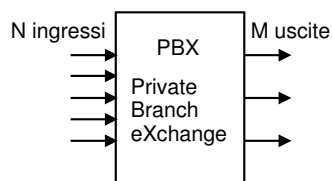


# Sistema di servizio, teoria del traffico, e delle reti

Qui trovano spazio argomenti di *ingegneria delle reti*, a carattere prevalentemente teorico, mentre le realizzazioni pratiche sono sviluppate nei capitoli che seguono. Tali aspetti si basano sui concetti di probabilità affrontati al capitolo 6, e dopo aver caratterizzato in tal senso il *traffico* informativo, sono fornite le metodologie di dimensionamento per collegamenti condivisi da più flussi, ai fini del conseguimento di prestazioni individuate come la probabilità di blocco nei sistemi di servizio orientati alla perdita, nei casi di popolazione finita ed infinita, ed il tempo medio di servizio per i sistemi orientati al ritardo, come nel caso di coda infinita e servente unico. Nell'ultima sezione poi, dopo aver discusso brevemente dei compromessi da affrontare nella progettazione di un sistema di comunicazione a pacchetto, viene sviluppata una tassonomia che permette di inquadrare le diverse architetture di rete in un contesto unitario, che ne mette in luce le differenti peculiarità.

## 22.1 Distribuzione binomiale per popolazione finita

Iniziamo con il chiederci quante linee uscenti  $M$  siano necessarie ad un centralino con  $N$  interni, in modo che la probabilità di trovare tutte le linee occupate sia inferiore ad un valore massimo, chiamato *grado di servizio*<sup>1</sup>. Per trovare il risultato, calcoliamo prima la probabilità che tutte le linee uscenti siano occupate, assumendo noti  $N$  ed  $M$ .



Affrontiamo il problema in termini ancor più generali, chiedendoci quale sia la probabilità  $p_B(k)$  che un numero  $k$  di persone (su  $N$ ) sia contemporaneamente al telefono. Assumiamo che ognuno degli  $N$  interni abbia una probabilità  $p$  di telefonare, ossia passi il  $p \cdot 100\%$  del suo tempo al telefono, e che le telefonate siano statisticamente

<sup>1</sup>Il termine grado di servizio esprime un concetto di *qualità*, ed è usato in contesti diversi per indicare differenti grandezze associate appunto alla qualità dei servizi di telecomunicazione, vedi pag. 639. Nel caso presente, una buona qualità corrisponde a una bassa probabilità di occupato.

indipendenti. Allora, ci saranno in media  $Np$  telefoni occupati, e la probabilità che un ben preciso gruppo di  $k$  individui telefoni (e  $N - k$  no), è pari a

$$p^k q^{N-k}$$

in cui  $q = 1 - p$ . Dato che il numero di differenti modi di scegliere  $k$  oggetti tra  $N$  è pari al *coefficiente binomiale*

$$\binom{N}{k} = \frac{N!}{k! (N - k)!} = \frac{N(N - 1) \cdots (N - k + 1)}{k!}$$

allora la probabilità di avere  $k$  (qualsiasi) persone al telefono è pari a

$$p_B(k) = \binom{N}{k} p^k q^{N-k} \tag{22.1}$$

Risultando che  $\sum_{k=0}^N p_B(k) = 1$ , la funzione  $p_B(k)$  rappresenta una densità di probabilità di v.a. discreta, detta anche variabile aleatoria di *Bernoulli* o *binomiale*<sup>2</sup>.

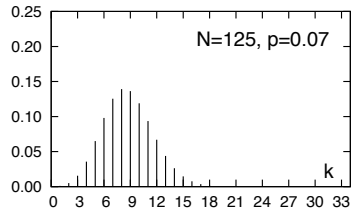
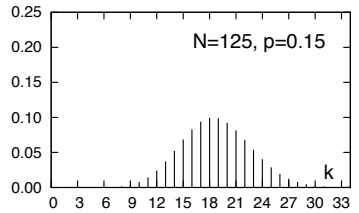
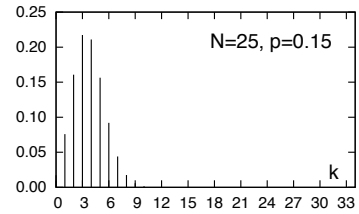
Al variare di  $k$  si ottengono tutte le probabilità cercate, rappresentate nella figura a lato nel caso in cui  $p = 0.15$  e  $N = 25$ , oppure  $N = 125$ . Nel secondo caso, si utilizza anche il valore  $p = 0.07$ , che produce una concentrazione di  $p_B(k)$  attorno a valori  $k$  inferiori; valori di  $p$  ancora più piccoli producono una d.d.p. che decresce monotonamente per  $k > 0$ . Infine, osserviamo che non si possono avere più di  $N$  utenti al telefono.

Per conoscere il numero di linee necessarie a garantire una probabilità di *congestione* (o di blocco)  $P_B$  inferiore ad un massimo, si sommano (partendo *da destra*) i valori di probabilità  $p_B(k)$ , finché non si supera la probabilità prefissata: allora  $M$  sarà pari all'ultimo indice  $k$ . Infatti in tal modo la probabilità che ci siano più di  $M$  interni a voler telefonare è pari a

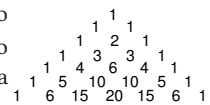
$$Pr(k > M) = \sum_{k=M+1}^N p_B(k) = \sum_{k=M+1}^N \binom{N}{k} p^k q^{N-k} < P_B$$

La distribuzione binomiale è detta anche *delle prove ripetute* poiché può essere usata per calcolare la probabilità di un certo numero di eventi favorevoli, a seguito della ripetizione dello stesso fenomeno aleatorio<sup>3</sup>.

Densità di Bernoulli



<sup>2</sup>Infatti i termini  $\binom{N}{k}$  sono pari ai coefficienti della potenza di un binomio  $2(p + q)^N$ , calcolabili anche facendo uso del triangolo di *Pascal* (ma definito prima da *Tartaglia*, e prima ancora da *Hayyām*), mostrato per riferimento a



<sup>3</sup>Infatti si applica ad un qualunque fenomeno aleatorio rappresentato dalla ripetizione di un secondo fenomeno aleatorio *soggiacente*, come ad esempio il lancio ripetuto di monete o di dadi: in questi casi,

**Intensità di traffico** Il valor medio della distribuzione Binomiale è  $m_B = Np$ , e la varianza  $\sigma_B^2 = Npq$ . Tornando al caso del centralino, il numero medio di linee occupate è  $Np$ : tale quantità rappresenta *l'intensità di traffico offerto medio*, che si misura in ERLANG: ad esempio, un traffico medio di 3 Erlang corrisponde ad osservare in media 3 linee occupate. Il rapporto

$$\frac{\sigma_B^2}{m_B} = \frac{Npq}{Np} = q < 1$$

è un indice di come la variabile aleatoria *traffico* si distribuisce attorno alla media. Il caso di Bernoulli in cui  $\frac{\sigma_B^2}{m_B} < 1$  è rappresentativo di un traffico *dolce*, che deriva dall'ipotesi di popolazione finita, e che si sostanzia nel fatto che all'aumentare delle linee occupate, diminuisce la probabilità di una nuova chiamata, in quanto diminuiscono le persone *non* al telefono.

**Esercizio** Una linea telefonica risulta occupata per l'80 % del tempo, e le telefonate non durano mai più di 5 minuti. Provando a chiamarla con una cadenza fissa di un tentativo ogni 10 minuti, determinare

1. la probabilità di trovare libero *entro* 3 tentativi
2. la probabilità di trovare libero *almeno* una volta in due ore
3. la probabilità di trovare libero *esattamente* tre volte in due ore

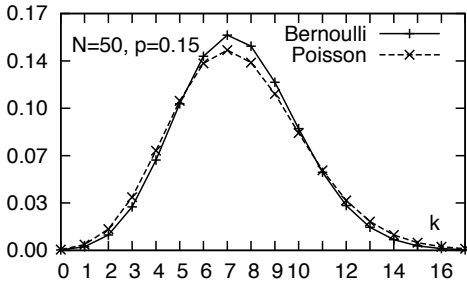
Indichiamo con  $p = 0.2$  la probabilità di successo di un singolo tentativo, e con  $q = 1 - p = 0.8$  quella di fallimento, identificando così il problema nel contesto delle *prove ripetute*.

1. Assumendo gli eventi indipendenti, la prob. di trovare libero entro tre tentativi è la somma delle prob. degli eventi favorevoli, ossia subito libero, oppure al secondo, od al terzo tentativo, ovvero  $p + p \cdot q + p \cdot q \cdot q = .2 + .2 \cdot .8 + .2 \cdot .8 \cdot .8 = 0.488 = 48.8$  %.
2. In due ore si effettuano  $\frac{120}{10} = 12$  tentativi. Conviene in questo caso valutare la probabilità dell'evento complementare  $p_0$ , quello di fallire tutti i tentativi, pari a  $p_B(k)|_{k=0}$ , ovvero  $p_0 = \binom{12}{0} p^0 q^{12} = \frac{12!}{12!} \cdot 8^{12} = 0.0687195$ , e quindi la prob.  $p_1$  di libero almeno una volta vale  $p_1 = 1 - p_0 = 93.12$  %.
3. Trovare libero esattamente tre volte infine ha probabilità  $\binom{12}{3} p^3 q^9 = \frac{12 \cdot 11 \cdot 10}{3 \cdot 2} \cdot .2^3 \cdot .8^9 = 0.23$ .

## 22.2 Distribuzione di Poisson

Al crescere del numero  $N$  di utenti, l'utilizzo della distribuzione Binomiale può risultare disagiata, per via dei fattoriali, e si preferisce trattare il numero di conversazioni attive

ha senso chiedersi con che probabilità una funzione della v.a. soggiacente acquisisce un certo valore, per un certo numero di volte. *Esempio*: si voglia calcolare la probabilità di osservare 3 volte testa, su 10 lanci di una moneta. Applicando la (22.1), si ottiene  $p_B(3) = \binom{10}{3} p^3 q^7 = 120 \cdot .5^3 \cdot .5^7 = 0.117$ , ovvero una probabilità dell'11,7 %. Come ulteriore esempio, citiamo l'uso della distribuzione binomiale per calcolare la probabilità di errore complessiva in una trasmissione numerica realizzata mediante un collegamento costituito da  $N$  tratte collegate da ripetitori rigenerativi, come illustrato al § 18.3.2.



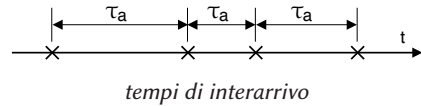
$k$  come una variabile aleatoria di POISSON, la cui densità di probabilità ha espressione

$$p_P(k) = e^{-\alpha} \frac{\alpha^k}{k!} \quad (22.2)$$

ed è caratterizzata da valor medio e varianza  $m_p = \sigma_p^2 = \alpha$ . La Poissoniana costituisce una buona approssimazione della ddp di Bernoulli, adottando per la prima lo stesso valor

medio della seconda  $m_p = m_B$ , ossia  $\alpha = Np$ , come mostrato in figura.

Più in generale, la densità (22.2) è impiegata per descrivere la probabilità che si verifichino un numero di eventi *indipendenti e completamente casuali* di cui è noto solo il numero medio  $\alpha$ (<sup>4</sup>). D'altra parte, al tendere di  $N$  ad  $\infty$  il modello Bernoulliano adottato finora perde di validità: infatti nel caso di una popolazione infinita il numero di nuove chiamate *non diminuisce* all'aumentare del numero dei collegamenti in corso. In questo caso gli eventi corrispondenti all'inizio di una nuova chiamata sono invece considerati *indipendenti e completamente casuali*, e descritti unicamente in base ad una *frequenza media di interarrivo*  $\lambda$  che rappresenta la velocità (come *richieste per unità di tempo*) con cui si presentano le nuove chiamate<sup>5</sup>. L'inverso di  $\lambda$  rappresenta un tempo, ed esattamente  $\bar{\tau}_a = 1/\lambda$  è il *valor medio* della variabile aleatoria  $\tau_a$  costituita dall'intervallo di tempo tra l'arrivo di due chiamate.



Con queste definizioni è possibile riferire la v.a. di Poisson ad un intervallo temporale di osservazione  $T$ , durante il quale si presentano un numero medio  $\alpha$  di chiamate<sup>6</sup> pari a  $\alpha = \lambda T$ . Pertanto, possiamo scrivere la d.d.p. della v.a. Poissoniana come

$$p_P(k)|_T = e^{-\lambda T} \frac{(\lambda T)^k}{k!}$$

che indica la probabilità che in un tempo  $T$  si verifichino  $k$  eventi (indipendenti e completamente casuali) la cui frequenza media è  $\lambda$ (<sup>7</sup>).

<sup>4</sup>Usando il modello Poissoniano la probabilità che (ad esempio) si stiano svolgendo *meno* di 4 conversazioni contemporanee è pari pertanto a  $p_P(0) + p_P(1) + p_P(2) + p_P(3) = e^{-\alpha} \left( 1 + \alpha + \frac{\alpha^2}{2} + \frac{\alpha^3}{6} \right)$ .

<sup>5</sup>La trattazione può facilmente applicarsi a svariate circostanze: dalla frequenza con cui si presentano richieste di collegamento ad una rete di comunicazioni, alla frequenza con cui transitano automobili sotto un cavalcavia, alla frequenza con cui particelle subatomiche transitano in un determinato volume, alla frequenza con cui gli studenti si presentano a lezione...

<sup>6</sup>Esempio: se da un cavalcavia osserviamo (mediamente)  $\lambda = 3$  auto/minuto, nell'arco di  $T = 2$  minuti, transiteranno (in media)  $3 \cdot 2 = 6$  autovetture.

<sup>7</sup>Esempio: sapendo che l'autobus (completamente casuale!) che stiamo aspettando ha una frequenza di passaggio (media) di 8 minuti, calcolare: **A**) la probabilità di non vederne nessuno per 15 minuti e **B**) la probabilità che ne passino 2 in 10 minuti.

