + l(\omega, z)\} \quad \forall z \notin F$$

v) Se $|F| < M$ allora vai a (ii) altrimenti stop

Esempio: sorgente $s = A$



F	$D(B)$	$D(C)$	$D(D)$	$D(E)$	$D(F)$
{A}	2 (A-B)	∞	1 (A-D)	4 (A-E)	∞
{A, D}	2 (A-B)	∞	1 (A-D)	3 (A-D-E)	3 (A-D-F)
{A, D, B}	2 (A-B)	6 (A-B-C)	1 (A-D)	3 (A-D-E)	3 (A-D-F)
{A, D, B, E}	2 (A-B)	6 (A-B-C)	1 (A-D)	3 (A-D-E)	3 (A-D-F)
{A, D, B, E, F}	2 (A-B)	5 (A-D-F-C)	1 (A-D)	3 (A-D-E)	3 (A-D-F)
{A, D, B, E, F, C}	2 (A-B)	5 (A-D-F-C)	1 (A-D)	3 (A-D-E)	3 (A-D-F)



I percorsi sono a costo minimo, ma

- c'è un solo percorso sorgente-destinazione il che limita il throughput in caso di traffico sbilanciato

- difficoltà di adattamento a condizioni dinamiche del traffico

Nota: dall'algoritmo di Dijkstra deriva l'algoritmo di Viterbi per la decodifica dei codici convulzionali.

4.2.2 Algoritmo Bellman-Ford

È un algoritmo centralizzato proposto nel 1957. Contrariamente a Dijkstra non fissa un nodo di partenza, bensì quello di destinazione. Definiamo:

- d : nodo di destinazione
- h : indica l'iterazione
- D_i^h : distanza da nodo i a destinazione fissata al passo h

Step dell'algoritmo

i) inizializzazione

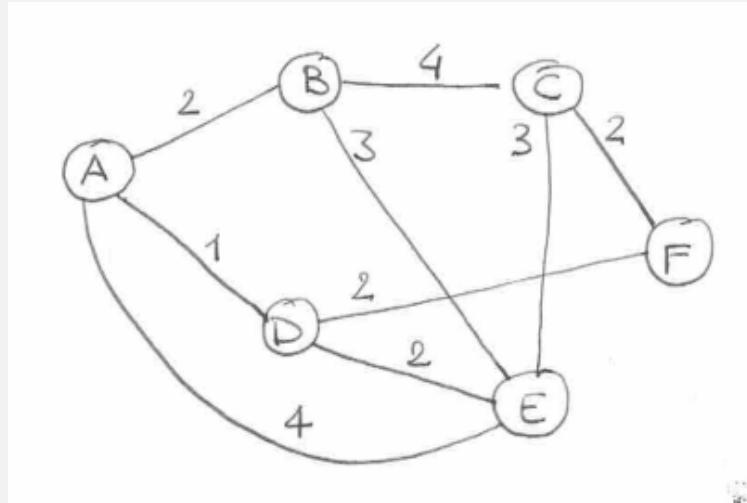
$$\begin{aligned} D_d^h &= 0 \quad \forall h \\ D_i^0 &= \infty \quad \forall i \neq d \end{aligned}$$

ii) selezione del nodo i nel percorso verso la destinazione da k

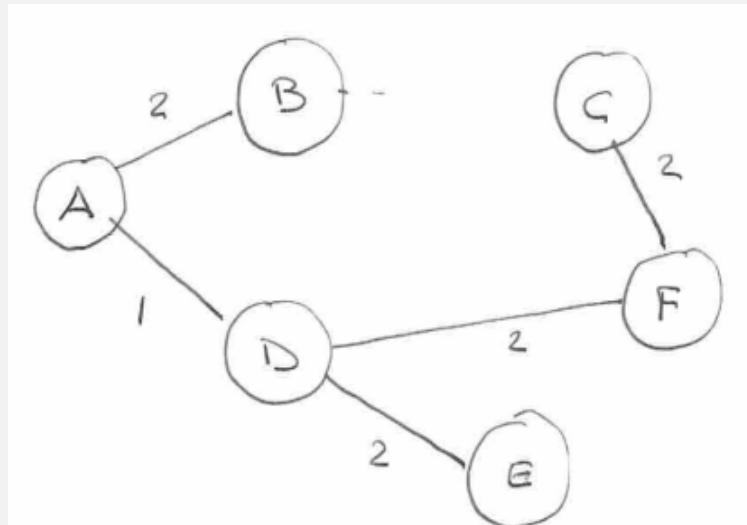
$$\begin{aligned} D_k^{h+1} &= \min\{l(i, j) + D_j^h, D_k^h\} \quad \forall k \neq d \\ i &= \operatorname{argmin} D_k^{h+1} \end{aligned}$$

iii) se $D_i^{h+1} = D_i^h \quad \forall i$ stop, altrimenti vai a (ii)

Esempio: destinazione $d = A$



h (passo)	D_A^h	D_B^h	D_C^h	D_D^h	D_E^h	D_F^h
0	0	∞	∞	∞	∞	∞
1	0	2 (B-A)	∞ (C-A)	1 (D-A)	4 (E-A)	∞ (F-A)
2	0	2 (B-A)	6 (C-B-A)	1 (D-A)	3 (E-D-A)	3 (F-D-A)
3	0	2 (B-A)	5 (C-F-D-A)	1 (D-A)	3 (E-D-A)	3 (F-D-A)
4	0	2 (B-A)	5 (C-F-D-A)	1 (D-A)	3 (E-D-A)	3 (F-D-A)



4.2.3 Algoritmo Distance Vector

È uno degli algoritmi proattivi distribuiti più popolari. Rappresenta una versione distribuita di Bellman-Ford.

Ogni nodo mantiene una tabella di instradamento che per ogni destinazione memorizza:

- l'indirizzo del destinatario
- l'indirizzo del prossimo nodo (next hop) a cui inviare il pacchetto verso il destinatario
- il costo dell'intero collegamento per raggiungere la destinazione
- perciò si ha un vettore di distanze (costi) verso i possibili nodi di destinazione

Step dell'algoritmo

- i) I nodi connessi direttamente si scambiano i distance vector

$$\text{DV}^{(i)} \quad \text{distance vector del nodo } i$$

- ii) Quando un nodo i riceve il $\text{DV}^{(i)}$ del nodo j verifica se per ogni possibile destinazione k si ha

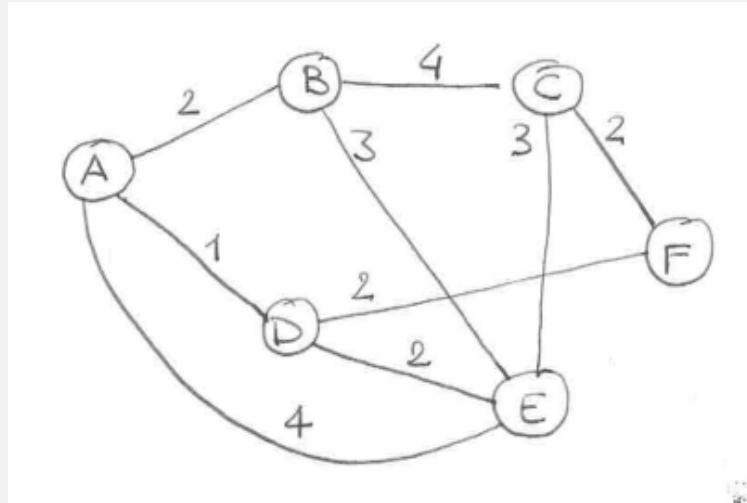
$$l(i, j) + \text{DV}_k^{(i)} < \text{DV}_k^{(j)}$$

Se ciò è vero, allora i aggiorna la sua tabella di instradamento verso la k con

$$\text{DV}_k^{(i)} = \text{DV}_k^{(j)} + l(i, j)$$

e come "next hop" passa da j per arrivare a k ; altrimenti distanza e next hop rimangono inalterati

