Quality of Service on IP Networks

Arquitectura e Gestão de Redes

Current Internet Services

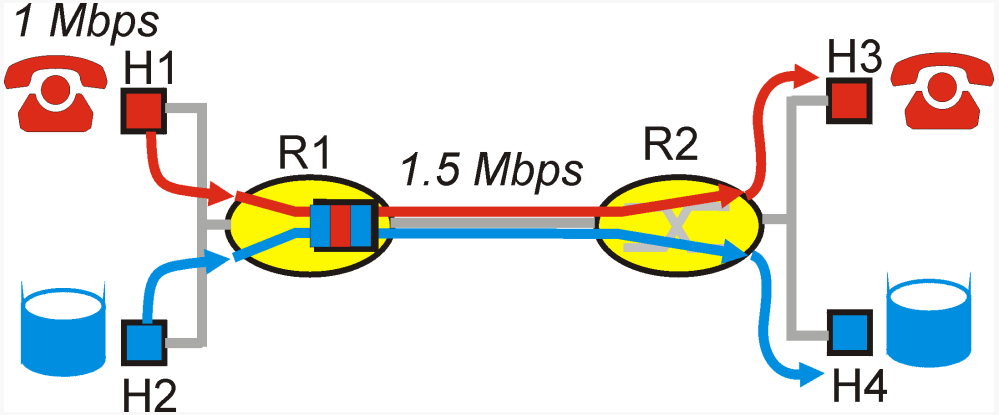
- **Internet supports a lot of services, besides the basic packet exchange service**
  - **Interactive games**
  - **Real time audio/video**
  - **High definition moving images**
  - **Complex databases**
  - **Cloud**
  - **Peer-to-peer**
  - **Mobile services**

Service Requirements

- **Packet loss**
  - **Some applications (e.g., real-time audio/video) support losses**
    - **Voice is more tolerant to losses than video**
  - **Others (file transfer and telnet, for example) require 100% of transmission success**
    - **Use TCP**
- **Bandwidth**
  - **Some applications (ex., multimedia) require a minimum bandwidth to work**
    - **Full queues**
    - **High delays and some losses**
  - **Others ("elastic", like e-mail or file transfer) use the bandwidth they can get**
- **Delays**
  - **Some applications (ex., VoIP, multi-user games) require low delays**
  - **Others (non real-time) do not impose limits to the end-to-end delay**

Principles for QoS Guarantee

- **Example: 1Mbps VoIP and FTP share a 1.5 Mbps connection**
  - **FTP burts can congestion the router, causing audio losses**
  - **Audio traffic should have priority**

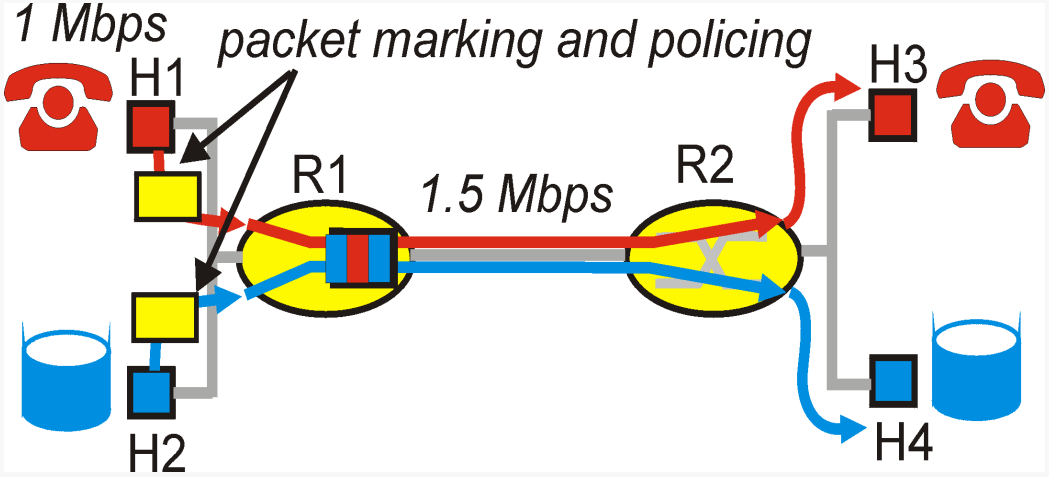


Principle 1

The router can only distinguih between different traffic classes if packets are marked; new packet scheduling mechanisms are needed

Principles for QoS Guarantee

- **What happens if applications misbehave (audio sends more than the declared rate)?**
  - **policing: forces sources to adhere to pre-established BW**
- **Marking and policing at the network entry**

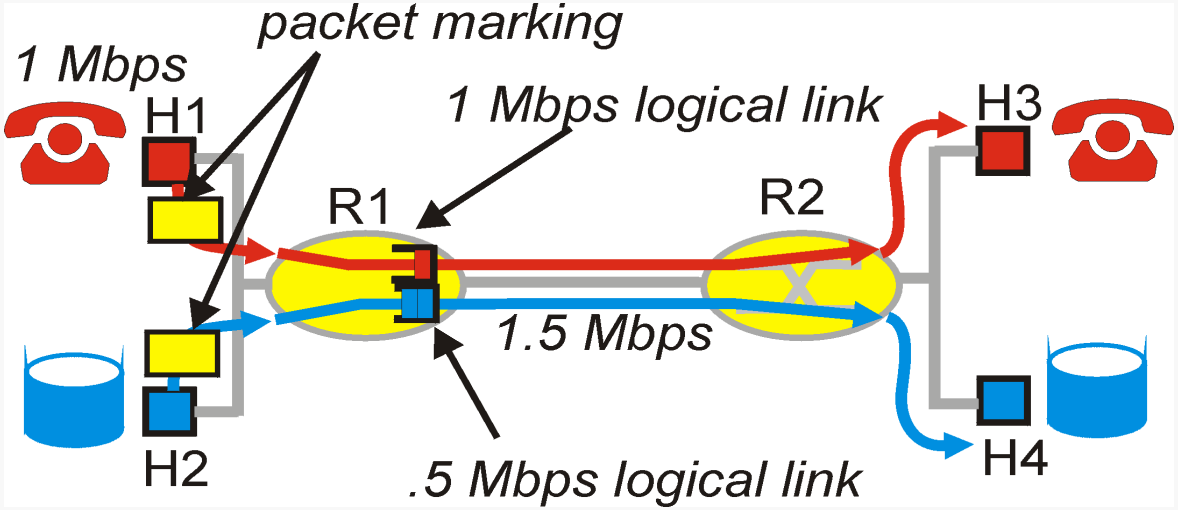


Principle 2

Protecting a traffic class from the others.

Principles for QoS Guarantee

- **Assigning a fixed BW to the flow: ineficient use of the link if the flow does not use the whole assigned BW**

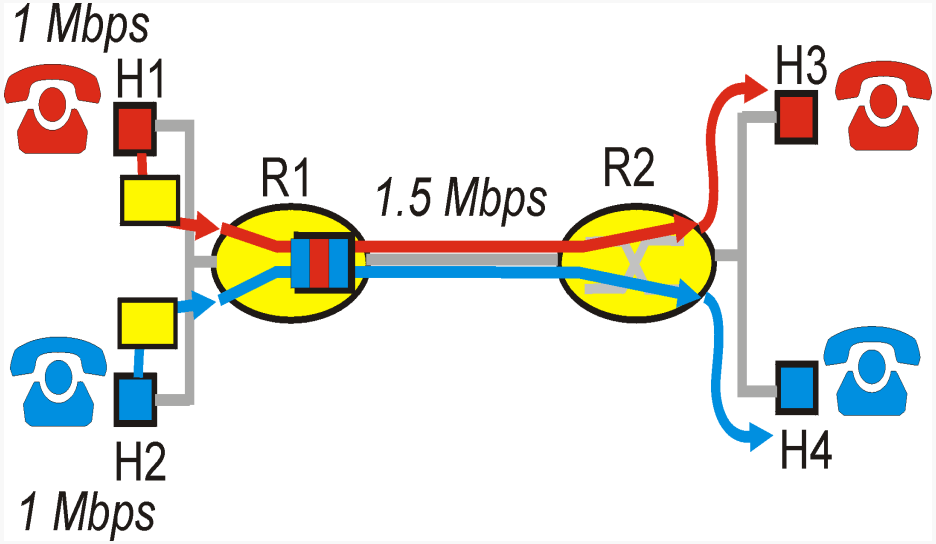


Principle 3

Resources should be used in the most efficient way

Principles for QoS Guarantee

- **It is not possible to support requests that go beyond the link capacity**



Principle 4

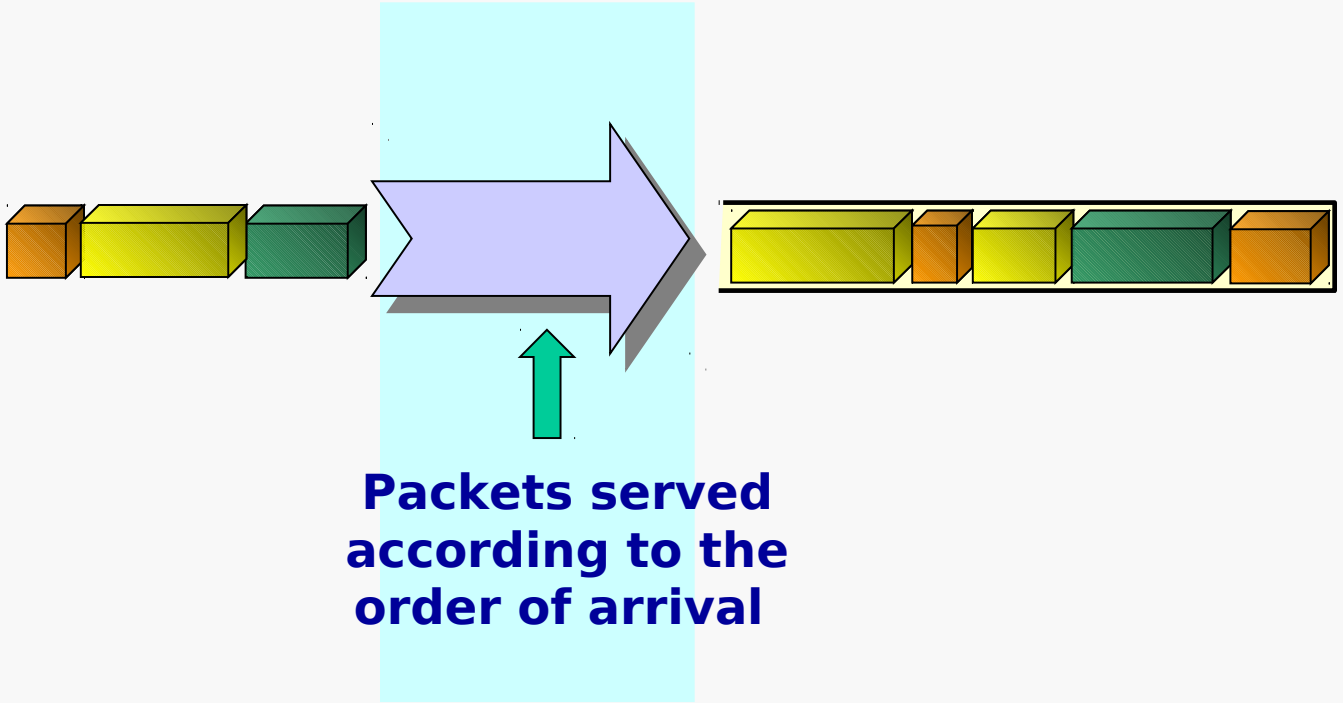
Call admission: a flow declares its needs, the network can block a call (ocuppied signal) if it cannot support it

IP Network with QoS Support

Scheduling algorithms

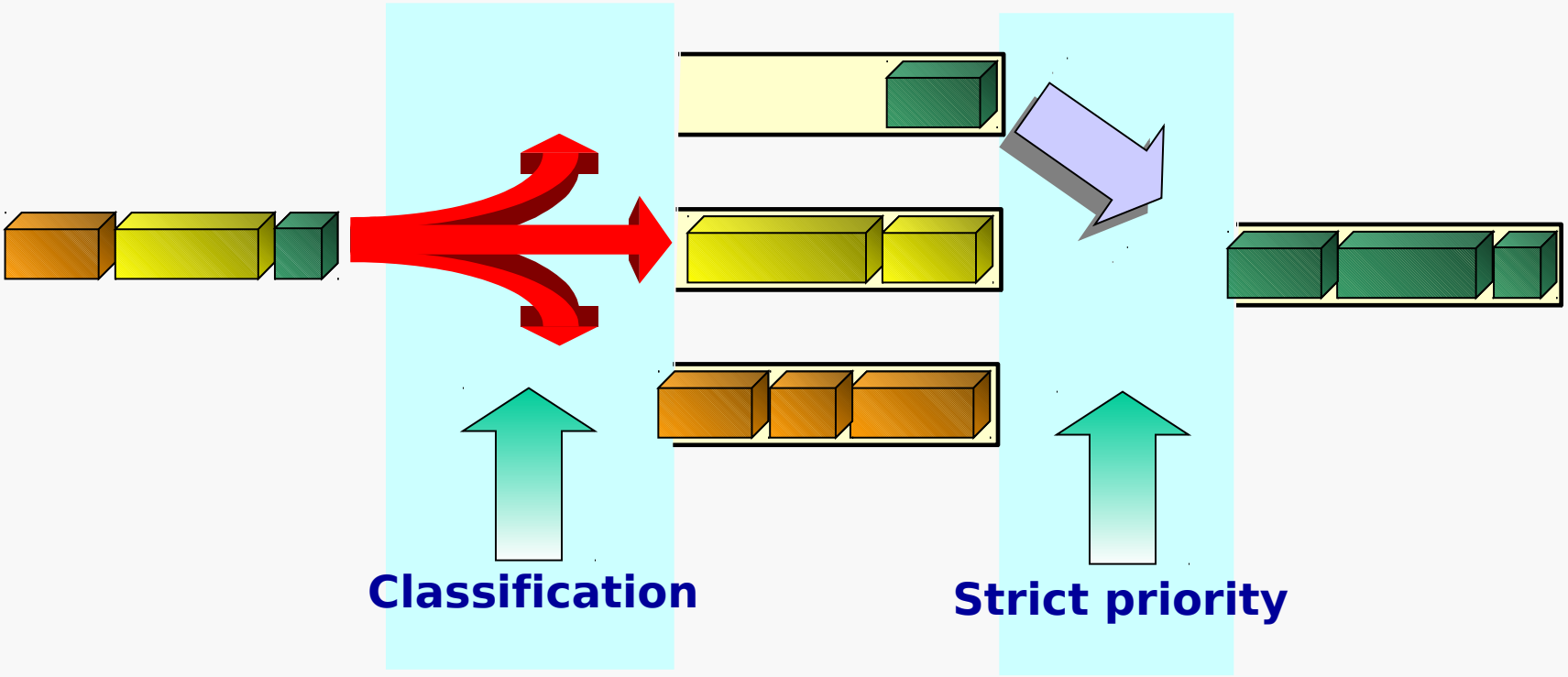
- **Decide the service order of the packets, when packets belong to different flows**
- ***Work conserving* scheduling algorithms guarantee that the server is always occupied when there are packets to transmit**
  - **FIFO**
  - **Strict priority**
  - **Fair Queuing,**
  - **Weighted Fair Queuing.**

