

Piano dei dati 2

mercoledì 23 agosto 2023 15:56

Accodamenti

Gli accodamenti si creano nelle porta d'ingresso ed uscita, questi dipendono dalla quantità di traffico di rete, dalla velocità della struttura di commutazione e della linea (*terminazione fisica*). Le code costituiscono un problema nel momento in cui la memoria nelle code all'interno delle porte (*buffer*) esaurisce. Una situazione in cui il **tasso di inoltra tra porta d'ingresso-uscita** è N volte, numero delle porte d'ingresso(= numero porte d'uscita), più veloce del **tasso di trasmissione della linea** l'accodamento è trascurabile. Abbiamo due casi in cui l'accodamento diventa un problema:

(1) Accodamento in ingresso

Supponiamo che la velocità dei collegamenti siano uguali e che un pacchetto viene trasferito dalla porta d'ingresso a quella d'uscita nello stesso tempo che necessita a farlo arrivare alla linea d'ingresso, tale trasferimento avviene secondo una politica FCFS (*first come first served*).



Due pacchetti in testa alle rispettive code di ingresso sono destinati alla stessa porta di uscita... Se la struttura di commutazione sceglie di trasferire il primo pacchetto della coda in alto a sinistra, allora l'altro, nella coda in basso a sinistra, destinato alla stessa porta, dovrà attendere (abbiamo detto prima che in una struttura di commutazione a rete interconnessa può essere inoltrato un pacchetto alla volta). Non solo, i pacchetti che sono nella stessa coda, presenti dietro il pacchetto che attende, anche se vogliono andare in una porta libera, non possono perché hanno davanti un pacchetto bloccato...

Blocco in testa alla coda (HOL): un pacchetto nella coda d'ingresso deve attendere il trasferimento attraverso la struttura di commutazione, anche se la propria porta di destinazione è libera, in quanto risulta bloccato da un altro pacchetto che lo precede.

(1) Accodamento in uscita

Riprendiamo la situazione in cui il **tasso di inoltra tra porta d'ingresso-uscita** è N volte, numero delle porte d'ingresso(= numero porte d'uscita), più veloce del **tasso di trasmissione della linea**, ma che i pacchetti siano diretti alla medesima uscita, sappiamo che la porta d'uscita può trasmettere sul collegamento in uscita un solo pacchetto per volta: da qui le code si allungano e i buffer si saturano. A questo punto quando un pacchetto arriva e lo spazio è esaurito viene deciso se scartarlo, ovvero adottare una **politica Drop-Tail**, inoltre viene deciso se scartarne solo uno o più fra quelli già in coda e con quale criterio.

La politica Drop-Tail più utilizzata è la politica RED.

La formula per indicare una grandezza corretta dei buffer in un router è:

$$B = (RTT * C) / \sqrt{N}, \text{ dove } C = \text{capacità del collegamento}$$

Abbiamo parlato di code, in particolare prendiamo in considerazione le code d'uscita...*come scegliamo l'ordine di trasmissione, ovvero in che ordine preleviamo i pacchetti dalla coda?*

Scheduling dei pacchetti

- **FCFS**, la priorità dei pacchetti viene impostata in base al loro arrivo. Il primo arrivato è il primo ad essere servito, la coda di conseguenza è una coda FIFO.
- **Coda di Priorità**, viene adottata una strategia di classificazione, ovvero possiamo classificare i pacchetti dalla loro natura in code diverse, ad esempio classifichiamo i pacchetti dedicati alla gestione della rete e quelli per gli utenti. La strategia ha due modalità di gestione:
 - **Preemptive**: i pacchetti della coda con priorità bassa vengono trasmessi fino a che non vi sono pacchetti nella coda con priorità alta;
 - **Non-Preemptive**: l'invio dei pacchetti nelle code, fino a che non sono vuote, non viene fermato, bisogna svuotare prima la coda corrente per poi passare alla successiva;