Esempio: $i = A$, vicini $j = B, D, E$



k	$DV^{(B)}$	$DV^{(D)}$	$DV^{(E)}$
A	2	1	4
B	0	∞	3
C	4	∞	3
D	∞	0	2
E	3	2	0
F	∞	2	∞

Distance vector

Situazione iniziale		
k	$DV^{(A)}$	$NH^{(A)}$
A	0	A
B	2	B
C	∞	/
D	1	D
E	4	E
F	∞	/

Entra $DV^{(B)}$		
k	$DV^{(A)}$	$NH^{(A)}$
A	0	A
B	2	B
C	6	B
D	1	D
E	4	E
F	∞	/

Entra $DV^{(D)}$		
k	$DV^{(A)}$	$NH^{(A)}$
A	0	A
B	2	B
C	6	B
D	1	D
E	3	D
F	3	D

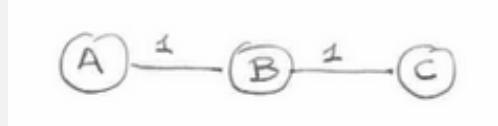
Entra $DV^{(E)}$		
k	$DV^{(A)}$	$NH^{(A)}$
A	0	A
B	2	B
C	6	B
D	1	D
E	3	D
F	3	D

In maniera automatica e distribuita si costruiscono la tabella di routing per ogni nodo verso gli altri nodi. È necessario bilanciare la periodicità con cui si trasmette il

DV: compromesso fra dinamicità dell'instradamento e l'overhead di segnalazione.

Un problema del DV é il **count-to-infinity**

Esempio: 3 nodi in cascata



- iniziale:

$$\begin{aligned} - \text{DV}_C^{(A)} &= 2, & \text{NH}_C^{(A)} &= B \\ - \text{DV}_c^{(B)} &= 1, & \text{NH}_c^{(B)} &= C \end{aligned}$$

- Supponiamo si rompa/cessi il collegamento B-C:

$$\begin{aligned} - \text{DV}_C^{(A)} &= 2, & \text{NH}_C^{(A)} &= B \\ - \text{DV}_c^{(B)} &= \infty, & \text{NH}_c^{(B)} &= / \end{aligned}$$

- I scambio

$$\begin{aligned} - \text{DV}_C^{(A)} &= \infty, & \text{NH}_C^{(A)} &= / \\ - \text{DV}_c^{(B)} &= \text{DV}_C^{(A)} + l(B, A) = 3, & \text{NH}_c^{(B)} &= A \end{aligned}$$

- II scambio

$$\begin{aligned} - \text{DV}_C^{(A)} &= \text{DV}_C^{(B)} + l(B, A) = 4, & \text{NH}_C^{(A)} &= B \\ - \text{DV}_c^{(B)} &= \infty, & \text{NH}_c^{(B)} &= / \end{aligned}$$

- ...

4.2.4 Algoritmo Link-State

- Migliora il DV eliminando il problema del count-to-infinity
- La topologia della rete é disseminata periodicamente mediante un pacchetto LSP (link-state packet)
- Ogni LSP contiene l'indirizzo del nodo, quello dei nodi vicini e il costo della linea verso i nodi vicini
- L'LSP viene propagato in maniera controllata (controlled flooding) al fine di inviarlo a tutte le linee di uscite tranne la linea verso il nodo da cui é stato ricevuto
- Ogni LSP contiene un numero di sequenza progressivo per poter verificare quale é piú aggiornato e dopo questo considerare l'informazione sullo stato dei collegamenti troppo vecchio (max age)

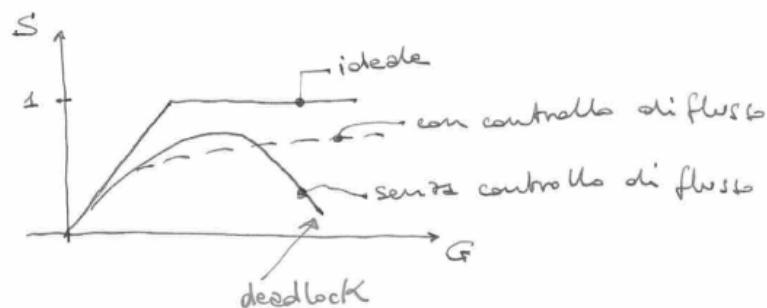
- Se un nodo si rompe, allora manca la propagazione dell'informazione. Per questo i vicini vengono scoperti tramite un pacchetto di "hello" che periodicamente un nodo invia ai propri vicini (1 hop) da dalla propria linea di uscita. "hello" contiene la lista dei vicini.
- Chi riceve "hello" verifica se il proprio ID è presente nella lista e in tal caso si appresta ad ottenere da lui lo stato del collegamento tramite LSP. Se non è presente nella lista memorizza il vicino per potergli trasmettere successivamente lo stato del collegamento via LSP.
- Esistono altri algoritmi di routing
 - ad-hoc on-demand distance vector (AODV) (reattivo)
 - hot potato

Chapter 5

Controllo di flusso

Il controllo di flusso risolve problemi quali i seguenti:

- quando il traffico offerto è elevato ma porzione di rete potrebbe non sostenerlo
- quando i buffer delle code hanno lunghezza limitata e sono pieni i pacchetti vengono scartati
- le ritrasmissioni aumentano ulteriormente il traffico offerto: cala il throughput e aumenta il ritardo (*congestione*)



Il controllo di flusso deve:

- risolvere la degradazione delle prestazioni all'aumentare del traffico offerto
- garantire equità di servizio in congestione
- armonizzare le velocità di trasmissione fra le varie parti della rete

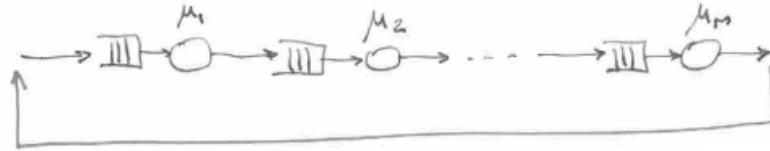
Esistono due tipi di controlli di flusso:

- **end-to-end**: a livello di trasporto; molto usato soprattutto nelle comunicazioni a circuito virtuale
- **hop-to-hop** (o back pressure): a livello di collegamento o di rete, si propaga sui collegamenti a monte

5.1 Finestra mobile

È un controllo di flusso end-to-end. Si impone che il numero di pacchetti su una singola comunicazione sia fissato a priori e pari alla lunghezza della finestra H . I pacchetti sono numerati da 1 a H e H rappresenta il massimo numero di pacchetti pendenti nella rete appartenenti al circuito.

L'analisi della finestra mobile è complessa ma può essere semplificata considerando il **sistema a code chiuse**.



Questa condizione è estrema per il controllo di flusso perché fà sì che ci sia in continuazione un pacchetto in ingresso essendo una coda chiusa (numero di pacchetti costante e massimo sul circuito). Ipotizziamo che:

- non ci siano arrivi dall'esterno $I_{si} = 0 \quad \forall i$
- non ci siano partenze verso l'esterno $I_{id} = 0 \quad \forall i$

Applicando il **teorema di Jackson** e la conservazione del flusso, la probabilità di stato $\underline{n} = (n_1, n_2, \dots, n_M)$ risulta:

$$\mathbb{P}\{\underline{n}\} = \mathbb{P}\{0\} \prod_{i=1}^M \left(\frac{\lambda_i}{\mu_i}\right)^{n_i}$$

$\mathbb{P}\{0\}$ va interpretato come un coefficiente di normalizzazione (se il numero di pacchetti è costante sarebbe $\mathbb{P}\{0\} = 0$).