*Soluzione:* si ha  $\lambda = 1/8$  passaggi/minuto e quindi: **A**)  $p_P(0)|_{15} = e^{-\frac{15}{8}} = 0.15$  pari al 15%; **B**)  $p_P(2)|_{10} = e^{-\frac{10}{8}} \frac{(\frac{10}{8})^2}{2} = 0.224$  pari al 22.4%

### 22.2.1 Variabile aleatoria esponenziale negativa

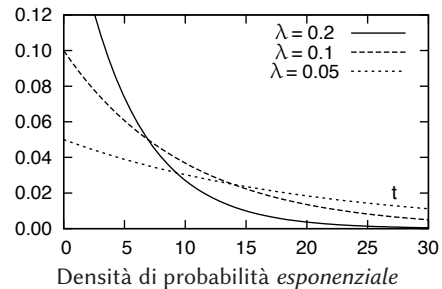
La descrizione statistica che la ddp di Poisson fornisce per *il numero* di eventi che si verificano in un (generico) tempo  $t$  è strettamente legata al considerare gli eventi come *indipendenti, identicamente distribuiti*, e per i quali *l'intervallo di tempo* tra l'occorrenza degli stessi è una determinazione di variabile aleatoria *completamente casuale*<sup>8</sup>, descritta da una densità di probabilità *esponenziale negativa*<sup>9</sup>, espressa analiticamente come

$$p_E(t) = \lambda e^{-\lambda t}$$

valida per  $t \geq 0$ , e mostrata in figura; tale v.a. è caratterizzata dai momenti<sup>10</sup>  $m_E = \frac{1}{\lambda}$  e  $\sigma_E^2 = \frac{1}{\lambda^2}$ . La probabilità che il tempo di attesa di una v.a. esponenziale superi un determinato valore  $t_0$ , è allora calcolabile come

$$Pr(t > t_0) = \int_{t_0}^{\infty} \lambda e^{-\lambda t} dt = -e^{-\lambda t} \Big|_{t_0}^{\infty} = e^{-\lambda t_0} \quad (22.3)$$

e questo risultato ci permette di verificare il legame con la Poissoniana<sup>11</sup>.



<sup>8</sup>Da un punto di vista formale per *eventi completamente casuali* si intende che gli eventi stessi *non hanno memoria* di quando siano accaduti l'ultima volta, permettendo di scrivere

$$Pr(t > t_0 + \theta / t > t_0) = Pr(t > \theta)$$

ossia la probabilità di attendere altri  $\theta$  istanti, avendone già attesi  $t_0$ , non dipende da  $t_0$ . Per verificare che la ddp esponenziale consente di soddisfare questa condizione svolgiamo i passaggi, applicando al terzultimo la (22.3):

$$\begin{aligned} Pr(t > t_0 + \theta / t > t_0) &= \frac{Pr(t > t_0 + \theta; t > t_0)}{Pr(t > t_0)} = \frac{Pr(t > t_0 + \theta)}{Pr(t > t_0)} = \\ &= \frac{e^{-\lambda(t_0 + \theta)}}{e^{-\lambda t_0}} = e^{-\lambda \theta} = Pr(t > \theta) \end{aligned}$$

<sup>9</sup>La ddp esponenziale è spesso adottata come un modello approssimato ma di facile applicazione per rappresentare un tempo di attesa, ed applicato ad esempio alla durata di una conversazione telefonica, oppure all'intervallo tra due malfunzionamenti di un apparato.

<sup>10</sup>Per quanto riguarda il valor medio  $m_E = \int_0^{\infty} t \lambda e^{-\lambda t} dt$  possiamo procedere *per parti*, ossia applicando la regola  $\int_a^b f'(t) g(t) dt = f(t) g(t) \Big|_a^b - \int_a^b f(t) g'(t) dt$ , avendo posto  $f'(t) = e^{-\lambda t}$  e  $g(t) = \lambda t$ : si ottiene allora

$$m_E = -\frac{1}{\lambda} e^{-\lambda t} \cdot \lambda t \Big|_0^{\infty} - \int_0^{\infty} -\frac{1}{\lambda} e^{-\lambda t} \cdot \lambda dt = -0 + 0 - \frac{1}{\lambda} e^{-\lambda t} \Big|_0^{\infty} = \frac{1}{\lambda}$$

essendo  $\lim_{t \rightarrow \infty} e^{-\lambda t} \cdot \lambda t = 0$ . Per  $\sigma_E^2 = \int_0^{\infty} t^2 \lambda e^{-\lambda t} dt - (m_E)^2$ , il primo integrale (sempre procedendo per parti) fornisce  $\int_0^{\infty} t^2 \lambda e^{-\lambda t} dt = \frac{2}{\lambda^2}$ , e dunque  $\sigma_E^2 = \frac{2}{\lambda^2} - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}$ .

<sup>11</sup>Consideriamo un ospedale in cui nascono *in media* 6 bimbettini al giorno (o 0.25 nascite l'ora), e consideriamo l'intervallo tra questi eventi come una v.a. completamente casuale. Se assumiamo che la probabilità di  $k$  nascite in un tempo  $T$  sia descritta da una v.a. di Poisson, ossia a cui compete una probabilità  $p_p(k) = e^{-\lambda T} \frac{(\lambda T)^k}{k!}$ , allora la probabilità che durante un tempo  $T$  non avvenga nessuna nascita,

**Esempio** Se la durata media di una telefonata è di 5 minuti, e la durata complessiva è completamente casuale, quale è la probabilità che la stessa duri più di 20 minuti?

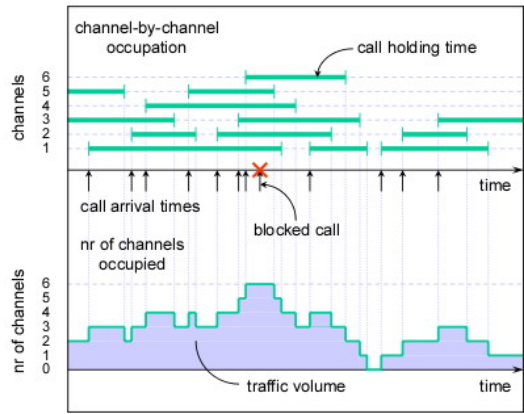
*Risposta:* ci viene fornito un tempo di attesa medio  $\tau_a$ , a cui corrisponde una frequenza di servizio  $\lambda = \frac{1}{\tau_a}$ , e quindi la soluzione risulta  $Pr(t > 20) = \int_{20}^{\infty} \frac{1}{\tau_a} e^{-t/\tau_a} dt = e^{-20/5} = 0.0183 = 1.83\%$ .

Un corollario<sup>12</sup> della (22.3) è che, se  $t_0 \rightarrow 0$ , allora la probabilità che si verifichi un evento entro un tempo  $t_0$ , è *direttamente proporzionale* (a meno di un infinitesimo di ordine superiore di  $t_0$ ) al valore di  $t_0$ , ossia

$$Pr(t \leq t_0)|_{t_0 \rightarrow 0} = \lambda t_0 + o(t_0) \tag{22.4}$$

### 22.3 Sistema di servizio orientato alla perdita

Un *sistema di servizio* è una entità in grado di accogliere delle *richieste di servizio*, ovvero eventi che definiscono il cosiddetto *processo di ingresso* al sistema, fino al raggiungimento della capacità limite, determinata dal numero  $M$  di *serventi* di cui il sistema dispone<sup>13</sup>. Una volta occupati tutti i serventi, e finché non se ne libera qualcuno, le successive richieste possono essere poste in coda, individuando così un sistema *orientato al ritardo* (che affrontiamo al § 22.4), oppure rifiutate (vedi la figura a fianco), come avviene per i sistemi *orientati alla perdita*. Scopo della presente sezione è quindi quello di determinare il numero di serventi necessario a garantire una *probabilità di rifiuto* della richiesta di servizio pari ad un valore che descrive il *grado di servizio* che si intende fornire.



richieste di servizio e occupazione serventi nel tempo

#### 22.3.1 Frequenza di arrivo e di servizio

Mentre il processo di ingresso è descritto in termini della *frequenza media* di arrivo  $\lambda$ , il tempo medio di occupazione dei serventi (indicato come *processo di servizio*) è

dovrebbe corrispondere a calcolare  $p_p(0)$ , ovvero  $e^{-\lambda T} \frac{(\lambda T)^0}{0!} = e^{-\lambda T}$ , che è esattamente il risultato che fornisce la v.a. esponenziale per la probabilità  $Pr(t > T)$  che non vi siano nascite per un tempo  $T$ .

<sup>12</sup>La dimostrazione della (22.4) si basa sulla considerazione che  $Pr(t \leq t_0) = 1 - Pr(t > t_0)$ , e sulla espansione in serie di potenze  $e^x = 1 + x + x^2/2 + x^3/3! + \dots$  che si riduce a  $e^x = 1 + x + o(t_0)$  se  $x \rightarrow 0$ . Pertanto la (22.3) diviene  $Pr(t > t_0)|_{t_0 \rightarrow 0} = 1 - \lambda t_0 + o(t_0)$ , e quindi  $Pr(t \leq t_0) = 1 - 1 + \lambda t_0 + o(t_0) = \lambda t_0 + o(t_0)$ .

<sup>13</sup>Gli esempi dalla vita reale sono molteplici, dal casello autostradale presso cui arrivano auto richiedenti il servizio del casellante ( $M$ =numero di caselli aperti), al distributore automatico di bevande (servente unico), all'aereo che per atterrare richiede l'uso della pista (servente unico)... nel contesto delle telecomunicazioni, il modello si applica ogni qualvolta vi siano un numero limitato di risorse a disposizione, come ad esempio (ma non solo!) il numero di linee telefoniche uscenti da un organo di commutazione, od il numero di *time-slot* presente in una trama PCM, od il numero di operatori di una *call-center*...

descritto nei termini del *tempo medio di servizio*  $\tau_S$ , ovvero dal suo inverso  $\mu = 1/\tau_S$ , pari alla *frequenza media* di servizio. Nella trattazione seguente si fa l'ipotesi che entrambi i processi (di ingresso e di servizio) siano descrivibili in termini di v.a. a distribuzione esponenziale<sup>14</sup>, ovvero che le durate degli eventi "nuova richiesta" e "servente occupato" siano *completamente casuali*<sup>15</sup>.

### 22.3.2 Intensità media di traffico

Il rapporto  $A_o = \frac{\lambda}{\mu}$  è indicato come *intensità media* del traffico *offerto*<sup>16</sup> e descrive quanti serventi (in media) *sarebbero* occupati ad espletare le richieste arrivate e non ancora servite, nel caso in cui  $M$  fosse infinito. L'aggettivo *offerto* indica la circostanza che, essendo invece  $M$  finito, alcune richieste non sono accolte, ed  $A_o$  risulta diverso dal traffico  $A_s$  che può essere effettivamente *smaltito*. L'unità di misura dell'intensità di traffico è l'ERLANG, il cui valore indica appunto il numero medio di serventi occupati.

**Esempio** Ad un centralino giungono una media di  $\lambda = 3$  chiamate al minuto, e la durata media di una conversazione è  $1/\mu = 3$  minuti. In tal caso l'intensità media di traffico risulta  $A_o = 3 \cdot 3 = 9$  Erlang, corrispondenti al potenziale impegno di una *media* di 9 centralinisti (e nove linee telefoniche).

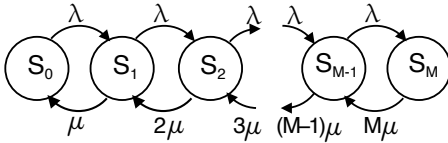
### 22.3.3 Probabilità di rifiuto

La teoria che porta a determinare la probabilità che una nuova richiesta di servizio non possa essere accolta a causa dell'esaurimento dei serventi si basa sulla descrizione di un cosiddetto *processo di nascita e morte*, che rappresenta da un punto di vista statistico l'evoluzione di una popolazione, nei termini di una frequenza di nascita (nuova conversazione) e di morte (termine della conversazione). Istante per istante, il numero esatto di individui della popolazione può variare, ma in un istante a caso, possiamo pensare alla numerosità della popolazione come ad una variabile aleatoria discreta, descritta in base ai valori di probabilità  $p_k$  che la popolazione assommi esattamente a  $k$  individui. La determinazione di questi valori  $p_k$  dipende dalla caratterizzazione dei processi di ingresso e di servizio, e nel caso in cui questi siano descritti da v.a. esponenziali (o poissoniane, a seconda se ci riferiamo ai tempi medi di interarrivo/partenza, od al loro numero medio per unità di tempo) si può procedere nel modo che segue.

<sup>14</sup>L'ipotesi permette di valutare la probabilità che l'intervallo temporale tra due eventi di ingresso sia superiore a  $\theta$ , in base alla (22.3), come  $e^{-\lambda\theta}$  (ad esempio, la prob. che tra due richieste di connessione in ingresso ad una centrale telefonica passi un tempo almeno pari a  $\theta$ ); allo stesso modo, la probabilità che il servizio abbia una durata maggiore di  $\theta$  è pari a  $e^{-\mu\theta}$  (ad esempio, la prob. che una telefonata duri più di  $\theta$ ).

<sup>15</sup>Le ipotesi poste fanno sì che i risultati a cui giungeremo siano conservativi, ovvero il numero di serventi risulterà maggiore od uguale a quello realmente necessario; l'altro caso limite (di attese deterministiche) corrisponde a quello in cui il tempo di servizio non varia, ma è costante, come ad esempio il caso del tempo necessario alla trasmissione di una cella ATM di dimensioni fisse. In questi casi, la stessa intensità media di traffico  $A_o = \frac{\lambda}{\mu}$  può essere gestita con un numero molto ridotto di serventi; nella realtà, ci si troverà in situazioni intermedie.

<sup>16</sup>Si noti che il pedice  $o$  è una "o" e non uno "0", ed identifica appunto l'aggettivo *offerto*.



Descriviamo innanzitutto l'evoluzione dello stato del sistema, in cui il numero di *serventi occupati* evolve aumentando o diminuendo di una unità alla volta (come per i

processi di nascita e morte), con l'ausilio della figura, dove il generico stato \$S\_k\$ rappresenta la circostanza che \$k\$ serventi siano occupati, circostanza a cui compete una probabilità \$p\_k = Pr(S\_k)\$.