First In First Out (FIFO)



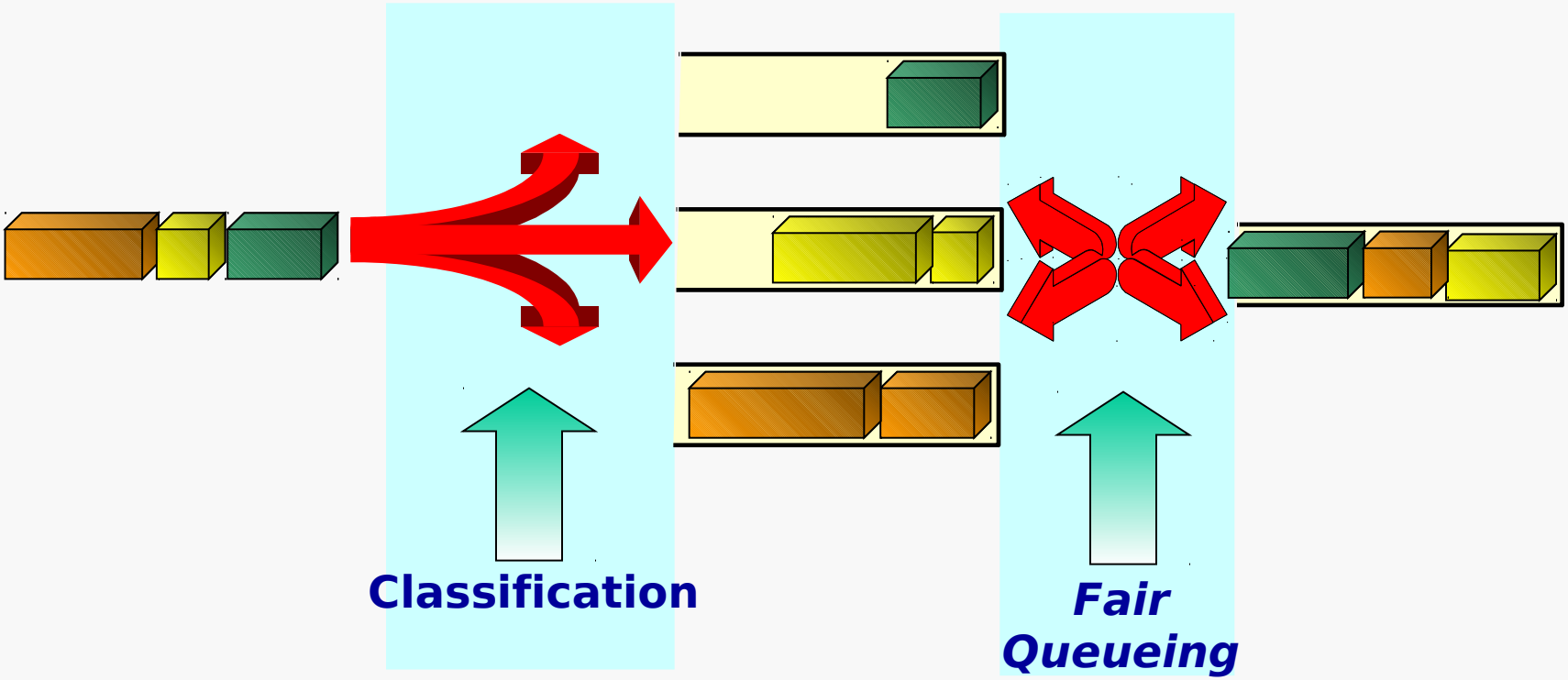
- Does not invlove ordering
- Does not allow QoS differentiation
- Flows with more traffic receive more service
- In finite length queues, flows with smaller size packets receive more service

Priority Queueing



- Involve traffic classification, according to the priority
- Higher priority traffic is always served before the lower priority one
- Allows QoS differentiation
- High priority flows can avoid low priority ones from being served

Fair Queueing (FQ)



- Involve traffic classification in different queues
- Transmission bandwidth is equally distributed among the non-empty queues
- Allows QoS deployment

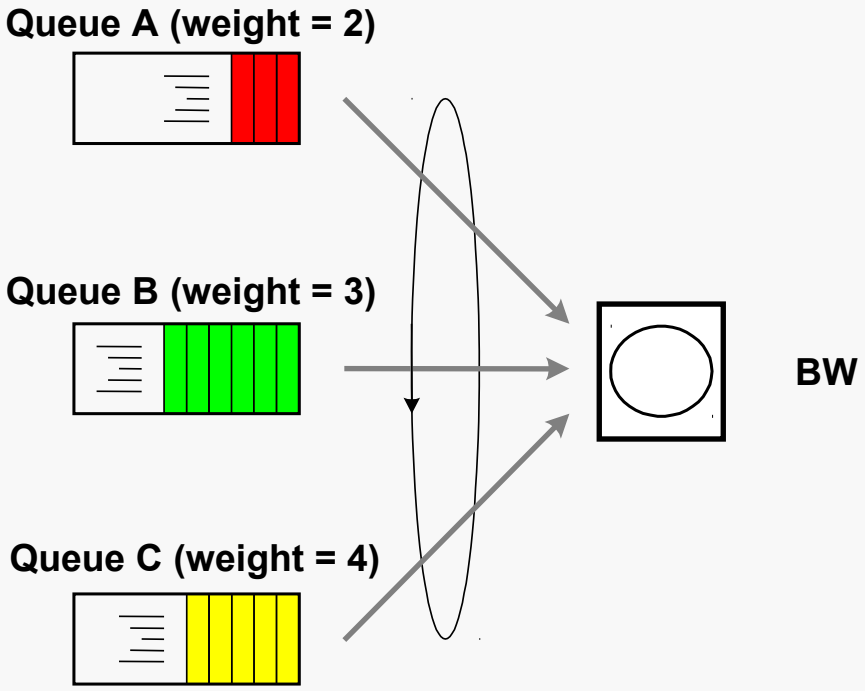
Weighted Fair Queuing (WFQ)

Each queue receives a percentage of the connection bandwidth that is <u>at least</u> equal to its weight divided be the sum of the weights of all queues

$$R_A = \frac{2}{2+3+4} BW$$

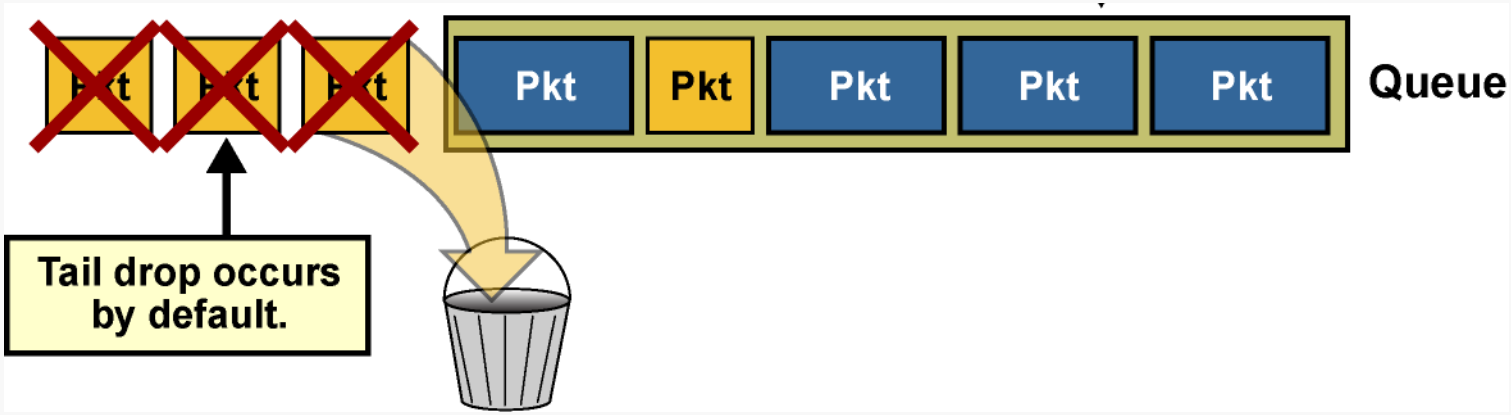
$$R_B = \frac{3}{2+3+4} BW$$

$$R_C = \frac{4}{2+3+4} BW$$



Packet discarding policies

Tail Drop



- **Router interfaces experience congestion when the output queue is full**
  - **Additional incoming packets are dropped**
  - **Dropped packets may cause significant application performance degradation**
  - **Tail drop has significant drawbacks**

Random Early Detection (RED)

Tail drop can be avoided if congestion is prevented

RED is a mechanism that randomly drops packets before a queue is full

RED increases drop rate as the average queue size increases

RED result:

- **TCP sessions slow to the approximate rate of output-link bandwidth**
- **Average queue size is small (much less than the maximum queue size)**
- **TCP sessions are desynchronized by random drops**

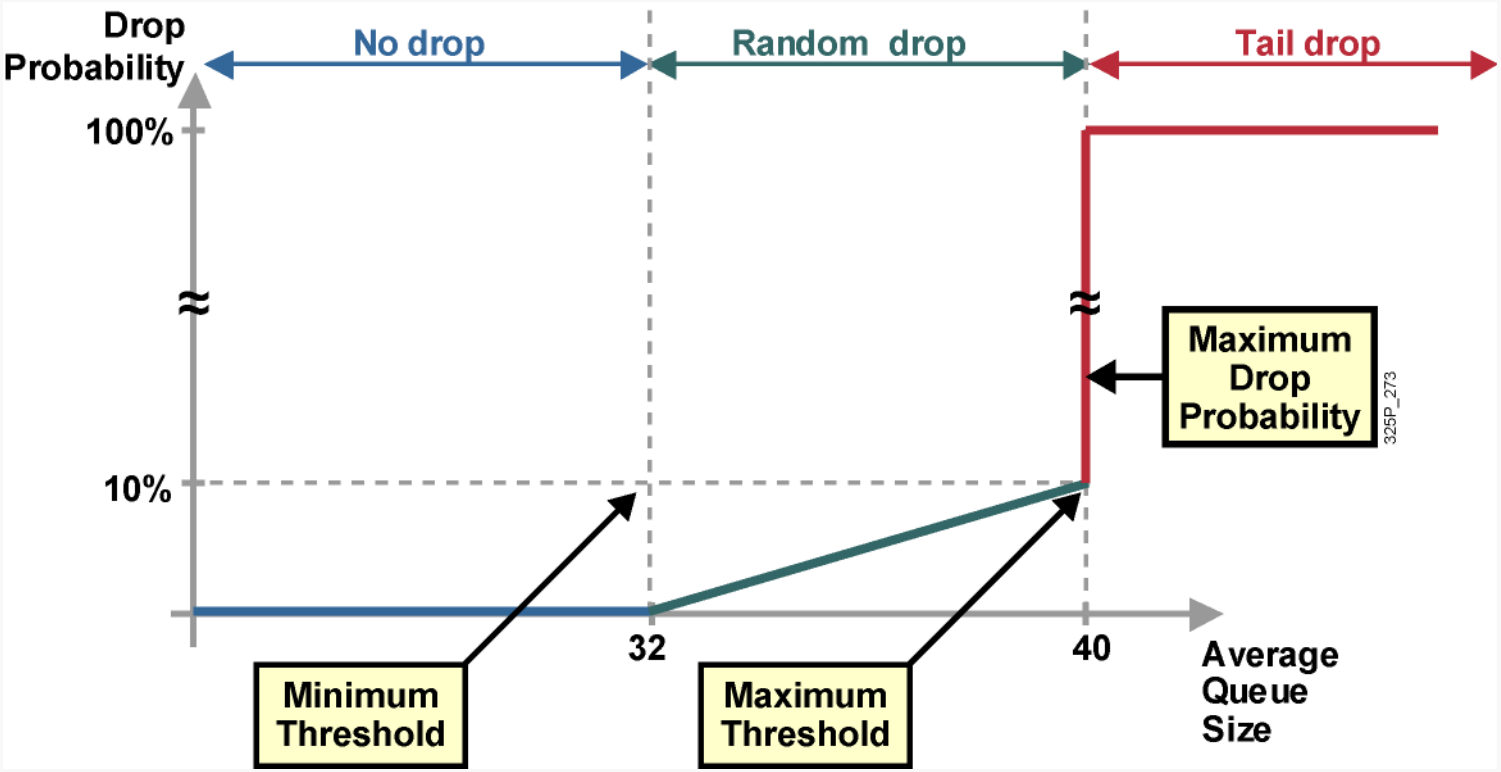
RED Modes

RED has three modes:

- **No drop:** When the average queue size is between 0 and the minimum threshold
- **Random drop:** When the average queue size is between the minimum and the maximum threshold
- **Full drop (tail drop):** When the average queue size is above the maximum threshold

Random drop should prevent congestion (prevent tail drops).