$$\mathbb{P}\{\text{evento certo}\} = \sum_{\underline{n}: \sum_{i=1}^M n_i = H} \mathbb{P}\{0\} \prod_{i=1}^M \left(\frac{\lambda_i}{\mu_i}\right)^{n_i} = 1$$

da cui

$$\mathbb{P}\{0\} = g(H, M)^{-1}$$

con

$$g(H, M) = \sum_{\underline{n}: \sum_{i=1}^M n_i = H} \prod_{i=1}^M \left(\frac{\lambda_i}{\mu_i}\right)^{n_i}$$

questa funzione viene chiamata **funzione di Buzen**. Essa può essere determinata anche per via ricorsiva:

$$g(h, m) = g(h, m-1) + \frac{\lambda_m}{\mu_m} g(h-1, m)$$

con

$$g(h, 1) = \left(\frac{\lambda_n}{\mu_n}\right)^h$$

$$g(-1, m) = 0$$

$$g(0, m) = 1$$

$$g(h, 0) = 0$$

$$h = 0, 1, \dots, H$$

$$m = 1, 2, \dots, M$$

Per la conservazione del flusso, $\lambda_i = \mu_i$ e, come dimostra la tabella seguente, $g(h, m)$ prende i valori del triangolo di tartaglia:

h	m				
	1	2	3	4	5
0	1	1	1	1	1
1	1	2	3	4	5
2	1	3	6	10	15
3	1	4	10	20	35
4	1	5	15	35	70

Si puó dimostrare che:

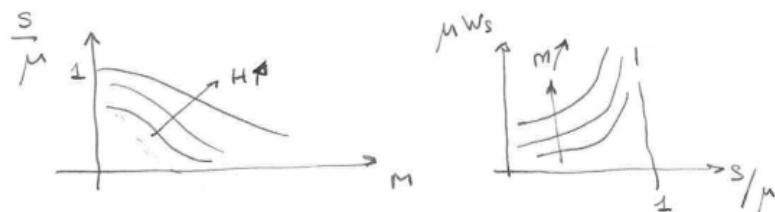
$$\begin{aligned}\mathbb{P}\{n_i \geq k\} &= \left(\frac{\lambda_i}{\mu_i}\right)^k \frac{g(H-k, M)}{g(H, M)} \\ \mathbb{P}\{n_i \geq H+1\} &= 0 \\ \mathbb{E}\{n_i\} &= \sum_{k=0}^{+\infty} \left(\frac{\lambda_i}{\mu_i}\right)^k \frac{g(H-k, M)}{g(H, M)}\end{aligned}$$

Throughput i -esimo sistema a coda:

$$\begin{aligned}S_i &= \mu_i \mathbb{P}\{n_i \geq 1\} \\ &= \mu_i \left(\frac{\lambda_i}{\mu_i}\right)^1 \frac{g(H-1, M)}{g(H, M)} \\ &= \lambda_i \frac{g(H-1, M)}{g(H, M)}\end{aligned}$$

In congestione vale $\lambda_i = \mu_i = \mu \forall i$, perció:

$$\begin{aligned}S &= \mu \frac{g(H-1, M)}{g(H, M)} \quad \forall i \\ g(h, m) &= \binom{h+m-1}{m-1} = \frac{(h+m-1)!}{(m-1)!h!} \\ \frac{S}{\mu} &= \frac{(H+M-2)!}{(M-1)!(H-1)!} \frac{(M-1)!H!}{(H+M-1)!} = \frac{H}{H+M-1} \\ \mu W_s &= \frac{n-1}{1 - \frac{S}{\mu}}\end{aligned}$$



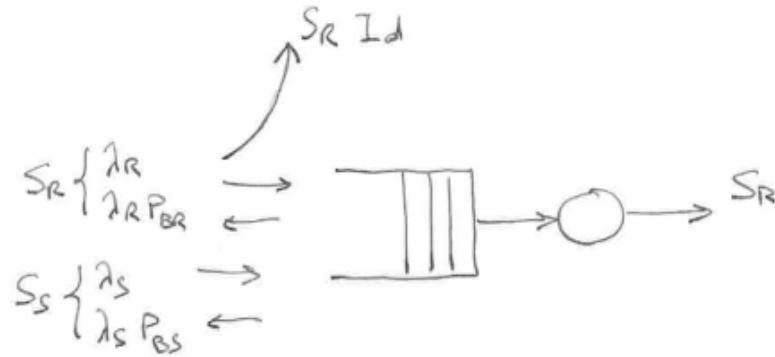
Per ottimizzare il throughput e il tempo di permanenza è necessario bilanciare H rispetto a M e μ .

$$\frac{d}{d(\frac{S}{\mu})} \frac{\frac{S}{\mu}}{\mu W_s} = 0 \Leftrightarrow \frac{S}{\mu} = \frac{1}{2} \Rightarrow H = M - 1$$

5.2 Blocco sull'ingresso

È un controllo di flusso hop-to-hop. I pacchetti nella rete di coda possono arrivare dall'esterno (in numero n_s) o da altri nodi della rete (in numero n_r). Indichiamo con Y la lunghezza della coda. Si limitano i pacchetti dall'esterno bloccandoli forzatamente quando il loro numero supera la soglia $N_s < Y$ privilegiando così quelli già dentro la rete che possono invece accedere a tutto il buffer Y . Il numero complessivo di pacchetti $n_s + n_r$ può essere maggiore di Y e ci sarà una probabilità di blocco.

Ad un nodo della rete si sommano un flusso da altri nodi della rete (S_r) ed un flusso dall'esterno (S_s).



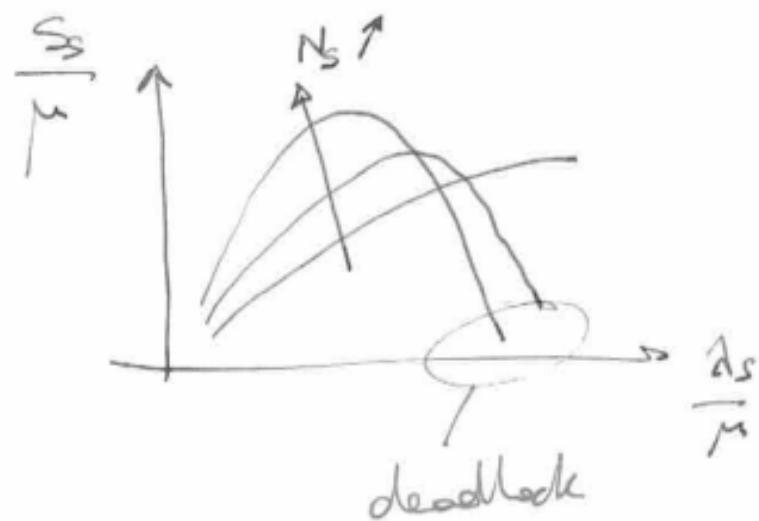
dove $P_{BR} = \mathbb{P}\{\text{blocco pkt interni alla rete}\}$ e $P_{BS} = \mathbb{P}\{\text{blocco pkt esterni alla rete}\}$.
Per la conservazione del flusso:

$$S'_r + S'_s - S_r I_0 = S_r$$

Lo studio in presenza di alcune ipotesi semplificatorie indica una capacità di trasposto della rete:

$$\frac{S - s}{\mu} = \frac{\lambda_s}{\mu} (1 - P_{BS})$$

dove P_{BS} è in funzione dei tassi di arrivo, della soglia N_s e della coda Y .



Per evitare deadlock si utilizzano N_s piccoli rispetto a Y .

Chapter 6

Standard implementativi

Passiamo da una trattazione generale al particolare standard, con particolare riferimento alle reti locali e metropolitane.

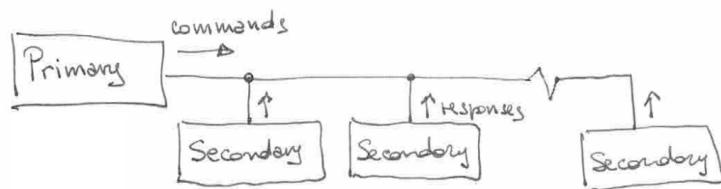
6.1 Protocollo High-level Data Link Control (HDLC)

È un protocollo di livello 2 (sul Data Link). Ha 3 modalità di funzionamento:

- Normal Response Mode (NRM)
- Asynchronous Response Mode (ARM)
- Asynchronous Balance Mode (ABM)

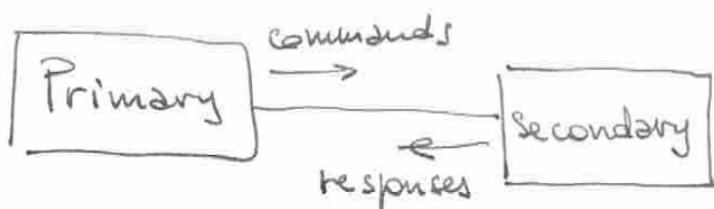
6.1.1 Normal Response Mode (NRM)

È utilizzato per le comunicazioni punto-multipunto, cioè da un nodo verso tanti nodi. Si prevede una stazione primaria che comunica verso più stazioni secondarie, dando loro dei comandi e attendendo da loro delle risposte. Inoltre, la stazione primaria si occupa dell'inizializzazione a controllo, del recupero degli errori e della sincronizzazione.



