



# QoS in Internet



## Oltre il Best-Effort



## L'Internet attuale

- ❑ L'Internet attuale fornisce un servizio best-effort a tutte le applicazioni.
- ❑ Non fa alcuna promessa sulla qualità del servizio (QoS) che un'applicazione riceve
- ❑ Un'applicazione riceverà qualsiasi livello di prestazioni che la rete è in grado di offrire in quel momento.
- ❑ L'Internet attuale non permette alle applicazioni multimediali (sensibili al ritardo) di richiedere alcun trattamento speciale.



## L'Internet attuale

- ❑ Ai router i pacchetti sono trattati tutti allo stesso modo, compresi quelli sensibili al ritardo.
- ❑ Per rovinare la qualità di una chiamata IP in corso in Internet basta una quantità sufficiente di traffico che mandi la rete in congestione.
- ❑ Ciò causerà l'aumento dei ritardi e la perdita dei pacchetti.



## L'Internet attuale

- ❑ Spesso in internet ogni connessione esiste solo per i due host alle sue estremità, che identificano tutti i pacchetti che si scambiano come appartenenti alla connessione stessa. Tali pacchetti, una volta usciti dall'host sorgente e prima di entrare in quello di destinazione, perdono la loro "reciproca parentela" e diventano entità indipendenti.
- ❑ Di conseguenza, le risorse della rete sono assorbite in modo del tutto incontrollato dai vari flussi di pacchetti e le prestazioni ottenute variano in modo quasi casuale a seconda del livello momentaneo di congestione.



## Qualità del Servizio (QoS)

- ❑ Alcune connessioni sono involontariamente favorite dalla rete, mentre altre sono penalizzate; questo è il prezzo da pagare per avere una rete con architettura semplice e priva di tariffazione.
- ❑ I modelli di *QoS* per IP sono stati introdotti proprio per cambiare questa situazione e per dare la possibilità agli utenti (eventualmente paganti) di richiedere alla rete determinate prestazioni garantite.

## QoS

- ❑ La *QoS* descrive il livello di prestazione che deve essere assicurato per una particolare applicazione.
- ❑ La *QoS* può essere definita in modo :
  - "assoluto" (***Performance Assurance***), definendo i valori che devono essere rispettati da un insieme di parametri "prestazionali" (es. ritardo massimo, probabilità di perdita di pacchetti, ecc),
  - o "relativo" (***Service Differentiation***), definendo le modalità di trattamento di una classe di traffico rispetto alle altre (es. livello di priorità di servizio, livello di priorità di scarto, ecc.).

## QoS

- ❑ Una rete è in grado di garantire un fissato livello di *QoS* in due modi:
  - *Overprovisioning*: le risorse di rete sono dimensionate in modo che, nelle condizioni peggiori, il carico sia inferiore alla soglia minima necessaria a soddisfare i contratti concordati con gli utenti;
  - *Admission Control*: il traffico è controllato preventivamente e sarà accettato solo se le risorse di rete saranno sufficienti a garantire i livelli di qualità richiesti dagli utenti.



## QoS

- ❑ La *QoS* può essere garantita:
  - per *flusso*, per connessione, per chiamata
  - per *aggregati* (o *classi*) di flussi, di connessioni, di chiamate.
- ❑ Un ***flusso*** è definito dalla 5-pla: *Source IP address, Source port number, Destination IP address, Destination port number, Protocol*.
- ❑ Una ***classe o aggregato di traffico*** comprende un insieme di flussi aventi requisiti simili di *QoS*.

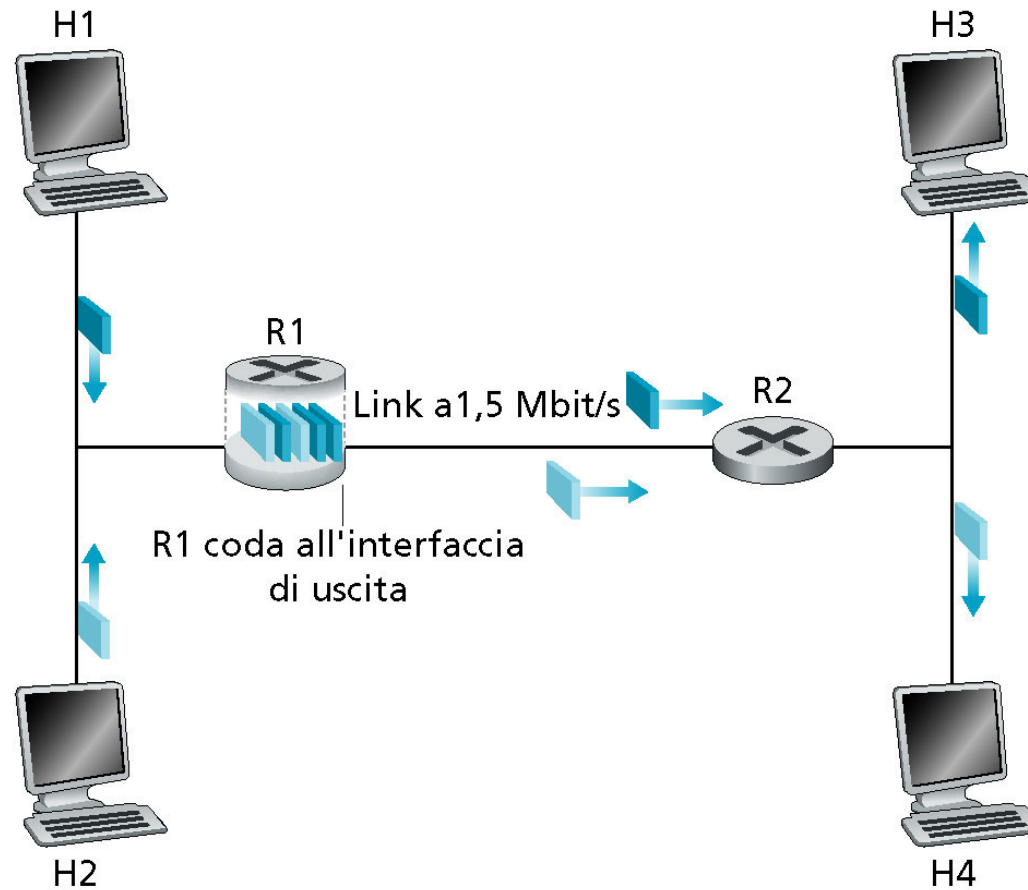
## QoS

- ❑ Una politica di *QoS* orientata al trattamento dei singoli flussi (*per flow QoS*) assicura prestazioni migliori sia per quanto riguarda i singoli flussi che per quanto riguarda l'utilizzazione delle risorse di rete. Ha inoltre una notevole complessità dovuta ai meccanismi di segnalazione e alla necessità di monitorare i singoli flussi di pacchetti in rete nonché una bassa scalabilità.
- ❑ Una politica di *QoS* orientata al trattamento degli aggregati di flussi (*aggregate traffic QoS*) fornisce prestazioni non ottimali ma ha una complessità minore ed è maggiormente scalabile.

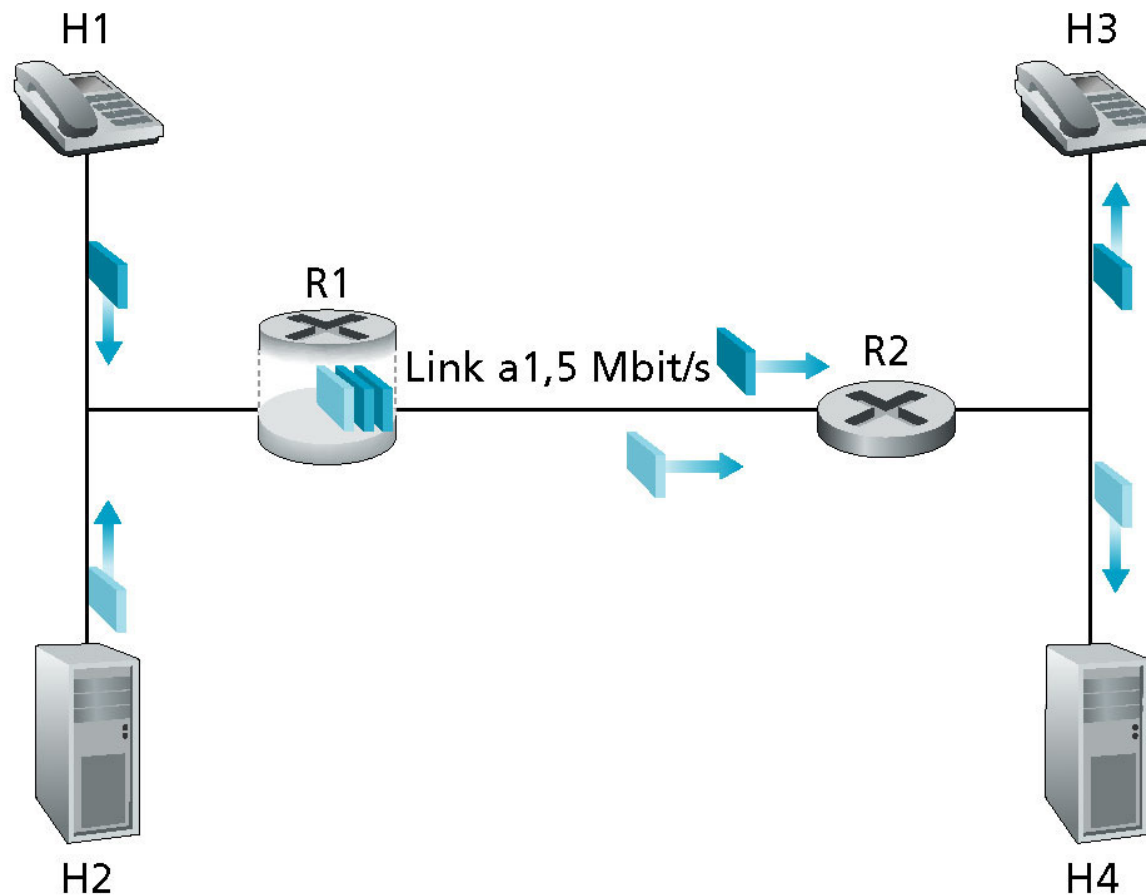


- ❑ In questa lezione vedremo nuovi componenti strutturali che possono essere aggiunti all'architettura di Internet per proteggere un'applicazione da fenomeni come la congestione

# Scenario di riferimento



## Esempio 1: un'applicazione audio a 1 Mbs e un trasferimento FTP.





- ❑ Supponiamo di avere un'applicazione audio a 1 Mbs che condivide il link a 1,5 Mbs fra R1 e R2 con un'applicazione FTP che sta trasferendo file da H2 a H4.
- ❑ Nell'Internet best-effort i pacchetti audio e FTP sono mescolati nella coda in uscita da R1 e tipicamente trasmessi nell'ordine FIFO.
- ❑ Se una raffica di pacchetti FTP riempisse la coda causerebbe eccessivo ritardo o perdita di pacchetti audio per la saturazione della coda



- ❑ Come si può risolvere il problema?
- ❑ L'applicazione FTP non ha vincoli di tempo quindi potremmo dare una priorità ai pacchetti audio in R1
- ❑ Grazie alla priorità un pacchetto audio nel buffer di uscita di R1 dovrebbe essere trasmesso prima di qualsiasi pacchetto FTP
- ❑ Il link da R1 a R2 appare così come un link dedicato al traffico audio e FTP lo usa solo quando non c'è traffico audio accodato



## I principio

- ❑ Perché R1 possa distinguere fra traffico audio e pacchetti FTP nella sua coda, ciascun pacchetto dovrà essere contrassegnato come appartenente a una delle due "classi" di traffico.

### I principio

La marcatura dei pacchetti permette a un router di distinguere fra pacchetti appartenenti a due diverse classi di traffico.





- ❑ In realtà la marcatura esplicita è un mezzo con il quale distinguere i pacchetti. Il contrassegno portato da un pacchetto non può, di per sé, implicare che un pacchetto riceva una data QoS.
- ❑ La marcatura è un meccanismo per distinguere i pacchetti.
- ❑ Il modo in cui un router distingue tra i pacchetti per trattarli in modo diverso è una decisione politica



## Esempio II: un'applicazione audio che si comporta in modo scorretto e un trasferimento FTP.

- ❑ Supponiamo ora che in qualche modo il router sappia che deve dare la priorità ai pacchetti dell'applicazione audio a 1 Mbs.
- ❑ Poiché la velocità del link in uscita è 1,5 Mbs anche se i pacchetti FTP ricevono una bassa priorità, essi riceveranno in media 0,5 Mbs.