RED Drop Profiles



Weighted Random Early Detection (WRED)

WRED can use multiple RED profiles.

Each profile is identified by:

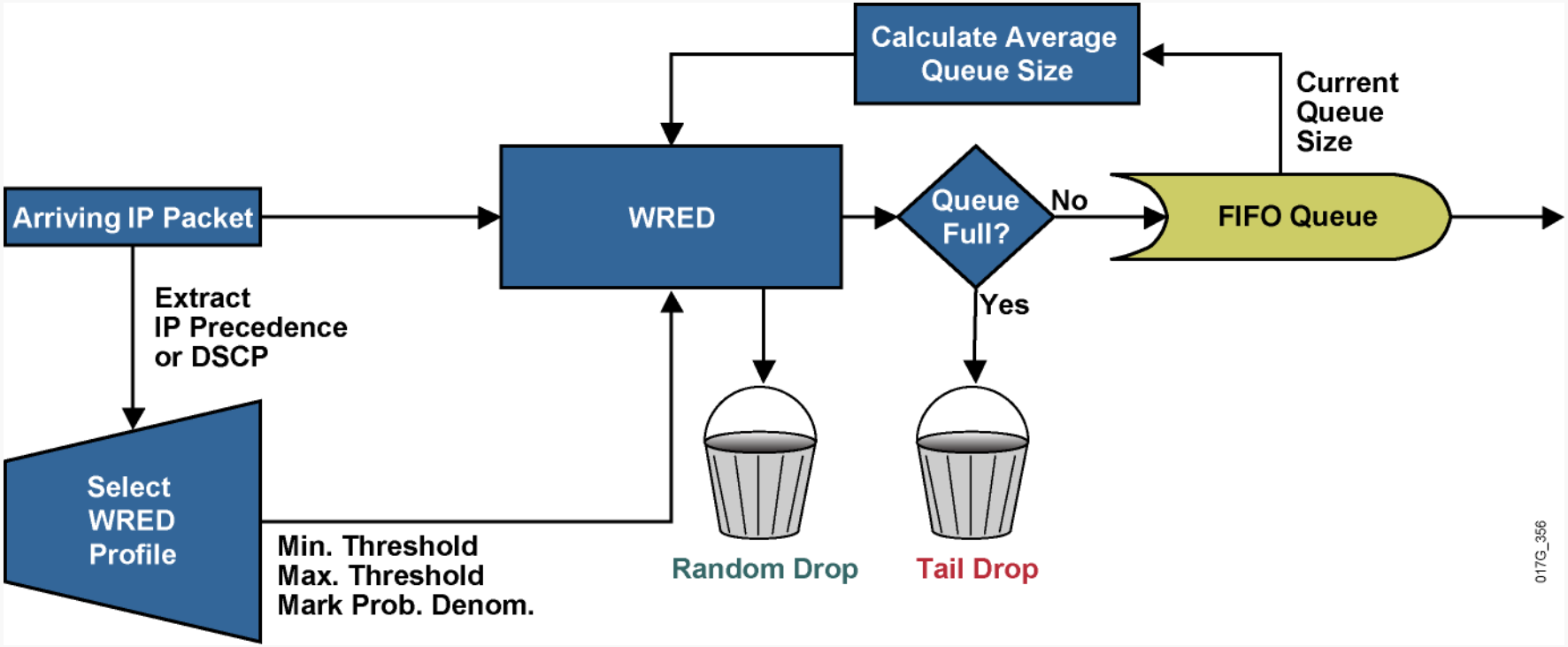
- **Minimum threshold**
- **Maximum threshold**
- **Mark probability denominator**

WRED profile selection is based on:

- **IP precedence (8 profiles)**
- **DSCP (64 profiles)**

WRED drops less important packets more aggressively than more important packets.

WRED Building Blocks



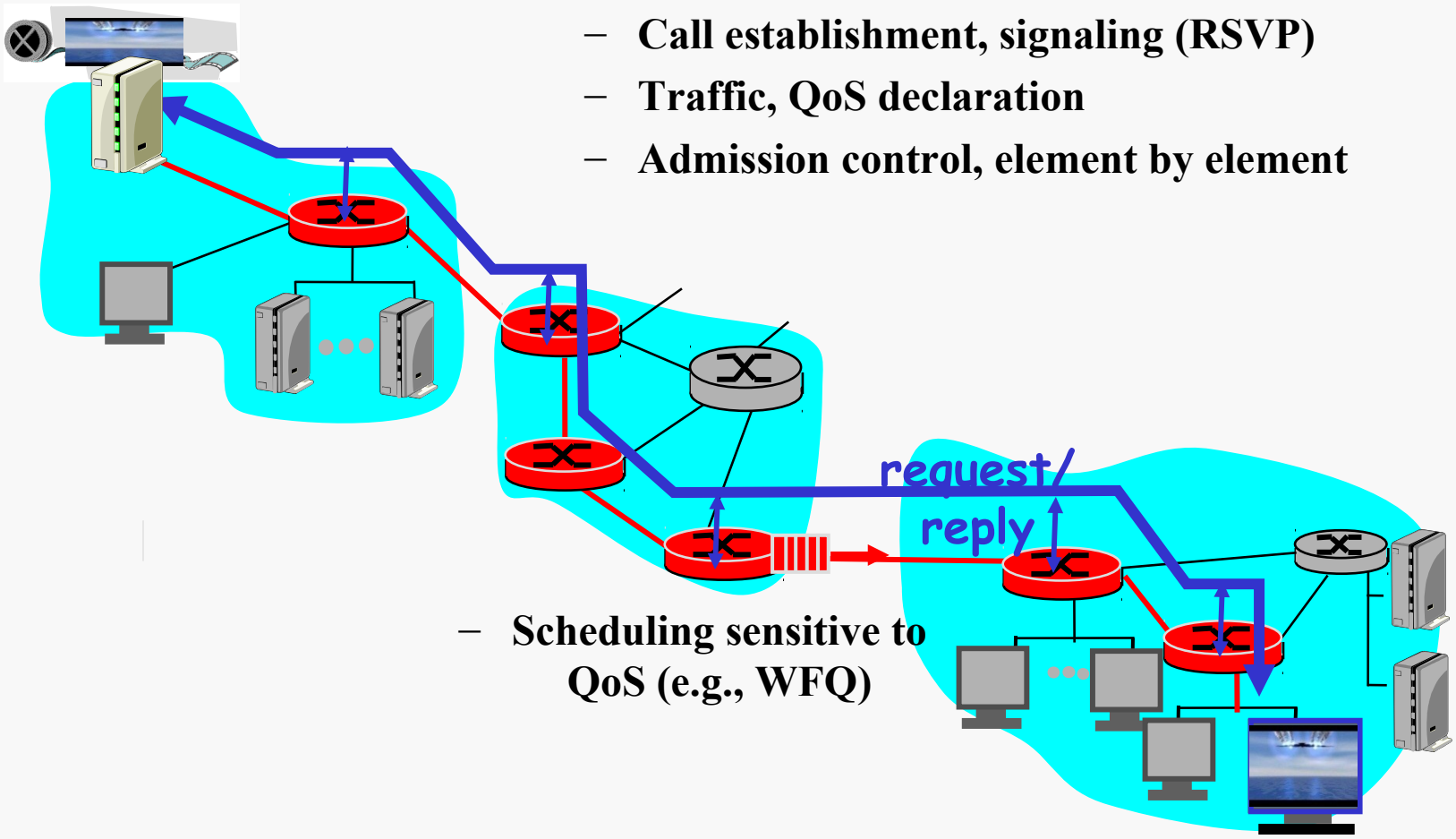
Integrated Services Architecture

Integrated Services (IntServ)

- **For flows requiring QoS, it is necessary to reserve resources on the path from source(s) to destination(s)**
- **Reservations are made on a per flow basis**
- **The RSVP signaling protocol is used, allowing:**
  - **The source to describe the characteristics of the IP packet flow**
  - **The destinations to describe the reservation they want**
  - **The routers to know how to process the packet flow in order to cope with the required reservation(s)**
- **The network implements a *call admission control mechanism***
  - **Flows that do not obtain a reservation are treated as "*best effort*" traffic**

Scenario that needs QoS support

- **Resources reservation**
  - Call establishment, signaling (RSVP)
  - Traffic, QoS declaration
  - Admission control, element by element



- Scheduling sensitive to QoS (e.g., WFQ)

Call admission control

The starting session must:

- **Declare its QoS requirements**
  - **R-spec: defines the required QoS**
- **Characterize the traffic that will be sent to the network**
  - **T-spec: defines the traffic characteristics**

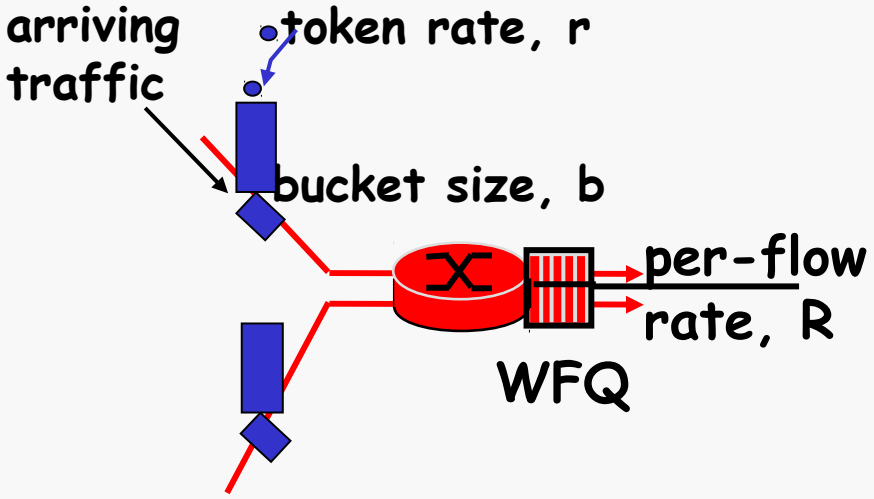
A signaling protocol is necessary to transport R-spec and T-spec to the routers (where reservation is made)

- **RSVP [RFC 2205]**

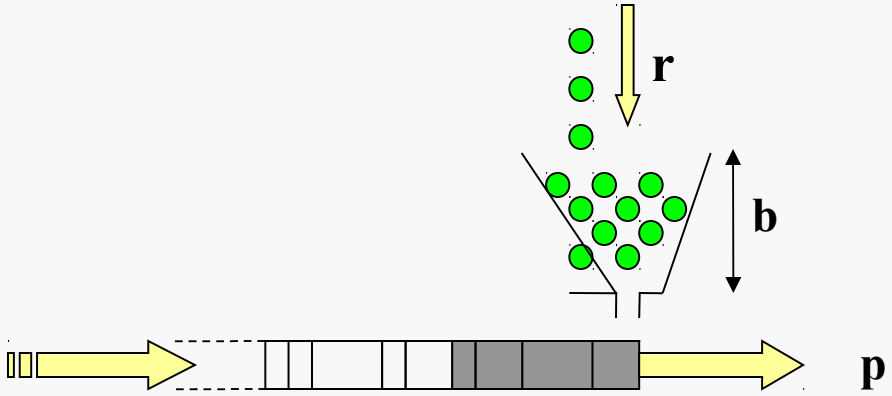
IntServ service classes

- *Controlled Load* (RFC 2211)
  - Provides a service similar to the "*best effort*" service on a uncongestioned network
  - End stations should feel that a high percentage of the packets is delivered with very small *queueing delays* at the routers
- *Guaranteed Service* (RFC 2212)
  - Guaranteed a maximum delay for all IP packets

- In both cases, the source has to condition its packets to a "*token bucket*" model



Token Bucket model



r **= token filling rate (bytes/s)**

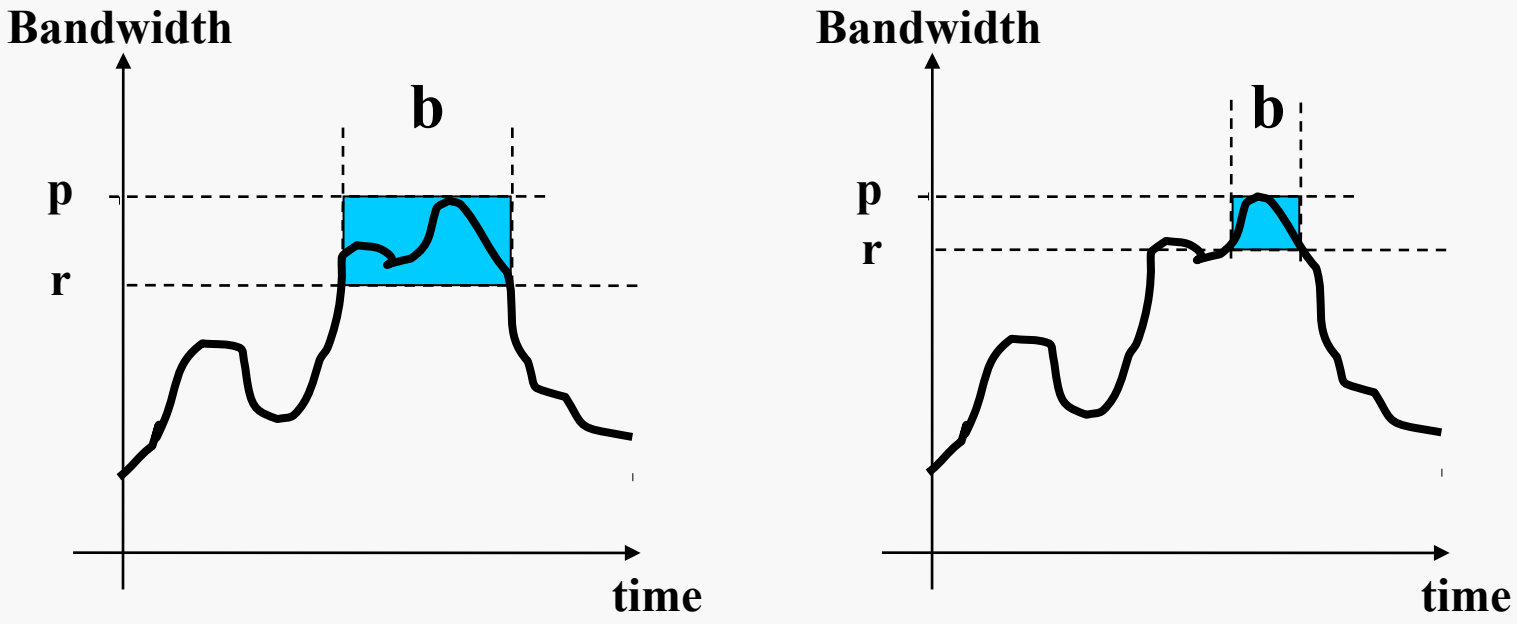
b **= bucket size (bytes)**

p **= maximum sending rate (bytes/s)**

M **= maximum packet size (bytes)**

m **= minimum packet size (bytes) – any packet with a size smaller than *m* will be tra«eated as having size *m***