6.1.2 Asynchronous Response Mode (ARM)

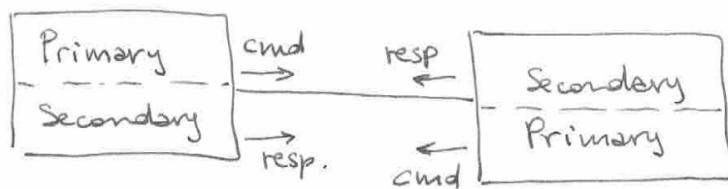
È utilizzato per le comunicazioni punto-punto, cioè da un nodo verso un altro nodo. Si prevede una stazione primaria che comunica e una stazione secondaria che risponde.



In modalità NRM e ARM la stazione primaria è unica e una stazione secondaria può trasmettere uno o più pacchetti dopo aver ricevuto il controllo dalla stazione primaria.

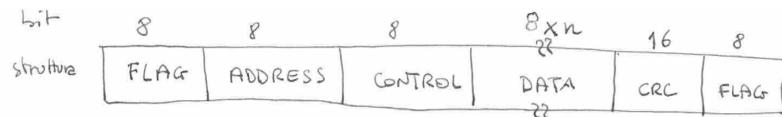
6.1.3 Asynchronous Balance Mode (ARM)

È utilizzato anch'esso per le comunicazioni punto-punto. In questa modalità, tutte le stazioni possono assumere il ruolo di primaria o secondaria all'occorrenza. Ciascuna stazione può iniziare come primaria quando necessario e il destinatario si pone come secondaria fino all'ultimo pacchetto della sessione.



6.1.4 Struttura pacchetto HDLC

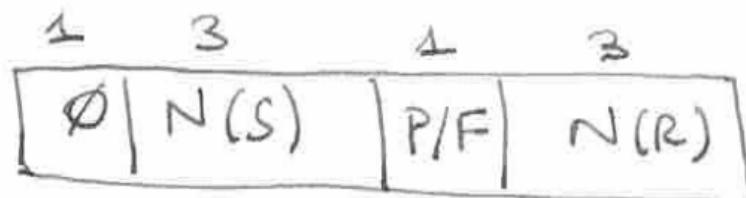
Occorre strutturare l'informazione che le stazioni trasmettono. In questo senso, i pacchetti sono formati da campi formati da un certo numero di bit.



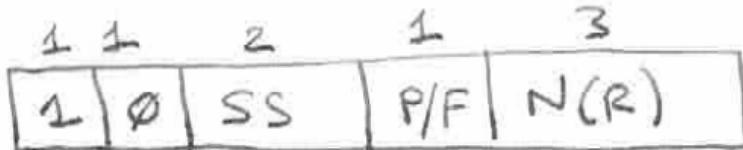
- **Flag** (8 bit): serve ad identificare l'inizio e la fine del pacchetto e per estrarre il sincronismo. Ha una struttura fissa (01111110).

Siccome questa combinazione di flag può trovarsi anche in altri campi, si utilizza la tecnica del *bit stuffing*: se l'informazione richiede 6 uni consecutivi, si inserisce uno zero dopo 5 uni

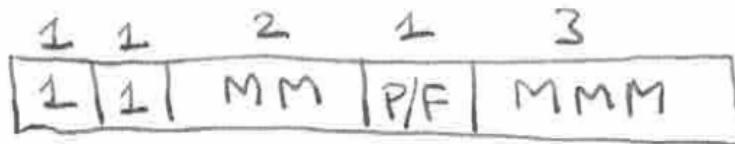
- **Address** (8 bit): contiene l'indirizzo della stazione secondaria (NRM, ARM) o della stazione destinataria (ABM)
- **Control** (8 bit): ha tre tipi di pacchetto
 - informazione: per trasmissione dati



- supervisor: per segnalazione e controllo



- unnumbered: per segnalazione e controllo



dove

- $N(S)$: identificativo ultimo pacchetto trasmesso
- $N(R)$: identificativo pacchetto che ci si attende di ricevere e dipende dal campo SS
- SS: rappresenta lo stato della stazione

SS	Stato	$N(R)$ e suo significato
00	Ready to Receive (RR)	ACK a tutti i pacchetti ricevuti fino a $N(R)$
01	Reject (REJ)	NACK a partire da $N(R)$
10	Receiver Not Ready (RNR)	Si blocca la trasmissione e ACK fino a $N(R) - 1$
11	Selective Reject (SREJ)	ARQ Selective Repeat chiedendo la ritrasmissione del solo pacchetto $N(R)$

- P/F (Poll/Final): questo bit è usato dalla stazione primaria per il polling (P). Se è uguale a 1, la stazione primaria forza la risposta a una secondaria, la quale deve rispondere con $F = 1$.
In NRM, P/F fa polling fra stazioni secondarie.
In ABM, P/F forza la risposta dell'unica stazione secondaria.
- MMMMM: rappresenta 32 combinazioni che permettono di instaurare una comunicazione di tipo datagram
Nota: quando si trasmette un pacchetto di informazione c'è il meccanismo di ACK attraverso gli SS. Questo meccanismo fa risparmiare segnalazione e viene detto *piggy back*

Ciò serve per implementare un ARQ di tipo Go-Back-N dove la lunghezza della finestra N è uguale a 8 e consente la trasmissione continua fino a 8 pacchetti.

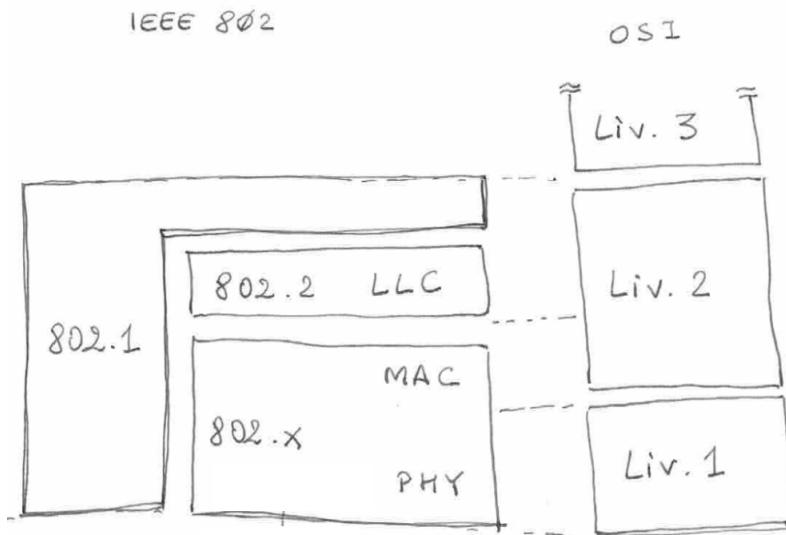
6.1.5 Topologia delle reti

Come mostrato in modo illustrativo in [questa sezione](#) si utilizzano principalmente tre topologie di reti:

- Bus
- Anello (Ring)
- Albero (Tree)

6.2 Comitato IEEE 802

Fra i compiti dell'IEEE (Institute of Electrical and Electronics Engineers) c'è anche la definizione di specifiche e standard per sistemi. In particolare lo standard 802 copre i primi due strati dell'ISO.