- ❑ Ma cosa succede se l'applicazione audio comincia a inviare pacchetti al tasso di 1,5 Mbs o oltre?
- ❑ In questo caso i pacchetti FTP non riceveranno alcun tipo di servizio sul link R1-R2.
- ❑ Idealmente sarebbe desiderabile qualche grado di isolamento tra i flussi, per proteggere un flusso da un altro che si comporta in modo scorretto



## II principio

### II principio

E' desiderabile fornire un grado di isolamento tra i flussi di traffico, in modo che un flusso non subisca gli effetti avversi di un altro flusso che si comporta in modo scorretto

Si possono seguire due approcci:

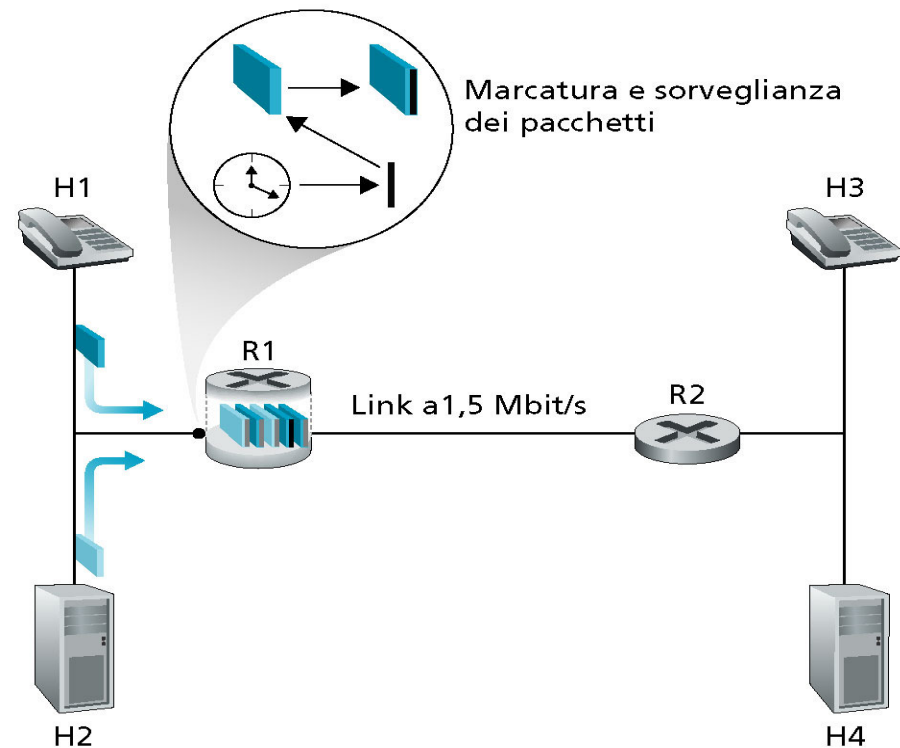
- ❑ Sorvegliare i flussi di traffico
- ❑ Assegnare ai flussi dei canali logici separati



## Sorvegliare i flussi di traffico

- ❑ Se un flusso di traffico deve soddisfare certi criteri (per es. il flusso audio non superi la velocità di picco di 1 Mbs) allora un meccanismo di sorveglianza può essere posto per assicurare che questo criterio sia rispettato.
- ❑ Se l'applicazione esaminata si comporta in modo scorretto, il meccanismo di sorveglianza prenderà alcune decisioni. (es. scartando o ritardando i pacchetti che violano i criteri)

- Il Token Bucket è il meccanismo di sorveglianza (policing) più usato



Legenda:



Misurazione e sorveglianza

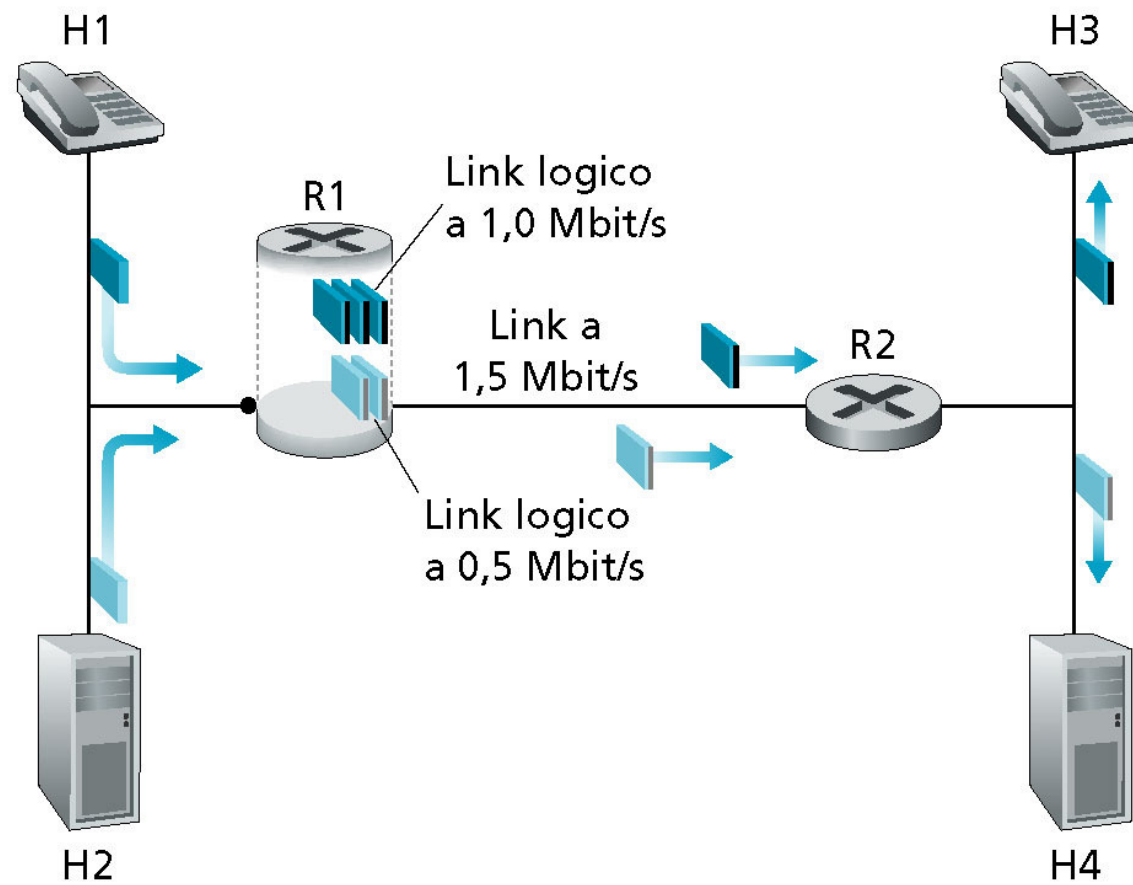


Contrassegni



## Assegnare ai flussi dei canali logici separati

- ❑ L'altro approccio per fornire l'isolamento tra i flussi prevede un meccanismo di scheduling dei pacchetti che a livello di link assegni esplicitamente a ciascun flusso una quantità fissa di larghezza di banda.
- ❑ Es. In R1 al traffico audio potrebbe essere assegnato 1 Mbs e al flusso FTP 0,5 Mbs. Così da vedere dei link logici con capacità di 1,0 e 0,5 Mbs.







- ❑ Con un preciso controllo della larghezza di banda allocata a livello di link, un flusso può impiegare solo la larghezza di banda che gli è stata assegnata: in particolare non può utilizzare la banda che al momento non è usata da altre applicazioni.
- ❑ E' desiderabile però utilizzare la banda nel modo più efficiente possibile.



## III principio

### III principio

Mentre si fornisce l'isolamento tra i flussi, è desiderabile usare le risorse (es. buffer e larghezza di banda) il più efficientemente possibile.



## Esempio III: due applicazione audio a 1 Mbs su un link sovraccarico a 1,5 Mbs.

- ❑ Supponiamo, ora, di avere due connessioni audio a 1 Mbs che trasmettono i loro pacchetti su un link di 1,5 Mbs
- ❑ La velocità combinata dei due flussi (2 Mbs) eccede la capacità del link.
- ❑ Anche con classificazione e marcatura (I principio), isolamento dei flussi (II principio) e condivisione della banda non usata (III principio), di cui non c'è traccia, chiaramente non si può fare molto.



- ❑ Non c'è abbastanza banda per soddisfare le necessità delle applicazioni
- ❑ Se esse si spartissero in modo uguale la capacità di banda del link riceverebbero 0,75 Mbs ciascuna.
- ❑ Le applicazioni perderebbero il 25% dei pacchetti trasmessi rendendole inutilizzabili a causa della bassa qualità ricevuta.



- ❑ Ad un flusso che richiede una minima qualità di servizio per essere considerato “utilizzabile”, la rete dovrebbe garantire le condizioni per poter essere utilizzata oppure per impedirne l’utilizzo stesso.
- ❑ La rete telefonica è un esempio di rete che blocca le chiamate facendo risultare all’utente un segnale di occupato.
- ❑ Non c’è guadagno nel permettere a un flusso di accedere alla rete se non riceve la necessaria QoS da essere considerato “utilizzabile”
- ❑ Implicita con la necessità di fornire una QoS garantita a un flusso è la necessità che esso dichiari i suoi requisiti di QoS.



- Il processo di ammissione che fa sì che un flusso dichiari i suoi requisiti di QoS e che indichi alla rete di accettare il flusso oppure di bloccarlo è detto processo di **ammissione della chiamata**.

#### IV principio

E' necessario un processo di ammissione della chiamata in cui i flussi dichiarano i loro requisiti di QoS per poter essere ammessi alla rete (alla QoS richiesta) o bloccati dalla rete (se la QoS richiesta non può essere fornita)



## Sommario dei principi

QoS for networked applications

packet classification

Isolation: scheduling  
and policing

high resource  
utilization

Call admission



## Meccanismi di scheduling e policing





## Meccanismi di scheduling

- ❑ I pacchetti appartenenti a diversi flussi della rete sono *multiplati* insieme e accodati per la trasmissione ai buffer di uscita associati con un link.
- ❑ Il modo in cui i pacchetti delle code sono selezionati per la trasmissione sul link è detto: **modalità di scheduling del link**
- ❑ La modalità di scheduling nel link gioca un ruolo importante nel fornire le garanzie di QoS

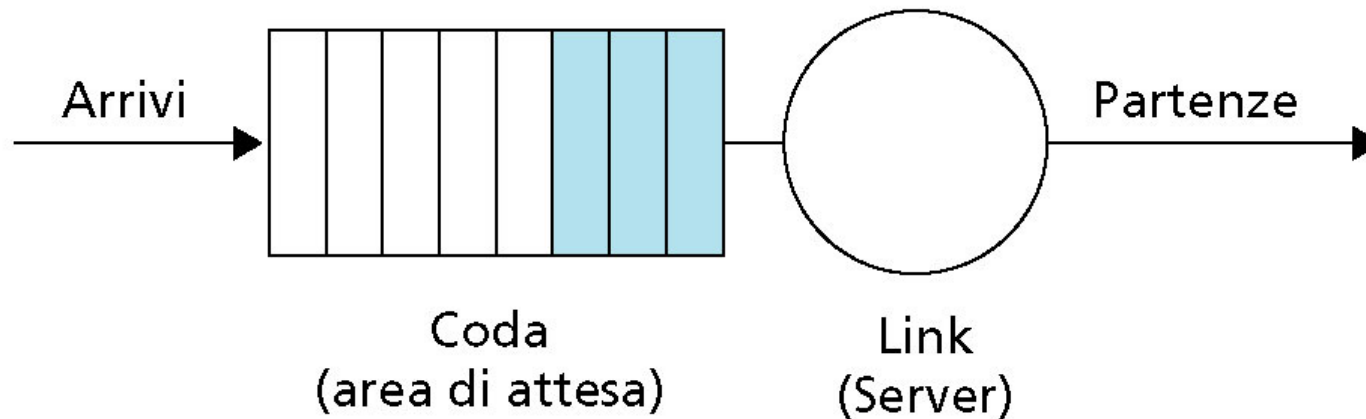


## Meccanismi di scheduling

- ❑ Consideriamo alcune delle più importanti modalità di scheduling di link:
  - First In First Out (FIFO)
  - Accodamento Prioritario
  - Round Robin
  - Accodamento equamente pesato (WFQ)



## First In First Out

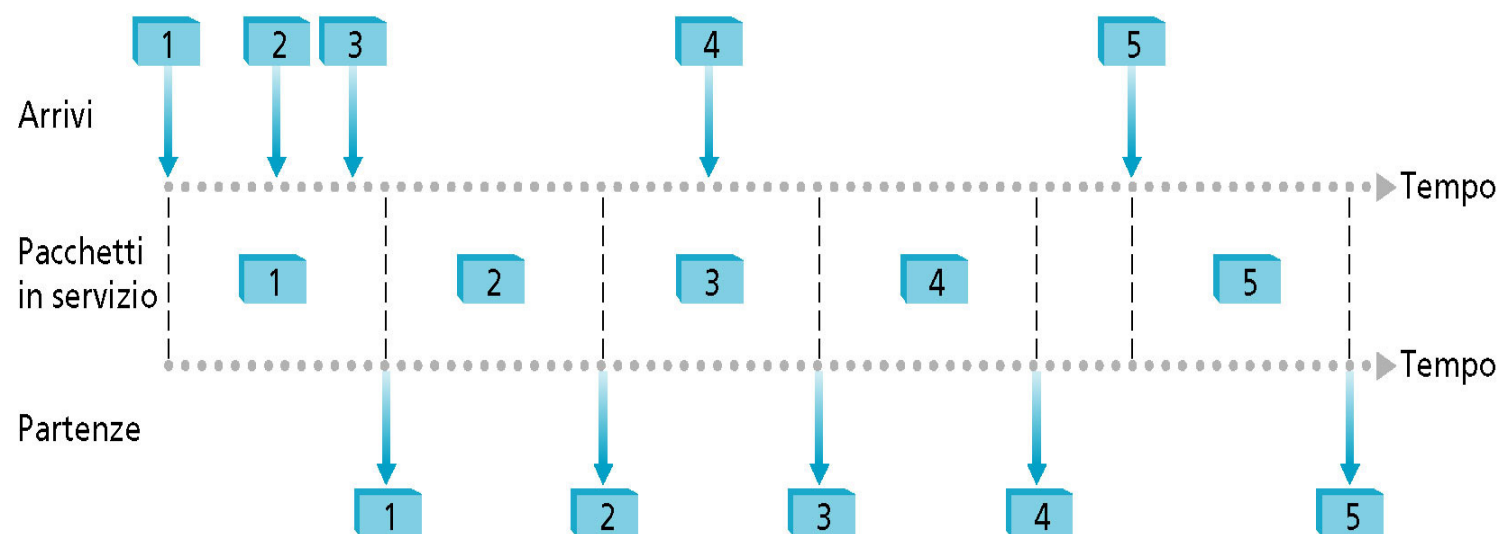


I pacchetti vengono accodati nel buffer se il link è occupato, ma se lo spazio nel buffer non è sufficiente per contenere il pacchetto in arrivo, la **politica di scarto del pacchetto** della coda determina se:

- ☐ Il pacchetto deve essere scartato
- ☐ Altri pacchetti possono essere rimossi dalla coda per fare spazio al pacchetto in arrivo.