Gli stati del grafo sono collegati da archi etichettati con la frequenza \$\lambda\$ delle transizioni tra gli stati, ovvero dal ritmo con cui si passa da \$S\_k\$ a \$S\_{k+1}\$ a causa di una nuova richiesta, indipendente (per ipotesi) dal numero di serventi già occupati, e dal ritmo \$(k + 1) \cdot \mu\$ con cui si torna da \$S\_{k+1}\$ ad \$S\_k\$, a causa del termine del servizio espletato da uno tra i \$k + 1\$ serventi occupati, e proporzionale quindi a questo numero<sup>17</sup>. Se \$\lambda\$ e \$\mu\$ non variano nel tempo, una volta esaurito un transitorio iniziale il sistema di servizio si troverà in *condizioni stazionarie*, permettendoci di scrivere le *equazioni di equilibrio statistico*

$$\lambda p_k = \mu (k + 1) p_{k+1} \quad \text{con } k = 0, 1, 2, \dots, M - 1 \tag{22.5}$$

che eguagliano la frequenza media con cui il sistema evolve dallo stato \$k\$ verso \$k + 1\$, alla frequenza media con cui avviene la transizione inversa<sup>18</sup>. La (22.5) può essere riscritta come

$$p_{k+1} = \frac{\lambda}{\mu (k + 1)} p_k = \frac{A_0}{(k + 1)} p_k$$

che applicata ricorsivamente, porta a scrivere

$$p_k = \frac{A_0^k}{k!} p_0 \tag{22.6}$$

Non resta ora che trovare il modo per dare un valore a \$p\_0\$, e questo è oltremodo semplice, ricordando che deve risultare<sup>19</sup> \$1 = \sum\_{m=0}^M p\_m = p\_0 \sum\_{m=0}^M \frac{A\_0^m}{m!}\$, e quindi

$$p_0 = \left( \sum_{m=0}^M \frac{A_0^m}{m!} \right)^{-1} \tag{22.7}$$

Nei due casi distinti in cui i serventi siano in numero finito (e pari ad \$M\$) od infinito (\$M = \infty\$) otteniamo rispettivamente il caso cercato, ed un caso limite. Se poniamo

<sup>17</sup>Pensiamo ad un ufficio postale visto dall'esterno: la frequenza media \$\lambda\$ con cui entrano nuove persone non dipende da quanti siano già all'interno, mentre invece la frequenza con la quale escono dipende sia dal tempo medio \$1/\mu\$ di permanenza allo sportello, che dal numero di sportelli (serventi) \$M\$ in funzione. La differenza con il caso che stiamo trattando deriva dal fatto che l'ufficio postale è un sistema a coda, e dato che la coda *c'è praticamente sempre* (ossia i serventi sono generalmente tutti occupati) possiamo dire che la frequenza media di uscita è proprio \$M\mu\$.

<sup>18</sup>E' un po' come se il numero medio di nuove richieste per unità di tempo \$\lambda\$ si distribuisse, in accordo alle probabilità \$p\_k\$, tra tutti gli stati possibili del sistema: come dire che del totale di \$\lambda\$, una parte \$\lambda p\_0\$ trovano il sistema vuoto, una parte \$\lambda p\_1\$ con un solo occupante, eccetera. Per quanto riguarda le richieste servite per unità di tempo, la frequenza di uscita dal sistema è quella che si otterrebbe con un unico servente, moltiplicata per il numero di serventi occupati. Dato che questa ultima quantità è una grandezza probabilistica, la reale frequenza di uscita \$\mu\_r\$ può essere valutata come valore atteso, ossia \$\mu\_r = \sum\_{k=1}^M \mu \cdot k \cdot p\_k\$

<sup>19</sup>Usiamo il pedice \$m\$ anziché \$k\$ per non creare confusione nella (22.8)

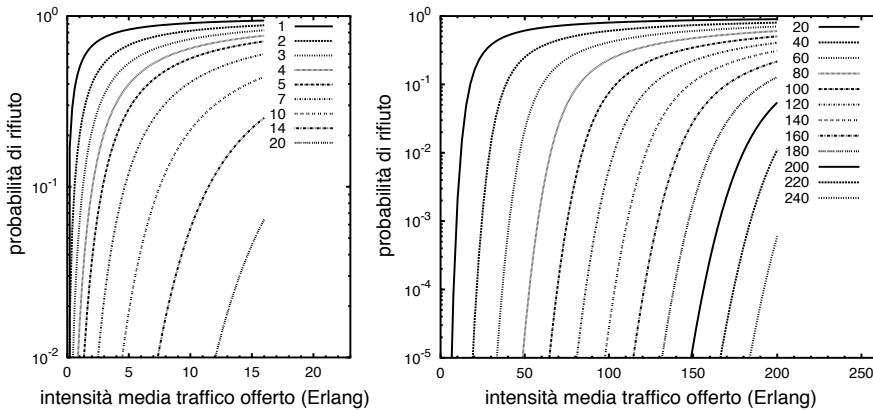


Figura 22.1: Valori della probabilità di blocco  $P_B$  in un sistema orientato alla perdita, al variare di  $A_0$ , per il numero di serventi indicato sulle curve

$M = \infty$ , tenendo conto dell'espansione in serie  $\sum_{m=0}^{\infty} \frac{A_0^m}{m!} = e^{A_0}$ , si ottiene che la (22.7) fornisce appunto  $p_0 = e^{-A_0}$ , e la (22.6) diviene  $p_k = e^{-A_0} \frac{A_0^k}{k!}$ , che come riconosciamo immediatamente è proprio la ddp di Poisson (22.2) con valore medio  $A_0$ <sup>20</sup>. Se invece poniamo  $M$  finito, la sommatoria che compare in (22.7) non corrisponde ad una serie nota, e dunque rimane come è, fornendo il risultato

$$p_k = Pr(S_k) = \frac{\frac{A_0^k}{k!}}{\sum_{m=0}^M \frac{A_0^m}{m!}}$$

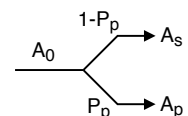
Notiamo ora che  $p_M$  è la probabilità che tutti i serventi siano occupati, pari dunque alla probabilità che una nuova richiesta di servizio sia rifiutata. Chiamiamo allora questo valore *probabilità di Blocco, di Rifiuto o di Perdita*, la cui espressione prende il nome di **FORMULA B DI ERLANG**, del primo tipo, di ordine  $M$  ed argomento  $A_0$ :

$$P_B = Pr(S_M) = p_M = \frac{\frac{A_0^M}{M!}}{\sum_{m=0}^M \frac{A_0^m}{m!}} = E_{1,M}(A_0) \tag{22.8}$$

L'andamento di  $P_B$  in funzione di  $M$  e di  $A_0$  è graficato in Fig. 22.1, e mostra come (ad esempio) per una intensità di traffico offerto pari a 40 Erlang, siano necessari più di 50 serventi per mantenere una  $P_B$  minore dell'1%, che salgono a più di 60 per una  $P_B = 10^{-3}$ .

### 22.3.4 Efficienza di giunzione

In presenza di una intensità media di traffico offerto  $A_0$ , ed una probabilità di perdita  $P_p = P_B$ , solamente il  $(1 - P_p) \cdot 100\%$  delle richieste è smaltito, e quindi  $A_0$  si ripartisce tra l'intensità media di *traffico smaltito*  $A_s = A_0 (1 - P_p)$ , e l'intensità media



<sup>20</sup>Questo risultato è in perfetto accordo con la (22.2), quando abbiamo sostituito alla ddp di Bernoulli quella di Poisson, mantenendo inalterato il numero medio di serventi occupati, che ora indichiamo con  $A_0$ , come definito al § 22.3.2.

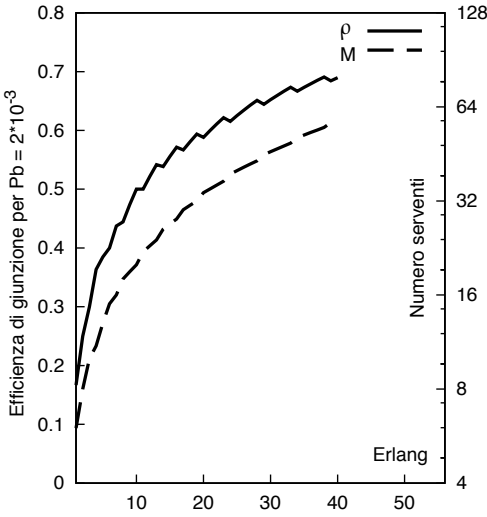


Figura 22.2: Efficienza di giunzione

l'efficienza aumenta con l'intensità di traffico offerto, e per questo i collegamenti (*giunzioni*) in grado di smaltire un numero più elevato di connessioni, garantiscono anche una maggiore economicità di esercizio.

### 22.3.5 Validità del modello

Le considerazioni espone si riferiscono ad una ipotesi di traffico completamente casuale con tempi di interarrivo e di servizio esponenziali<sup>22</sup>, ossia con un processo di traffico incidente di Poisson. In queste ipotesi, il rapporto  $\frac{\sigma_p^2}{m_p} = 1$  tra la varianza e la media delle distribuzioni di Poisson, è rappresentativo appunto di un traffico *completamente casuale*.

Del tutto diversa può risultare l'analisi nel caso di una giunzione usata solo nel caso di trabocco del traffico da una giunzione piena. In questo caso  $\lambda$  non è più costante, anzi aumenta con l'aumentare delle connessioni già avvenute, tipico di *traffico a valanga*<sup>23</sup>.

**Esempio** Un numero molto elevato di sorgenti analogiche condivide uno stesso mezzo trasmissivo, caratterizzato da una capacità complessiva netta di 25.6 Mbps. Le sorgenti sono campionate a frequenza  $f_c = 21.33$  KHz e con una risoluzione di 12 bit/campione; ogni sorgente trasmette ad istanti casuali per un tempo casuale, quindi gli intervalli di interarrivo e di servizio sono entrambi v.a. a distribuzione esponenziale negativa, di valor medio rispettivamente  $\lambda = 20$  richieste/minuto e  $\frac{1}{\mu} = 4.25$  minuti.

<sup>21</sup>ovvero, all'aumentare del traffico offerto,  $M$  aumenta più lentamente di  $A_0$ . Ad esempio, dalla figura si può verificare che se per  $A_0 = 10$  occorrono circa 21 serveri, per una intensità doppia  $A_0 = 20$  il numero di serveri necessario a mantenere la stessa  $P_B$  risulta poco più di 32.

<sup>22</sup>In effetti, è stato dimostrato che i risultati ottenuti per i sistemi di servizio orientati alla perdita possono essere considerati validi anche nel caso di tempi di servizio a distribuzione qualsiasi, non necessariamente esponenziale.

<sup>23</sup>Un esempio di tale tipo di traffico potrebbe essere... l'uscita da uno stadio (o da un cinema, una metropolitana,...) in cui il flusso di individui non è casuale, ma aumenta fino a saturare le vie di uscita.

di *traffico perso*  $A_p = A_0 P_p$ . Possiamo definire un coefficiente di utilizzazione, o efficienza

$$\rho = \frac{A_s}{M} = \frac{A_0}{M} (1 - P_p)$$

che rappresenta la percentuale di impegno dei serveri, e di cui la figura 22.2 mostra l'andamento al variare di  $A_0$ , per una  $P_B$  assegnata e pari a  $2 \cdot 10^{-3}$ , assieme al numero di serveri necessario a garantire tale probabilità di blocco.

Come si può osservare, una volta fissato il grado di servizio, all'aumentare del numero di serveri il traffico smaltito cresce più in fretta di quanto non crescano i serveri<sup>21</sup>, cosicché (a parità di  $P_p$ ) l'efficienza

1. Determinare la  $f_b$  di una sorgente nelle fasi di attività;
2. determinare il numero massimo di sorgenti contemporaneamente attive;
3. determinare il grado di servizio (Probabilità di rifiuto) ottenibile con il mezzo trasmissivo indicato;
4. indicare la capacità da aggiungere al collegamento per garantire un grado di servizio cento volte migliore.

- Risposte**
1.  $f_b = \frac{\text{bit}}{\text{campione}} \cdot \frac{\text{campioni}}{\text{secondo}} = 12 \cdot 21.33 \cdot 10^3 = 256 \text{ Kbps}$ ;
  2. Il numero massimo di sorgenti contemporaneamente attive coincide con il numero di serventi  $M$  del collegamento, e quindi  $M = \frac{25.6 \cdot 10^6}{256 \cdot 10^3} = 100$  serventi;
  3. L'intensità media di traffico offerto risulta pari a  $A_o = \frac{\lambda}{\mu} = \frac{20}{1/4.25} = 85$  Erlang, e pertanto dalle curve di Fig. 22.1 si trova una probabilità di rifiuto pari a circa  $10^{-2}$ ;
  4. Si richiede quindi una probabilità di rifiuto 100 volte inferiore, e cioè pari a  $10^{-4}$ : si ottiene che la banda deve essere aumentata del 20%. Infatti, dalle curve di Fig. 22.1 si osserva che ciò richiede (a parità di  $A_o$ ) almeno 120 (circa) serventi, 20 in più, pari ad una capacità aggiuntiva di  $20 \cdot 256 \cdot 10^3 = 5.12 \text{ Mbps}$ .

## 22.4 Sistemi di servizio orientati al ritardo

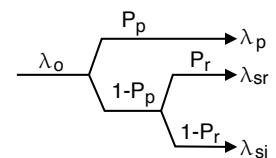
Mentre i sistemi orientati alla perdita rappresentano il modo di operare delle reti di telecomunicazione a *commutazione di circuito*, in cui ogni connessione impegna in modo esclusivo alcune risorse di rete, che una volta esaurite producono un *rifiuto* della richiesta di connessione, i sistemi *orientati al ritardo* sono rappresentativi di reti a *commutazione di pacchetto*, in cui i messaggi sono suddivisi in unità elementari (detti pacchetti, appunto) la cui ricezione non deve più avvenire in tempo reale, e che condividono le stesse risorse fisiche (degli organi di commutazione e di trasmissione) con i pacchetti di altre comunicazioni. Pertanto, l'invio di un pacchetto può essere *ritardato* se il sistema di servizio è in grado di gestire delle *code di attesa*, in cui accumulare le richieste che eccedono il numero di serventi a disposizione, e da cui prelevare (con ritardo) i pacchetti stessi non appena si rendano disponibili le risorse trasmissive necessarie.