Lo standard IEEE 802.1 fornisce una descrizione generale, collega tra di loro gli standard e li mette in relazione col modello OSI. Altri standard (indicati con $x > 2$ nell'immagine) sono:

x	Nome protocolli aziendali simili
3	Ethernet II
4	Token bus
5	Token ring
6	DQDB
11	WLAN

6.2.1 IEEE 802.2 - Logical Link Control (LLC)

È un protocollo derivato dall'HDLC. Ha due tipi di servizio:

- senza connessione → datagram
- con connessione → circuito virtuale

Permette di instaurare fino a 128 collegamenti logici distinti consentendo punto-punto, punto-multipla, broadcast in modo datagram.

Il pacchetto LLC è strutturato in questo modo:



- **Destination Service Access Point (DSAP)** [8 bit]: indica il numero di collegamento don 7 bit, piú 1 bit per indicare se il destinatario é singolo o multiplo

- **Source Service Access Point (SSAP)** [8 bit]: indica il collegamento logico sorgente con 7 bit, più 1 bit per indicare se è un pacchetto di comando o di risposta
- **Control** (8 bit): come il campo control di HDLC in modalità ABM

Il controllo degli errori è effettuato a livello MAC con un campo CRC a 32 bit.

6.2.2 IEEE 802.3 - Ethernet II

Questo protocollo presenta un accesso casuale CSMA 1-persistent in una versione modificata che garantisce anche la rilevazione in ogni stazione delle connessioni. Con questa nuova tecnica, detta CSMA/CD (Collision Detect), si ascolta il canale anche durante la trasmissione per abortire immediatamente la trasmissione, bloccare l'altro trasmettitore e avvertire tutte le stazioni che parte di un pacchetto è errata.

Questa tecnica funziona sulle reti cablate, ma presenta problemi sulle reti radio per via dell'aleatorietà dovuta alla propagazione wireless della potenza ricevuta. Per questo motivo si usa questa tecnica solo nelle reti a bus o ad albero per le quali le probabilità di errore sono molto piccole e si evitano ACK/NACK.

In caso di collisione si ritrasmette dopo un tempo di backoff casuale $\mathcal{U}(0, (2^r - 1)t_B)$, dove r è il numero delle ritrasmissioni, $t_B = 2\tau_M$, ovvero due volte il massimo ritardo di propagazione fra le stazioni. L'802.3 prevede diverse tipologie di trasmissione, tant'è che si indica con

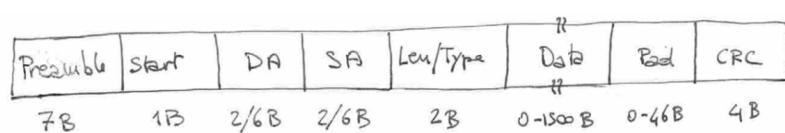
$$X \text{ Base } Y$$

dove:

- X è la velocità di trasmissione in Mbit/s
- Base è la trasmissione del segnale in banda base
- Y è la lunghezza massima del cavo

Year	Standard	Transmission Medium	Maximum speed	Maximum length
1983	10base5	Coaxial cable	10 Mbit/s	500 m
1985	10base2	Coaxial cable	10 Mbit/s	185 m
1990	10base-T	Twisted pair (Cat. 3)	10 Mbit/s	100 m
1995	100base-T4	Twisted pair (Cat. 3)	100 Mbit/s	100 m
1995	100base-TX	Twisted pair (Cat. 5)	100 Mbit/s	100 m
1995	100base-FX	Optical Fiber (MMF)	100 Mbit/s	2000 m
1998	100base-T2	Twisted pair (Cat. 3)	100 Mbit/s	100 m
1998	1000base-SX	Optical Fiber (MMF)	1000 Mbit/s	500 m
1998	1000base-LX	Optical Fiber (SMF)	1000 Mbit/s	5000 m
1999	1000base-TX	Twisted pair (Cat. 5)	1000 Mbit/s	100 m

Il pacchetto nell'IEEE 802.3 ha questa struttura:



- **Preamble** (7 byte): serve per la sincronizzazione. È un'alternanza di zeri e di uni (01010...)
- **Start** (1 byte): interrompe l'alternanza di uni e di zeri e termina il Preamble.
- Destination Address (**DA**) [2/6 byte in 802.3; 6 byte in Ethernet]: è l'indirizzo MAC di livello 2, ovvero 6 campi formati da 2 cifre esadecimale divise da due punti (:)

$\underbrace{ab : 24 : 23}_{\text{produttore}} : \underbrace{ee : ff}_{\text{seriale}} : 10$

- **Source Address (SA)**: è l'indirizzo MAC di livello 2. Medesima struttura del DA
- **Length (802.3) / Type (Ethernet) [2 byte]**: numero di byte nel pacchetto LLC / tipo di dati nel pacchetto
- **Data** (fino a 1500 byte): pacchetto LLC
- **Pad** (fino a 46 byte): serve per il meccanismo di rilevazione delle collisioni
- **CRC** (4 byte): usato per la correzione degli errori

Data e Pad formano la **transmission unit** e la loro somma è compresa tra 46 e 1500 byte

6.2.3 IEEE 802.4 - Token bus

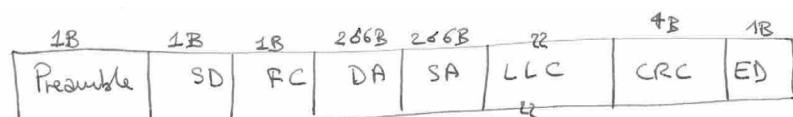
È un protocollo controllato distribuito per reti locali con topologia a bus o ad albero. Ogni stazione:

- conosce l'identificativo della stazione successiva e della precedente
- può trasmettere quando ha il token o quando interrogata da un'altra stazione che ha il token
- quando ha il token lo tiene per un tempo massimo (**token holding time, THT**) e poi il token passa alla stazione successiva

Le stazioni concordano un tempo massimo per passare il token fra tutti i nodi dell'anello virtuale: questo tempo è chiamato **target token rotation time (TTRT)**.

È previsto un meccanismo di accesso a contesa con cui nuove stazioni entrano nell'anello virtuale.

Il pacchetto nell'IEEE 802.4 ha questa struttura:



- **Preamble** (1 byte): serve per la sincronizzazione
- **Start Delimiter (SD)** [1 byte]: indica l'inizio del pacchetto

- **Frame Control (FC)** [1 byte]: serve per mandare una segnalazione come ad esempio
 - richiedere il token
 - mandare una sollecitazione al successore
 - chiedere chi é la stazione successiva
 - risolvere una contention
 - impostare un successore
 - dare prioritá a un pacchetto
 - ...
- **Destination Address (DA)** [2/6 byte]
- **Source Address (SA)** [2/6 byte]
- **LLC** (fino a 8182 byte se DA, SA a 2 byte; fino a 8174 byte se DA, SA a 6 byte)
- **CRC** (4 byte): usato per la correzione degli errori
- **End Delimiter (ED)** (1 byte): indica la fine del pacchetto

6.2.4 IEEE 802.5 - Token ring

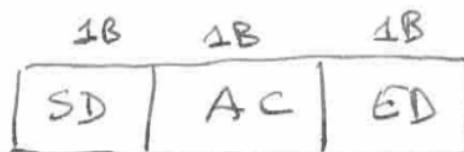
É un protocollo ad accesso controllato per reti locali con topologia ad anello monodirezionale e un meccanismo per bypassare stazioni spente. Una stazione trasmette quando riceve un token che sequenzialmente passsa da una stazione all'altra. Il ritardo per la rigenerazione é di 1 bit.

Ogni stazione rilascia il token alla successiva se ha finito di trasmettere o dopo che é passato il tempo a lei concesso in una fase di negoziazione.