Token Bucket model



- **In an interval *Δt*, the number of admitted packets (*pΔt)* should be less than or equal to *(rΔt + b)***

Guaranteed Services

It is possible to guarantee a maximum delay for the flow:

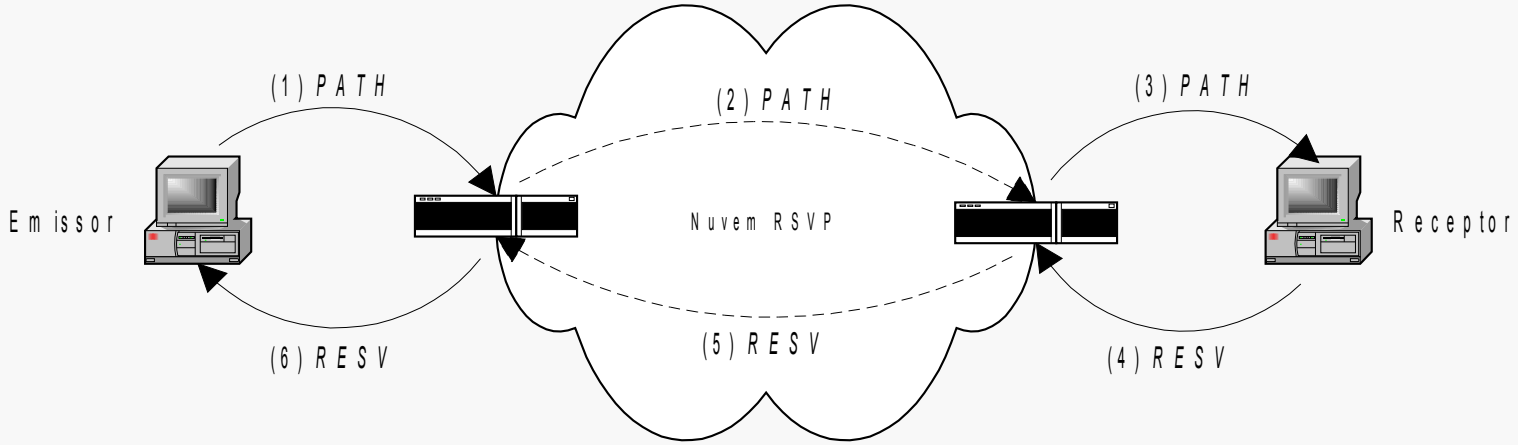
$$\text{Delay}_{max} = \begin{cases} \left[\frac{b-M}{R} \times \frac{p-R}{p-r}\right] + \frac{M+C_{tot}}{R} + D_{tot} & \text{if } p > R \geq r \\ \frac{M+C_{tot}}{R} + D_{tot} & \text{if } R \geq p \geq r \end{cases}$$

C – delay of the packets due to the flow parameters
D – delay introduced by the network nodes

C_{tot} and D_{tot} are updated by routers on RSVP PATH messages

RSVP (*Resource Reservation Protocol*)

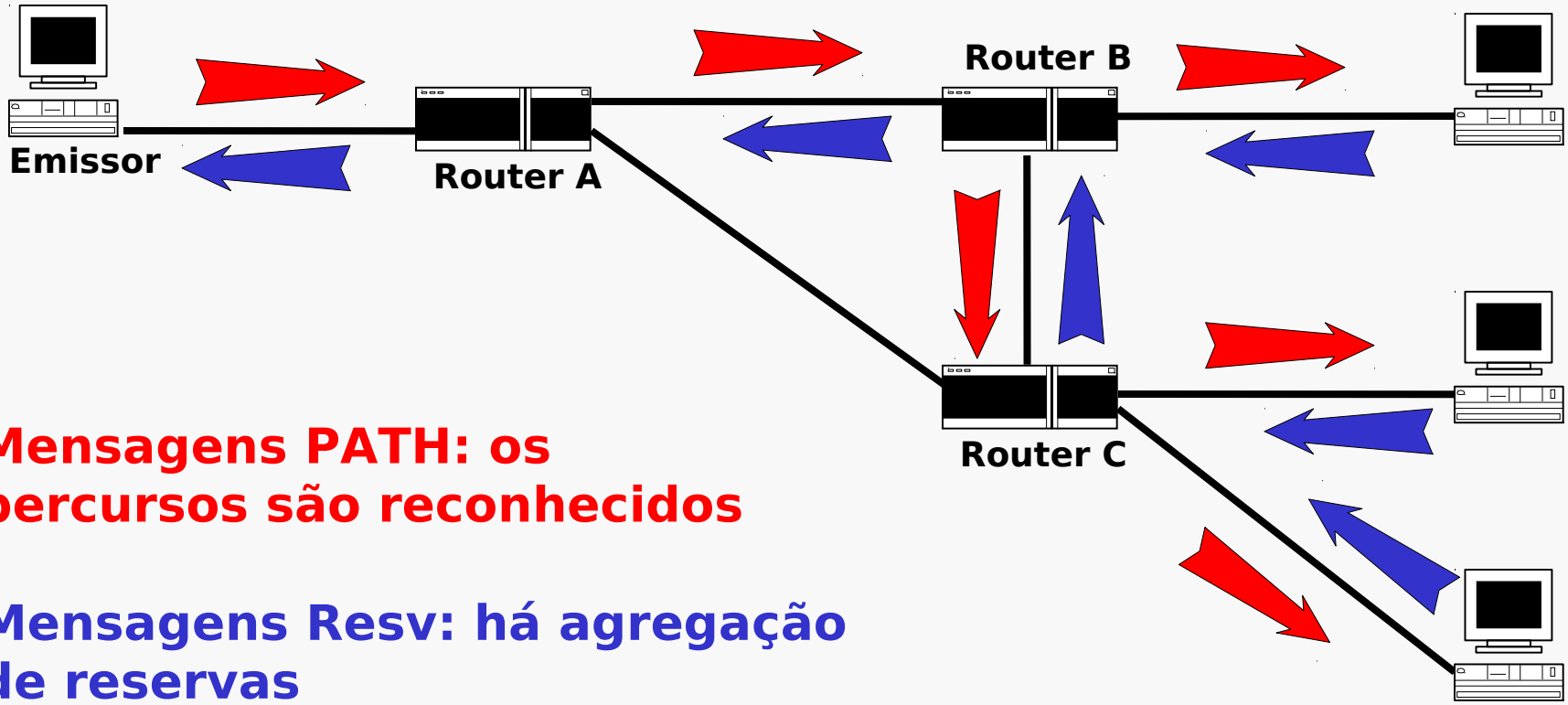
- **RFC 2205**
- **Encapsulated in IP; *protocol type* = 46 (0x2E)**
- **Signaling is based on the exchange of PATH and RESV messages**
  - **PATH announces the source traffic characteristics**
  - **RESV establishes the reservations requested by the receivers**
  - **If the reservation cannot be established, a RESV ERR message is sent**
- **Reservation states should be periodically refreshed (*soft states*)**



RSVP operation

- **Source-network signaling**
  - ***Path message:*** **torna a presença do emissor conhecida aos routers**
  - **Anulação do caminho: apaga dos routers o estado do caminho correspondente ao emissor**
- **Sinalização receptor-rede**
  - ***Mensagem reservation:*** **reserva recursos do(s) emissor(es) até ao receptor**
  - **Anulação da reserva: apaga as reservas feitas pelo receptor**
- **Sinalização rede-sistema terminal**
  - ***path error***
  - ***reservation error***

RSVP - Funcionamento Básico

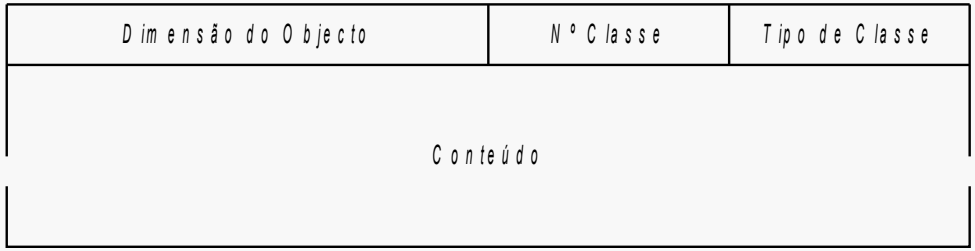


Mensagens PATH: os percursos são reconhecidos

Mensagens Resv: há agregação de reservas

Formato das Mensagens RSVP

Formato de cada objecto RSVP:



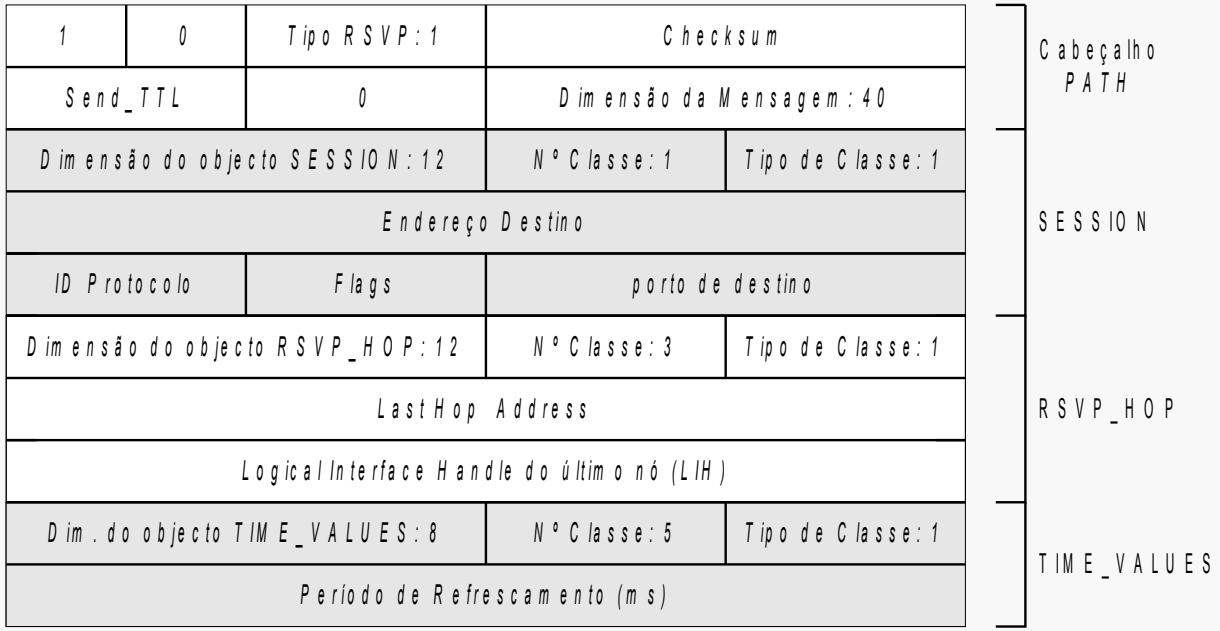
Mensagem *Path*: sinalização *emissor-rede*

- Comunica a informação do emissor e a informação de encaminhamento correspondente ao caminho inverso até ao emissor
- *PATH* (*Type* = 0x01)
  - Tspec ("flow traffic specification"): contém os parâmetros que descrevem a fonte de tráfego baseados no modelo "Token Bucket"
- Conteúdo da mensagem *path*:
  - *address:* destino unicast ou grupo multicast
  - *flowspec:* especificação dos requisitos de LB
  - *filter flag:* se for igual a 1, grava as identidades dos emissores localizados a montante (para permitir a filtragem de pacotes com base na fonte)
  - *Salto anterior:* ID do router/host localizado a montante
  - *refresh time:* intervalo de tempo ao fim do qual esta informação expira

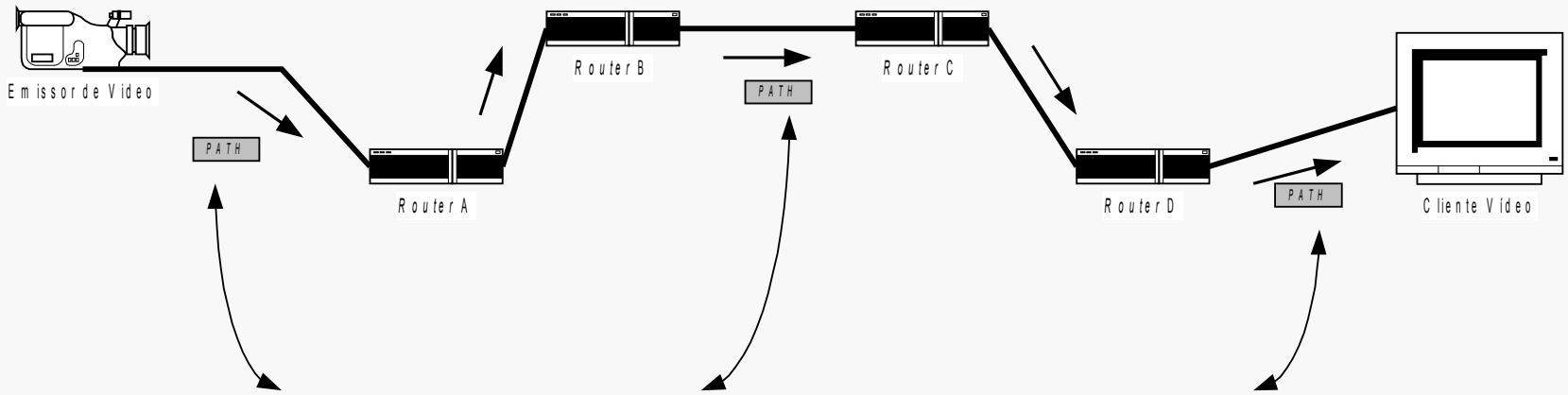
RSVP: mensagem PATH

Esta mensagem inclui três objectos RSVP obrigatórios (para além do FLOWSPEC):