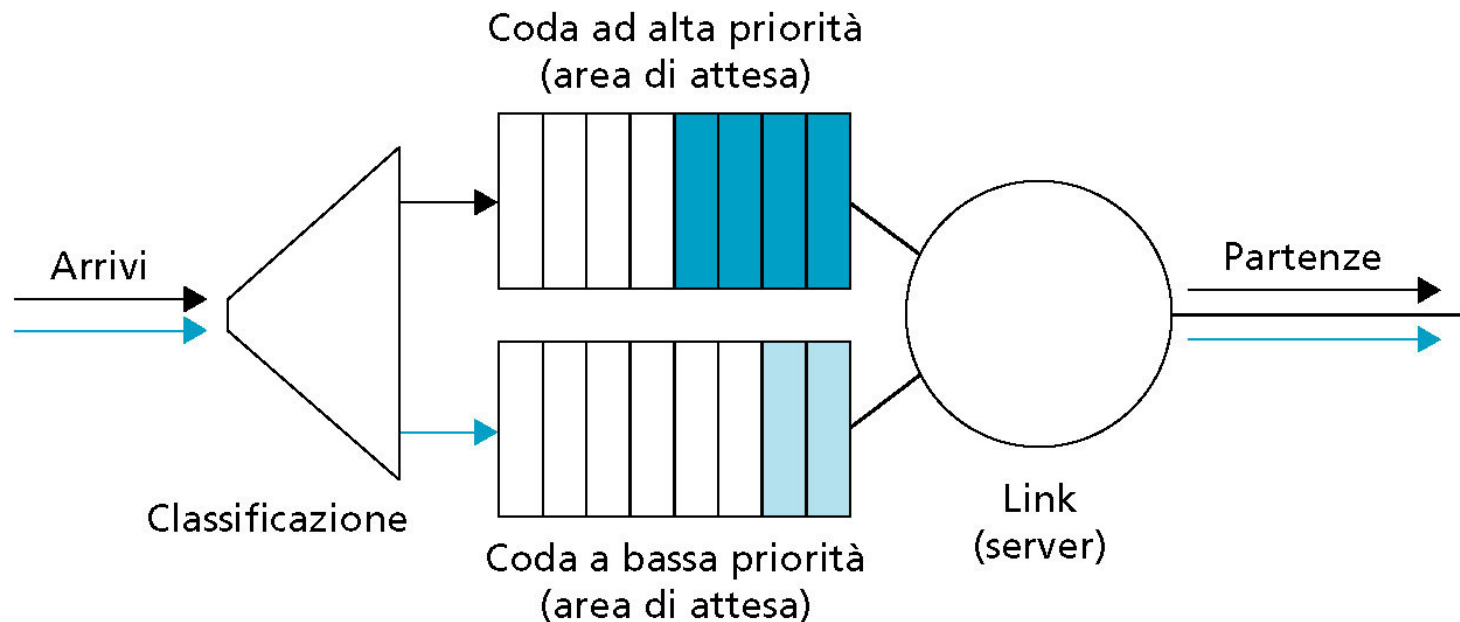


# FIFO





## Accodamento prioritario



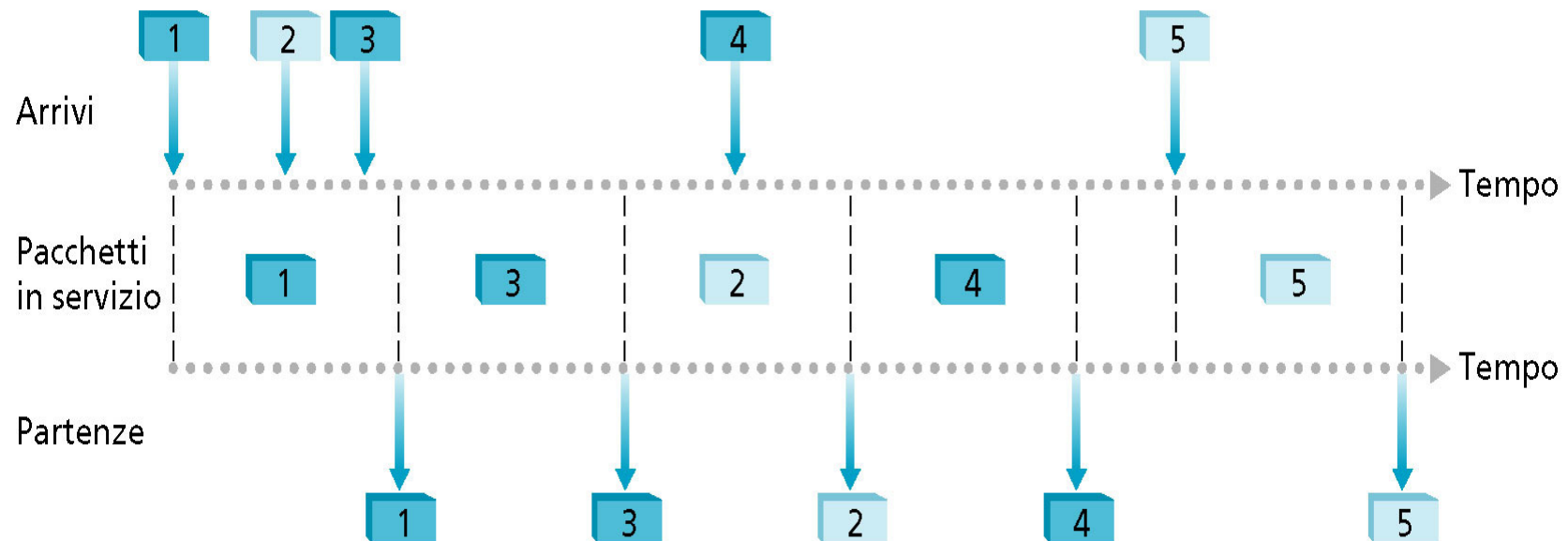
Con questa tecnica i pacchetti che arrivano sul link di uscita sono classificati all'interno di due o più code di priorità nelle code d'uscita.



## Accodamento prioritario

- ❑ La classe di priorità del pacchetto può dipendere da una marcatura esplicita che è riportata nell'intestazione del pacchetto (es. ToS IPv4)
- ❑ Nella scelta del pacchetto da trasmettere, tale modalità, trasmetterà un pacchetto della più alta classe di priorità la cui coda non sia vuota.
- ❑ La scelta tra i pacchetti di una stessa classe di priorità, di solito, è effettuata in modo FIFO.

# Accodamento prioritario



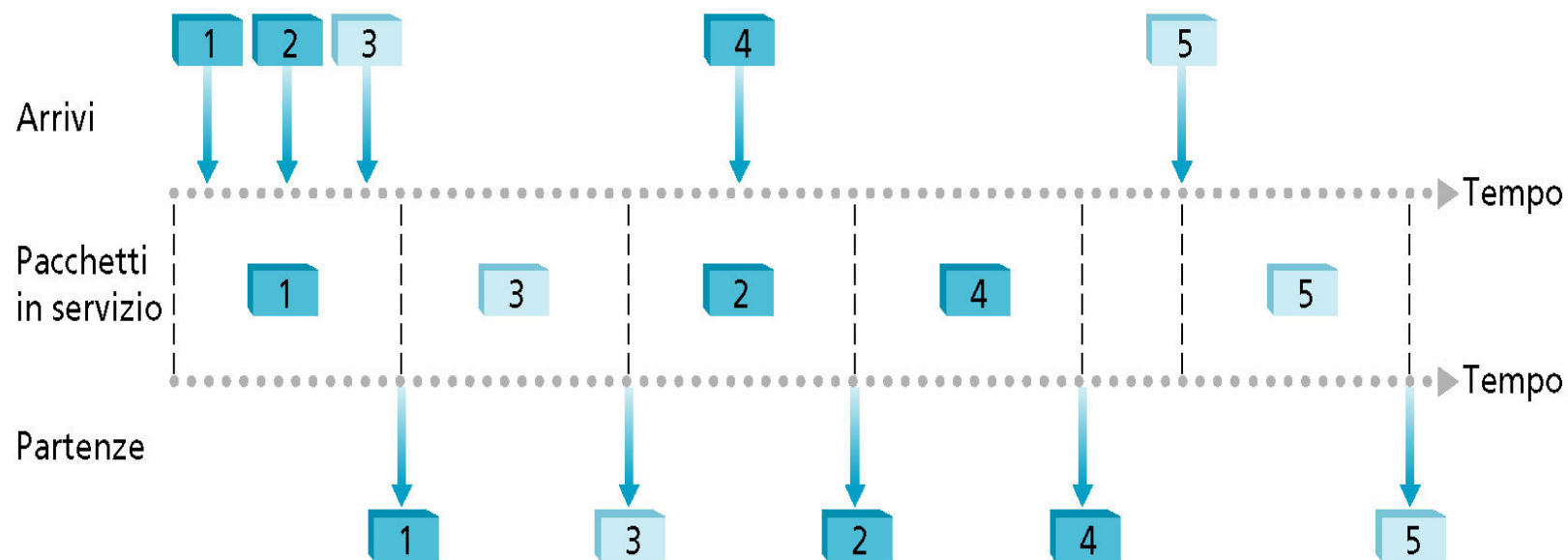


## Round Robin

- ❑ I pacchetti, come nel caso di accodamento prioritario, sono ancora smistati in classi
- ❑ Invece di avere una stretta priorità assegnata alle classi la modalità round robin alterna i servizi tra le classi
- ❑ E' una modalità di accodamento Work-Conserving



# Round Robin



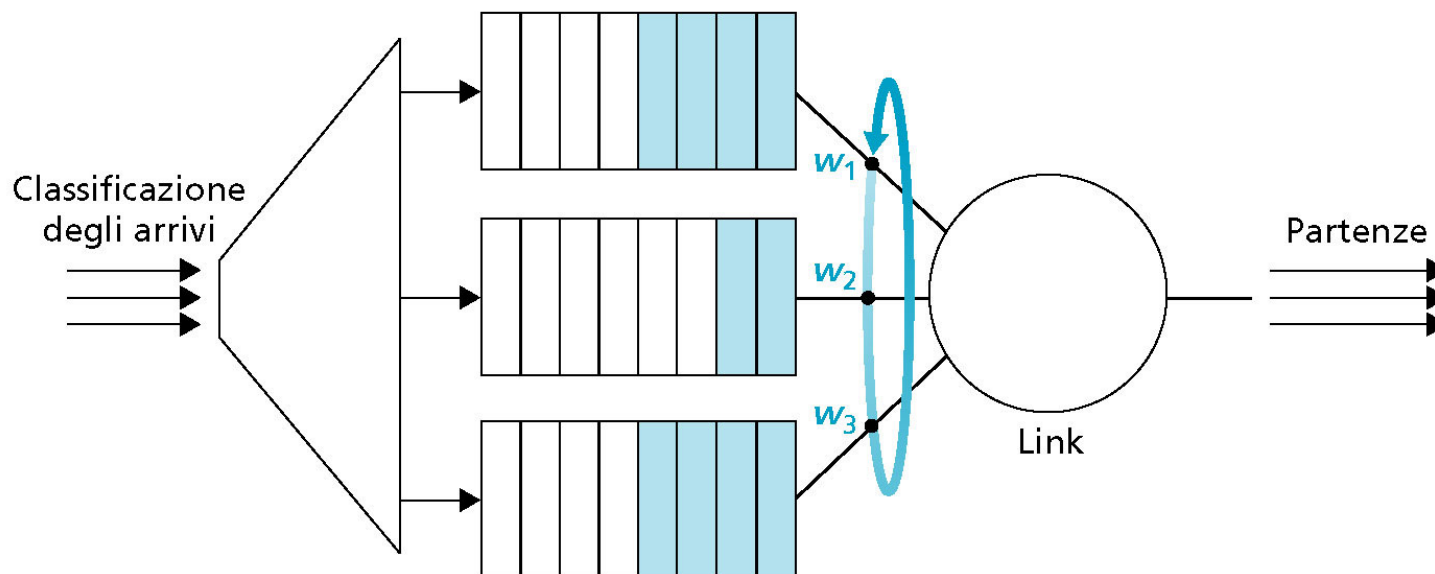


## Weighted Fair Queuing (WFQ)

- ❑ E' una astrazione generalizzata dell'accodamento round robin che ha trovato un impiego considerevole nelle architetture per la QoS
- ❑ I pacchetti in arrivo sono ancora classificati e vengono accodati nella loro area di attesa della classe appropriata
- ❑ Come nella modalità round robin servirà ancora le classi in modo ciclico
- ❑ E' una modalità di lavoro work-conserving



# Weighted Fair Queuing (WFQ)





## Weighted Fair Queuing (WFQ)

- ❑ A ciascuna classe  $i$  è assegnato un peso  $w_i$
- ❑ Alla classe  $i$  è garantito di ricevere una frazione di servizio uguale a  $w_i/(\sum w_j)$  dove la somma al denominatore è effettuata su tutte le classi che hanno pacchetti accodati
- ❑ Per un link con velocità di trasmissione  $R$ , la classe  $i$  raggiungerà sempre un throughput di almeno  $R * w_i/(\sum w_j)$ .