- (*)Round Robin e (**)Accodamento Equo-Ponderato, (*)i pacchetti sono suddivisi in classi, senza una rigida priorità di servizio, ma con un'alternanza tra le classi. La forma più semplice di Round Robin implementa un'alternanza della trasmissione, ovvero se ad esempio abbiamo due classi, prima viene inviato un pacchetto della classe 1, successivamente uno della classe 2, poi nuovamente un pacchetto di classe 1 e così via. Nella modalità conservativa (work-conserving Round Robin), il collegamento non resta mai inattivo fino a quando ci sono pacchetti, di qualsiasi classe da trasmettere, infatti se la coda di una classe è vuota viene immediatamente controllata la successiva.
- (**)Nello schema di Accodamento Equo-Ponderato (WFQ) i pacchetti in arrivo sono classificati e accodati in base alla classe, offre un servizio a tipo ciclico ed è conservativo; differisce rispetto a Round Robin: **a ciascuna classe "i" viene assegnato un peso w_i** , in qualsiasi momento in cui nella coda i sono presenti pacchetti che devono essere spediti, WFQ garantisce che questi ricevano sempre e comunque una frazione di servizio pari a:

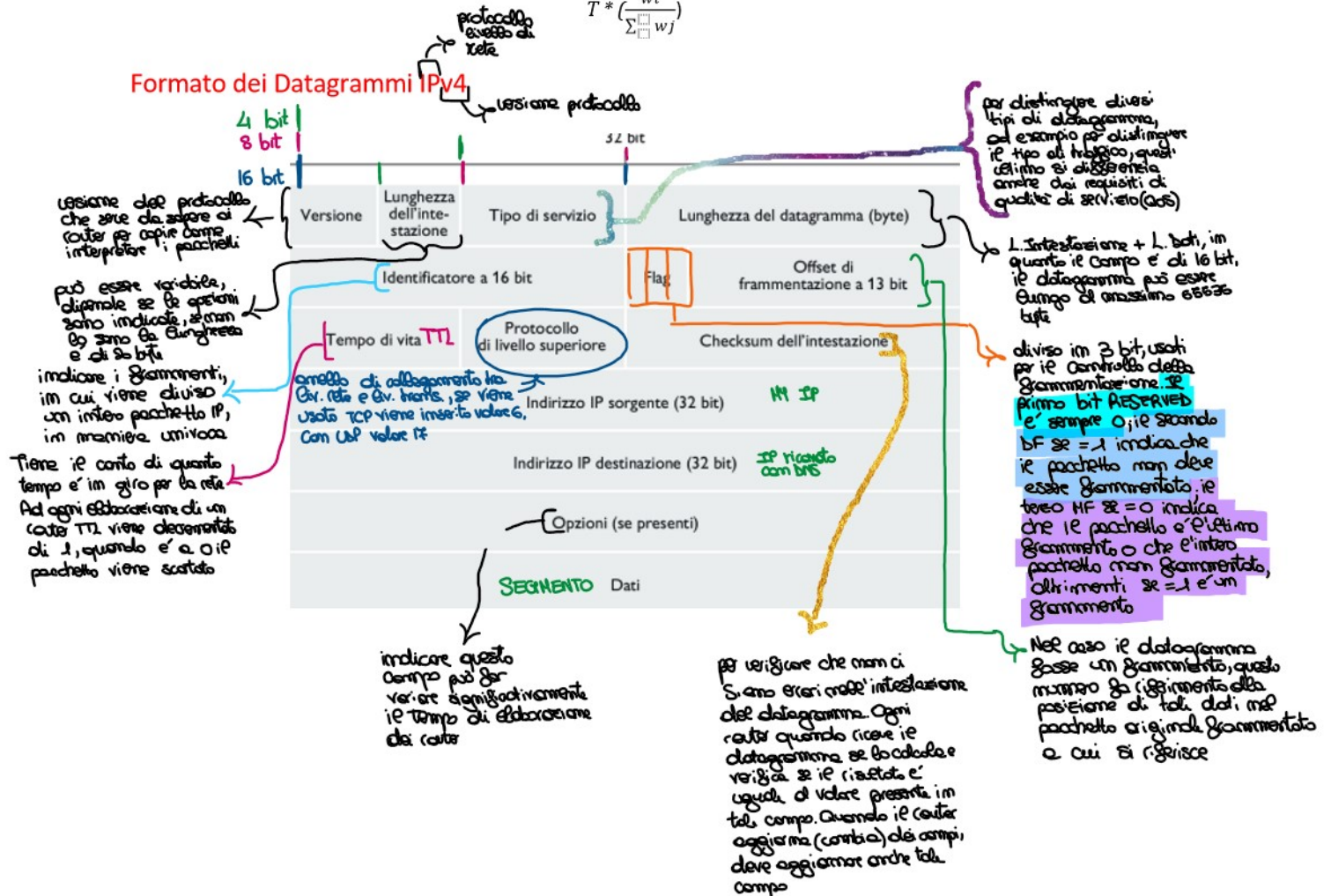
$$\frac{w_i}{\sum w_j}$$

dove al denominatore è indicata la somma dei pesi di tutte le classi che hanno pacchetti accodati

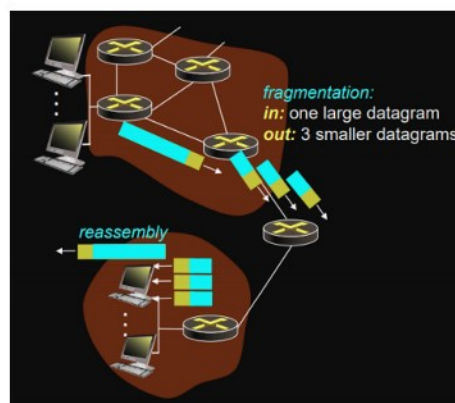
Concludiamo che per un collegamento con capacità trasmissiva T , la classe i avrà sempre un rendimento almeno pari a:

$$T * \left(\frac{w_i}{\sum w_j} \right)$$

Formato dei Datagrammi IPv4



Frammentazione dei Datagrammi IPv4