Nell'IEEE 802.5 esistono due tipologie di pacchetti:

- pacchetto token
- pacchetto informativo

La struttura del pacchetto token é la seguente:



- **Start Delimiter (SD)** [1 byte]: indica l'inizio del pacchetto
- **Access Control (AC)** [1 byte]: gli 8 bit sono organizzati in questo modo:

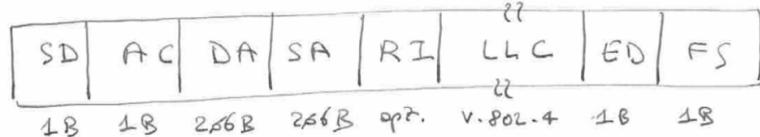
PPPTMRRR

dove

- P stabiliscono il livello di prioritá del traffico
- T identifica se il pacchetto é di tipo token o di tipo informativo

- M è utilizzato per il monitoraggio
- RRR sono utilizzati per la prenotazione del livello di priorità
- **End Delimiter (ED)** [1 byte]: indica la fine del pacchetto

La struttura del pacchetto informativo non termina dopo l'Access Control, ma contiene tutti i campi relativi al pacchetto:



- **Start Delimiter (SD)** [1 byte]: indica l'inizio del pacchetto
- **Access Control (AC)** [1 byte]
- **Destination Address (DA)** [2/6 byte]
- **Source Address (SA)** [2/6 byte]
- **Routing Information (RI)**: opzionale. È usato per collegare fra loro diversi anelli (bridge)
- **LLC** (fino a 8182 byte se DA, SA a 2 byte; fino a 8174 byte se DA, SA a 6 byte)
- **End Delimiter (ED)** (1 byte): indica la fine del pacchetto
- **Frame Status (FS)** [1 byte]: informa il trasmettitore sullo stato del pacchetto mediante due bit dell'AC che sono impostati dal ricevitore

Il token non viene rilasciato immediatamente dopo la trasmissione dell'ultimo pacchetto, ma solo dopo che questo ha effettuato il giro completo dell'anello.

6.2.5 IEEE 802.11 - WLAN

Le WLAN operano sulle bande libere ISM (Industrial, Scientific and Medical), ovvero quelle attorno ai 2.4GHz e ai 5GHz. L'ERP, che è un parametro che identifica la potenza emessa nella direzione di massima, è di 100mW, offrendo coperture tra gli 80 e i 100 metri.

Il protocollo 802.11 hanno visto varie generazioni, alcune delle quali sono:

- 802.11: 1Mbit/s
- 802.11b 11 Mbit/s
- 802.11n: 300 Mbit/s (Wi-Fi 4)
- 802.11ac: 500 Mbit/s (Wi-Fi 5)
- 802.11ax: 1 Gbit/s (Wi-Fi 6)

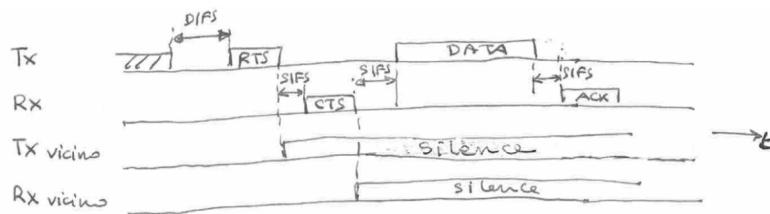
Le ultime versioni dello standard utilizzano antenne multiple per migliorare le prestazioni in presenza dei fenomeni aleatori nel mezzo radio.

Lo standard IEEE 802.11 prevede diverse modalità di funzionamento.

- **Independent Basic Service Set (IBSS)**: i terminali in copertura radio reciproca si indirizzano direttamente fra loro (peer-to-peer)
- **Basic Service Set (BSS)**: un dispositivo hardware copre una certa area e serve da punto di accesso (Access Point) per l'intera rete. Ci sono due possibili modalità di funzionamento:
 - **Distributed Coordination Function (DCF)**: tutte le comunicazioni passano per l'Access Point
 - **Point Coordination Function (PCF)**: l'Access Point interrompe periodicamente la modalità DCF per interrogare i terminali a polling
- **Extended Service Set (ESS)**: più BSS con Access Point collegati fra loro mediante cavo o radio

Il protocollo di accesso usato nel DCF è casuale CSMA. Siccome con mezzi radio il Collision Detect non è utilizzabile per via dell'aleatorietà nella potenza ricevuta, si usa la tecnica di Collision Avoidable (CSMA/CA): essa prevede una 4-way handshaking che risolve i problemi del terminale nascosto ed esposto:

- un trasmettitore fa carrier sensing per un tempo DIFS (Distributed Inter-Frame Service) prima di trasmettere
- poi informa i vicini e il destinatario che è in procinto di trasmettere con un pacchetto RTS (Request To Send)
- i trasmettitori che ricevono il RTS non trasmettono
- il ricevitore dopo un tempo SIFS (Short Inter-Frame Space) informa di essere pronto a ricevere con un pacchetto CTS (Clear To Send) e i suoi vicini che lo ricevono non trasmettono



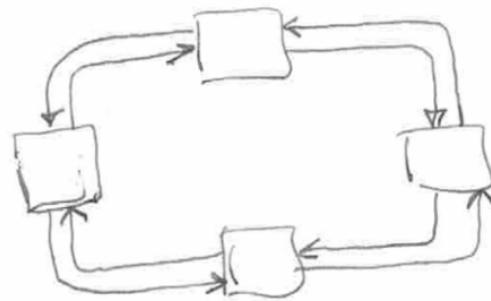
Per quanto riguarda il tempo di backoff si usa un contatore CW (Contention Window). Si parte da un valore minimo $CW = 7$ e per ogni trasmissione fallita cresce $CW = 2(CW+1) - 1$. Ciò implica un back-off esponenziale dato che si raddoppia ad ogni ritrasmissione.

$$t_B \sim \mathcal{U}(0, CW)t_{\text{base}}$$

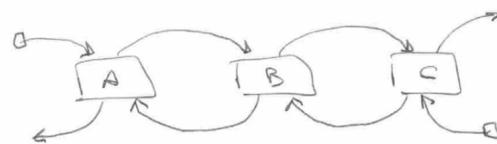
Durante il backoff una stazione non trasmette dati, ma può ricevere CTS e rispondere con ACK. L'effetto del terminale nascosto è evitato sui dati, non su RTS/CTS, che però sono pacchetti corti e con molta probabilità non crea problemi.

6.2.6 Altri protocolli

- **Fiber Distributed Data Interface (FDDI)**: MAN con due anelli controrotanti. Permette di gestire traffico asincrono, sincrono o isocrono



- **Distributed Queue Dual Bus (DQDB, IEEE 802.6):** MAN con traffico isocrono e due bus monodirezionali



Chapter 7

Internetworking

L'internetworking si occupa dell'interconnessione di reti allo scopo di rendere disponibili a tutti gli utenti e le macchine interconnesse i servizi offerti dalla rete. Vengono utilizzati protocolli e strategie dell'Internet Engineering Task Force (IETF) descritti in documenti chiamati Request For Comment (RFC).

I dispositivi hardware per far transitare i pacchetti tra le reti si differenziano tra di loro sulla base del livello sul quale operano e quindi sulla loro complessità realizzativa.

- **repeater:** ripetitore a livello fisico
 - rigenera il segnale modificandone il formato per adattarli a reti con strati fisici diversi
 - è usato per suddividere una rete in sezione più piccole
 - ha 2 porte. La versione multiporta si chiama **hub**
- **bridge:** ripetitore a livello di Data Link
 - fa comunicare reti con diversi protocolli di livello 2 e diverse velocità di trasmissione
 - è un ripetitore logico, non fisico
 - deve avere qualche conoscenza di dove si trova il destinatario e dell'instradamento per raggiungerlo per evitare di generare traffico verso uscite non necessarie. Quindi si dice che fa operazioni di **filtraggio**
 - ha 2 porte (collega due reti). La versione multiporta si chiama **switch**
- **router:** ripetitore a livello Network
 - fa comunicare reti diverse (internetworking)
 - realizza l'instradamento dei pacchetti
 - può comprendere operazioni più sofisticate di processamento dei pacchetti (e.g.: controllo di legittimità, gestione della qualità del servizio, tariffazione, ...)
- **gateway:** ripetitori per livelli superiori a quello di Network
 - viene utilizzato per il processamento ad alto livello dei pacchetti

7.1 Livello 2

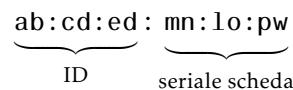
7.1.1 Bridge

Il bridge deve essere trasparente e avere velocità di processamento tale da analizzare i pacchetti provenienti dalla rete più veloce a cui è interconnesso. Vista la varietà di reti interconnesse al bridge occorre un buffer finito e dimensionato per memorizzare i pacchetti in attesa di essere trasmessi e per effettuare il controllo di flusso (blocco su ingresso. A livello 2 l'instradamento dipende dall'indirizzo MAC del destinatario.