In questo caso lo schema che esemplifica la ripartizione dei flussi di richieste si modifica come in figura, dove è evidenziato come la frequenza di richieste  $\lambda_o$  si suddivide tra la frequenza delle richieste perse  $\lambda_p$ , quelle servite con ritardo  $\lambda_{sr}$ , e quelle servite immediatamente  $\lambda_{si}$ , in funzione della probabilità di perdita  $P_p$  e di ritardo  $P_r$ . Nei termini di queste quantità, valgono le relazioni:

$$\lambda_p = P_p \lambda_o; \quad \lambda_{sr} = P_r (1 - P_p) \lambda_o; \quad \lambda_{si} = (1 - P_r) (1 - P_p) \lambda_o$$

Indicando con  $\tau_S = \frac{1}{\mu}$  il tempo medio di servizio di ogni richiesta, (che non comprende quindi il tempo di accodamento), si definisce, come già noto, una intensità di traffico offerto  $A_o = \frac{\lambda_o}{\mu} = \lambda_o \tau_S$ , che deve risultare

$$A_o = A_p + A_{sr} + A_{si} \quad \text{e quindi} \quad A_{sr} = \frac{\lambda_{sr}}{\mu}, \quad A_{si} = \frac{\lambda_{si}}{\mu}$$



Considerando il caso in cui la coda abbia una lunghezza finita e pari ad  $L$ , osserviamo che, a prima vista, anche le  $L$  richieste successive all'impegno di tutti gli  $M$  serventi sono accolte (e poste in coda), come se i serventi fossero divenuti  $M + L$ . In realtà l'analisi fornisce risultati differenti, in quanto le richieste accodate devono essere ancora servite, e quindi il calcolo della  $P_p$  non è una diretta estensione dei risultati ottenuti per i sistemi orientati alla perdita. E' comunque abbastanza semplice verificare<sup>24</sup> che ora la  $P_p$  risulta inferiore alla  $P_B$  del caso senza coda, e pertanto l'intensità di traffico smaltito  $A_s = A_{sr} + A_{si} = (1 - P_p) A_o$  aumenta, a parità di offerta.

### 22.4.1 Risultato di Little

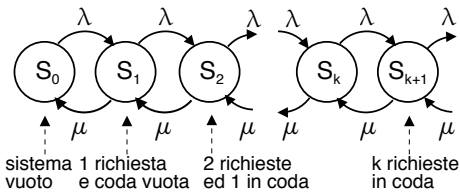
Si tratta di un risultato molto generale, valido per qualsiasi distribuzione dei tempi di interarrivo e di servizio, la cui applicazione può tornare utile nell'analisi, e che recita:

*Il numero medio  $\bar{N}$  di utenti contemporaneamente presenti in un sistema di servizio è pari al prodotto tra frequenza media di smaltimento delle richieste  $\lambda_s$  ed il tempo medio di permanenza  $\tau_p$  dell'utente nel sistema*

e quindi in definitiva  $\bar{N} = \lambda_s \cdot \tau_p$ . Nell'applicazione al caso di servizi orientati alla perdita, si ha  $\tau_p = \tau_s$ , mentre nei servizi a coda risulta  $\tau_p = \tau_c + \tau_s$  in cui  $\tau_c$  rappresenta il tempo medio di coda.

### 22.4.2 Sistemi a coda infinita ed a servente unico

Prima di fornire risultati più generali, svolgiamo l'analisi per questo caso particolare, in cui la frequenza di richieste perse  $\lambda_p$  è nulla, dato che una coda di lunghezza infinita



le accoglie comunque tutte. Da un punto di vista statistico un tale sistema è descritto mediante il diagramma di nascita e morte riportato a fianco, in cui ogni stato  $S_k$  rappresenta  $k$  richieste nel sistema, di cui una sta ricevendo servizio e  $k - 1$  sono accodate.

Per procedere nell'analisi si applica lo stesso principio di equilibrio statistico già adottato a pag. 770 dove si asserisce che, esaurito un periodo transitorio iniziale, la frequenza media delle transizioni tra  $S_k$  e  $S_{k+1}$  deve eguagliare quella da  $S_{k+1}$  ad  $S_k$ . Indicando con  $p_k = Pr(S_k)$  la probabilità che il sistema contenga  $k$  richieste, l'equilibrio statistico si traduce nell'insieme di equazioni

$$\lambda_0 p_k = \mu p_{k+1} \quad \text{con } k = 0, 1, 2, \dots, \infty \tag{22.9}$$

Infatti, in base alle stesse considerazioni svolte nella prima parte della nota 18 di pag. 770,  $\lambda_0 p_k$  è pari alla frequenza media (frazione di  $\lambda_0$ ) con cui il numero di richieste accolte passa da  $k$  a  $k + 1$ ; essendo il servente unico, la frequenza di servizio è sempre

<sup>24</sup>Se  $P_B$  è la probabilità di blocco derivante dalla disponibilità di  $M$  serventi, una frequenza di richieste pari a  $P_B \cdot \lambda_o$  non può essere servita immediatamente; adottando una coda, la frequenza delle richieste non servite immediatamente  $P_B \cdot \lambda_o$  è uguale a  $\lambda_o (P_p + P_r (1 - P_p))$ , ed eguagliando le due espressioni si ottiene  $P_p = \frac{P_B - P_r}{1 - P_r}$ , che è sempre minore di  $P_B$ .

$\mu = \frac{1}{\tau_s}$ , indipendentemente dal numero di richieste accodate, e dunque  $\mu p_{k+1}$  è proprio la frequenza media con cui il sistema passa da  $k+1$  a  $k$  richieste accolte.

La relazione (22.9) è di natura ricorsiva, e può esprimersi come

$$p_k = \left(\frac{\lambda_o}{\mu}\right)^k p_0 = A_o^k p_0$$

Per determinare il valore  $p_0 = Pr(S_0)$ , uguale alla probabilità che il sistema sia vuoto, ricordiamo<sup>25</sup> che deve risultare

$$1 = \sum_{k=0}^{\infty} p_k = \sum_{k=0}^{\infty} p_0 A_o^k = p_0 \frac{1}{1 - A_o}$$

da cui otteniamo  $p_0 = 1 - A_o$  e dunque

$$p_k = (1 - A_o) A_o^k$$

che corrisponde ad una densità di probabilità esponenziale discreta.

Siamo ora in grado di determinare alcune grandezze di interesse:

**Probabilità di ritardo**  $P_r$ : risulta pari alla probabilità che il sistema non sia vuoto, e cioè che ci sia già almeno una richiesta accolta, ed è pari a<sup>26</sup>

$$P_r = 1 - p_0 = 1 - (1 - A_o) = A_o$$

Ricordiamo di aver già definito l'efficienza come il rapporto  $\rho = \frac{A_s}{M}$  tra il traffico smaltito ed il numero dei serventi; nel nostro caso  $M = 1$  e  $A_s = A_o$ : dunque  $\rho = A_o$ . Pertanto, il risultato  $P_r = A_o = \rho$  indica come, al tendere ad 1 dell'efficienza, la probabilità di ritardo tenda anch'essa ad 1.

**Lunghezza media di coda** indicata con  $\bar{L}$ : risulta essere semplicemente il valore atteso del numero di richieste presenti nel sistema, ovvero<sup>27</sup>

$$\bar{L} = E\{k\} = \sum_{k=0}^{\infty} k p_k = (1 - A_o) \sum_{k=0}^{\infty} k A_o^k = \frac{A_o}{1 - A_o}$$

da cui risulta che per  $A_o \rightarrow 1$  la coda tende ad una lunghezza infinita.

**Tempo medio di permanenza** indicato con  $\tau_p$ , e scomponibile nella somma  $\tau_p = \tau_s + \tau_c$  tra il tempo medio di servizio ed il tempo medio di coda. Possiamo applicare qui il risultato di Little  $\bar{N} = \lambda_s \cdot \tau_p$ , che esprime la relazione tra numero medio  $\bar{N}$  di richieste presenti, frequenza di smaltimento (qui pari a quella di

<sup>25</sup>Nella derivazione del risultato si fa uso della relazione  $\sum_{k=0}^{\infty} \alpha^k = \frac{1}{1-\alpha}$ , nota con il nome di *serie geometrica*, e valida se  $\alpha < 1$ , come infatti risulta nel nostro caso, in quanto necessariamente deve risultare  $A_o = \frac{\lambda_o}{\mu} < 1$ ; infatti se il servente è unico una frequenza di arrivo maggiore di quella di servizio preclude ogni speranza di funzionamento, dato che evidentemente il sistema non ha modo di smaltire in tempo le richieste che si presentano.

<sup>26</sup>Ricordiamo che  $p_0$  è la probabilità che il sistema sia vuoto, e dunque  $1 - p_0$  quella che *non* sia vuoto.

<sup>27</sup>si fa uso della relazione  $\sum_{k=0}^{\infty} k \alpha^k = \alpha \sum_{k=0}^{\infty} k \alpha^{k-1} = \alpha \frac{\partial}{\partial \alpha} \sum_{k=0}^{\infty} \alpha^k = \alpha \frac{\partial}{\partial \alpha} \frac{1}{1-\alpha} = \frac{\alpha}{(1-\alpha)^2}$

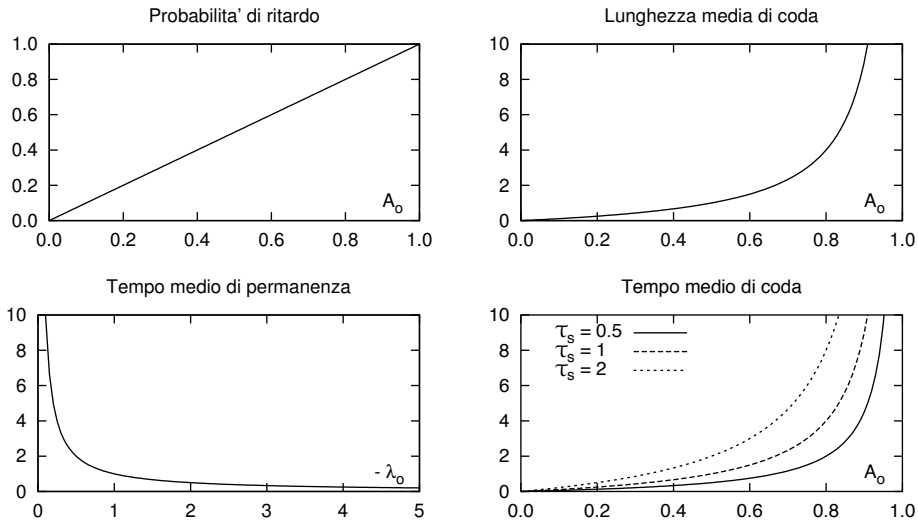


Figura 22.3: Grandezze di interesse per il sistema a coda infinita ed unico servente

offerta<sup>28</sup>), e tempo medio di permanenza; infatti accade che  $\bar{N} = \bar{L}$ , ed utilizzando il risultato  $\bar{L} = \frac{A_0}{1-A_0}$  si ottiene

$$\tau_p = \frac{\bar{N}}{\lambda_s} = \frac{\bar{L}}{\lambda_0} = \frac{A_0}{1-A_0} \frac{1}{\lambda_0} = \frac{\lambda_0}{\mu} \frac{1}{\lambda_0} \frac{1}{1-\lambda_0/\mu} = \frac{1}{\mu - \lambda_0}$$

da cui si osserva che, se la frequenza di offerta tende al valore della frequenza di servizio, il tempo medio di permanenza tende ad  $\infty$ .

**Tempo medio di coda** si calcola come

$$\tau_c = \tau_p - \tau_s = \frac{1}{\mu - \lambda_0} - \frac{1}{\mu} = \frac{\mu - \mu + \lambda_0}{\mu(\mu - \lambda_0)} = \frac{A_0}{\mu(1-A_0)} = \frac{1}{\mu} \frac{\rho}{1-\rho} = \tau_s \frac{\rho}{1-\rho}$$

Questo risultato mostra che il tempo medio di coda è legato al tempo medio di servizio e all'efficienza di giunzione, confermando ancora i risultati per  $\rho \rightarrow \infty$ .

La fig. 22.3 mostra l'andamento delle grandezze appena calcolate.

### 22.4.3 Sistemi a coda finita e con più serventi

Riportiamo solo i risultati, validi se entrambi i processi di ingresso e di servizio sono esponenziali con frequenza media  $\lambda_0$  e  $\mu$ , la coda è lunga  $L$ , i serventi sono  $M$  e le sorgenti infinite.

**Probabilità di k richieste nel sistema**

$$p_k(A_0) = \begin{cases} \frac{A_0^k}{k! \alpha(A_0)} & 0 \leq k \leq M \\ \frac{A_0^k}{M^{k-M} M! \alpha(A_0)} & M \leq k \leq M + L \end{cases}$$

<sup>28</sup>Non può essere  $\lambda_s > \lambda_0$ , perché si servirebbero più richieste di quante se ne presentano. Se fosse invece  $\lambda_s < \lambda_0$ , la coda crescerebbe inesorabilmente e sarebbe quindi inutile.

in cui  $\alpha(A_0) = \frac{1}{\rho_0(A_0)} = \sum_{k=0}^{M+L} \frac{A_0^k}{k!}$  e  $A_0 = \frac{\lambda_0}{\mu}$ . Si noti come per  $0 \leq k \leq M$  ed  $L = 0$  si ottenga lo stesso risultato già esposto per i sistemi orientati alla perdita, mentre per  $M = 1$  ed  $L = \infty$  ci si riconduca al caso precedentemente analizzato.

**Probabilità di ritardo**

$$P_r = \sum_{k=M}^{M+L} p_k(A_0) = p_M(A_0) \frac{1 - \rho^{L+1}}{1 - \rho} \quad \text{in cui} \quad \rho = \frac{A_0}{M}$$

**Probabilità di perdita**

$$P_p = p_{M+L}(A_0) = \frac{A_0^{M+L}}{M^L M! \cdot \alpha(A_0)}$$

**Tempo medio di coda**

$$\tau_c = \tau_S \frac{P_r - L \cdot P_{M+L}(A_0)}{M - A_0}$$

La Figura 22.4 descrive la probabilità di perdita per un sistema a servente singolo (a sinistra) e con 10 serventi (a destra), in funzione dell'intensità di traffico offerto e della lunghezza di coda  $L$ , così come risulta dalla applicazione delle formule riportate. Nel caso di trasmissione di pacchetti di lunghezza fissa, per i quali il tempo di servizio è fisso e *non* a distribuzione esponenziale<sup>29</sup>, i risultati ottenuti costituiscono una *stima conservativa* delle prestazioni del sistema (che potranno cioè essere migliori). L'analisi delle curve permette di valutare con esattezza il vantaggio dell'uso di una coda (a spese del tempo di ritardo). Infatti, aumentando il numero di posizioni di coda si mantiene una probabilità di blocco accettabile anche per traffico intenso.