- **SESSION – Identifica a sessão pelo endereço IP destino, porto destino e ID protocolo**
- **RSVP_HOP – Indica ao próximo router o endereço IP e porta emissora**
- **TIME_VALUES – Indica o período de tempo entre envios de mensagens PATH**



RSVP PATH (Exemplo)



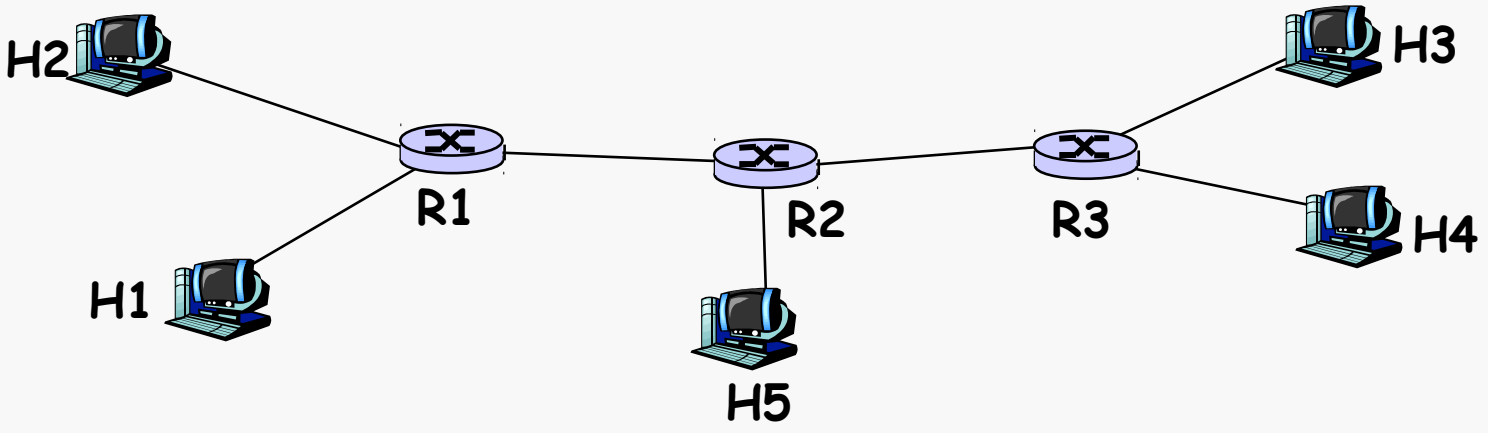
Vs.:4	iHL:5	Serviço	Dim. Total: 60
Identificação		Flg	Offset do Fragmento
Time to Live	Protocolo: 46		Checksum do cabeçalho
Endereço Fonte: **Servidor de Vídeo**			
Endereço Destino: **Cliente Vídeo**			
1	0	Tipo: 1	Checksum
Send_TTL	0		Dim. Mensagem: 40
Dim. SESSION: 12		Nº Classe: 1	Tipo Cl.: 1
Endereço Destino: **Cliente Vídeo**			
ID Protocolo	Flags		Porto de destino
Dim. RSVP_HOP: 12		Nº Classe: 3	Tipo Cl.: 1
Last Hop Address: **Servidor de Vídeo**			
Logical Interface Handle do último nó (LIH)			
Dim. TIME_VALUES: 8		Nº Classe: 5	Tipo Cl.: 1
Período de Refrescamento (ms)			

Vs.:4	iHL:5	Serviço	Dim. Total: 60
Identificação		Flg	Offset do Fragmento
Time to Live	Protocolo: 46		Checksum do cabeçalho
Endereço Fonte: **Servidor de Vídeo**			
Endereço Destino: **Cliente Vídeo**			
1	0	Tipo: 1	Checksum
Send_TTL	0		Dim. Mensagem: 40
Dim. SESSION: 12		Nº Classe: 1	Tipo Cl.: 1
Endereço Destino: **Cliente Vídeo**			
ID Protocolo	Flags		Porto de destino
Dim. RSVP_HOP: 12		Nº Classe: 3	Tipo Cl.: 1
Last Hop Address: **Router B**			
Logical Interface Handle do último nó (LIH)			
Dim. TIME_VALUES: 8		Nº Classe: 5	Tipo Cl.: 1
Período de Refrescamento (ms)			

Vs.:4	iHL:5	Serviço	Dim. Total: 60
Identificação		Flg	Offset do Fragmento
Time to Live	Protocolo: 46		Checksum do cabeçalho
Endereço Fonte: **Servidor de Vídeo**			
Endereço Destino: **Cliente Vídeo**			
1	0	Tipo: 1	Checksum
Send_TTL	0		Dim. Mensagem: 40
Dim. SESSION: 12		Nº Classe: 1	Tipo Cl.: 1
Endereço Destino: **Cliente Vídeo**			
ID Protocolo	Flags		Porto de destino
Dim. RSVP_HOP: 12		Nº Classe: 3	Tipo Cl.: 1
Last Hop Address: **Router D**			
Logical Interface Handle do último nó (LIH)			
Dim. TIME_VALUES: 8		Nº Classe: 5	Tipo Cl.: 1
Período de Refrescamento (ms)			

RSVP: audioconferência simples

- **H_1, H_2, H_3, H_4, H_5 simultaneamente emissores e receptores**
- **Grupo multicast: m_1**
- **Não há filtragem: os pacotes de qualquer emissor são encaminhados**
- **Ritmo do áudio: *b***
- **Apenas é possível uma árvore de encaminhamento multicast**

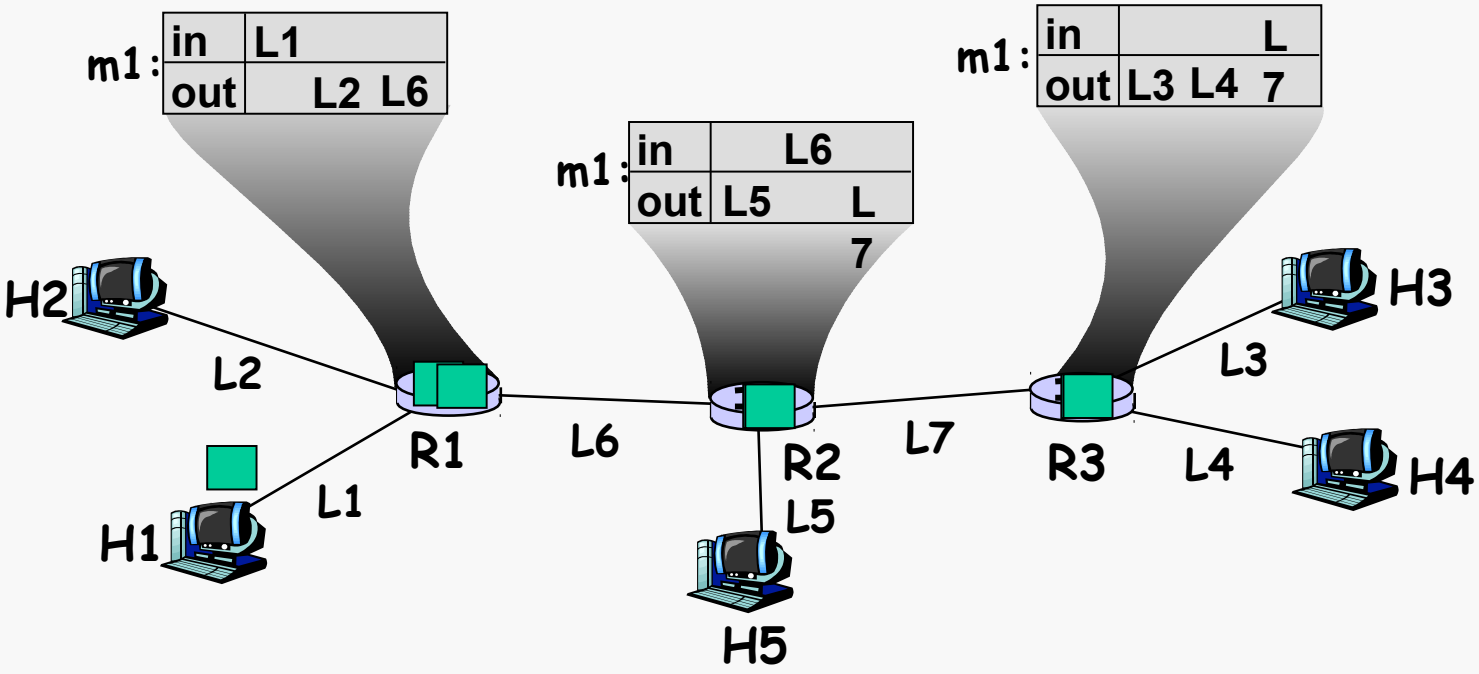


RSVP: construindo o estado do caminho

- H_1, …, H_5 enviam mensagens para *m1:*

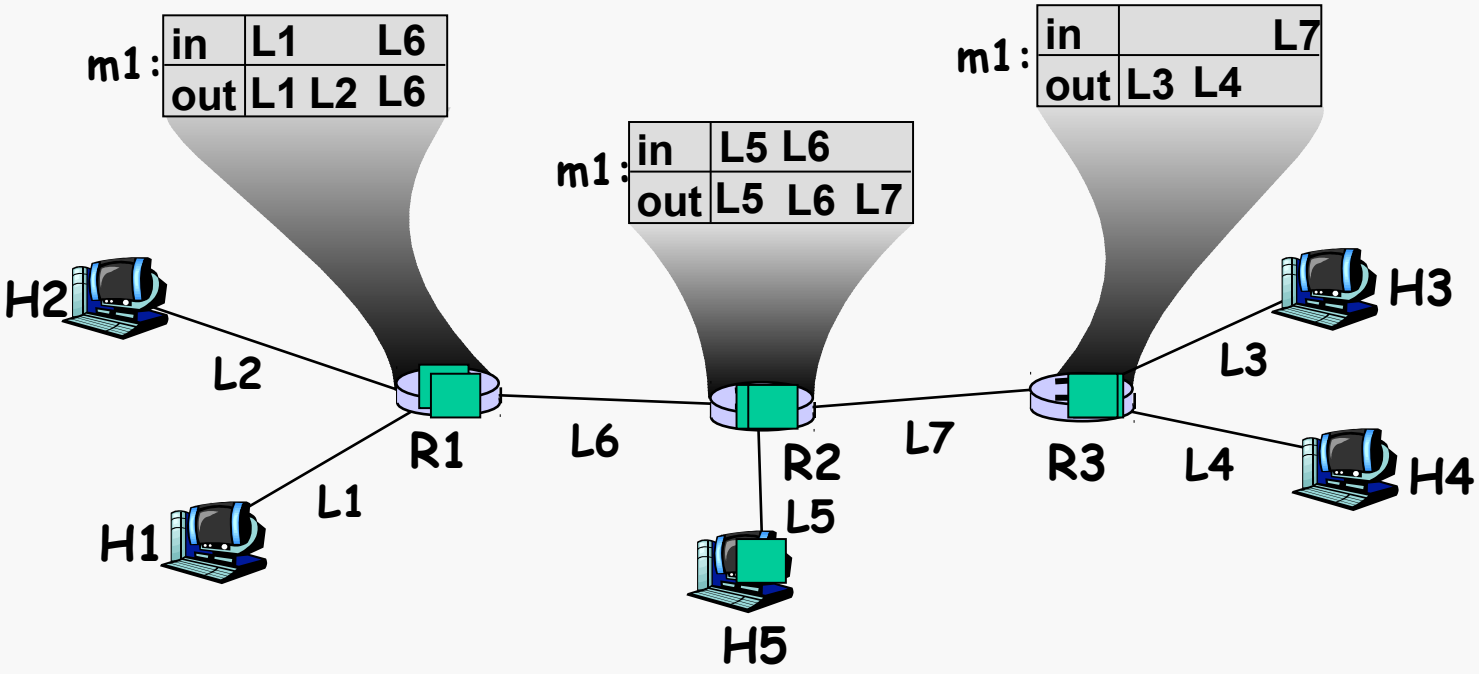
  (address=m_1, Tspec=*b*, filter-spec=no-filter, refresh=100)