7.1.2 Indirizzo MAC

Il Media Access Control (MAC) ha un indirizzo di livello 2 composto da 2 byte o 6 byte (comunemente 6 byte) che indirizza univocamente una scheda di rete a livello mondiale. Viene tipicamente espresso utilizzando cifre esadecimali.

- 3 byte per l'ID produttore
- 3 byte per il seriale della scheda del produttore



Ha le seguenti caratteristiche:

- spazio di indirizzamento globale 2^{48} indirizzi
 - spazio di indirizzamento per ogni produttore: 2^{24} indirizzi
 - indirizzo di broadcast: FF:FF:FF:FF:FF:FF
- È l'indirizzo di destinazione, non di sorgente. Permette l'invio di un pacchetto a tutte le stazioni di livello 2

7.1.3 Instradamento del bridge

L'instradamento a livello di bridge può essere fatto in due modi:

- **a percorsi fissati:** ogni possibile percorso rete sorgente - rete destinazione è mappato a priori. Ciò significa che:
 - ogni stazione è associata, sulla base del proprio indirizzo MAC, a una rete
 - tutti i bridge devono conoscere le tabelle di associazione stazione - rete

Questo metodo presenta due problemi:

- non è adattivo (la tabella non varia al variare delle condizioni di rete)
- possibile duplicazione di risorse e di pacchetti

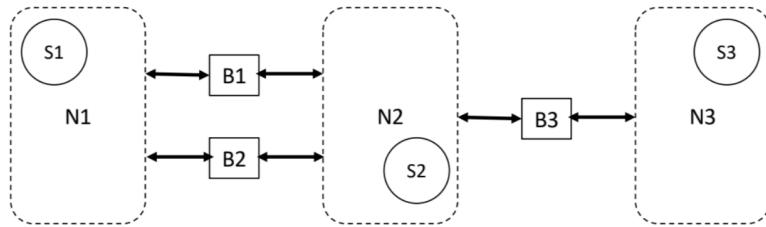


FIGURE 7.1

	N1	N2	N3
N1	X	B1	B1
N2	B1	X	B3
N3	B3	B3	X

- **con autoapprendimento:** per rendere adattivo il sistema a percorsi fissati, ogni bridge viene dotato della capacità di compilare autonomamente la tabella di associazione
 - l'associazione è basata sull'analisi dei pacchetti che transitano attraverso il bridge tramite estrazione dell'indirizzo MAC sorgente
 - l'associazione viene cancellata dopo un tempo chiamato *timeout* dall'ultima acquisizione. Per stazioni mobili le associazioni sono indicizzate e può essere presente una sola associazione per MAC address

Questo metodo presenta un problema: ogni bridge può conoscere solo la parte della tabella di associazione riferita alle reti ad esso collegate

C'è un metodo per evitare la propagazione delle tabelle di associazione e di instradamento fra reti:

- viene cercata all'interno della tabella di associazione la porta corrispondente a MAC_DEST, cioè la porta verso la rete che ospita la stazione indirizzata da MAC_DEST
- se la porta destinataria è la stessa di arrivo, il pacchetto non viene ripetuto, altrimenti il pacchetto viene ripetuto sulla porta identificata
- se nessuna porta destinataria è identificata (e.g.: per un timeout di associazione), allora il filtraggio non è possibile e conservativamente si ripete il pacchetto sulla porta diversa da quella di arrivo per propagare nella rete

Il MAC_DEST broadcast è trattato allo stesso modo. C'è il rischio di loop che sono eliminabili mediante la costruzione automatica di un albero di bridge e permettendo ai pacchetti di muoversi solo lungo l'albero (no cicli).

B1	PORT	B3	PORT
S1	1	S1	1
S2	2	S2	1
S3	2	S3	2

TABLE 7.1: Esempio di associazione degli indirizzi MAC delle stazioni alla porta di un bridge (1: porta di sinistra; 2: porta di destra) in riferimento alla figura 7.1

7.1.4 Spanning Tree

L'algoritmo spanning tree é un algoritmo introdotto nel IEEE 802.1 per la creazione dell'albero di bridge. Le condizioni iniziale sono le seguenti:

- ad ogni bridge viene assegnato un identificativo univoco (prioritá - unicita)
- l'indirizzo MAC é unico e noto per ciascun bridge
- ogni porta é identificata univocamente

L'algoritmo spanning tree consiste nei seguenti passaggi:

1. viene scelto inizialmente il bridge radice prendendo quello con identificativo piú basso
2. ogni bridge determina quale sua porta, detta *porta radice*, é nella direzione a costo inferiore (velocitá maggiore) verso la radice e fra tutti viene scelto quello a costo inferiore
3. ogni bridge lascia attiva la sua porta radice e tutte le porte selezionate per interconnettere una rete, le altre vengono bloccate e restano inutilizzate
4. il processo di costruzione dello spanning tree avviene attraverso lo scambio di informazioni tra bridge mediante il Bridge Protocol Data Unit (BPDU) che contengono informazioni sul bridge radice e sulle funzioni di costo. Tale procedura é ripetuta periodicamente

7.2 Livello 3

7.2.1 Internet Protocol (IP)

L'Internet Protocol é il protocollo di livello 3 alla base di Internet, la rete a pacchetto attualmente piú diffusa al mondo. IP é un protocollo di tipo datagram sviluppato inizialmente per creare una rete per interconnettere postazioni della difesa americana e mantenere la connessione anche se una parte venisse distrutta, dando vita alla commutazione a pacchetto. Inoltre, IP é un protocollo a pacchetti senza connessione e di tipo *best effort*, cioé che non garantisce alcuna forma di affidabilitá della comunicazione. In questo protocollo non c'é nessun tipo di controllo di errore, controllo di flusso e controllo di congestione: l'affidabilitá puó essere realizzata dai protocolli di trasporto del livello 4. Esistono due versioni del protocollo IP: l'originaria v4 e la piú recente v6, quest'ultima nata dall'esigenza di gestire meglio il crescente numero di dispositivi connessi ad Internet. IP nasce prima del modello OSI e si occupa anche della frammentazione: riceve dal livello 4 dei pacchetti di 64KB e li frammenta in segmenti, che riassembra in ricezione per formare i 64KB da restituire al livello 4. I pacchetti sono composti da un header e dai dati.

7.2.2 Indirizzi IPv4

Gli indirizzi IPv4 sono composti da 32 bit. Vengono tipicamente espressi utilizzando una notazione a punti (dot notation) che separano quattro campi di 8 bit ciascuno (e.g.: 192.168.1.6). Esistono 4 diversi schemi di indirizzamento (rete e stazione):

	Bit iniziali	Bit reti	Bit host	range indirizzi
Classe A	0	7	24	da 0.0.0.0 a 127.255.255.255
Classe B	10	14	16	da 128.0.0.0 a 191.255.255.255
Classe C	110	21	8	da 192.0.0.0 a 223.255.255.255
Classe D	1110	0	28	da 224.0.0.0 a 239.255.255.255
Usi futuri	11110			

Gli indirizzi pubblici sono rilasciati e regolamentati dall'ICANN (Internet Corporation for Assigned Names and Numbers) e da altri enti internazionali. Gli enti detentori di una classe possono suddividere ulteriormente la classe. Per ogni rete due indirizzi sono dedicati: uno per identificare la rete stessa e uno per il broadcast:

- XXX.YYY.ZZZ.0: rete
- XXX.YYY.ZZZ.255: broadcast

tutti gli altri sono indirizzi unicast. Nelle diverse classi alcuni indirizzi sono riservati per applicazioni intranet (e.g.: nella classe C sono 192.168.0.0 - 192.168.255.255) o per scopi intrahost (e.g.: nella classe A sono 127.0.0.0 - 127.255.255.255, con 127.0.0.1 detto *localhost* per ogni host anche privo di scheda di rete per scrivere applicazioni di rete utilizzando l'host stesso).