## Meccanismi di Policing

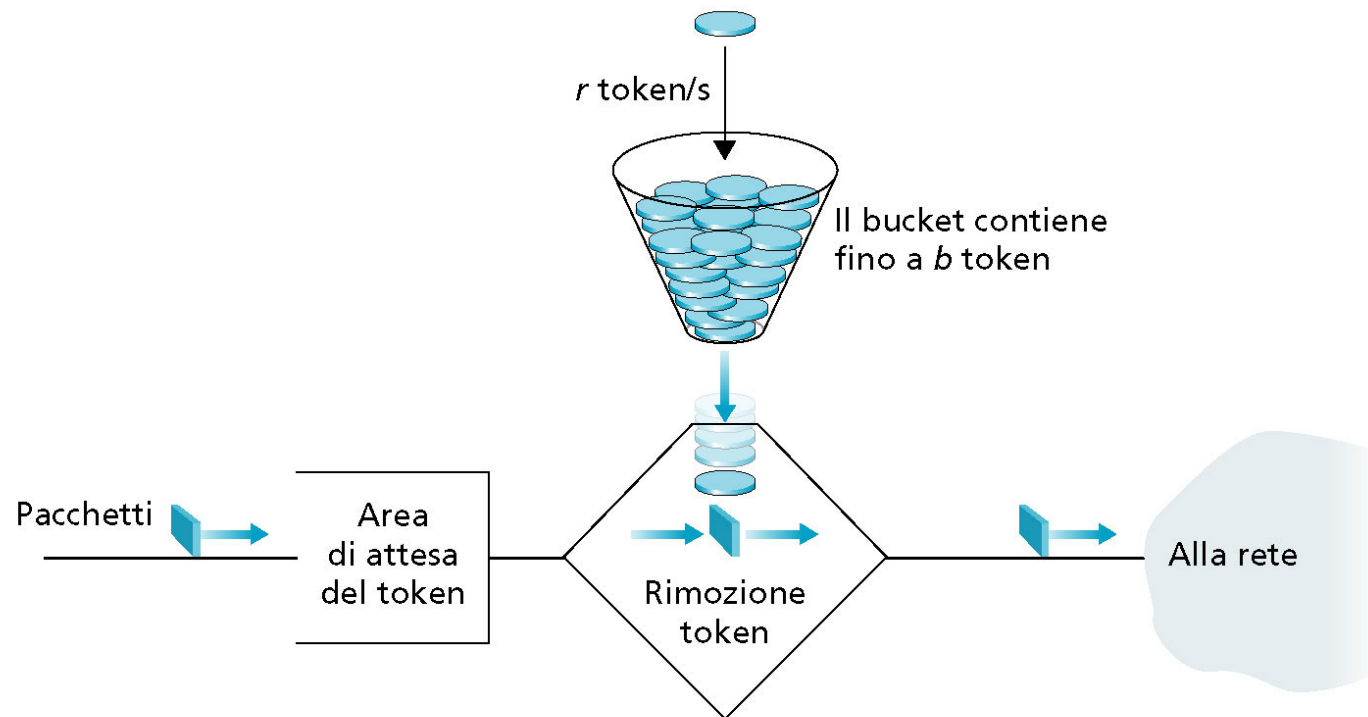
- ❑ Il policing è la regolazione della velocità a cui a un flusso è permesso di iniettare pacchetti nella rete.
- ❑ Possiamo identificare tre importanti criteri di sorveglianza:
  - Velocità media
  - Velocità di picco
  - Dimensione della raffica (burst)



## Token Bucket

- ❑ Un token bucket è un contenitore che può contenere fino a  $b$  token (gettoni)
- ❑ I token sono aggiunti al contenitore come segue:
  - Nuovi token sono sempre generati a una velocità di  $r$  token al secondo
  - Se il contenitore contiene meno di  $b$  token al momento della generazione di un token, esso verrà aggiunto al contenitore altrimenti viene ignorato ed il contenitore rimarrà con  $b$  token (cioè pieno).

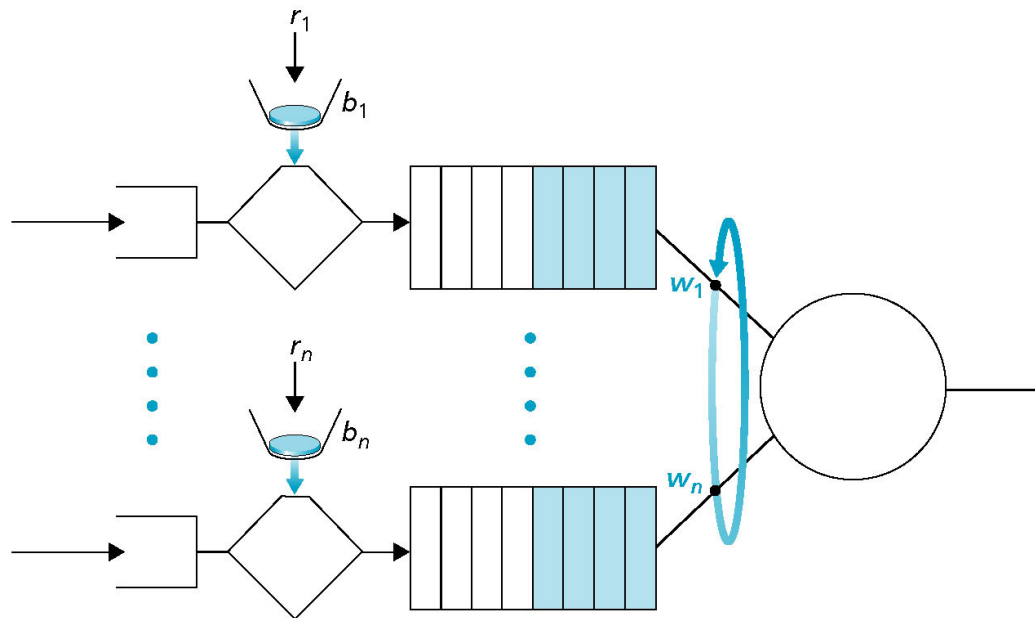
# Token Bucket



Numero massimo di pacchetti spediti dalla rete in un qualsiasi intervallo  $t$ :  $rt+b$ .

# Flussi multiplati con scheduling WFQ

Ritardo massimo  
dimostrabile in coda



$$d_{\max} = \frac{b_1}{R \cdot w_1 / \sum w_j}$$

if  $r < R \cdot w_1 / \sum w_j$





## Architetture per la QoS:

- ❑ Integrated Services (Servizi Integrati)
- ❑ Differentiated Services (Servizi Differenziati)



## Servizi Integrati

- ❑ Nel 1990 viene formato un gruppo di lavoro dell'IETF sui servizi integrati
- ❑ Esso pose l'attenzione sulla definizione di un minimo insieme di requisiti necessari per aiutare la modalità di inoltramento corrente di internet: il best effort
- ❑ L'IntServ mira a fornire garanzie per-flusso a sessioni di applicazioni individuali.

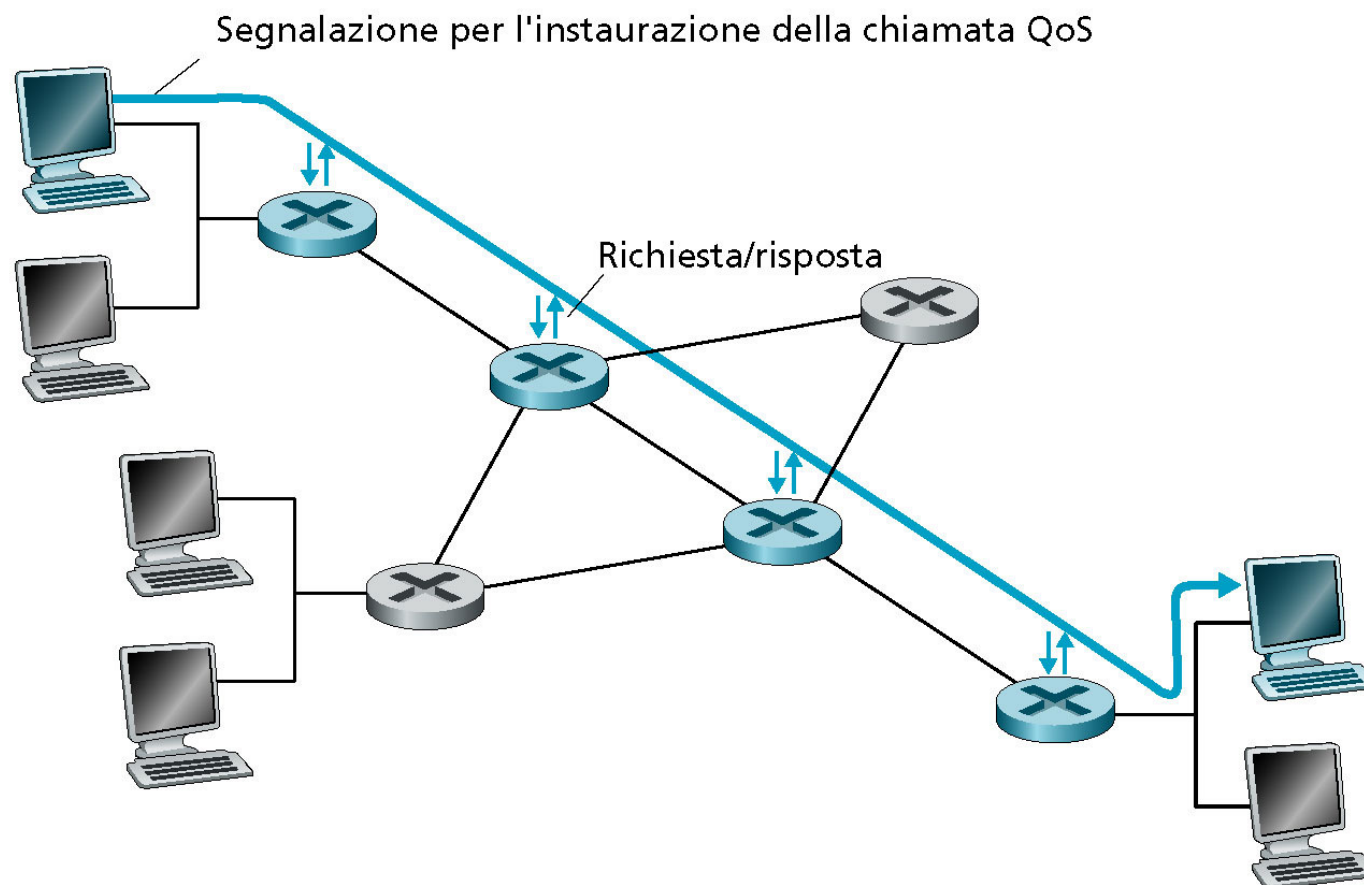


## Servizi Integrati (IntServ)

- ❑ Definisce nuove classi di servizio da affiancare al best effort
- ❑ L'idea è che ogni classe si base sulla richiesta di particolari requisiti di QoS
- ❑ Questa architettura fornisce servizi orientati al singolo flusso basati su una comunicazione orientata alla connessione



- ❑ L'architettura IntServ si basa su due concetti base
  - **Risorse riservate.** A ogni router è richiesto di conoscere la quantità delle sue risorse (buffer, larghezza di banda) già riservate
  - **Impostazione della chiamata.** Una sessione che richiede garanzie di QoS deve prima essere in grado di prenotare risorse sufficienti a ciascun router della rete sul suo percorso sorgente-destinazione, per assicurare i suoi requisiti di QoS.





## Flow descriptor

- ❑ Ogni richiesta di prenotazione richiede al suo interno un oggetto attraverso cui l'host sorgente specifica la QoS richiesta ed identifica il flusso di dati cui riservare quella QoS.
- ❑ Tale oggetto è detto **flow descriptor** costituito:
  - **Flowspec**: specifica la QoS desiderata
  - **Filter spec**: identifica il flusso di dati al quale riservare la QoS indicata nel flowpsec
- ❑ In ogni nodo ogni richiesta di prenotazione delle risorse interagisce con due entità locali:
  - **Admission control**
  - **Policy control**



## Admission control

- ❑ L'admission control verifica se la richiesta può essere esaudita, cioè se sono presenti risorse sufficienti a garantire la QoS specificata nel flowspec senza incorrere nel rischio che si deteriori la QoS riservata agli altri flussi di dati che in quel momento stanno attraversando il nodo in oggetto.



## Policy control

- ❑ Il policy control verifica invece che l'host richiedente sia autorizzato ad inoltrare tale richiesta, ed in più memorizza dei dati necessari successivamente alla tariffazione del servizio offerto.