- Supondo que H_1 envia a primeira mensagem *path*



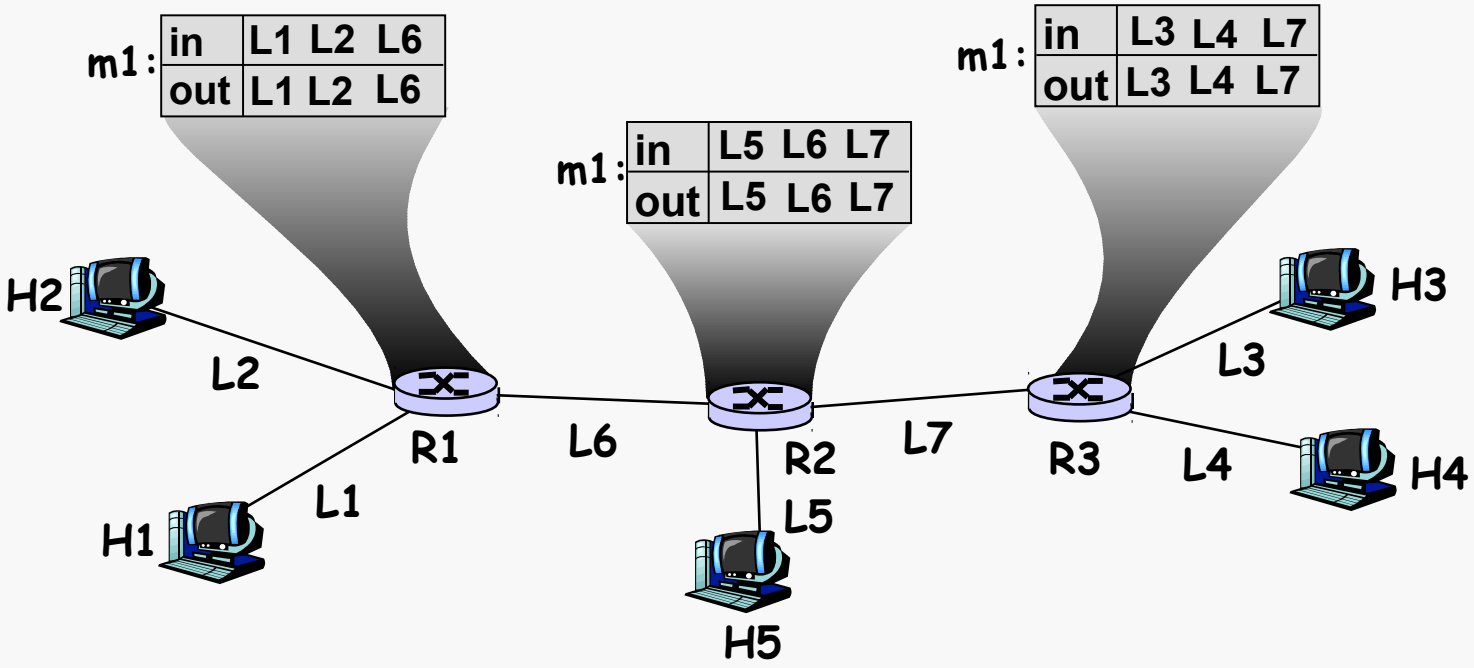
RSVP: construindo o estado do caminho

- a seguir, H_5 envia uma mensagem *path*



RSVP: construindo o estado do caminho

- H_2, H_3, H_5 enviam mensagens *path*, completando as tabelas de estado do *path*



Mensagem reservation: sinalização *receptor-rede*

- **Conteúdo da mensagem *reservation*:**
  - ***desired bandwidth:*** **parte do RSpec**
  - ***filter type:*** **filtra que pacotes podem utilizar a reserva**
    - **no filter: quaisquer pacotes endereçados ao grupo multicast podem usar a reserva**
    - **fixed filter: apenas os pacotes provenientes de um conjunto específico de emissores podem usar a reserva**
    - **dynamic filter: os emissores cujos pacotes podem ser encaminhados através da ligação mudarão (por escolha do receptor) ao longo do tempo.**
  - ***filter spec:*** **identifica os pacotes pertencentes a uma reserva**
- **As reservas irão fluir na direcção upstream, do receptor até aos emissores, reservando recursos e criando estados adicionais nos routers relacionados com as especificações dos receptores.**

RSVP: mensagem RESERVATION

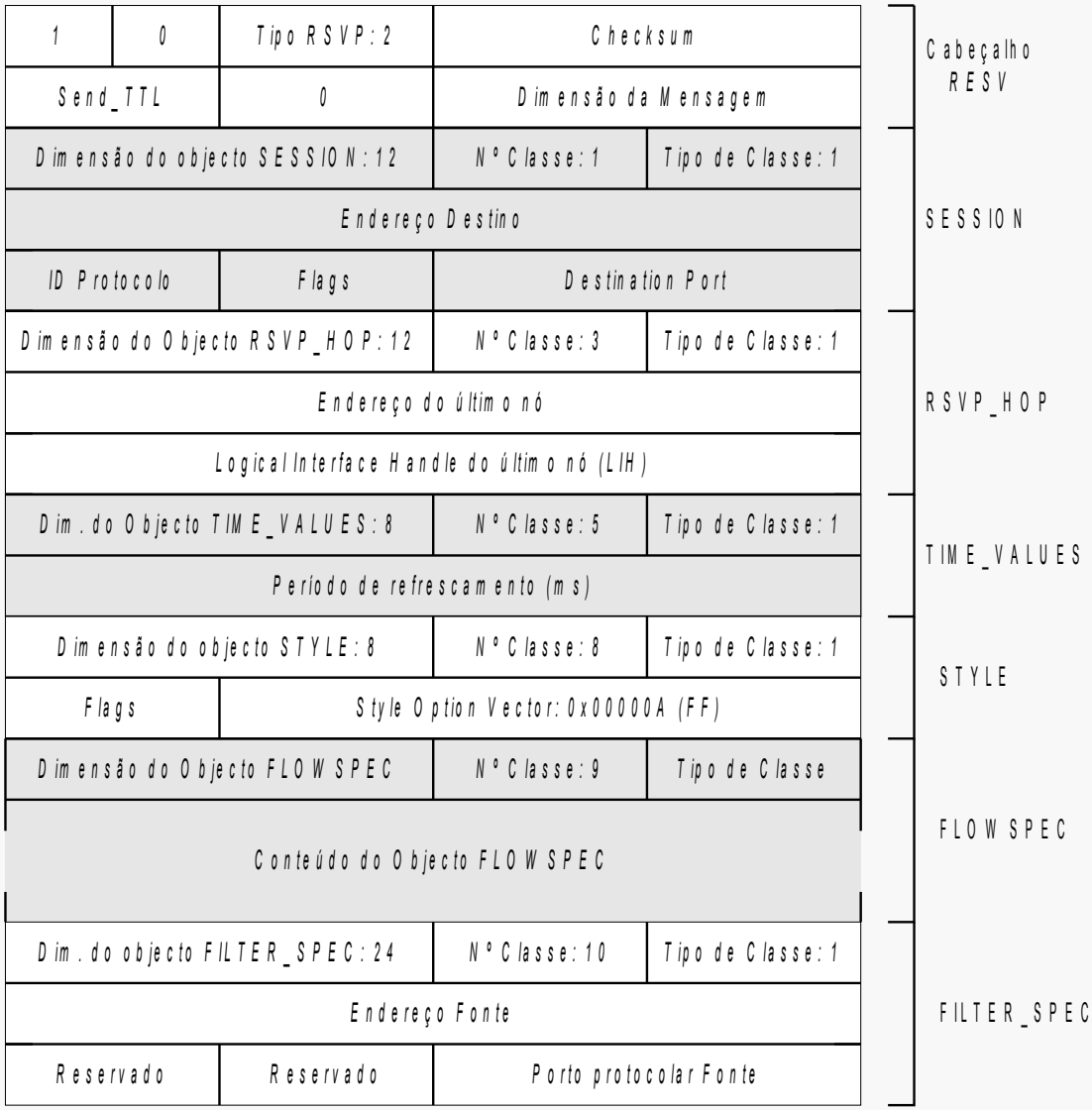
- **RESV (*Type* = 0x02)**
  - **Tspec: o mesmo que foi recebido na mensagem PATH**
  - **FilterSpec ("*filter specification*"): contém o descritor do fluxo que permite os routers identificarem os pacotes pertencentes a esta reserva (endereço origem, endereço destino, tipo de protocolo, número de porto origem, número de porto destino, uma qualquer combinação destes)**
  - **Rspec ("*flow reservation specification*"): contém os parâmetros que descrevem a reserva que o receptor pretende ver suportado**
    - **o Rspec é especificado se o receptor pretender um serviço do tipo "*guaranteed service*"; quando não é especificado, significa que o receptor pretende um serviço do tipo "*controlled load*"**

RSVP: mensagem RESV

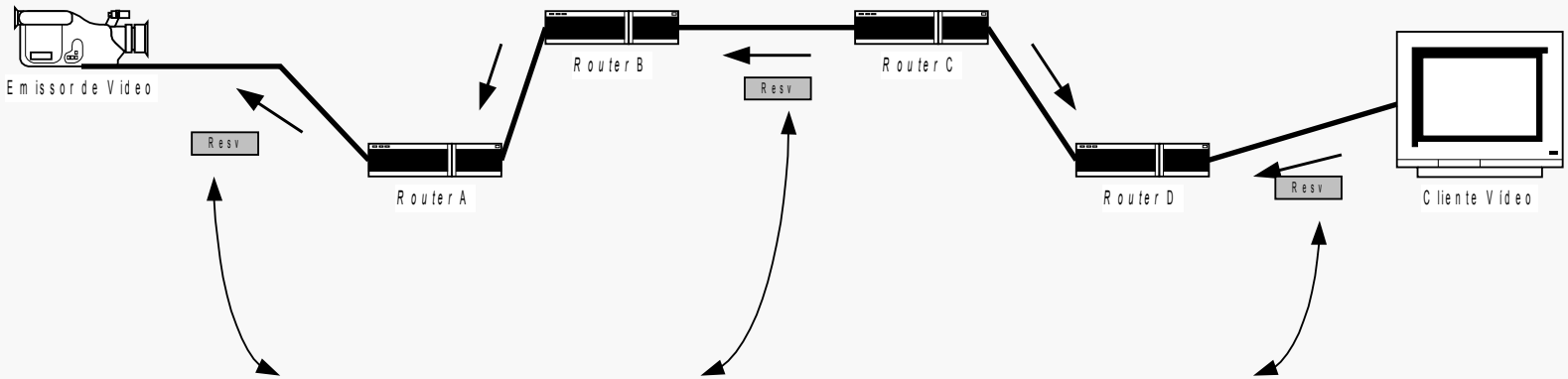
STYLE – Identifica o estilo de reserva (receptor especifica valor de reserva para qualquer emissor, por emissor ou conjunto de emissores)

FLOWSPEC – Inclui o TSpec e o RSpec

FILTER_SPEC – Indica a informação necessária para o classificador de pacotes



RSVP RESV (Exemplo)



Vs.:4	iHL:5	Serviço	Dimensão Total
Identificação		Flg	Offset do Fragmento
Time to Live	Protocolo: 46		Checksum do cabeçalho
Endereço Fonte: **Router A**			
Endereço Destino: **Servidor de Vídeo**			
1	0	Tipo: 2	Checksum
Send_TTL	0		Dimensão da Mensagem
Dimensão do SESSION: 12		Nº Classe: 1	Tipo Cl.: 1
Endereço Destino: **Cliente Vídeo**			
Protocol Id	Flags		Porto protocolar Destino
Dim. do RSVP_HOP: 12		Nº Classe: 3	Tipo Cl.: 1
Endereço do último nó: **Router A**			
Logical Interface Handle do último nó (LIH)			
Dim. do TIME_VALUES: 8		Nº Classe: 5	Tipo Cl.: 1
Período de Refrescamento (ms)			
Dim. do Objecto STYLE: 8		Nº Classe: 8	Tipo Cl.: 1
Flags	Style Option Vector: 0x00000A (FF)		
Dimensão do FLOW SPEC		Nº Classe: 9	Tipo Classe
Conteúdo do Objecto FLOW SPEC			
Dim. do FILTER_SPEC: 12		Nº Cl.: 10	Tipo Cl.: 1
Endereço Fonte: Servidor de Vídeo			
Reservado	Reservado		Porto protocolar Fonte

Vs.:4	iHL:5	Serviço	Dimensão Total
Identificação		Flg	Offset do Fragmento
Time to Live	Protocolo: 46		Checksum do cabeçalho
Endereço Fonte: **Router C**			
Endereço Destino: **Router B**			
1	0	Tipo: 2	Checksum
Send_TTL	0		Dimensão da Mensagem
Dimensão do SESSION: 12		Nº Classe: 1	Tipo Cl.: 1
Endereço Destino: **Cliente Vídeo**			
Protocol Id	Flags		Porto protocolar Destino
Dim. do RSVP_HOP: 12		Nº Classe: 3	Tipo Cl.: 1
Endereço do último nó: **Router C**			
Logical Interface Handle do último nó (LIH)			
Dim. do TIME_VALUES: 8		Nº Classe: 5	Tipo Cl.: 1
Período de Refrescamento (ms)			
Dim. do Objecto STYLE: 8		Nº Classe: 8	Tipo Cl.: 1
Flags	Style Option Vector: 0x00000A (FF)		
Dimensão do FLOW SPEC		Nº Classe: 9	Tipo Classe
Conteúdo do Objecto FLOW SPEC			
Dim. do FILTER_SPEC: 12		Nº Cl.: 10	Tipo Cl.: 1
Endereço Fonte: Servidor de Vídeo			
Reservado	Reservado		Porto protocolar Fonte

Vs.:4	iHL:5	Serviço	Dimensão Total
Identificação		Flg	Offset do Fragmento
Time to Live	Protocolo: 46		Checksum do cabeçalho
Endereço Fonte: **Cliente Vídeo**			
Endereço Destino: **Router D**			
1	0	Tipo: 2	Checksum
Send_TTL	0		Dimensão da Mensagem
Dimensão do SESSION: 12		Nº Classe: 1	Tipo Cl.: 1
Endereço Destino: **Cliente Vídeo**			
Protocol Id	Flags		Porto protocolar Destino
Dim. do RSVP_HOP: 12		Nº Classe: 3	Tipo Cl.: 1
Endereço do último nó: **Cliente Vídeo**			
Logical Interface Handle do último nó (LIH)			
Dim. do TIME_VALUES: 8		Nº Classe: 5	Tipo Cl.: 1
Período de Refrescamento (ms)			
Dim. do Objecto STYLE: 8		Nº Classe: 8	Tipo Cl.: 1
Flags	Style Option Vector: 0x00000A (FF)		
Dimensão do FLOW SPEC		Nº Classe: 9	Tipo Classe
Conteúdo do Objecto FLOW SPEC			
Dim. do FILTER_SPEC: 12		Nº Cl.: 10	Tipo Cl.: 1
Endereço Fonte: Servidor de Vídeo			
Reservado	Reservado		Porto protocolar Fonte