7.2.3 NETMASK

Se il destinatario é nella stessa rete (intranet) allora il pacchetto puó essere messo sulla rete e il destinatario lo vedrá; altrimenti (internet) occorre inviare il pacchetto fuori affidandolo alla macchina che identifica l'accesso al mondo esterno, detta **gateway**. Gli indirizzi IP di sorgente e destinatario, però, non sono sufficienti per capire se il payload é destinato sulla stessa rete o all'esterno. Per fare ciò, si utilizza il **NETMASK**: esso permette di risolvere il problema di cosa é locale e cosa é remoto verificando se IP_DEST é dentro l'intervallo di indirizzi della rete locale. La rete di un indirizzo IP é l'AND logico fra l'indirizzo IP e la NETMASK.

Esempio

IP_DEST	195.32.62.2
NETMASK	255.255.255.0

IP_DEST AND NETMASK = 195.32.62.0: nella rete ci sono tutti gli indirizzi 195.32.62.X

Esistono dei NETMASK notevoli:

- 255.255.255.0: classe C
- 255.255.255.192: divide la classe C in 4 sottoreti, ciascuna con $2^6 - 2$ ID
- 255.255.255.128: divide la classe C in 2 sottoreti, ciascuna con $2^7 - 2$ ID
- 255.255.255.252: é la piú piccola sottorete, usabile con $2^2 - 2$ ID
- 255.255.255.255: rappresenta il solo host

- 255.255.0.0: classe B
- 255.0.0.0: classe A
- 0.0.0.0: intero spazio degli indirizzi IP

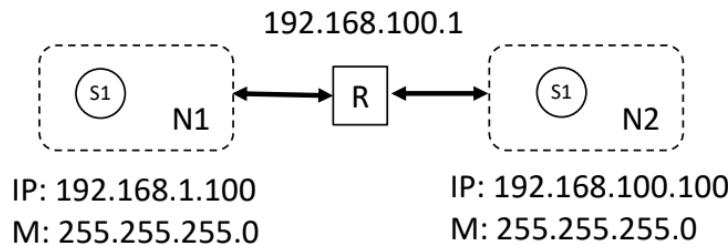
7.2.4 Router

Supponiamo di voler inviare un pacchetto di livello 3.

- se il destinatario é nella rete locale, allora é possibile operare una trasmissione a livello 2
- altrimenti é necessario affidare il pacchetto ad un router

Ogni nodo ed ogni router posseggono una tabella di instradamento che indichi il router (next hop) a cui affidare i pacchetti che non appartengono alla rete locale. Esistono diversi protocolli di instradamento per la compilazione della tabella (e.g.: RIP, OSPF, BGP, ...)

IP_NET	NET_MASK	NEXT_HOP
192.168.100.0	255.255.255.0	192.168.100.1



7.2.5 Protocollo ARP

Una volta noto il successivo nodo di destinazione di livello 3, é necessario effettuare una comunicazione di livello 2 (tra macchine locali), dunque é necessario un meccanismo di conversione tra indirizzo di livello 3 e di livello 2. Questa operazione é delegata all'Address Resolution Protocol (ARP), che funziona in questo modo:

1. l'entitá interessata alla associazione invia sulla rete locale un pacchetto broadcast (indirizzo MAC FF:FF:FF:FF:FF:FF) chiedendo la risoluzione dell'IP desiderato
2. la stazione con l'IP indicato nel pacchetto broadcast risponde all'entitá sorgente sfruttando il MAC sorgente del pacchetto stesso
3. l'entitá sorgente riceve il pacchetto ed effettua la risoluzione

Per evitare eccessivo traffico di segnalazione, le risoluzioni ARP vengono mantenute in una tabella che viene aggiornata periodicamente.

7.2.6 DHCP

Ogni nodo deve essere configurato per poter operare in una qualsiasi rete. A questo scopo esiste il Dynamic Host Configuration Protocol (DHCP), un protocollo applicativo che permette a un nodo di apprendere autonomamente la configurazione della rete a cui si connette. Opera in questo modo:

1. l'host interessato invia una richiesta di configurazione in broadcast
2. se è presente un server DHCP nella rete, questo risponde all'host sorgente con la configurazione da adottare e il tempo di validità della configurazione stessa

Scaduto il tempo di validità della configurazione (*lease time*), in assenza di comunicazioni da parte dell'host, il server DHCP considera la macchina spenta.

7.2.7 ICMP

L'Internet Control Message Protocol (ICMP) è il protocollo di segnalazione di IP. I primi 32 bit sono fissi:

- 8: tipo messaggio
- 8: codice messaggio
- 16: checksum

La parte seguente dipende dal tipo di segnalazione. Fra le più frequenti vi sono i seguenti:

- rapporto di errori
 - destination unreachable: pacchetto scartato per non aver raggiunto la destinazione
 - time exceed: pacchetto scartato da router per aver terminato il TTL
 - parameter error: parametro header non interpretabile
- test di raggiungibilità e prestazioni
 - echo request / echo reply: raggiungibilità e stima tempo A/R
- controllo congestione: diminuisce la frequenza di trasmissione dei pacchetti
- cambiamento routing
 - redirect: informa la macchina di usare un altro router per spedire il pacchetto verso una destinazione
- richiesta parametri di rete
 - Address Mask Request / Reply: per ottenere il NETMASK

7.2.8 IPv6

IPv6 nasce per superare alcuni limiti di IPv4:

- spazio di indirizzamento esteso a 16B (128 bit) superando i concetti di classi per maggiore flessibilità e con gerarchie per semplificare le tabelle di instradamento
- IPv4 ha il type of service ma i router non ne tengono conto e pertanto il funzionamento è best effort. IPv6 supporta la QoS in maniera nativa
- gestione degli header più semplice per un controllo più veloce
- funzioni di sicurezza e di controllo di integrità
- implementa ICMPv6 che ha alcune funzionalità aggiuntive rispetto a v4

7.3 Livello 4

7.3.1 TCP

Il Transmit Control Protocol (TCP) è il livello di trasporto tipicamente usato con IP (spesso non si distingue fra livelli 3 e 4 e si parla di TCP/IP). Ha lo scopo di rendere affidabile le comunicazione end-to-end, cosa non possibile con IP a causa, ad esempio, delle code finite nei router. Il protocollo TCP crea una connessione virtuale fra sorgente e destinatario per ogni flusso di pacchetti, i pacchetti sono numerati in modo sequenziale e si implementa un meccanismo ARQ basato su ACK e timeout. Il flusso di pacchetti è identificato da:

- indirizzi (sorgente, destinazione) di livello 3
- porte (sorgente, destinazione) di livello 4

TCP implementa:

- fasi di connessione e disconnessione
- trasferimento dei segmenti
- controllo degli errori
- controllo di flusso
- controllo della congestione (meccanismo a finestra)

Il TCP accetta pacchetti di lunghezza arbitraria, li frammenta in segmenti di lunghezza 64KB accettabili da IP, li riordina e li riassembra. TCP sovraintende alla connessione end-to-end e decide la velocità con cui immettere i pacchetti nel circuito virtuale di livello 4 che instaura. Questo consente di controllare il flusso e la congestione nella rete. Essendo un protocollo di comunicazione end-to-end, opera solo in modalità unicast e non multicast o broadcast.

7.3.2 UDP e RTP

L'User Datagram Protocol (UDP) è un altro modo per effettuare il trasporto mediante datagram senza circuito virtuale. Sono le applicazioni a livello superiore che riordinano i pacchetti e chiedono la ritrasmissione.

Il Real Time Protocol (RTP) è un modo di fare trasporto per gestire dati isocroni (e.g.: video e audio per videoconferenza). È simile a UDP, ma fornisce alle applicazioni pacchetti ordinati e cadenzati anche accorpando sorgenti diverse (e.g.: video e audio).