- ❑ Le informazioni contenute negli oggetti flowspec e filter spec vengono rispettivamente utilizzate per configurare i parametri di due moduli:
  - Packet scheduler
  - Packet classifier



## Packet classifier

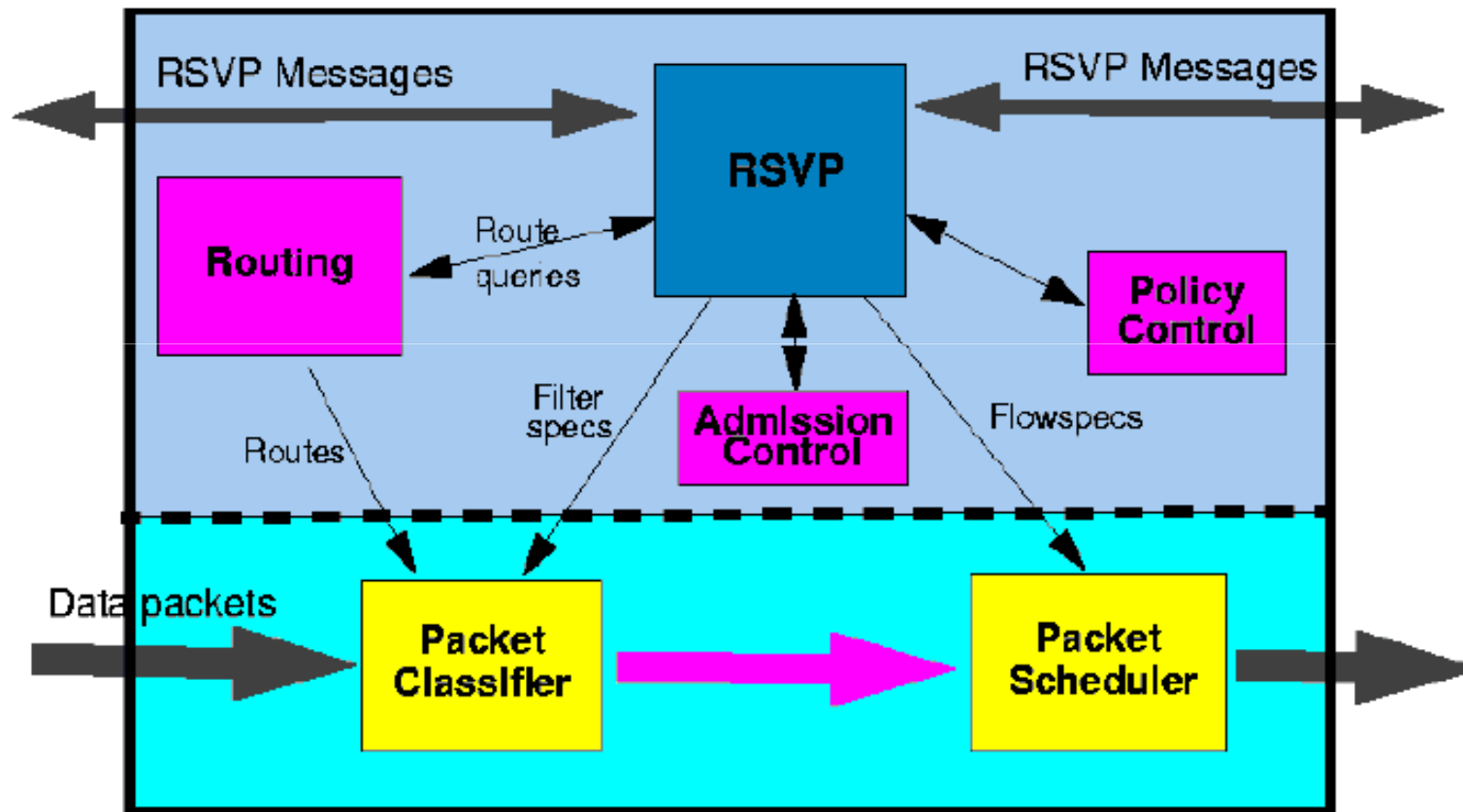
- ❑ Ha il compito di dividere i pacchetti che giungono al nodo sulla base della QoS loro assegnata, esso in pratica identifica i vari flussi di dati cui sono riservate QoS diverse e li accoda in modo opportuno.
- ❑ Il classificatore utilizzato è di tipo multi-field (MF) cioè classifica i pacchetti entranti in base ad una combinazione di campi dell'intestazione del pacchetto IP.



## Packet scheduler

- ❑ Si occupa dell'inoltro dei pacchetti sulle interfacce uscenti dei nodi, basandosi sull'utilizzo di una o più code gestite secondo una politica prestabilita.
- ❑ Esso deve inviare i datagrammi IP sul mezzo fisico in maniera da rispettare la QoS negoziata selezionando i pacchetti delle code ed instradandoli in rete al momento opportuno.

# Componenti in un nodo IP





- ❑ Il flowspec in una richiesta di prenotazione è in generale composto da due set di parametri:
  - Rspec: che specifica la QoS richiesta e
  - Tspec: che descrive le caratteristiche del flusso di dati cui assegnare la QoS richiesta.



## Classificazione delle applicazioni

- ❑ La struttura degli IntServ ha classificato le applicazioni nelle seguenti categorie:
  - Elastic Traffic
  - Real Time Traffic



## Elastic Traffic

- ❑ E' il traffico tradizionale delle reti TCP/IP
- ❑ È indifferente alle variazioni di ritardo di transito dei pacchetti
- ❑ Applicazioni come Telnet, FTP, Web browsing rientrano in questa categoria
- ❑ Il modello di servizio best-effort è accettabile per queste applicazioni



## Real Time Traffic

- ❑ Le applicazioni real-time sono le applicazioni dette playback application, cioè applicazioni in cui in fase di ricezione è necessario riprodurre i pacchetti partiti in un istante esatto detto playback point
- ❑ Questo induce la necessità di un delay bound da non superare affinché i pacchetti ricevuti siano ancora validi.
- ❑ Necessita un trattamento preferenziale rispetto alle applicazioni elastiche





□ Le applicazioni Real Time si dividono in due sottocategorie:

- Intolleranti: che scartano completamente un pacchetto scaduto
- Tolleranti: che riescono ad adeguarsi a variazioni del ritardo compensando con una perdita di qualità

I parametri di QoS richiesti sono:

- Throughput
- Delay
- Jitter
- Perdita di pacchetti



## Protocollo RSVP

- ❑ Il protocollo RSVP (ReSerVation Protocol) permette di prenotare risorse all'interno della rete internet in modo da garantire una certa QoS a determinati flussi all'interno di una data sessione.
- ❑ Una sessione è un insieme di uno o più flussi di dati individuato da una certa destinazione ed un certo protocollo di trasporto
- ❑ Per poter realizzare i propri obiettivi il protocollo RSVP usa dei messaggi di controllo incapsulati in pacchetti IP ed instradati dai router alla stessa stregua dei normali datagrammi IP.



- ❑ È stato ideato per venire incontro alle esigenze delle applicazioni multimediali
- ❑ E permettere a tali applicazioni di poter usufruire di un servizio di trasporto dati con una certa qualità garantita e negoziata precedentemente all'invio dei dati stessi
- ❑ Quando un'applicazione su un terminale IP richiede per un flusso di dati una certa QoS il protocollo RSVP si occupa di inoltrare tale richiesta a ciascun nodo della rete IP



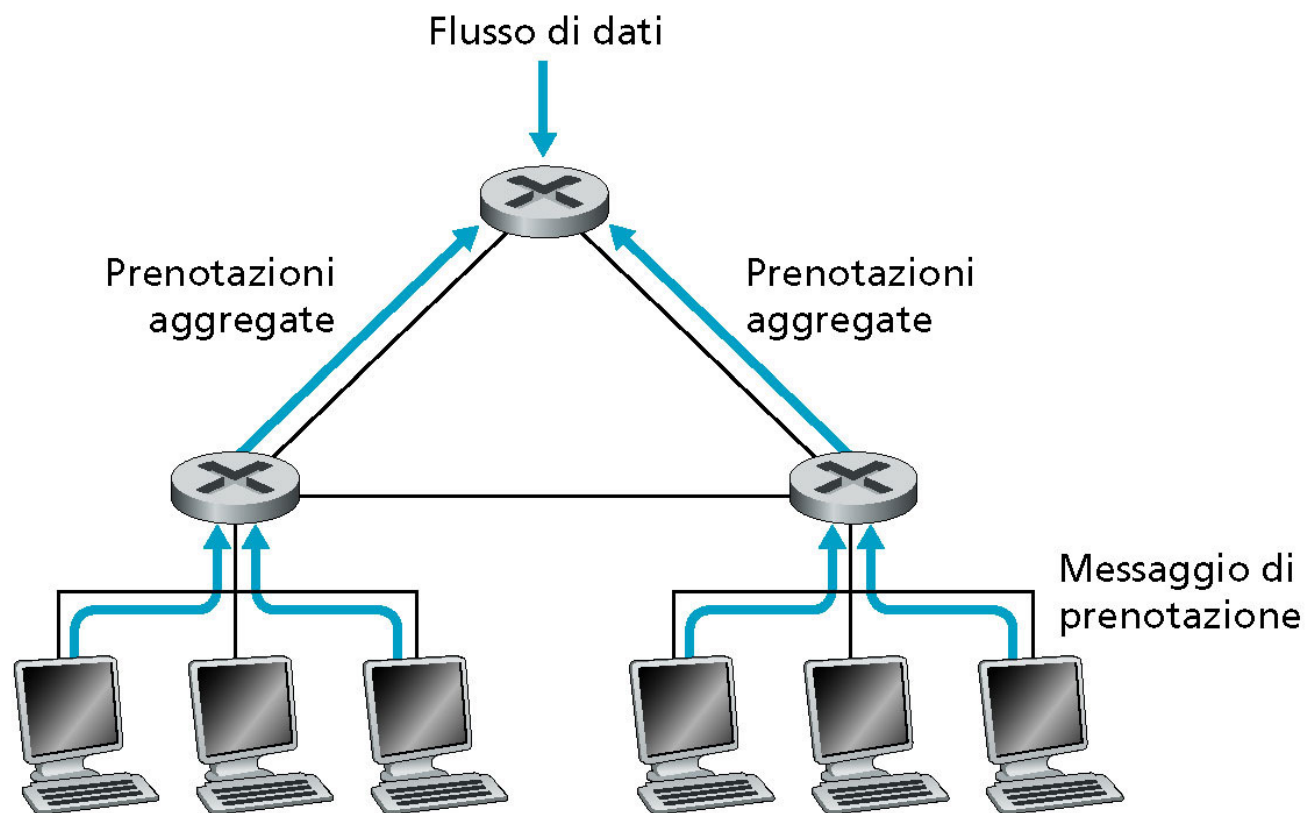
- ❑ Il protocollo introduce quindi un meccanismo di discriminazione tra i vari flussi di dati in quanto consente di riservare a determinate applicazioni un trattamento privilegiato



## Caratteristiche del protocollo

Le caratteristiche salienti del protocollo sono:

- È un protocollo di controllo che si poggia sullo strato di rete IP
- È in grado di prenotare risorse all'interno della rete sia per trasmissioni unicast che multicast
- È orientato al ricevitore, ossia è l'host ricevente che, sulla base della QoS con cui desidera ricevere i dati, decide la quantità di risorse da prenotare
- Si adatta in maniera veloce e robusta ai cambiamenti dei percorsi





- Non interferisce con i protocolli di routing, si limita solo ad utilizzare le informazioni presenti nelle tabelle di routing
- Trasporta in maniera trasparente i messaggi per il controllo del traffico; vale a dire che esso non interpreta i dati in essi contenuti ma li trasferisce soltanto ad appositi moduli
- Non specifica come la rete fornisca la banda prenotata dai flussi



## Messaggi fondamentali di RSVP

- ❑ I messaggi fondamentali utilizzati dal protocollo RSVP sono due:
  - PATH
  - RESV