Ad esempio, per  $P_b = 1\%$  ed  $M = 1$ , osserviamo che una coda con  $L = 20$  posizioni gestisce un traffico di  $A_0 = 0.83$  Erlang, contro gli  $A_0 = 0.11$  Erlang del caso senza coda.

<sup>29</sup>In una trasmissione a pacchetto, operata a frequenza binaria  $f_b$  e con pacchetti di lunghezza media  $\bar{L}_p$  bit, il tempo medio di servizio per un singolo pacchetto è pari a quello medio necessario alla sua trasmissione, e cioè  $\tau_S = \bar{L}_p / f_b$ .

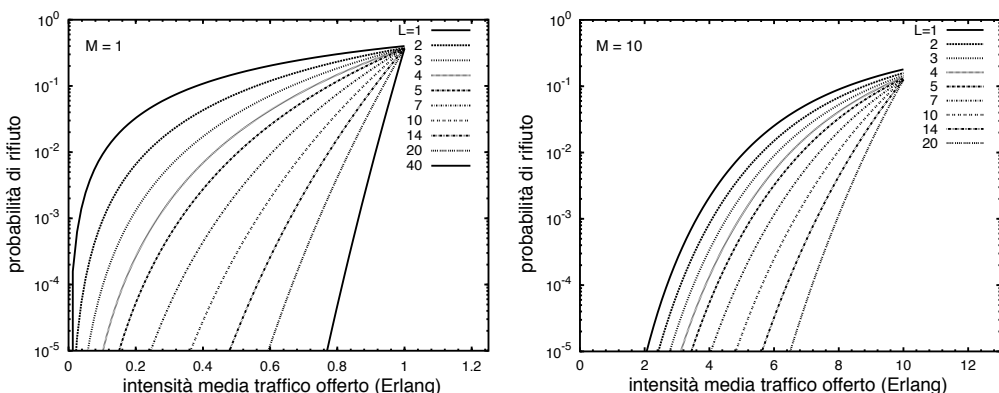


Figura 22.4: Probabilità di perdita per un sistema a coda finita con uno o dieci serventi

Ciò corrisponde ad un aumento dell'efficienza di  $\frac{0.83}{0.11} = 7.54$  volte. D'altra parte ora il tempo medio di coda (calcolato in modo conservativo applicando la relazione per coda infinita) è  $\tau_c = \tau_S \frac{\rho}{1-\rho} = \frac{0.83}{1-0.83} \tau_S = 4.9 \tau_S$ , ed è quindi aumentato (rispetto a  $\tau_S$ ) di quasi 5 volte.

**Esercizio** Un nodo di una rete per dati effettua la moltiplicazione di pacchetti di dimensione media di 8 KByte<sup>30</sup> su collegamenti con velocità binaria  $f_b = 100$  Mbps<sup>31</sup>

- 1) Determinare il tempo medio di servizio di ogni singolo pacchetto;
- 2) determinare il tempo medio di interarrivo  $\tau_a$  tra pacchetti corrispondente ad un traffico di ingresso di 1200 pacchetti/secondo, e l'associata intensità  $A_o$ ;
- 3) assumendo che la dimensione dei pacchetti sia una v.a. con densità esponenziale negativa, così come il tempo di interarrivo tra pacchetti, e che la memoria del moltiplicatore sia così grande da approssimare le condizioni di coda infinita, determinare il ritardo medio di un pacchetto, ossia il tempo medio trascorso tra quando un pacchetto si presenta in ingresso al nodo e quando ne esce;
- 4) calcolare la quantità di memoria necessaria ad ospitare i dati che si accumulano in un intervallo temporale pari al ritardo medio, considerando pacchetti di lunghezza fissa e pari alla media.

**Risposte** 1) Il tempo medio di servizio di un pacchetto è pari al tempo occorrente per trasmettere:  $\tau_S = \frac{1}{\mu} = \text{durata di un bit} \cdot \frac{\text{bit}}{\text{pacchetto}} = \frac{1}{10^8} \left[ \frac{\text{secondi}}{\text{bit}} \right] \cdot 1024 \left[ \frac{\text{byte}}{\text{pacchetto}} \right] \cdot 8 \left[ \frac{\text{bit}}{\text{byte}} \right] \approx 655 \mu\text{sec};$

2)  $\tau_a = \frac{1}{\lambda} = \frac{1}{1200} = 833 \mu\text{sec}; A_o = \frac{\lambda}{\mu} = 1200 \cdot 655 \cdot 10^{-6} = 0.786$  Erlang;

3) Le condizioni poste corrispondono a quelle di traffico poissoniano e sistema a singolo servente e coda infinita, per il quale la teoria fornisce per il tempo di permanenza il risultato  $\tau_p = \frac{1}{\mu - \lambda_o} = \frac{1}{\frac{10^6}{655} - \frac{10^6}{833}} = \frac{1}{326} \approx 3$  msec;

4) La memoria necessaria è pari al prodotto tra il tempo medio di permanenza ed il numero di bit che si accumulano in quel periodo, ovvero  $3 \cdot 10^{-3} [\text{sec}] \cdot 1200 \left[ \frac{\text{pacch}}{\text{sec}} \right] \cdot 1024 \left[ \frac{\text{byte}}{\text{pacch}} \right] \approx 3.7$  Kbyte.

## 22.5 Reti per trasmissione dati

In questa sezione illustriamo le particolarità legate alla *trasmissione dati*, e come possa essere vantaggiosamente sfruttata per conseguire la *maggiore efficienza* che i sistemi di servizio a coda presentano rispetto a quelli orientati alla perdita. Le particolari *modalità* e *funzioni* legate alle trasmissioni dati saranno classificate secondo uno schema che ne consente il confronto in termini di prestazioni e vincoli sulla realizzazione della rete. Infine, verranno formalizzati modelli operativi volti alla soluzione dei problemi di trasmissione dati, introducendo i concetti legati alle *architetture protocollari*, assieme ad alcuni esempi reali.

<sup>30</sup>1 byte = 8 bit, 1 K =  $2^{10} = 1024$ . Il "K" in questione è "un K informatico". Nel caso invece in cui ci si riferisca ad una velocità di trasmissione, il prefisso K torna a valere  $10^3 = 1000$ .

<sup>31</sup>In virtù di quanto esposto alla nota precedente, in questo caso  $1M = 10^6 = 1000000$ .

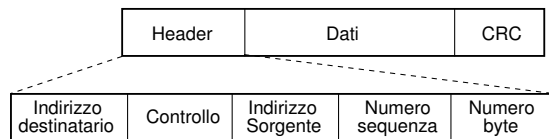
Le trasmissioni dati si prestano bene a comunicazioni in cui siano possibili ritardi temporali variabili, attuando una filosofia di tipo *ad immagazzinamento e rilancio* (STORE AND FORWARD) basata sul suddividere il messaggio in unità informative elementari denominate *pacchetti*, che possono essere inoltrati sulla rete di comunicazione assieme a quelli prodotti da altre trasmissioni. L'applicazione della stessa metodologia a trasmissioni (ad esempio) vocali non è scontato, in quanto la presenza di un ritardo variabile per la trasmissione dei pacchetti comporta problemi non trascurabili, a meno di attuare speciali meccanismi di priorità e prenotazione della banda.

### 22.5.1 Il pacchetto dati

Discutiamo brevemente, in termini generali, i possibili contenuti di un pacchetto dati; il suo formato effettivo dipenderà dal particolare protocollo di trasmissione adottato.

La prima osservazione da fare è che suddividendo il messaggio in pacchetti è necessario prevedere un aumento delle informazioni da trasmettere, dato che ognuno pacchetto dovrà contenere informazioni aggiuntive per consentire un suo corretto recapito e la sua ricombinazione con gli altri pacchetti dello stesso messaggio. Occorre inoltre affrontare gli ulteriori problemi tipici di una comunicazione dati, ovvero come contrastare gli errori di trasmissione, e come gestire le risorse di rete.

In termini generali, un pacchetto è composto da una *intestazione* (HEADER), dalla parte di messaggio che trasporta (*dati*), e da un campo *codice di parità* (CRC) necessario a ri-

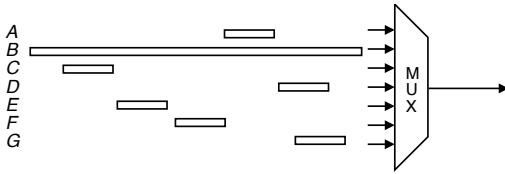


velare l'occorrenza di errori di trasmissione<sup>32</sup>. L'*header* a sua volta può essere suddiviso in campi, in cui trovano posto (tra le altre cose) gli *indirizzi* del destinatario e della sorgente, un *codice di controllo* che causa in chi lo riceve l'esecuzione di una procedura specifica, un *numero di sequenza* che identifica il pacchetto all'interno del messaggio originale, ed un campo che indica la *lunghezza* del pacchetto. Nonostante la presenza delle informazioni aggiuntive<sup>33</sup>, la trasmissione a pacchetto consegue una efficienza maggiore di quella a circuito, in quanto è attuata mediante sistemi a coda.

Può sembrare vantaggioso mantenere la dimensione dei pacchetti elevata, riducendo così la rilevanza delle informazioni aggiuntive, ma si verificano controindicazioni. Infatti, suddividere messaggi lunghi in pacchetti più piccoli garantisce l'inoltro di (altre) comunicazioni più brevi durante la trasmissione di messaggi lunghi, che altrimenti *bloccherebbero* i sistemi di coda se realizzate con un unico "pacchettone": in figura è mostrato un esempio in cui *B*, presentandosi in ingresso al multiplexer con lieve

<sup>32</sup>La sigla CRC significa *Cyclic Redundancy Check* (controllo ciclico di ridondanza) ed indica una parola binaria i cui bit sono calcolati in base ad operazioni algebriche (vedi § 15.6.3.3) attuate sui bit di cui il resto del messaggio è composto. Dal lato ricevente sono eseguite le stesse operazioni, ed il risultato confrontato con quello presente nel CRC, in modo da controllare la presenza di errori di trasmissione.

<sup>33</sup>L'entità delle informazioni aggiuntive rispetto a quelle del messaggio può variare molto per i diversi protocolli, da pochi bit a pacchetto fino ad un 10-20% dell'intero pacchetto (per lunghezze ridotte di quest'ultimo).



anticipo rispetto agli altri pacchetti più piccoli, ne impedisce l'inoltro, monopolizzando la linea di uscita per tutta la durata della sua trasmissione.

Infine, all'aumentare della lunghezza di un pacchetto aumenta proporzionalmente la probabilità di uno (o più) bit errati (vedi anche la formula (15.27) a pag. 468 e la discussione al § 22.6.2.3), e dunque l'uso di dimensioni contenute riduce le necessità di ritrasmissione.

## 22.5.2 Modo di trasferimento delle informazioni

È definito in base alla specificazione di 3 caratteristiche che lo contraddistinguono: lo *schema di multiploazione*, il *principio di commutazione* e l'*architettura protocollare*.

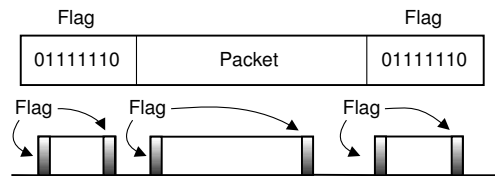
### 22.5.2.1 Schema di multiploazione

Al § 24.3.1 è descritto uno schema a divisione di tempo che prevede l'uso di una *trama* in cui trovano posto diverse comunicazioni vocali<sup>34</sup>, e che necessita di un funzionamento sincronizzato (o quasi) dei nodi di rete. La trasmissione *a pacchetto* invece non prevede l'uso esclusivo di risorse da parte delle singole comunicazioni e *non fa uso* di una struttura di trama; pertanto si rendono necessarie soluzioni idonee alla *delimitazione* dei pacchetti.

Ad esempio, i protocolli HDLC ed x.25<sup>35</sup> presentano pacchetti di dimensione variabile, e fanno uso di un byte di *flag* (vedi pag. 482) costituito dalla sequenza 01111110 in testa ed in coda, per separare tra loro i pacchetti di comunicazioni differenti.

Per evitare che i dati "propri" del pacchetto possano simulare un flag, in trasmissione viene inserito un bit 0 dopo 5 uni di fila, che (se presente) viene rimosso al ricevitore. Se dopo 5 uni c'è ancora un 1 (e poi uno zero), allora è un flag.

Nel caso in cui il pacchetto invece abbia una *dimensione fissa*<sup>36</sup>, ci si trova



ad operare in una situazione simile a quella in presenza di trama, tranne che... la trama non c'è, e dunque l'ordine dei pacchetti è qualsiasi, ma viene meno l'esigenza dei flag di delimitazione.

In entrambi i casi (lunghezza di pacchetto fissa o variabile) i nodi della rete non necessitano di operare in sincronismo tra loro; lo schema di multiploazione è quindi detto *a divisione di tempo senza organizzazione di trama, asincrono, con etichetta*. Il

<sup>34</sup>Come nel PCM telefonico, vedi § 24.3.1

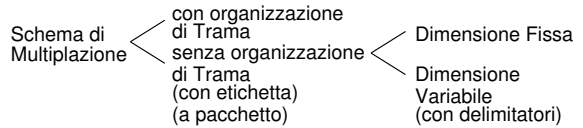
<sup>35</sup>A riguardo di questi due protocolli ormai *fuori moda*, si veda ad es.