Estilos de Reserva suportados pelo RSVP

- **"Fixed Filter" (Style Option Vector = 0x00000A)**
  - **O receptor especifica um valor de reserva por cada emissor**
- **"Wildcard Filter" (Style Option Vector = 0x000011)**
  - **O receptor especifica um valor de reserva único para receber o tráfego de qualquer emissor**
- **"Explicit Filter" (Style Option Vector = 0x000012)**
  - **O receptor especifica uma lista de emissores dos quais quer receber informação e um valor de reserva único para receber o tráfego dos emissores especificados**

- **Nas mensagens RSVP RESV:**
  - **O estilo de reserva é declarado pelo objecto STYLE**
  - **Os emissores são declarados no objecto FILTER_SPEC**

Parâmetros RSVP

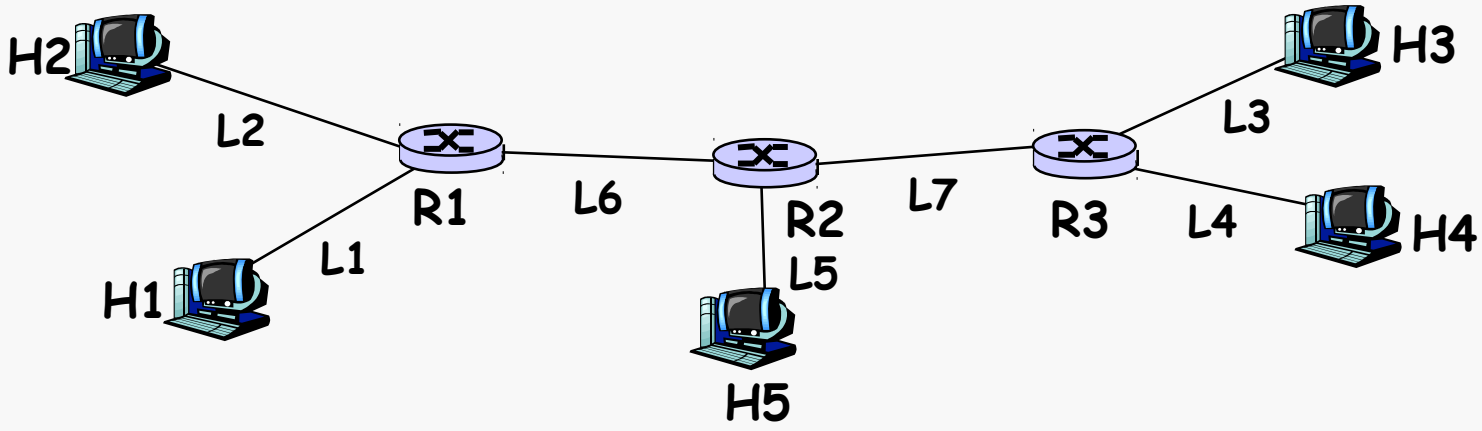
Parâmetros RSVP	Descrição
TOKENBUCKETRATE (r)	TSpec: Rate of arriving tokens
TOKENBUCKETSIZE (b)	TSpec: Size of bucket
PEAKRATE (p)	TSpec: Maximum bit rate of the flow
MINIMUMPOLICEDUNIT (m)	TSpec: Minimum packet size considered
MAXIMUMPACKETSIZE (M)	TSpec: Maximum packet size
RATE (R)	RSpec*: Reservation rate
DELAYSLACKTERM	RSpec*: Tolerance of the requested delay

*** RSpec é especificado apenas em *Guaranteed Services***

Reserva RSVP - Exemplo 1

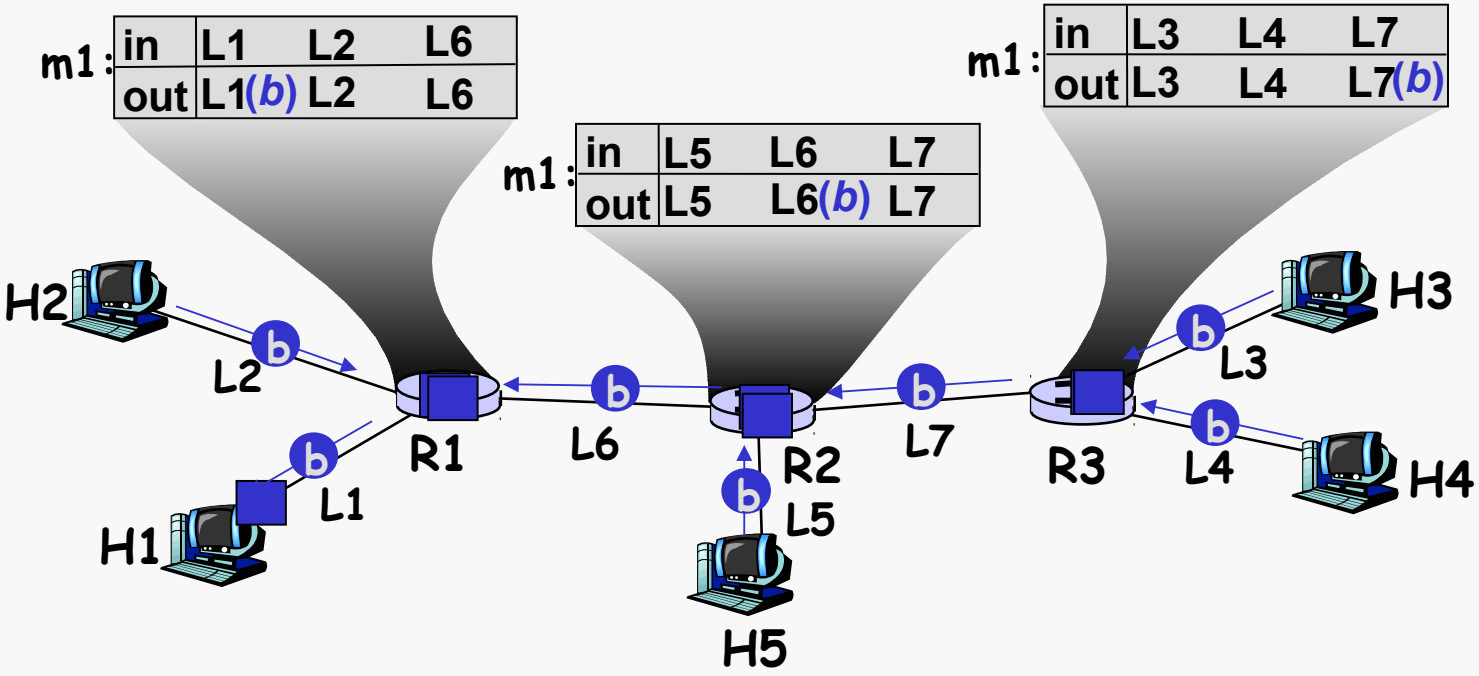
H_1 pretende receber áudio de todos os emissores

- A mensagem de reserva de H_1 circula ao longo da árvore até ás fontes
- H_1 apenas reserva LB suficiente para 1 stream de áudio
- A reserva é do tipo "no filter" – qualquer emissor pode usar a LB reservada



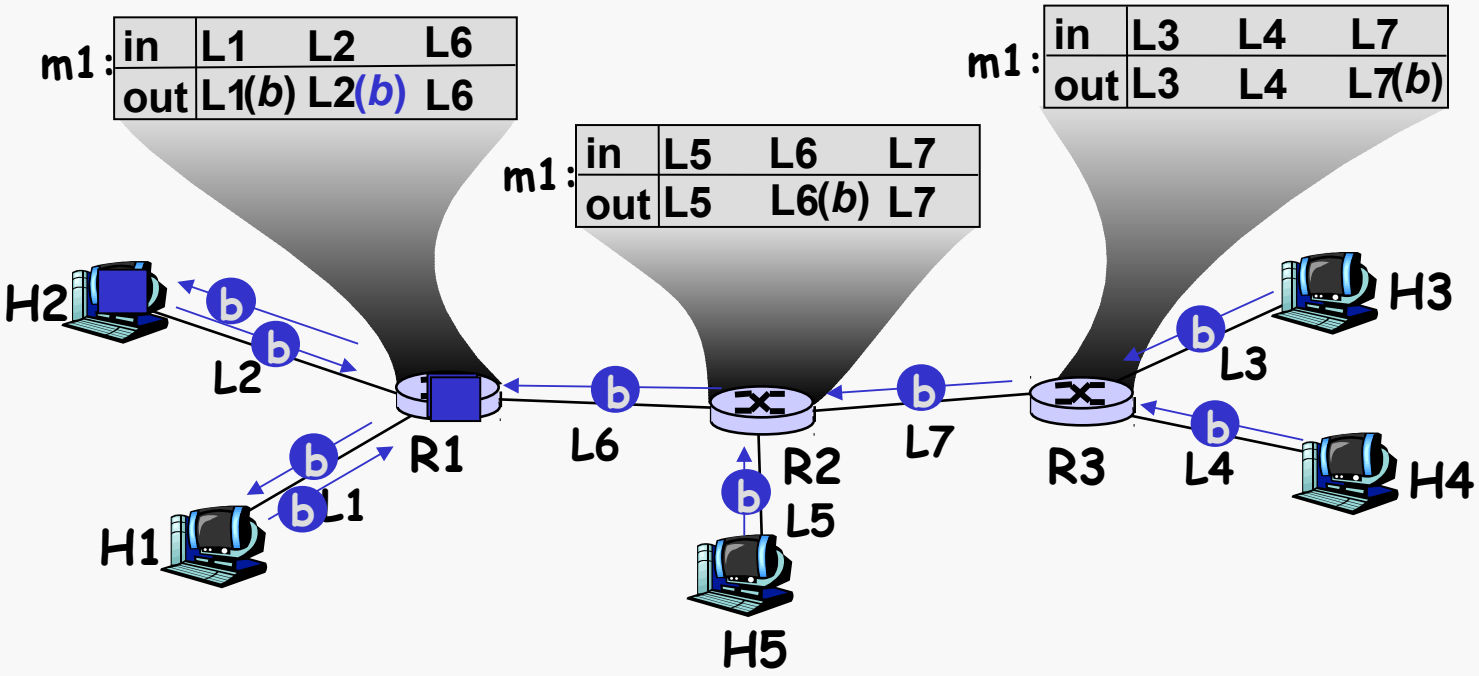
Reserva RSVP - Exemplo 1

- As mensagens de reserva de H_1 circulam ao longo da árvore até ás fontes
- Os routers e os hosts reservam a LB *b* necessária nas ligações downstream em direcção a H_1



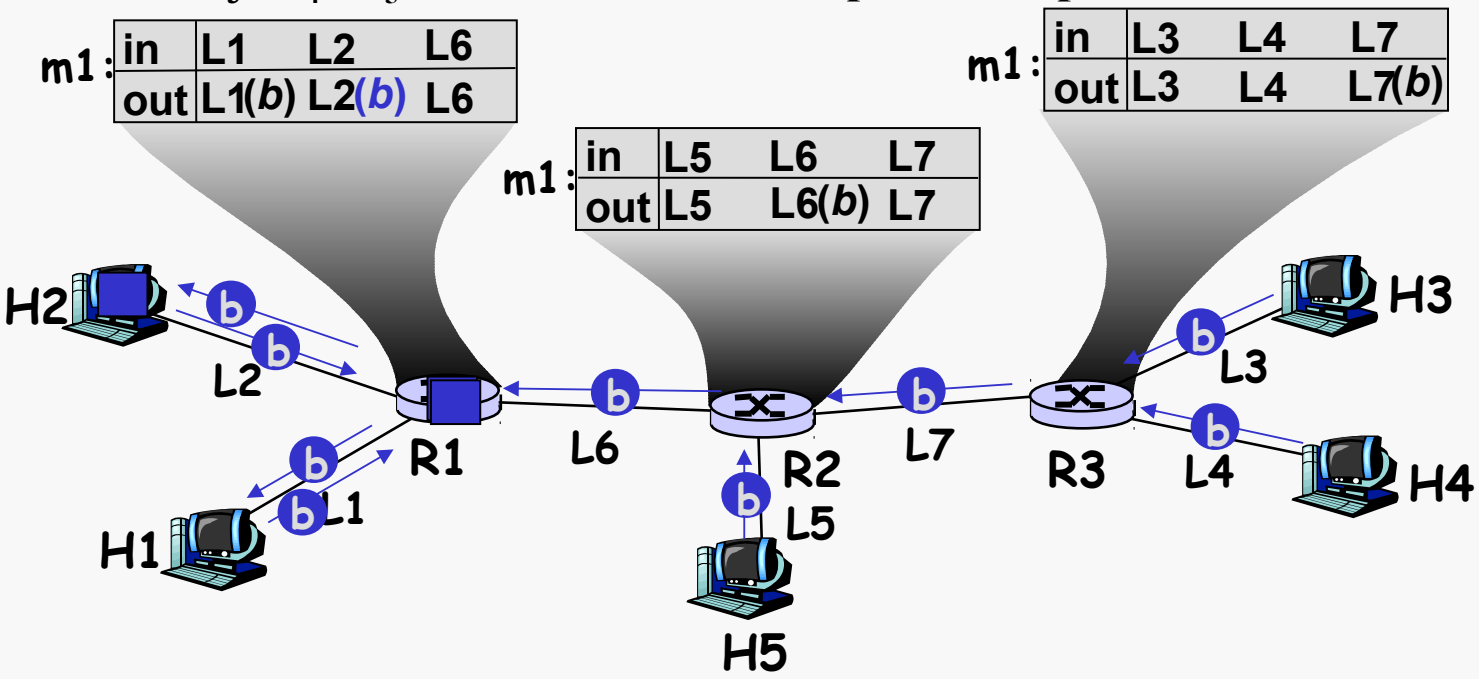
Reserva RSVP - Exemplo 1

- De seguida, H_2 efectua uma reserva de LB *b* do tipo no-filter
- H_2 encaminha para R_1, R_1 encaminha para H_1 e R_2
- R_2 não toma nenhuma acção, uma vez que a LB *b* já tinha sido reservada em L_6