## Messaggio di Path

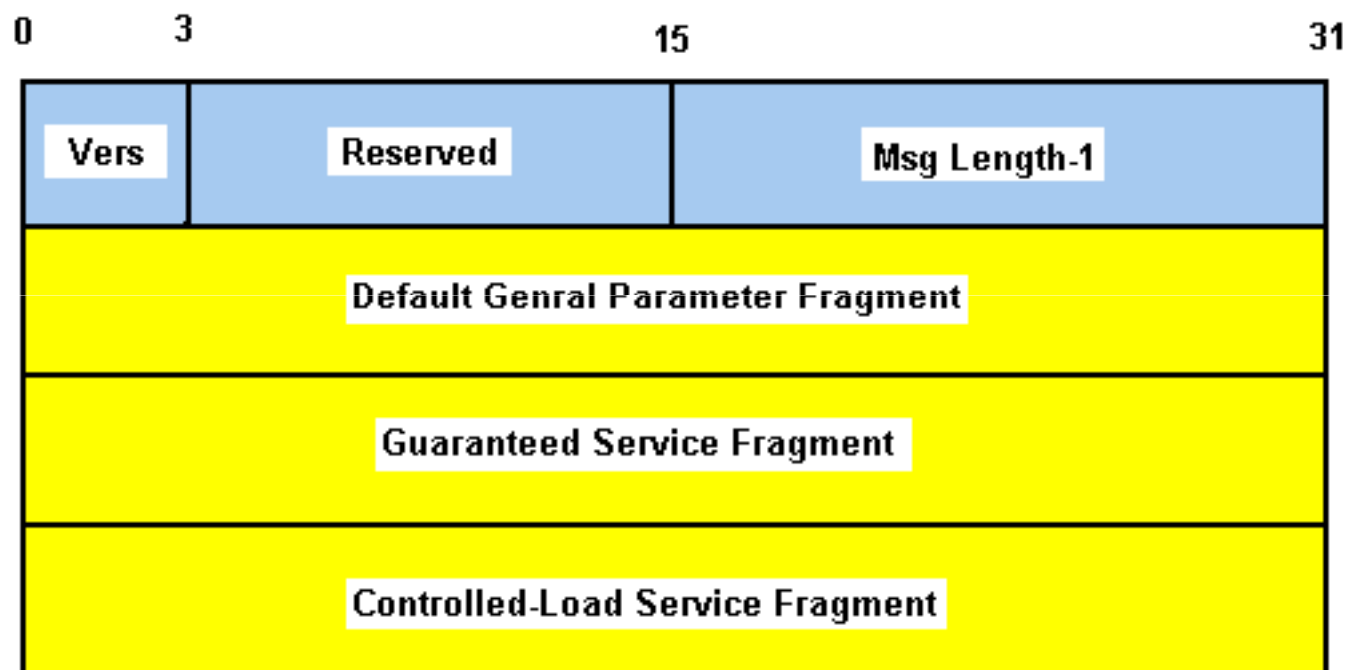
- ❑ Quando un host sorgente vuole utilizzare il protocollo RSVP, per consentire che il proprio flusso di dati venga ricevuto con Qos, invia un messaggio di Path che ha come indirizzo di destinazione l'indirizzo IP del ricevitore al quale vuole trasmettere.
- ❑ Questo messaggio viene instradato come un normale messaggio IP, ma ogni router che lo elabora deve immagazzinare alcune informazioni, dette Path State:
  - Il router deve tenere nota dell'indirizzo IP del nodo precedente, così che i messaggi di prenotazione RESV possa essere instradato correttamente
  - Il path trasporta inoltre le informazioni necessarie per l'attuazione del meccanismo di prenotazione denominato One Pass With Advertising (OPWA).



- ❑ *OPWA* si riferisce al modello di prenotazione nel caso in cui la sorgente include l'oggetto *ADSEPC* nel suo messaggio di *Path* per abilitare il ricevente a determinare il servizio end-to-end che risulterà da una data richiesta di prenotazione.
- ❑ L'*ADSPEC* è un oggetto opzionale che la sorgente può includere nel suo messaggio di *Path* per avvertire il ricevente circa le caratteristiche della comunicazione end-to-end. L'informazione può essere usata dal ricevente per determinare il livello di prenotazione richiesto per raggiungere il livello di *QoS* end-to-end desiderato.



# Formato dell'oggetto ADSPEC





- ❑ L'*ADSPEC* consiste di un messaggio di intestazione, un frammento che contiene i parametri generali di default denominato *Default General Parameters fragment*, ed almeno uno dei frammenti relativi alla classe di servizio che può essere selezionata dall'applicazione ricevente, e cioè almeno uno tra *Guaranteed Services fragment* e *Controlled-Load Services fragment*. L'omissione di uno dei due frammenti relativi ai servizi è un'indicazione al ricevitore che in servizio omesso non è disponibile
  
- ❑ Il *Default General Parameters fragment* include i seguenti campi, che sono aggiornati ad ogni router RSVP lungo il percorso per far pervenire i valori al ricevente:
  - ❑ Minimum Path Latency (somma delle latenze individuali dei link).
  - ❑ Path bandwidth (minimo delle bande dei link individuali lungo il percorso).
  - ❑ Global break bit
  - ❑ Integrated Services hop count
  - ❑ PathMTU – unità di trasmissione massima



## Messaggio di RESV

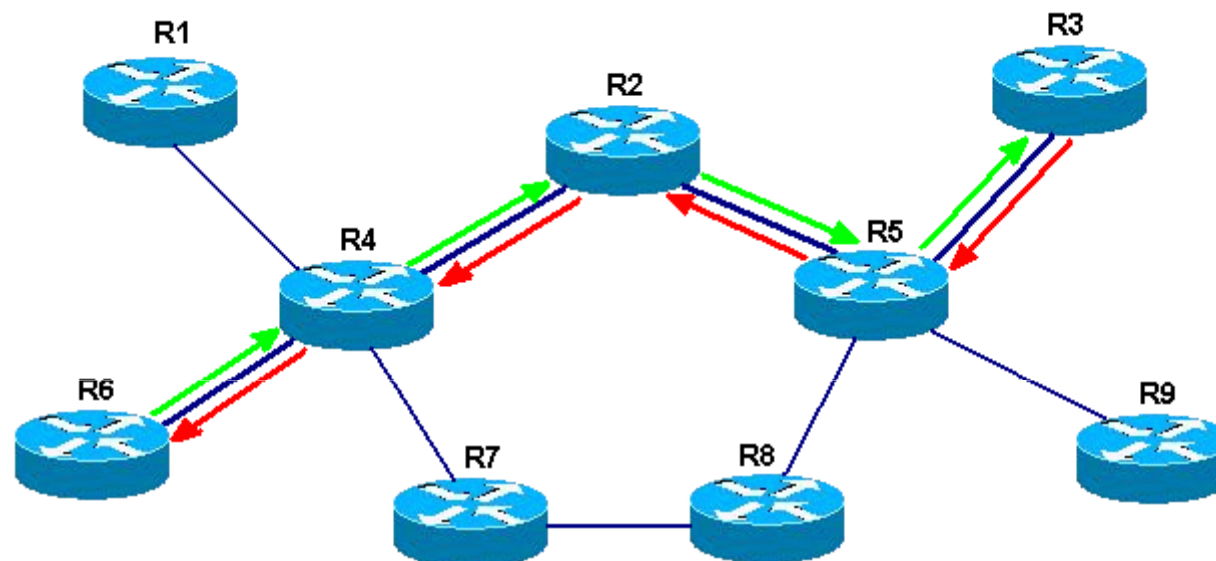
- ❑ Il messaggio Resv viene utilizzato dagli host riceventi per inoltrare le richieste di prenotazione delle risorse ai nodi che si trovano sul cammino verso la sorgente del flusso di dati oggetto della richiesta.
- ❑ Tale messaggio deve seguire “a ritroso” il cammino seguito dai dati dalla sorgente verso il ricevitore, per cui essi sono instradati hop-by-hop e recano come indirizzo di destinazione l’indirizzo del prossimo nodo presente sul cammino verso la sorgente.



- ❑ Ad ogni nodo la richiesta di prenotazione è passata ai moduli admission control e policy control
- ❑ Se uno dei due moduli non accetta la richiesta, nessuna risorsa viene allocata e viene inviato un messaggio di errore verso l'host ricevente che aveva originata la richiesta.
- ❑ Se invece la richiesta è accettata vengono intraprese due azioni:



- Attraverso le informazioni contenute nel flow descriptor vengono configurati i parametri del packet classifier e del packet scheduler ed allocate le risorse in loco
  - La richiesta viene ulteriormente propagata verso gli host sorgenti
- 
- L'insieme delle informazioni necessarie in un nodo a tenere allocate determinate risorse per una certa sessione identifica il cosiddetto Resv State.



Setup: Path (R6,R4,R5,R3)



Reply: Resv





## Meccanismo di refresh

- ❑ Le risorse che vengono allocate in un nodo per servire un certo flusso di dati in seguito ad un messaggio di prenotazione Resv non sono mantenute indefinitivamente, ma dopo un certo periodo di tempo (cleanup timeout) vengono liberate
- ❑ Affinché un certo host possa continuare a ricevere dati con la QoS precedentemente richiesta deve inviare allo scadere di un certo refresh timer dei messaggi di refresh in modo che le risorse per esso allocate rimangano tali.



## Altri tipi di messaggi

- ❑ Esistono altri tipi di messaggi che vengono utilizzati dal protocollo di prenotazione RSVP:
  - Messaggi di Teardown: sono utilizzati per liberare delle risorse che non servono più:
    - PathTear: si propaga verso tutti gli host riceventi un dato flusso a partire dal suo punto di origine provocando in ogni nodo la rimozione del Path State, ossia il rilascio delle risorse allocate
    - ResvTear: si propaga dal suo punto di inizio verso le sorgenti di un dato flusso e provoca in ogni nodo che attraversa la rimozione delle risorse allocate per il filter spec in esso contenute.
- ❑ Una richiesta di rilascio può essere effettuata o da un'applicazione su un host o da un router a seguito dello scadere del cleanup timeout.



## Messaggi di errore

- ❑ Vi sono due tipi di messaggi di errore:
  - PathErr: è semplicemente trasmesso verso l'host sorgente che ha causato l'errore e non provoca alcun cambiamento nei router che attraversa
  - ResvErr: sono più frequenti e generalmente causati dalla mancanza di risorse in rete necessarie per accogliere una data richiesta o dalla mancanza di requisiti per effettuare la prenotazione da parte del richiedente.



## Formato dei messaggi RSVP

- ❑ I messaggi RSVP sono incapsulati in normali pacchetti IP con valore di protocolID pari a 46. L'intestazione comune è costituita da 8 byte.

	0		4		8		16		24		31
RSVP Header	Version		Flags		Message Type			RSVP Checksum			
	Send TTL			Reserved			RSVP Length				
Obj Head	Object Length						Class-Num		C-Type		
RSVP Object 1	Object Data										
.....											
...											
Obj Head	Object Length						Class-Num		C-Type		
RSVP Object n	Object Data										
.....											



## PDU RSVP

- ❑ Vers: identifica la versione del protocollo in uso (attualmente la 1)
- ❑ Flag: non sono stati ancora definiti
- ❑ Message type: permette di specificare il tipo di messaggio RSVP con cui si ha a che fare (Path, Resv, PathTear, etc.)
- ❑ SendTTL: contiene il valore corrispondente al campo TTL (Time to Live) dell'intestazione IP
- ❑ RSVP Checksum: contiene un codice di controllo per la correzione di errore
- ❑ RSVP Length: esprime la lunghezza totale in byte del messaggio, includendo sia intestazione comune che gli oggetti variabili



## L'uso di RSVP con i Servizi Integrati

- ❑ La prima funzione nel modello dei servizi integrati è fornire servizi di QoS tramite una differenziazione del traffico diviso in classi. Il modello dei servizi integrati prevede due classi di servizio:
  - Controlled Load Services (CLS) e
  - Guaranteed Services (GS).



## Controlled Load Services

- ❑ La classe Controlled Load nasce per fornire un servizio in precedenza definito Better than Best Effort
- ❑ L'idea è quella di offrire ai pacchetti di questa classe lo stesso servizio che si otterrebbe con la modalità di invio best-effort, ma in condizione di rete scarica, ovvero assenza di congestione, garantendo un basso ritardo di accodamento ed una bassa probabilità di dropping in seguito ad overflow dei buffer di trasmissione.

## CLS

- ❑ In altre parole la sessione potrebbe considerare che una percentuale molto alta dei suoi pacchetti passerà con successo attraverso i router senza essere scartata e con ritardi di coda molto piccoli
- ❑ Cosa importante è che questa classe non specifica cosa costituisce una percentuale molto alta di pacchetti né quale qualità del servizio approssima quella di un elemento di rete scarico.





## Guaranteed Services

- ❑ L'unica classe che permette l'ottenimento di garanzie quantificabili è la classe Guaranteed
- ❑ Essa ha lo scopo di fornire un bound rigoroso e calcolabile a priori sul massimo ritardo di accodamento dei pacchetti.
- ❑ In generale durante una trasmissione il ritardo incontrato dal pacchetto è costituito da due parti:
  - Un ritardo fisso, dipendente dal percorso: il ritardo di propagazione
  - Un ritardo variabile, dovuto al tempo che il pacchetto trascorre nei buffer nei nodi all'interno della rete, detto ritardo di accodamento.

## GS

- ❑ Lo scopo della classe Guaranteed è quello di permettere la limitazione del ritardo variabile, l'unico ad essere in qualche modo controllabile dall'applicazione.
- ❑ L'informazione necessarie alle richieste di ciascuna classe di servizio, sono contenute all'interno della Flowspec che il ricevente invia verso l'host sorgente, nel messaggio Resv.
- ❑ La Flowspec, a sua volta, contiene due gruppi di informazione:
  - Tspec: La traffic Specification, ossia l'insieme dei parametri atti a caratterizzare i traffico
  - Rspec: La Request Specification, ossia l'insieme dei parametri con cui viene realizzata una richiesta esplicita di banda, per soddisfare le esigenze in termini di ritardo.

## Tspec

□ La Tspec è costituita da cinque parametri:

- Token rate:  $r$
- Token bucket size:  $b$
- Peak rate:  $p$
- Maximum datagram size:  $M$
- Minimum polices unit:  $m$

Il parametro  $M$  indica la dimensione massima dei datagrammi inviati.

Il parametro  $m$  serve ai fini della policy, per la quali i pacchetti inviati di dimensione inferiore a  $m$  verranno considerati pari ad  $m$ , per evitare problemi dovuti all'overhead sui messaggi.