[https://it.wikipedia.org/wiki/High-Level\\_Data\\_Link\\_Control](https://it.wikipedia.org/wiki/High-Level_Data_Link_Control) e

<https://it.wikipedia.org/wiki/X.25>

<sup>36</sup>Un modo di trasferimento con pacchetti di dimensione fissa è l'ATM (*Asynchronous Transfer Mode*) che viene descritto al § 23.2.

termine etichetta (o *label*) indica che ogni pacchetto deve recare con sé le informazioni idonee a ricombinarlo assieme agli altri dello stesso messaggio.

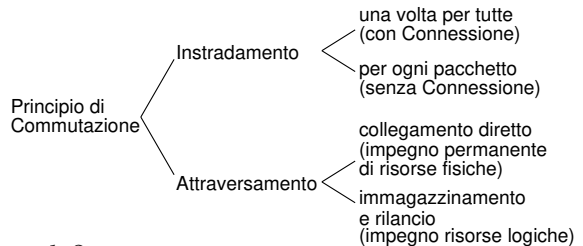


### 22.5.2.2 Principio di commutazione

È definito in base a come sono realizzate le due funzioni di *instradamento*, ovvero come individuare un percorso nella rete, e di *attraversamento*, ossia come permettere l'inoltro del messaggio tra le porte di ingresso e di uscita del commutatore.

Se l'*instradamento* (ROUTING) viene determinato una volta per tutte all'inizio del collegamento, il modo di trasferimento viene detto *con connessione*. Se al contrario l'instradamento avviene in modo indipendente per ogni pacchetto, il collegamento è detto *senza connessione* ed ogni pacchetto di uno stesso messaggio può seguire percorsi differenti.

L'*attraversamento* di un nodo di rete consiste invece nel *demultiplexare* le informazioni in ingresso e moltiplicarle di nuovo su uscite diverse: ciò può avvenire mediante un *collegamento diretto* o per *immagazzinamento e rilancio*.



Sulla base di queste considerazioni, definiamo:

**Commutazione di circuito:** l'instradamento avviene una volta per tutte prima della comunicazione, e l'attraversamento impegna in *modo permanente* ed *esclusivo* le *risorse fisiche* dei nodi della rete; è il caso della telefonia, sia POTS che PCM<sup>37</sup>.

**Commutazione di pacchetto a circuito virtuale:** L'instradamento è determinato *una volta per tutte* prima dell'inizio della trasmissione, durante una fase di *setup* delle risorse necessarie, conseguente alla *richiesta di connessione* effettuata da parte del nodo sorgente. Dopodiché i pacchetti di uno stesso messaggio seguono tutti lo stesso percorso, e l'attraversamento si basa sull'impegno di *risorse logiche*<sup>38</sup> ed avviene per *immagazzinamento e rilancio*. La trasmissione ha luogo dopo aver contrassegnato ogni pacchetto con un *identificativo di connessione* (IC) che individua un *canale virtuale*<sup>39</sup> tra

<sup>37</sup>Nel caso del POTS (vedi § 24.9.1) si creava un vero e proprio circuito elettrico (vedi anche pag. 826), e le risorse fisiche impegnate sono gli organi di centrale ed i collegamenti tra centrali, assegnati per tutta la durata della comunicazione in esclusiva alle due parti in colloquio. Nel caso del PCM (vedi § 24.3.1), le risorse allocate cambiano natura (ad esempio consistono anche nell'intervallo temporale assegnato al canale all'interno della trama) ma ciononostante vi si continua a far riferimento come ad una rete a *commutazione di circuito*.

<sup>38</sup>Le risorse impegnate sono dette *logiche* in quanto corrispondono ad entità concettuali (i *canali virtuali* descritti nel seguito).

<sup>39</sup>Il termine *canale virtuale* simboleggia il fatto che, nonostante i pacchetti di più comunicazioni viaggino "rimescolati" su di uno stesso mezzo, questi possono essere distinti in base alla comunicazione a

coppie di nodi di rete, che ne identifica l'appartenenza ad uno dei collegamenti in transito.

L'intestazione del pacchetto può essere ridotta, al limite, a contenere il solo IC del canale virtuale. L'attraversamento avviene consultando apposite tabelle (di *routing*), generate nella fase di setup che precede quella di trasmissione, in cui è indicata la porta di uscita per tutti i pacchetti appartenenti ad uno stesso messaggio. Facciamo un esempio, riferendoci allo schema di fig. 22.5: una sorgente, a seguito della fase di instradamento, invia i pacchetti con identificativo IC = 1 al primo nodo individuato dal routing. Consultando la propria tabella, il nodo trova che il canale virtuale 1 sulla porta di ingresso (PI) A si connette al c.v. 3 sulla porta di uscita (PU) C. Ora i pacchetti escono da C con IC = 3 ed una volta giunti al nodo seguente sulla PI A, escono dalla PU B con IC = 2 e giungono finalmente a destinazione. Notiamo che su di un collegamento tra due nodi, i numeri dei canali virtuali identificano in modo univoco il collegamento a cui appartengono i pacchetti, mentre uno stesso numero di canale virtuale può essere riutilizzato su porte differenti<sup>40</sup>.

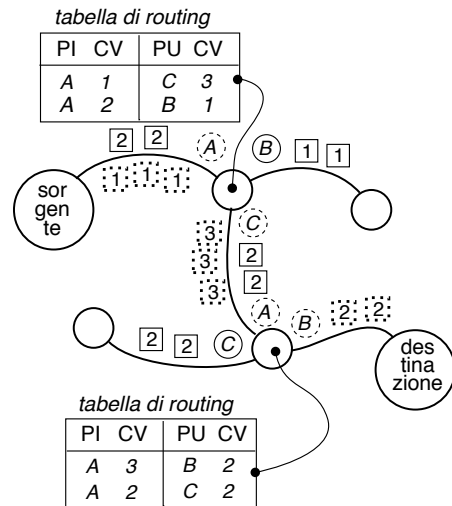


Figura 22.5: Commutazione di pacchetto a circuito virtuale

La concatenazione dei canali virtuali attraversati viene infine indicata con il termine *Circuito Virtuale* per similitudine con il caso di commutazione di circuito, con la differenza che ora il percorso individuato è definito solo in termini di tabelle e di etichette, e non di risorse fisiche (tranne che per la memoria della tabella).

Al termine della comunicazione, sul circuito virtuale viene inviato un apposito pacchetto di controllo, che provoca la rimozione del routing dalle tabelle.

**Congestione e controllo di flusso** Durante la fase di instradamento, il percorso nella rete è determinato in base alle condizioni di traffico del momento, ed eventualmente la connessione può essere rifiutata nel caso in cui la memoria di coda nei nodi coinvolti sia quasi esaurita, evento indicato con il termine di *congestione*.

D'altra parte, se alcune sorgenti origine dei canali virtuali già assegnati e che si incrociano in uno stesso nodo intermedio iniziano ad emettere pacchetti a frequenza più elevata del previsto, il nodo intermedio si congestiona (ossia esaurisce la memoria di transito) ed inizia a *perdere pacchetti*, penalizzando anche i canali virtuali delle altre connessioni che attraversano il nodo.

cui appartengono, grazie ai differenti IC (numeri) con cui sono etichettati; pertanto, è come se i pacchetti di una stessa comunicazione seguissero un proprio *canale virtuale* indipendente dagli altri.

<sup>40</sup>I numeri di c.v. sono negoziati tra ciascuna coppia di nodi durante la fase di instradamento, e scelti tra quelli non utilizzati da altre comunicazioni già in corso. Alcuni numeri di c.v. inoltre possono essere *riservati*, ed utilizzati per propagare messaggi di segnalazione inerenti il controllo di rete.

Per questo motivo, sono indispensabili strategie di *controllo di flusso* che permettano ai nodi di regolare l'emissione delle sorgenti. Il controllo di flusso è attuato anch'esso mediante pacchetti (di controllo), privi del campo di dati, ma contenenti un codice identificativo del comando che rappresentano. Ad esempio, un nodo non invia nuovi pacchetti di un circuito virtuale finché non riceve un *pacchetto di riscontro* relativo ai pacchetti precedenti. D'altra parte, nel caso di una rete congestionata, la perdita di pacchetti causa il mancato invio dei riscontri relativi, e dunque i nodi a monte cessano l'invio di nuovi pacchetti<sup>41</sup>. Dopo un certo periodo di tempo (TIMEOUT) il collegamento è giudicato interrotto e viene generato un pacchetto di *Reset* da inviare sul canale virtuale, e che causa, nei nodi attraversati, il rilascio delle risorse logiche (tabelle) relative al canale virtuale.

Discutiamo ora invece di un ulteriore possibile principio di commutazione:

**Commutazione di pacchetto a datagramma** Anche in questo caso *l'attraversamento* dei nodi avviene per *immagazzinamento e rilancio*, mentre la funzione di *instradamento* è svolta in modo distribuito tra i nodi di rete *per ogni pacchetto*, il quale (chiamato ora *datagramma*) deve necessariamente contenere l'indirizzo completo della destinazione. Infatti, in questo caso manca del tutto la fase iniziale del collegamento, in cui prenotare l'impegno delle risorse (fisiche o logiche) che saranno utilizzate<sup>42</sup>. Semplicemente, non è previsto alcun impegno a priori, ed ogni pacchetto costituisce un collegamento individuale che impegna i nodi di rete solo per la durata del proprio passaggio. L'instradamento avviene mediante tabelle presenti nei nodi, di tipo sia statico che dinamico (nel qual caso tengono conto delle condizioni di carico e di coda dei nodi limitrofi) che indicano le possibili porte di uscita per raggiungere la destinazione scritta sul pacchetto. Quest'ultimo quindi viene fatto uscire *senza nessuna alterazione* dalla porta di uscita.

Uno dei maggiori vantaggi dei datagrammi rispetto ai circuiti virtuali è una migliore resistenza ai guasti e malfunzionamenti: in questo caso infatti, a parte una eventuale necessità di ritrasmettere i pacchetti persi, il collegamento prosegue attraverso percorsi alternativi; inoltre l'elevato numero di percorsi alternativi, può permettere (in condizioni di carico leggero) di soddisfare brevi richieste di trasmissione a velocità elevate. Allo stesso tempo, in presenza di messaggi molto brevi, l'invio di un singolo datagramma è più che sufficiente, mentre nel caso a circuito virtuale le fasi di instaurazione ed abbattimento sarebbero state un lavoro in più da svolgere (tanto che ad es. l'X.25, che è nato a c.v., prevede anche il funzionamento a datagramma).

**Consegna ordinata e congestione** Uno dei maggiori problemi legati all'uso di datagrammi è che l'ordine di arrivo dei pacchetti può essere diverso da quello di

---

<sup>41</sup>In realtà vengono prima fatti dei tentativi di inviare nuovamente i pacchetti "vecchi". Questi ultimi infatti sono conservati da chi li invia (che può anche essere un nodo intermedio), finché non sono riscontrati dal ricevente. Quest'ultimo fatto può causare ulteriore congestione, in quanto restano impegnate risorse di memoria "a monte" della congestione che così *si propaga*.

<sup>42</sup>Per questo motivo, il collegamento è detto *senza connessione*.

partenza, potendo questi seguire percorsi differenti. Per questo motivo, nei datagrammi è presente un *numero di sequenza* che si incrementa ad ogni pacchetto trasmesso, ed alla destinazione sono predisposti dei *buffer*<sup>43</sup> di memoria nei quali ricostruire l'ordine esatto dei pacchetti.

Nel caso di un pacchetto mancante, il ricevente non sa se questo è semplicemente ritardato oppure è andato perso, rendendo problematico il controllo di flusso. In questo caso si produce un impegno anomalo dei buffer di ingresso, che non possono essere rilasciati perché incompleti, e ciò può causare il rifiuto dell'accettazione di nuovi pacchetti, provocando un impegno anomalo anche per i buffer di uscita di altri nodi, causando congestione<sup>44</sup>.

Prima di effettuare un trasferimento a datagramma, è opportuno (a parte il caso di messaggi composti da un singolo datagramma) verificare la disponibilità del destinatario finale, e preavvisarlo di riservare una adeguata quantità di memoria. Ad esempio, in Internet avviene proprio questo (vedi pag. 801).

Proseguiamo la descrizione delle reti per dati con l'ultima caratteristica di un modo di trasferimento:

### 22.5.2.3 Architettura protocollare

Definisce la stratificazione delle funzioni di comunicazione, sia per gli apparati terminali che per i nodi di transito, e di come queste interagiscono reciprocamente sia tra nodi diversi, che nell'ambito di uno stesso nodo. Alcune di queste sono già state introdotte, e le citiamo per prime, seguite da quelle più rilevanti illustrate di seguito:

- il *controllo di flusso*, che impedisce la saturazione dei buffer;
- la *consegna ordinata*, per riassemblare messaggi frammentati su più datagrammi;
- la *segmentazione e riassettaggio*, che definisce le regole per frammentare un messaggio in pacchetti e ricomporli, ad esempio in corrispondenza dei "confini" tra sottoreti con differente lunghezza di pacchetto;
- il *controllo di connessione*, che provvede ad instaurare la connessione, eseguire l'instradamento, impegnare le risorse, supervisionare il controllo di flusso, abbattere la connessione al suo termine;
- il *controllo di errore*, che provvede a riscontrare le unità informative, a rilevare gli errori di trasmissione, a gestire le richieste di trasmissione;
- l'*incapsulamento*, che aggiunge ai pacchetti di dati da trasmettere le informazioni di protocollo come l'header, gli indirizzi, il controllo di parità...

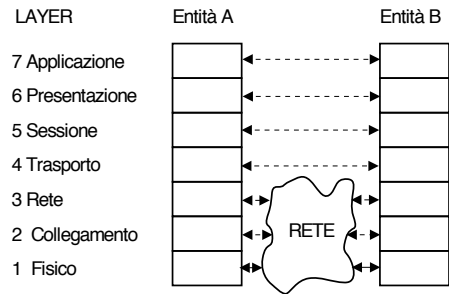
**Stratificazione ISO-OSI** Per aiutare nella schematizzazione delle interazioni tra le funzioni illustrate, l'*International Standard Organization* (ISO) ha formalizzato un

<sup>43</sup>Il termine *buffer* ha traduzione letterale "respingente, paracolpi, cuscinetto" ed è a volte espresso in italiano dalla locuzione *memoria tampone*.