L'incapsulamento a livello successivo prevede porre il datagramma all'interno di un frame, quest'ultimo ha un limite di grandezza per passare attraverso i collegamenti fisici che hanno unità massima di trasmissione che **denotiamo con valore MTU**. Ogni collegamento ha un valore differente di MTU.

Se un frame quindi risulta troppo grande, per il collegamento in cui deve passare, nella versione IPv4 viene frammentato.

Il compito di tale operazione è a carico del router che si accorge della differenza di dimensioni tra il dato frame e il valore MTU del collegamento d'uscita stabilito. Tali frammenti verranno poi ricostruiti, a formare il datagramma originario, arrivati sul sistema periferico del destinatario prima di trasmettere il payload al livello di trasporto, poiché TCP/UDP si aspetta un segmento integro; dato che il datagramma viaggia frammentato per la rete in maniera autonoma non provoca cali di prestazioni nell'elaborazione dei router.

Ogni frammento, tranne l'ultimo che è sempre il più piccolo, ha dimensioni multiple di 8 byte. Come visto in precedenza il protocollo IP usa 3 campi nell'intestazione per gestire il meccanismo della frammentazione: **Identificatore, Offset, More Fragment(MF)**.

Il destinatario riconosce di aver ricevuto un frammento in tali modi:

- Il pacchetto IP ha un offset = 0 ma ha MF = 1, è arrivato il primo frammento;
- Il pacchetto IP ha un offset != 0 ma ha MF = 0, è arrivato l'ultimo frammento;
- Il pacchetto IP ha un offset != 0 ed MF != 0, frammento di mezzo.

La dimensione totale dell'intero datagramma non si può dedurre fino a che non arriva l'ultimo frammento, appena arrivato facendo la somma degli offset abbiamo la lunghezza totale.

Inoltre il destinatario appena riceve il primo frammento inizializza un timer, se tutti i frammenti non arrivano entro lo scadere del tempo, li scarta tutti.