## Rspec

- ❑ Altri due parametri contenuti nell'Rspec sono utilizzati dalla classe Guaranteed:
  - Rate Richiesta:  $R$
  - Slack:  $S$  con  $S \geq 0$ ;
- ❑ Il parametro  $R$  serve ad effettuare una richiesta in termini di banda
- ❑ Il termine di correzione  $S$  indica la differenza tra il ritardo desiderato e il ritardo ottenuto tramite l'assegnazione della banda  $R$  ed è utilizzato dagli elementi di rete per ridurre l'assegnazione di banda al flusso.



## CLS vs GS

- ❑ La classe Controlled Load fornisce una garanzia di tipo qualitativo: per essa è previsto l'invio della sola  $T_{spec}$ , che viene utilizzata per effettuare l'admission control, senza alcuna esplicita richiesta di banda.
- ❑ La classe Guaranteed è l'unica ad usare il parametro  $R_{spec}$ . Essa fornisce garanzie quantificabili: ha lo scopo di fornire un bound matematicamente calcolabile sul massimo ritardo di accodamento.



## Servizi Differenziati

- ❑ Nel 1997 viene formato un gruppo di lavoro dell'IETF sui servizi differenziati
- ❑ Esso aveva il compito di risolvere il problema legato al fatto che le architetture IntServ si rivelarono adatte solo per piccole reti IP e non per una rete così grande come Internet
- ❑ I Servizi Differenziati mirano a fornire garanzie per aggregati di flussi caratterizzati da alcuni comportamenti comuni.



- ❑ Al loro interno i flussi sono costituiti da microflussi che rappresentano aggregati di diverse connessioni aventi caratteristiche comportamentali analoghe (stesso BA o Behavior Aggregate)
- ❑ La gestione di un flusso prescinde dalla sua composizione, da come sono organizzati i microflussi all'interno del flusso
- ❑ Da un lato si opera trattando analogamente tutti i microflussi appartenenti ad un unico BA, ma da un altro bisogna fare in modo che ogni flusso aggregato rispetti i livelli di servizio concordati con l'utente (SLA o Services Level Agreement).



- ❑ Nell'architettura prevista dai DiffServ si distinguono due regioni:
  - Una ai margini della rete (edge)
  - Una al centro della rete (core)
- ❑ Il traffico proveniente dagli utenti arriva ai margini della rete dove viene trattato e convogliato verso il centro
- ❑ L'idea dei DiffServ è l'aggregazione del traffico ai bordi della rete





- ❑ i pacchetti che arrivano dagli host, indipendentemente dalla connessione a cui appartengono ma tenuto conto dei loro requisiti in termini di risorse di rete (banda e ritardo), vengono

- raggruppati e
- "marcati"

con un identificatore comune, che permetterà in seguito ai router nella parte centrale di applicare a tali pacchetti le più appropriate politiche di gestione.



- ❑ In pratica, anziché far gestire alla parte interna della rete le singole connessioni (obbligando i router a riconoscere ogni singolo "micro flusso" di pacchetti), si gestiscono aggregati di connessioni (o "macro flussi") aventi caratteristiche simili.
- ❑ A grandi linee, si può dire che il principio con il quale il traffico viene gestito dalla rete è lo stesso nel caso *IntServ* e nel caso *DiffServ*, con la differenza che nel primo caso il controllo è fatto sulla singola connessione, mentre nel secondo caso è fatto su più connessioni considerate insieme.



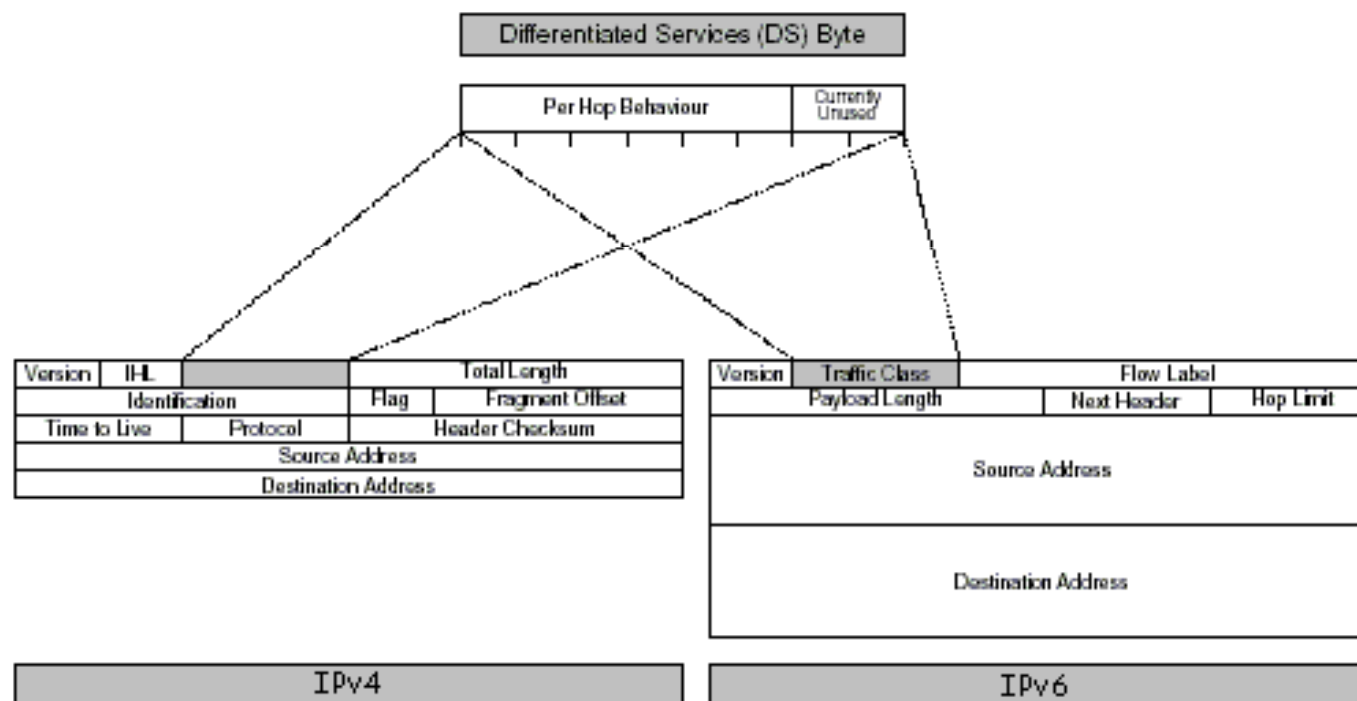
- ❑ I pacchetti vengono classificati e marcati in modo tale da poter essere trattati in modo diverso. Il trattamento differenziato del traffico è assolto da modalità di inoltro chiamate *Per Hop Behavior (PHB)* sui nodi lungo il percorso
- ❑ I domini di confine sono i più delicati per lo svolgimento delle politiche di qualità e differenziazione dei servizi, in quanto è lì che bisogna classificare i pacchetti, cioè creare la corrispondenza tra il campo *DS (DSFIELD)* e i pacchetti che hanno contrattato quel tipo di servizio e svolgere altre operazioni altrettanto importanti come *shaping, policing, dropping, remarking*



- ❑ La marcatura dei pacchetti e la distinzione tra flussi aggregati diversi è effettuata, rispettivamente, scrivendo ed esaminando un codice nel campo TOS (*Type Of Service*), contenuto nell'*header* di ogni pacchetto IP.
- ❑ Per la marcatura dei pacchetti, il così chiamato *DS-byte* (*codice DS o campo DS* per i servizi differenziati) nell'*header* di ogni pacchetto IP è mappato nell'ottetto Type-of-Service dell'Ipv4 o nell'ottetto Traffic Class dell'Ipv6



## Differentiated Services



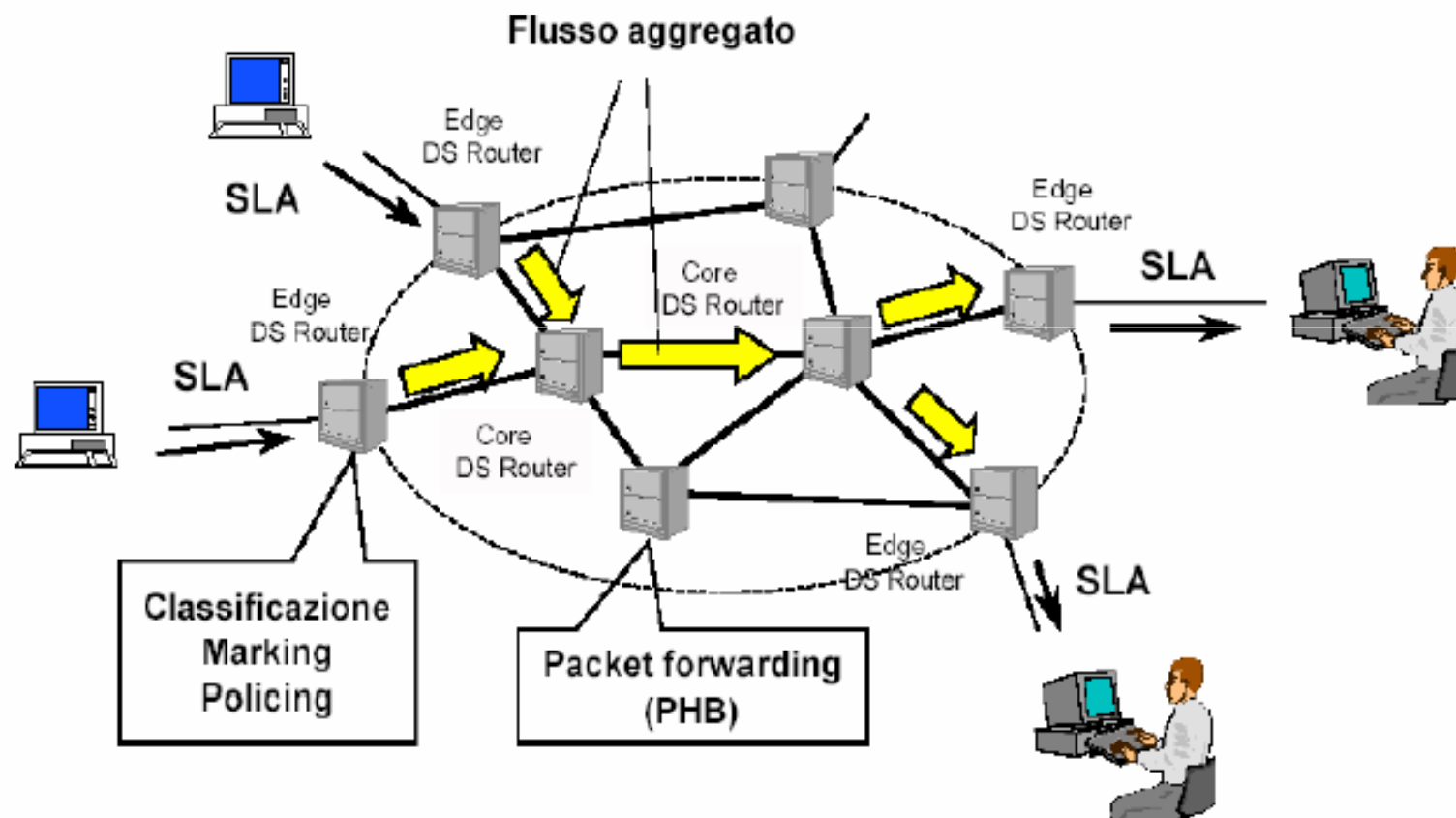


- Sei bit di questo byte, chiamati *Differentiated Services Code Point (DSCP)* sono usati per definire il *Per Hop Behavior (PHB)* che un pacchetto sperimenta in ogni router. I rimanenti due bit corrispondono al campo *currently unused (CU)* che è riservato per scopi non ancora specificati che possono venir definiti in seguito.
- Il significato dei bits individuati nel campo *DSCP* non è ancora standardizzato ed è parte di una discussione nel gruppo di lavoro dei *Servizi Differenziati (DiffServ)* dell'IETF.





- ❑ Secondo il modello *DiffServ*, la rete nel suo complesso è formata da più sotto-reti, amministrativamente indipendenti, chiamate *domini*.
- ❑ Ciascun dominio fornisce un servizio ai propri clienti (gli utenti finali oppure altri domini) e richiede a sua volta ai domini adiacenti un servizio, che consiste nel mettere a disposizione una certa quantità di risorse per la gestione del proprio traffico.
- ❑ Gli accordi tra i domini adiacenti consentono agli aggregati di attraversare la rete usufruendo della qualità desiderata.







- Un ***Service Level Agreement (SLA)*** specifica il contratto di traffico tra un utente ed il *service provider* e indica il livello di *QoS* del traffico generato dall'utente.
  
- Un *SLA* specifica i parametri di servizio, quali:
  - Parametri prestazionali (*throughput, loss probability, latency*);
  - Profilo di traffico (*Service Level Specification: SLS*) che deve essere rispettato dal flusso specificato dai parametri del *token bucket* ;
  - Trattamento del traffico in eccesso rispetto al profilo di traffico concordato;
  - *Marking service* (regole di marcatura dei pacchetti appartenenti ad un flusso);
  - *Shaping service* (regole di classificazione dei pacchetti appartenenti ad un flusso);
  - Insieme delle regole di *forwarding* che costituiscono il *PHB* che deve essere applicato ai pacchetti appartenenti al flusso nel transito nel dominio.