<sup>44</sup>La soluzione a questa "spirale negativa" si basa ancora sull'uso di un allarme a tempo (timeout), scaduto il quale si giudica interrotto il collegamento, e sono liberati i buffer.

modello concettuale per sistemi di comunicazione denominato *Open System Interconnection* (OSI)<sup>45</sup>, che individua una relazione gerarchica tra i protocolli. In particolare sono definiti sette *strati* o *livelli* (LAYERS) ognuno dei quali raggruppa un insieme di funzioni affini. Gli strati più elevati (4-7) sono indicati anche come *strati di utente*, in quanto legati a funzioni relative ai soli apparati terminali; gli *strati di transito* invece (1-3) riguardano funzioni che devono essere presenti anche nei nodi intermedi.

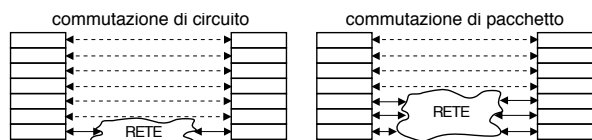
La relazione gerarchica individuata stabilisce tra due strati contigui un rapporto di tipo *utente-servizio*; ovvero lo svolgimento delle funzioni di strato superiore necessita dei servizi offerti dallo strato inferiore. A titolo di esempio, si pensi all'invio di un documento mediante un corriere espresso: ci si affida allora ad uno strato di trasporto che offre all'utente (strato di sessione) un servizio (appuntamento) di trasporto che ha il compito di "far apparire" il documento presso il destinatario. La sede locale del corriere si affida quindi alla propria divisione interna che gestisce la rete dei corrispondenti, la quale si affida a sua volta ai corrispondenti stessi, che hanno il compito di assistere alla consegna ed all'arrivo (collegamento) del documento. Il trasferimento fisico dello stesso può quindi avvenire mediante un ultimo strato funzionale (treno, nave, aereo, auto...) che provvede al recapito in base alle informazioni ricevute dallo strato di collegamento.



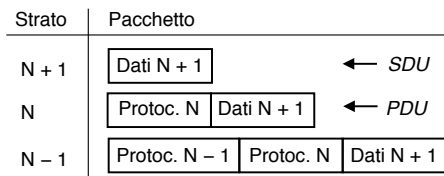
Per terminare l'esempio, facciamo notare come in ogni livello avvengano *due* tipi di colloqui (regolati da altrettanti protocolli): uno è *orizzontale*, detto anche *tra pari* (PEER-TO-PEER), come è ad esempio il contenuto del documento che spediamo, od i rapporti tra corrispondenti locali (che nel caso di un sistema di comunicazione corrisponde allo strato di collegamento, relativo ai protocolli tra singole coppie di nodi di rete); il secondo tipo di colloquio avviene invece in forma *verticale*, o *tra utente e servizio*, in quanto per realizzare le funzioni di uno strato *utente* ci si affida ad un *servizio* di comunicazione offerto dallo strato inferiore (che a sua volta può avvalersi dei servizi degli altri strati ancora inferiori)<sup>46</sup>.

<sup>45</sup>In virtù dell'intreccio di sigle, il modello di riferimento prende il nome (palindromo) di modello ISO-OSI.

<sup>46</sup>Il modo di trasferimento è completamente definito dopo che sia stato specificato in quale strato siano svolte le funzioni di commutazione e moltiplicazione. In una rete a commutazione di circuito, queste sono realizzate dallo strato fisico che, esaurita la fase di instradamento ed impegno di risorse fisiche, collega in modo trasparente sorgente e destinazione. Nella commutazione di pacchetto, invece, le funzioni di moltiplicazione e commutazione coinvolgono (per tutti i pacchetti del messaggio) tutti i nodi di rete interessati; si dice pertanto che i protocolli di collegamento e di rete devono *essere terminati* (nel senso di gestiti) da tutti i nodi di rete.



**Incapsulamento** La modalità con cui un protocollo tra pari di strato  $N$  affida i suoi dati ad un servizio di strato  $N - 1$ , si avvale (nella commutazione di pacchetto) della funzione di *incapsulamento*, di cui viene data una interpretazione grafica alla figura seguente. I dati che lo strato  $N + 1$  vuol trasmettere al suo pari, indicati anche come *Service Data Unit* (SDU), sono prefissi dalle informazioni di protocollo necessarie alla gestione del collegamento tra entità allo strato  $N$ . Questa nuova unità informativa prende il nome di *Protocol Data Unit* (PDU) per lo strato  $N$ , e viene passata in forma di SDU al servizio di collegamento offerto dallo strato  $N - 1$ , che ripete l'operazione



di incapsulamento con le proprie informazioni di protocollo, generando una nuova PDU (di strato  $N - 1$ ). Pertanto, lo strato fisico provvederà a trasmettere pacchetti contenenti tutte le informazioni di protocollo degli strati superiori.

**Indipendenza dei servizi tra pari dal servizio di collegamento** Quando uno strato affida il collegamento con un suo pari allo strato inferiore, quest'ultimo può mascherare al superiore la modalità con cui viene realizzato il trasferimento.

In particolare, se ci riferiamo all'interfaccia tra gli strati di trasporto e di rete, lo strato di rete può realizzare con il suo pari collegamenti con o senza connessione, mentre quello di trasporto offre allo stesso tempo (ma in modo indipendente) agli strati superiori un servizio con o senza connessione, dando luogo alle seguenti 4 possibilità:

Servizio di rete	SERVIZIO DI TRASPORTO	
	CIRCUITO VIRTUALE	DATAGRAMMA
Circuito Virtuale	SNA, X.25	Insolito
Datagramma	Arpanet, TCP/IP	Decnet

**SNA** (SYSTEM NETWORK ARCHITECTURE) è una architettura proprietaria IBM, in cui il trasferimento avviene in modo ordinato, richiedendo al livello di trasporto un circuito virtuale, che è realizzato da una serie di canali virtuali tra i nodi di rete. La stessa architettura è adottata anche dall'X.25, che costituisce l'insieme di protocolli che descrivono il funzionamento di reti pubbliche a commutazione di pacchetto, presenti in tutto il mondo: quella italiana prende il nome di ITAPAC.

**Arpanet** è l'architettura di Internet, in cui sebbene lo strato di rete operi con un principio di commutazione a datagramma, mediante il protocollo *IP* (INTERNET PROTOCOL), lo strato di trasporto (*TCP*, TRANSFER CONTROL PROGRAM) offre a quelli superiori un servizio con connessione, attuato mediante circuiti virtuali, in modo da garantire il corretto sequenziamento delle unità informative, ed offrire canali di comunicazione formalmente simili ai files presenti localmente su disco. Il mascheramento del servizio di rete interna a datagramma in un servizio con connessione avviene a carico dello strato *TCP* di trasporto presente nei nodi terminali, che appunto affronta il riassetto ordinato dei datagrammi ricevuti dallo strato di rete.



**Utilizzo del collegamento** Considerando l'intervallo di tempo  $t_T$  che intercorre tra la trasmissione di due trame consecutive, la trasmissione vera e propria dura solamente  $t_{Tx}$  istanti, dopodiché occorre attendere  $2 \cdot t_p$  istanti ( $t_p$  è il tempo di *propagazione*) prima di ricevere l'ACK. Trascurando gli altri tempi (di trasmissione dell'ACK, e di elaborazione delle trame), si definisce una efficienza di utilizzo

$$U = \frac{t_{Tx}}{t_T} \simeq \frac{t_{Tx}}{t_{Tx} + 2 \cdot t_p} = \frac{1}{1 + 2 \cdot t_p/t_{Tx}} = \frac{1}{1 + 2 \cdot a}$$

in cui il parametro  $a$  che determina il risultato, può assumere valori molto diversi, in funzione della velocità di trasmissione e della lunghezza del collegamento.

**Esempio** Una serie di trame di  $N = 1000$  bit viene trasmessa utilizzando un protocollo *send-and-wait*, su tre diversi collegamenti:

- un cavo ritorto di 1 km,
- una linea dedicata di 200 km,
- un collegamento satellitare di 50000 km.

Sapendo che la velocità di propagazione è di  $2 \cdot 10^8$  m/sec per i casi (a) e (b), e di  $3 \cdot 10^8$  m/sec per il caso (c), determinare l'efficienza di utilizzo  $U = \frac{1}{1+2 \cdot a}$ , per le due possibili velocità di trasmissione  $f_b$  di 1 kbps ed 1 Mbps.

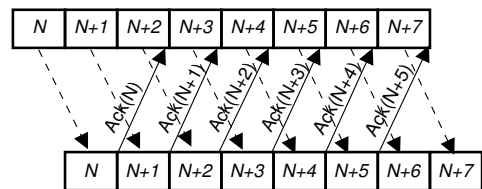
Il tempo necessario alla trasmissione  $t_{Tx} = \frac{N}{f_b}$  risulta pari ad 1 sec ed 1 msec alle velocità di  $10^3$  e  $10^6$  bps rispettivamente. Il tempo di propagazione  $t_p = \frac{\text{spazio}}{\text{velocità}}$  risulta pari a  $5 \cdot 10^{-6}$  sec,  $1 \cdot 10^{-3}$  sec e 0.167 sec nei tre casi (a), (b), e (c) rispettivamente. Pertanto:

- si ottiene  $a = \frac{t_p}{t_{Tx}} = 5 \cdot 10^{-6}$  e  $a = 5 \cdot 10^{-3}$  per le velocità di 1 kbps ed 1 Mbps rispettivamente, e quindi per entrambe  $U \simeq 1$ ;
- per  $f_b = 1$  kbps si ottiene  $a = 10^{-3}$  e quindi  $U \simeq 1$ , per  $f_b = 1$  Mbps risulta  $a = 1$  e quindi  $U = 0.33$ ;
- per le velocità di 1 kbps ed 1 Mbps si ottiene  $a = 0.167$  ed  $a = 167$  rispettivamente, a cui corrispondono efficienze pari a  $U = 0.75$  e  $U = 0.003$ .

Sulla base del risultato dell'esempio notiamo che, considerando fissa la dimensione di trama, le prestazioni di un collegamento nei confronti di un protocollo ARQ possono essere caratterizzate, oltre che dal parametro  $a$ , anche dal cosiddetto *Prodotto Banda-Ritardo*  $PBR = f_b \cdot t_p$ , che infatti nei sei casi in esame vale  $5 \cdot 10^{-3}$ , 5, 1,  $10^3$ , 160,  $1.6 \cdot 10^5$ . Pertanto, abbiamo dimostrato come la trasmissione *send and wait* possa essere idonea per basse velocità e/o collegamenti brevi, in virtù della sua semplicità realizzativa; in caso contrario, è opportuno ricorrere ad uno dei metodi seguenti.

### 22.6.2 Continuous RQ

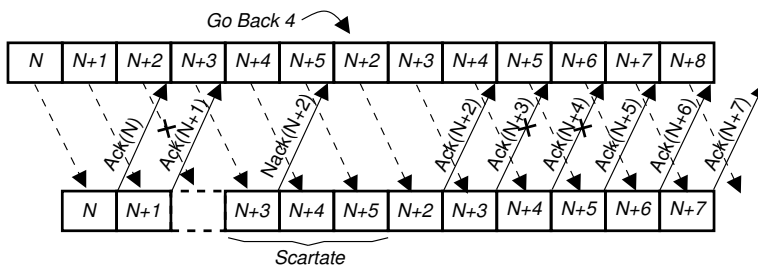
A differenza del protocollo *send-and-wait*, ora il trasmettitore invia le trame ininterrottamente, senza attendere la ricezione degli ACK. In presenza di trame ricevute correttamente, il ricevitore riscontra



positivamente le stesse, consentendo al trasmettitore di liberare i buffer di trasmissione. In presenza di trame ricevute con errori, la quantità di memoria tampone utilizzata al ricevitore determina la scelta di due possibili strategie di richiesta di ritrasmissione, denominate *go-back-N* e *selective-repeat*.

### 22.6.2.1 Go back N

In questo caso il ricevitore dispone di una sola posizione di memoria, dove trattiene la trama appena ricevuta per il tempo necessario al controllo di errore. In presenza di un errore di ricezione della trama  $N + i$ , rilevato<sup>48</sup> dopo la corretta ricezione di  $N + i + 1$ , il ricevitore invia un  $\text{NACK}(N + i)$ , chiedendo con ciò al trasmettitore di *andare indietro*, ed inizia a scartare tutte le trame con numeri maggiori di  $N + i$ , finché non riceve la  $N + i$ , e riprende le normali operazioni.



Se, trascorso un timeout, la  $N + i$  non è arrivata, si invia di nuovo un  $\text{NACK}(N + i)$ . Nel caso in cui invece si corrompa un ACK, le operazioni continuano regolarmente, e l'ACK successivo agisce da riscontro positivo anche per le trame per le quali non si sono ricevuti riscontri. Il trasmettitore deve quindi mantenere memorizzate tutte le trame trasmesse e non ancora riscontrate, fino ad un numero massimo, raggiunto il quale la trasmissione si arresta.

Una variante del metodo, idonea al caso in cui fenomeni di congestione di rete possano determinare la perdita dei NACK, prevede l'uso di un timer al trasmettitore, per re-inviare le trame non riscontrate.

### 22.6.2.2 Selective repeat

L'origine di questo nome deriva dal fatto che non è più richiesto al trasmettitore di *tornare indietro* completamente, ma è sufficiente ritrasmettere solamente la trama che ha dato origine ad errore in ricezione, grazie alla capacità del ricevitore di memorizzare temporaneamente più trame, anche se ricevute fuori sequenza.

Come si nota alla figura 22.7, a seguito della ritrasmissione della trama  $N + 2$  per cui si è ricevuto il NACK, il trasmettitore continua ad inviare nuove trame, fino al numero massimo previsto; in assenza di ulteriori ACK, un timer determina la ritrasmissione delle trame non riscontrate. Quando al ricevitore perviene la trama  $N + 2$ , questo emette un  $\text{ACK}(N + 5)$ , consentendo al trasmettitore di rilasciare tutta la memoria occupata dalle trame in sospenso, e di proseguire la trasmissione. La perdita di uno o più ACK è gestita

<sup>48</sup>Sottolineiamo nuovamente l'importanza dei numeri di sequenza, che permettono al ricevitore di capire il numero della trama corrotta, grazie alla discontinuità dei numeri stessi.