- Un *Per Hop Behavior (PHB)* stabilisce le regole con cui devono essere trattati i *datagrammi* nei router di un dominio *DS*. Un livello di *QoS* offerto da un dominio *DS* è dato dalla composizione dei *PHB* applicati nei router attraversati dai *datagrammi*.
  
- Oggi sono definiti due *PHB*:
  - *Expedited Forwarding (EF)*
  - *Assured Forwarding (AF)*.



## Expedited Forwarding

- ❑ Il *PHB Expedited Forwarding (EF)* è usato per costruire un servizio caratterizzato da
  - bassa probabilità di perdita
  - basso ritardo
  - bassa variabilità del ritardo (jitter)
  - banda garantita.
  
- ❑ Perdita, latenza e jitter sono tutte caratteristiche dovute alle code che il traffico sperimenta mentre attraversa la rete



- ❑ Fornire bassa perdita, latenza, jitter per qualche aggregato di traffico, significa assicurare che l'aggregato non veda mai delle code, o al più molto piccole. Le code nascono quando la rate del traffico in arrivo eccede la rate di quello in partenza a qualche nodo.
- ❑ Così un servizio che assicura che non ci saranno code per qualche aggregato è equivalente a limitare le rate così che, a ogni nodo di transito la rate di arrivo massima dell'aggregato sia minore della rate di partenza minima.



- ❑ La creazione di un tale servizio presenta due parti:
  - Configurare i nodi così che l'aggregato abbia una rate di partenza minima ben definita. Dove per "ben definita" si intende principalmente, indipendente dalla intensità di altro traffico a quel nodo.
  - Condizionare l'aggregato (attraverso policing e shaping) così che la sua rate di arrivo ad ogni nodo sia sempre minore della rate di partenza minima configurata del nodo.
  
- ❑ Il codepoint 101110 è il codepoint raccomandato per il *PHB EF*.



## Assured Forwarding

- ❑ L' *Assured Forwarding (AF) PHB* ha lo scopo di dare la possibilità ad un operatore di rete di definire un insieme di livelli di trasferimento (*Forwarding Assurances*) per differenziare il trattamento dei flussi in un dominio IP.
- ❑ Sono definiti quattro classi  $AF(AF_{xy}, x=1,2,3,4)$ ; per ognuna di esse è allocato in ogni router un diverso insieme di risorse (banda e buffer).
- ❑ Per ciascuna delle quattro classi sono definiti tre livelli di priorità di scarto dei *datagrammi* (*drop precedence*) ( $AF_{xy}, y=1,2,3$ ) in modo che in caso di congestione i *datagrammi* sono scartati nell'ordine stabilito dal livello di *drop precedence*.

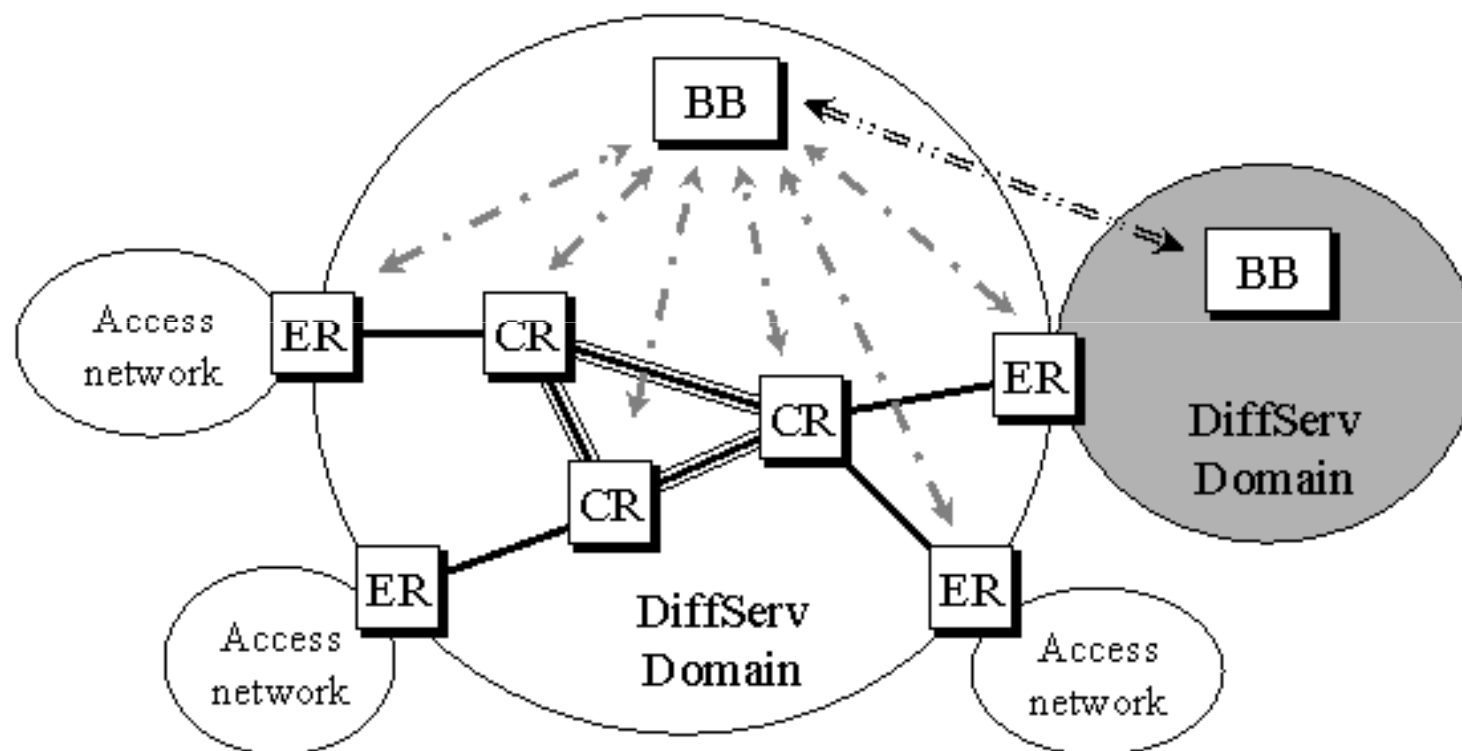


- ❑ I valori del campo DSCP sono:

Drop Precedence	Classe AF			
Low	010000	011000	100000	101000
Medium	010010	011010	100010	101010
High	010100	011100	100100	101100



## Modello di riferimento di una rete IP DiffServ







## Integrare o Differenziare

- ❑ Il nome adottato da IETF trae in inganno chi tenti di fare un confronto intuitivo tra "integrato" e "differenziato", dato che questi concetti potrebbero anche sembrare opposti:
  - IntServ: il nome nasce dall'idea di realizzare dei servizi dotati di *QoS integrati* con la rete, cioè in modo tale che la rete sia attivamente coinvolta nel fornire tali servizi
  - DiffServ: Il "differenziato" non va inteso in contrapposizione all'"integrato" degli *IntServ*, ma come attributo che evidenzia la presenza di più servizi *differenziati* (nel senso di "diversificati") forniti dalla rete.
- ❑ Il concetto, in conclusione, è lo stesso nei due casi e la terminologia è stata cambiata per evitare ambiguità.



## Vantaggi e Svantaggi

- ❑ La scalabilità, punto debole degli *IntServ*, è certamente garantita con l'approccio *DiffServ*, grazie al limitato numero di aggregati che possono attraversare la *core Network*.
- ❑ I *core* router, infatti, possono operare a velocità molto maggiori dovendo gestire solo pochi flussi di traffico diversi.
- ❑ Maggiore carico è posto sugli *edge* router, ai quali è lasciato il compito di classificare i pacchetti e aggregarli marcandoli con il codice opportuno nel campo TOS.



- ❑ Tuttavia, il numero di connessioni che arrivano ad un *edge* router è certamente limitato e molto inferiore a quello di connessioni gestite da un *core* router.
- ❑ L'architettura appare pertanto ben bilanciata.
- ❑ Il punto debole dei *DiffServ* è l'affidabilità con la quale le garanzie di *QoS* possono essere garantite.
- ❑ nel caso dei *DiffServ* mancano la segnalazione e la prenotazione dinamica delle risorse, che negli *IntServ* permettono una gestione molto più accurata della rete stessa.



- ❑ Il modello *IntServ* tende ad avvicinare il caotico mondo a commutazione di pacchetto, caratteristico di Internet, al più ordinato ed efficiente mondo delle comunicazioni a commutazione di circuito
- ❑ La complessità e il costo del secondo tipo di rete, ovviamente, è molto maggiore rispetto al primo, che si è rivelato senz'altro più adeguato fino a questo momento.

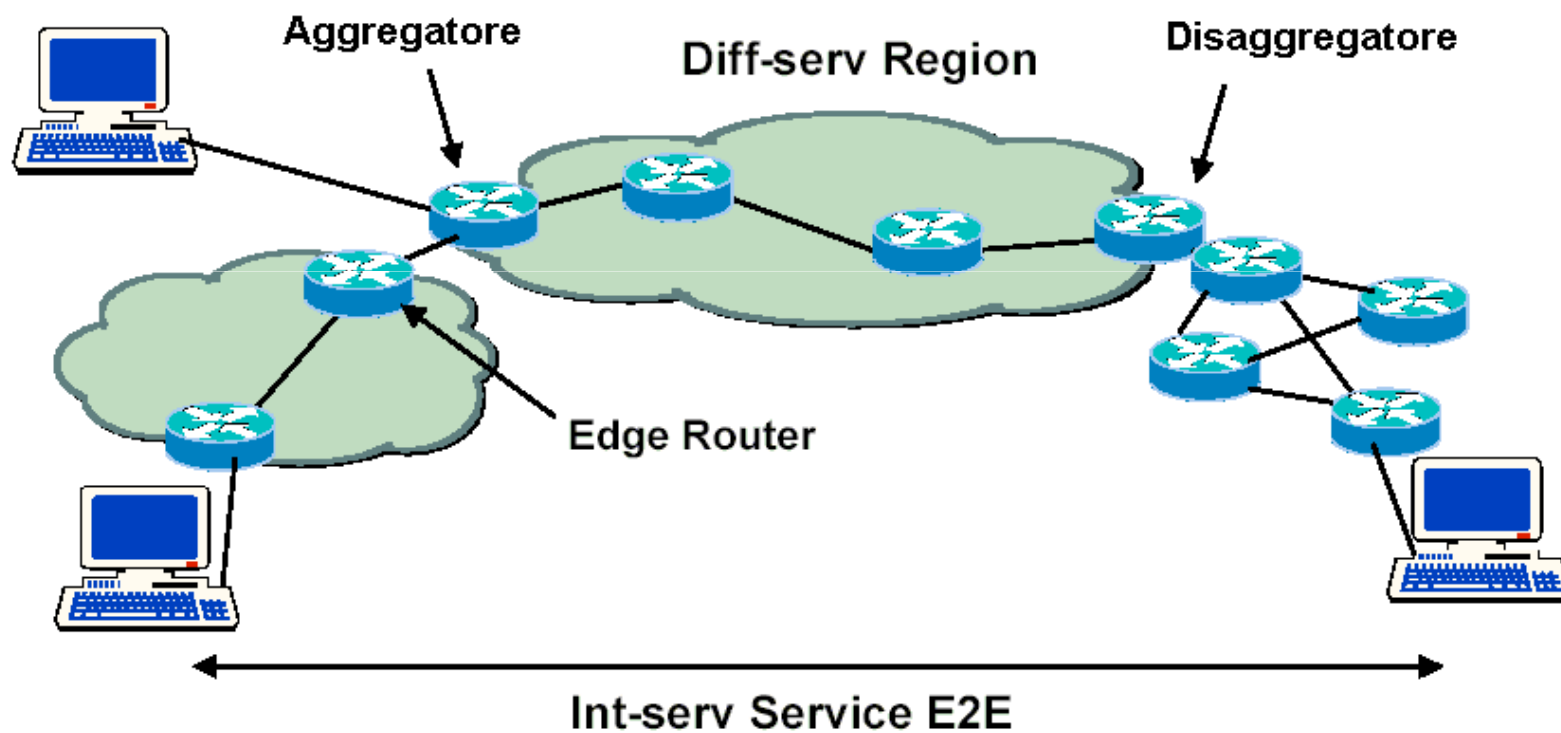


- ❑ Il punto debole del modello *IntServ*, in ultima analisi, è stato proprio cercare di ottenere i benefici della commutazione di circuito (dove il "circuito" è la "connessione" di *IntServ*) con apparecchiature nate per la commutazione di pacchetto, non abbastanza potenti per gestire su larga scala la rete così strutturata.
- ❑ In questo contesto, i *DiffServ* si posizionano a metà strada tra i due approcci, in quanto tentano ancora di gestire dei "circuiti commutati", in questo caso rappresentati dagli aggregati di pacchetti, ma in modo più rigido (con allocazione delle risorse statica o quasi) e meno complesso.



- ❑ Per reti di dimensione ridotta la maggiore flessibilità degli *IntServ* si può dimostrare più appropriata e non limitata da problemi di scalabilità.
- ❑ Un possibile scenario di applicazione dei due modelli, quindi, potrebbe prevedere la presenza di "isole" *IntServ* ai bordi della rete, all'interno delle quali la qualità sarebbe gestita in modo più efficiente; la dorsale, invece, dovrebbe necessariamente essere costituita da domini *DiffServ* in grado di consentire il transito di numerose connessioni raggruppate in pochi aggregati.

# IntServ-DiffServ





FINE