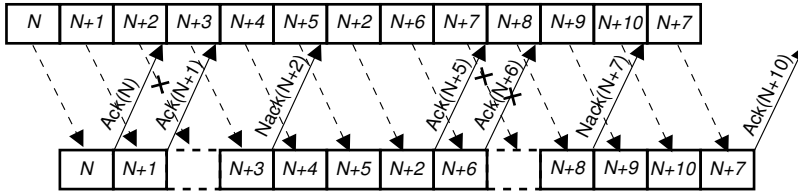


Figura 22.7: Protocollo di ritrasmissione *selective repeat*

allo stesso modo che per *go-back-N*, così come per ogni NACK inviato si inizializza un timer, allo scadere del quale ed in assenza di nuove trame ricevute, il NACK è re-inviato.

Dal punto di vista del ricevitore, questo è più complicato che nel caso *go-back-N*, dato che adesso occorre riordinare le trame ricevute, che possono arrivare sequenziate con un ordine diverso da quello naturale. Per questo motivo, anche il ricevitore deve predisporre delle memorie temporanee dove salvare le trame correttamente arrivate, successivamente a quella che invece conteneva errori, e di cui si attende la ritrasmissione.

### 22.6.2.3 Efficienza dei protocolli a richiesta automatica

Mentre con le tecniche FEC siamo obbligati ad aumentare la  $f_b$  dei dati trasmessi per poter inserire i bit di ridondanza, potrebbe sembrare che nel caso ARQ ciò non si verifichi. Ma anche se la ridondanza introdotta è effettivamente inferiore, la necessità di dover ritrasmettere l'intero pacchetto ricevuto errato è anch'essa fonte di riduzione della velocità trasmissiva. Valutiamo di quanto.

Indichiamo con  $p$  la probabilità di dover chiedere la ritrasmissione di una trama<sup>49</sup> per la quale si sono rilevati errori di trasmissione, e con  $m$  il numero totale di trasmissioni necessarie alla sua corretta ricezione. Osserviamo quindi che  $m$  è una variabile aleatoria discreta, caratterizzata dalle probabilità

$$\begin{aligned}
 p_M(1) &= Pr(m = 1) = 1 - p \\
 p_M(2) &= p(1 - p) \\
 p_M(3) &= p^2(1 - p) \\
 &\vdots \\
 p_M(m) &= p^{m-1}(1 - p)
 \end{aligned}$$

che descrivono come sia possibile ricevere la stessa trama come errata  $m - 1$  volte, finché all' $m$ -esima trasmissione non si rilevano più errori. Pertanto, il numero medio

<sup>49</sup>Nel caso in cui l'integrità della trama sia protetta da un codice a blocco  $(n, k)$  con  $d_m = l + 1$  (§ 15.6.2.1), la probabilità che la trama contenga più di  $l$  errori e che quindi venga accettata dal ricevitore anche se errata, vale approssimativamente  $P(l + 1, n) = \binom{n}{l+1} P_e^l$  (vedi formula (15.27)). Dato che il ricevitore accetta le trame che *non hanno* errori, oppure che hanno più di  $l$  errori, la probabilità che venga richiesta una ritrasmissione risulta

$$p = 1 - P(0, n) - P(l + 1, n)$$

Considerando ora che  $P(l + 1, n) \ll P(0, n)$  (vedi eq. 15.27), si ottiene

$$p \simeq 1 - P(0, n) = 1 - (1 - P_e)^n \simeq nP_e$$

in cui  $P_e$  è la probabilità di errore sul bit (dato che  $(1 - P_e)^n \simeq 1 - nP_e$  se  $nP_e \ll 1$ ).

di trasmissioni per una stessa trama è pari a

$$\begin{aligned}\bar{m} &= E\{m\} = \sum_{m=1}^{\infty} mp_M(m) = \sum_{m=1}^{\infty} mp^{m-1}(1-p) = \\ &= (1-p) \sum_{n=0}^{\infty} (n+1) \cdot p^n = (1-p) \frac{1}{(1-p)^2} = \frac{1}{1-p}\end{aligned}$$

in cui alla quarta eguaglianza si è posto  $n = m - 1$ , ed alla quinta si è utilizzato il risultato noto della serie geometrica<sup>50</sup> per ottenere

$$\begin{aligned}\sum_{n=0}^{\infty} (n+1) p^n &= \sum_{n=1}^{\infty} np^{n-1} = \sum_{n=0}^{\infty} np^{n-1} = \sum_{n=0}^{\infty} \frac{\partial}{\partial p} p^n = \\ &= \frac{\partial}{\partial p} \sum_{n=0}^{\infty} p^n = \frac{\partial}{\partial p} \frac{1}{1-p} = \frac{1}{(1-p)^2}\end{aligned}$$

Quindi, per trasmettere una frequenza binaria di  $f_b$  bps (comprensivi di CRC e *overhead* dei numeri di sequenza, vedi § 22.6.3.3), occorre in realtà inviare dati ad una velocità *media* pari a  $f_b/(1-p)$  bps<sup>51</sup>. Questo risultato approssimato si applica ad un protocollo di tipo *selective repeat*, e trascurando gli errori sul canale a ritroso.

### 22.6.3 Controllo di flusso

Si è illustrato come nei protocolli ARQ il trasmettitore, dopo un po' che non riceve nuovi ACK, cessa a sua volta di inviare trame, dato che esaurisce la memoria temporanea in cui memorizzare le trame già inviate ed in attesa di riscontro. Nel caso in cui il ricevitore non sia in grado di smaltire per tempo i dati ricevuti, può scegliere di sospendere temporaneamente l'invio di riscontri, con il risultato di rallentare la velocità di invio dei dati. Questo meccanismo prende il nome di *controllo di flusso*, per l'evidente analogia idraulica, in cui una condotta viene ristretta al fine di ridurre il flusso di liquido in transito.

Dato che il ritardo tra la sospensione dell'invio degli ACK e l'interruzione dell'invio di trame dipende dalla dimensione delle memorie temporanee, e che questa dimensione incide allo stesso tempo anche sulla efficienza di utilizzo temporale del collegamento in condizioni di ricezione a piena velocità, svolgiamo alcune riflessioni sull'argomento.

#### 22.6.3.1 Round trip time

Potrebbe essere tradotto come *tempo di girotondo*, ed è il tempo che intercorre tra l'inizio della trasmissione di una trama e l'arrivo del relativo ACK. La sua valutazione spesso si avvale della ipotesi di poter trascurare il tempo di trasmissione dell'ACK, e

<sup>50</sup>  $\sum_{k=0}^{\infty} \alpha^k = \frac{1}{1-\alpha}$  se  $|\alpha| < 1$

<sup>51</sup> Dato che  $p$  aumenta con  $n$  (vedi pag. 467), l'efficienza del protocollo ARQ *peggiora* con l'aumentare della dimensione delle trame. Questo risultato determina l'esigenza di ricercare una soluzione di compromesso, dato che l'incidenza dell'*overhead* sulla dimensione complessiva della trama invece *si riduce* all'aumentare di  $n$ .

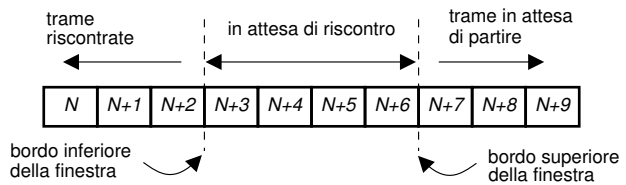
quindi si ottiene

$$RTT \simeq t_{Tx} + 2t_p$$

dove  $t_{Tx}$  è il tempo necessario a trasmettere il pacchetto, e  $t_p$  è il ritardo di propagazione.. Se la trasmissione avviene a velocità  $f_b$  allora in un tempo pari a  $RTT$  possono essere trasmessi  $N_{ba} = f_b \cdot RTT$  bit, che possono essere pensati come il numero di *bit in aria*<sup>52</sup>. Se la memoria temporanea del trasmettitore ha dimensioni  $W \geq N_{ba}$ , allora la trasmissione (senza errori) può avvenire senza soluzione di continuità, impegnando costantemente il collegamento.

### 22.6.3.2 Finestra scorrevole

La quantità massima di dati  $W$  che è possibile trasmettere senza ricevere un riscontro è indicata come *finestra di trasmissione* per il motivo che ora illustriamo. In figura si mostra un gruppo di trame oggetto di una trasmissione; quelle già trasmesse ed in attesa di riscontro (da  $N + 3$  a  $N + 6$  in figura) sono racchiuse tra due confini, i *bordi* della finestra. Ogni volta che ne viene trasmessa una, il bordo *superiore* della finestra è spostato a destra, *allargandola*; ogni volta che si riceve un riscontro, è il bordo *inferiore* ad essere spostato a destra, *restringendo* così la finestra. In definitiva, il termine *finestra* trae origine dal fatto che, allo spostarsi dei bordi inferiore e superiore, la finestra *si apre e si chiude*.



La condizione di *massima apertura* della finestra identifica la quantità di memoria necessaria al trasmettitore per realizzare un protocollo ARQ,

	<i>fin. di trasm.</i>	<i>fin. di ric.</i>	$N_{Max}$
send and wait	1	1	2
go back N	W	1	W+1
selective repeat	W	W	2W+1

che quindi può essere ri-classificato in questi termini come mostrato in tabella, dove la colonna *finestra di ricezione* indica anche i requisiti di memoria al lato ricevente<sup>53</sup>. Notiamo quindi che mentre per *send-and-wait* è sufficiente la memoria di una sola

<sup>52</sup>L'espressione "*bit in aria*" trae spunto dalla metafora di una coppia di giocolieri, posti ai due estremi di una piazza, che si lanciano una serie di clave. Il primo ne lancia in continuazione, e quando iniziano ad arrivare al secondo, questi le rilancia verso il primo. Nel momento in cui la clava partita per prima torna nelle mani del primo giocoliere, un certo numero di clave sono sospese a mezz'aria, e corrispondono approssimativamente al numero di bit trasmessi in un tempo di pari durata, con una frequenza pari al ritmo di lancio delle clave, e non ancora riscontrati.

<sup>53</sup>La ricezione di una sequenza di trame corrette determina l'avanzamento alternato dei due bordi della finestra al ricevitore: questa inizialmente è vuota, poi contiene solo la trama ricevuta (avanza bordo superiore), e quindi è di nuovo svuotata non appena viene trasmesso l'ACK (ed avanza il bordo inferiore). In presenza di errori, il bordo inferiore non avanza, ma resta fermo sulla trama ricevuta con errori, e di cui si attende la ritrasmissione. Mentre il trasmettitore continua ad inviare trame, il ricevitore le memorizza e fa avanzare il bordo superiore, finché non siano state ricevute tutte quelle trasmissibili senza riscontro, e pari alla dimensione massima della finestra in trasmissione.

trama, per *go-back-N* il trasmettitore deve ricordare fino ad un massimo di  $W$  trame in attesa di riscontro, e per *selective repeat* anche il ricevitore ha lo stesso vincolo, allo scopo di riordinare le trame ricevute fuori sequenza.

### 22.6.3.3 Numero di sequenza

Dato che non possono essere inviate più trame della dimensione della finestra, la loro numerazione può avvenire in forma ciclica, ossia utilizzando un contatore modulo  $N_{Max}$ , come indicato alla tabella precedente. Ad esempio, per *send-and-wait* è sufficiente un contatore binario ( $0 - 1$ ) perché, nel caso in cui l'ACK sia corrotto, il ricevitore possa riconoscere la trama ricevuta come duplicata anziché nuova; un ragionamento simile<sup>54</sup> determina la necessità di usare  $W + 1$  numeri ( $0 - W$ ) nel caso *go-back-N*, e  $2W + 1$  numeri ( $0 - 2W$ ) nel caso *selective repeat*.

L'uso di un numero di bit ridotto per indicare il numero di sequenza permette di limitare la dimensione dell'*overhead* di trama; ad esempio, con una finestra di dimensione 7, l'uso di *go-back-N* richiede 8 diversi numeri di sequenza, che quindi possono essere codificati utilizzando 3 bit.

---

<sup>54</sup>Se il trasmettitore invia tutte le  $W$  trame, ma tutti gli ACK sono corrotti, allora la  $(W + 1)$ -esima trama trasmessa è un duplicato della prima, ritrasmessa per time-out, ed il ricevitore può accorgersene solo se la trama reca un numero differente da quello della prima.

Per il caso *selective repeat*, vale un ragionamento simile, ma che per le differenze nella definizione del protocollo, porta ad un risultato diverso.

L'opera

# ***Trasmissione dei Segnali e Sistemi di Telecomunicazione***

è il risultato di un progetto ventennale di cultura libera, aggiornato di continuo ed evolutosi fino alla forma attuale. La sua disponibilità pubblica è regolata dalle norme di licenza CREATIVE COMMONS

*Attribuzione - Non commerciale -  
Condividi allo stesso modo*



<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.it>

e tutte le risorse relative al testo sono accessibili presso

<https://teoriadeisignali.it/libro/>

Puoi contribuire al suo successo promuovendone la diffusione e supportarne lo sviluppo attraverso una donazione, in buona parte devoluta ai progetti *open source*<sup>1</sup> che ne hanno resa possibile realizzazione e divulgazione. Ai donatori viene accordato un accesso *vitalizio* al formato PDF *navigabile* di tutte le edizioni presenti *e future*.

---

1

- . Lyx - <http://www.lyx.org/>
- . L<sup>A</sup>T<sub>E</sub>X - <https://www.latex-project.org/>
- . TeX Users Group - <https://tug.org/>
- . Inkscape - <http://www.inkscape.org/>
- . Gnuplot - <http://www.gnuplot.info/>
- . Octave - <http://www.gnu.org/software/octave/>
- . Geany - <https://www.geany.org/>
- . Linux - <https://www.linux.it/>
- . Free Software Foundation - <https://shop.fsf.org/>
- . GNOME Foundation - <https://www.gnome.org/>
- . Mozilla Foundation - <https://www.mozilla.org/it/>
- . Wikipedia - <https://it.wikipedia.org>
- . Internet Archive - <https://archive.org/about/>
- . Creative Commons - <https://creativecommons.it/chapterIT/>
- . WordPress - <https://it.wordpress.org/>
- . Phplist - <https://www.phplist.org/>