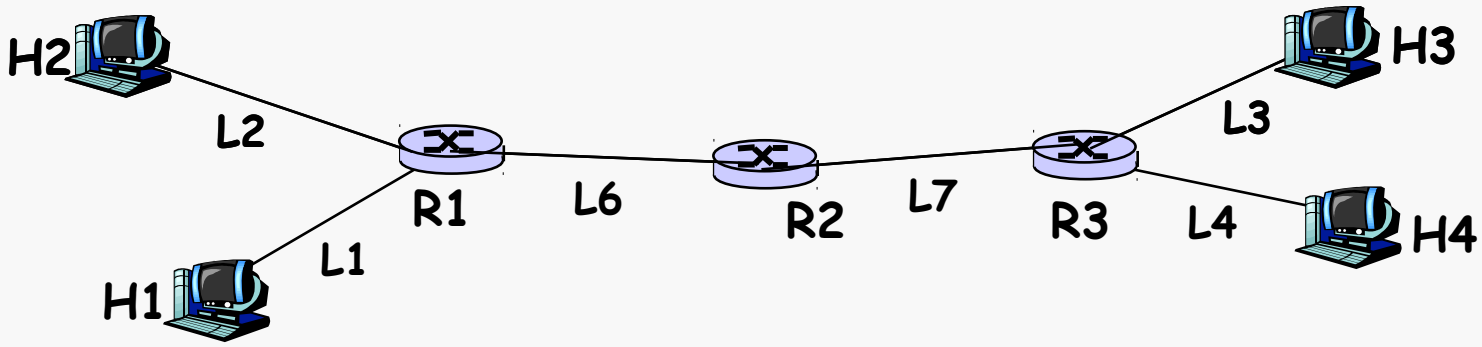
RSVP: questões sobre a reserva feita pelo receptor

- **E se houver múltiplos emissores (e.g., H_3, H_4, H_5) na mesma ligação (e.g., L_6)?**
  - **pacotes são intercalados de forma arbitrária**
  - **o fluxo de pacotes em L_6 é policiado por um *leaky bucket*: se o ritmo de envio de $H_3+H_4+H_5$ exceder b, ocorrerão perdas de pacotes**



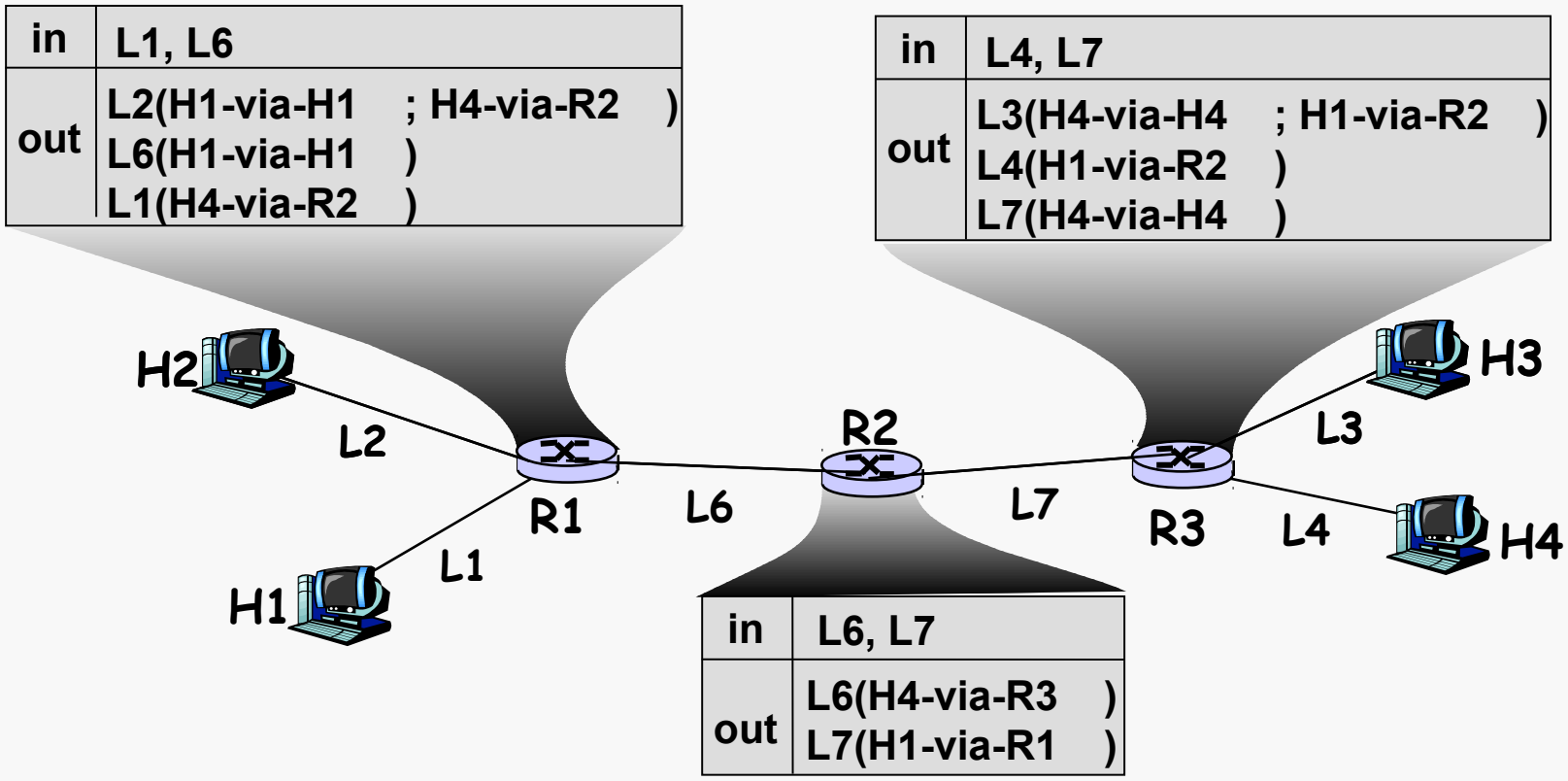
RSVP - Exemplo 2

- H_1, H_4 são apenas emissores
  - Enviam mensagens *path*
  - Os routers armazenam os emissores localizados a montante para cada ligação upstream
- H_2 pretende receber apenas de H_4



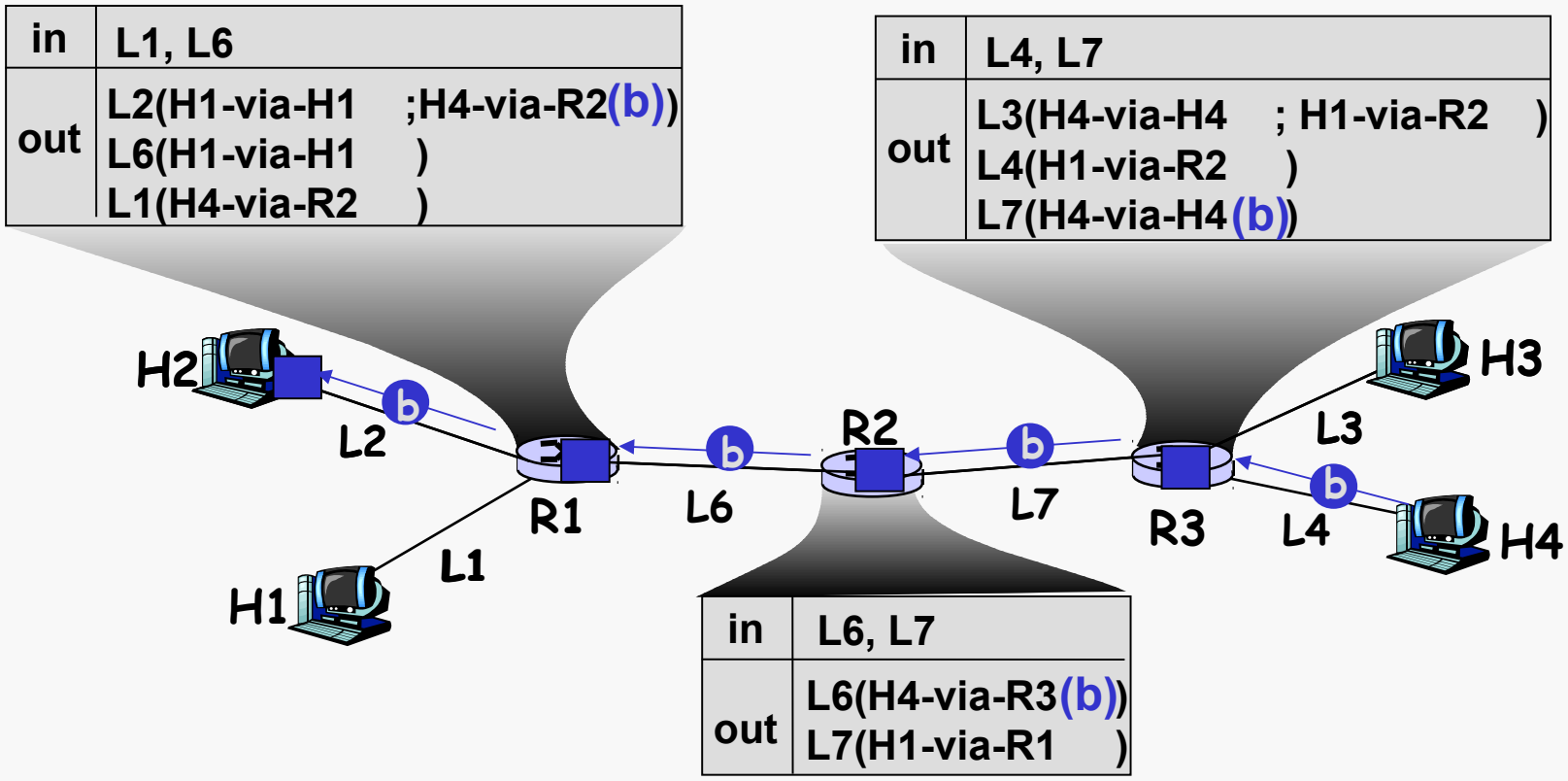
RSVP - Exemplo 2

- **H1, H4 são apenas emissores**
  - **Enviam mensagens *path***



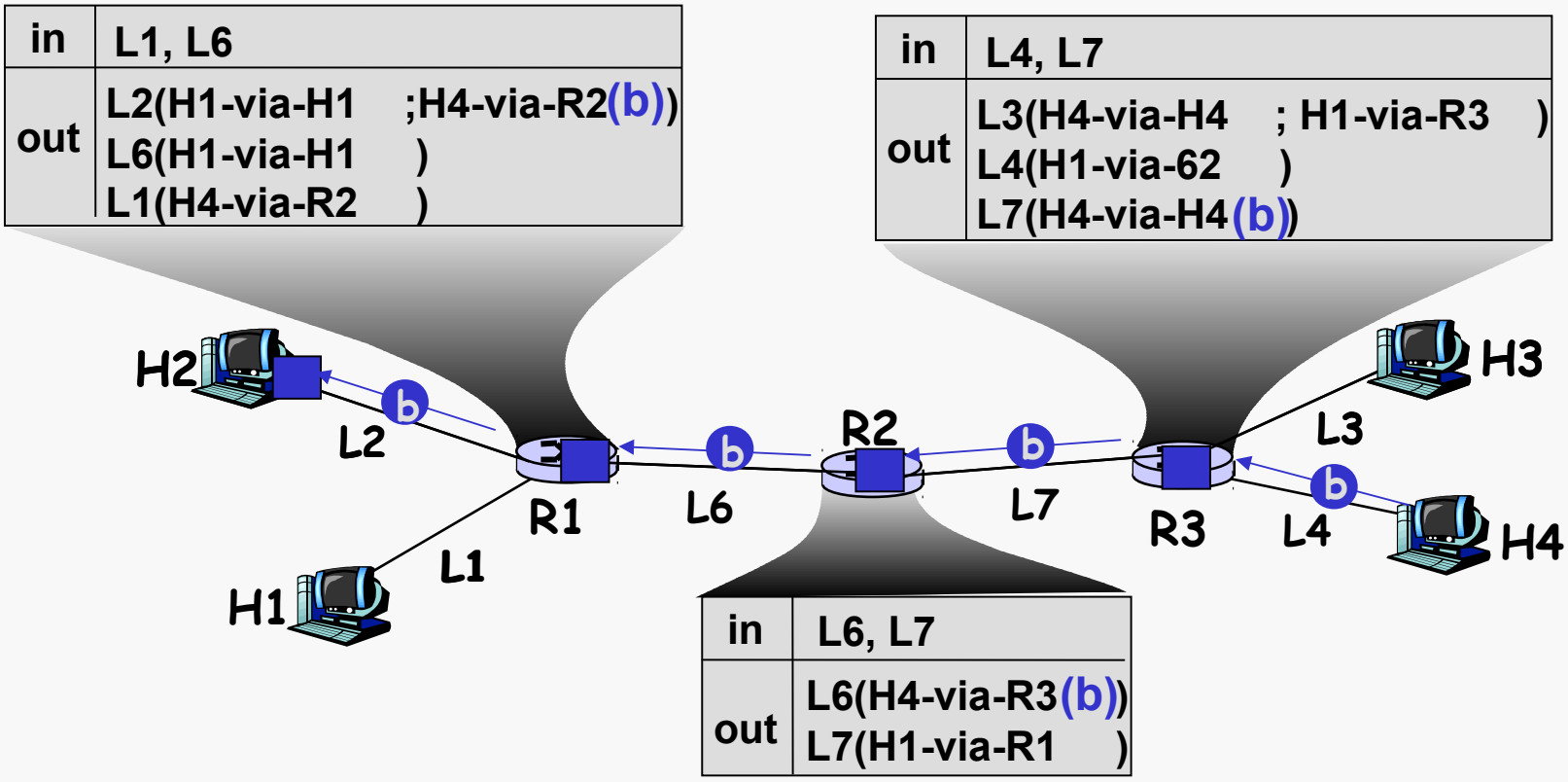
RSVP - Exemplo 2

- O receptor H_2 envia uma mensagem de reserva de uma LB *b* para a fonte H_4
  - Propagada para montante em direcção a H_4, reservando a LB *b*



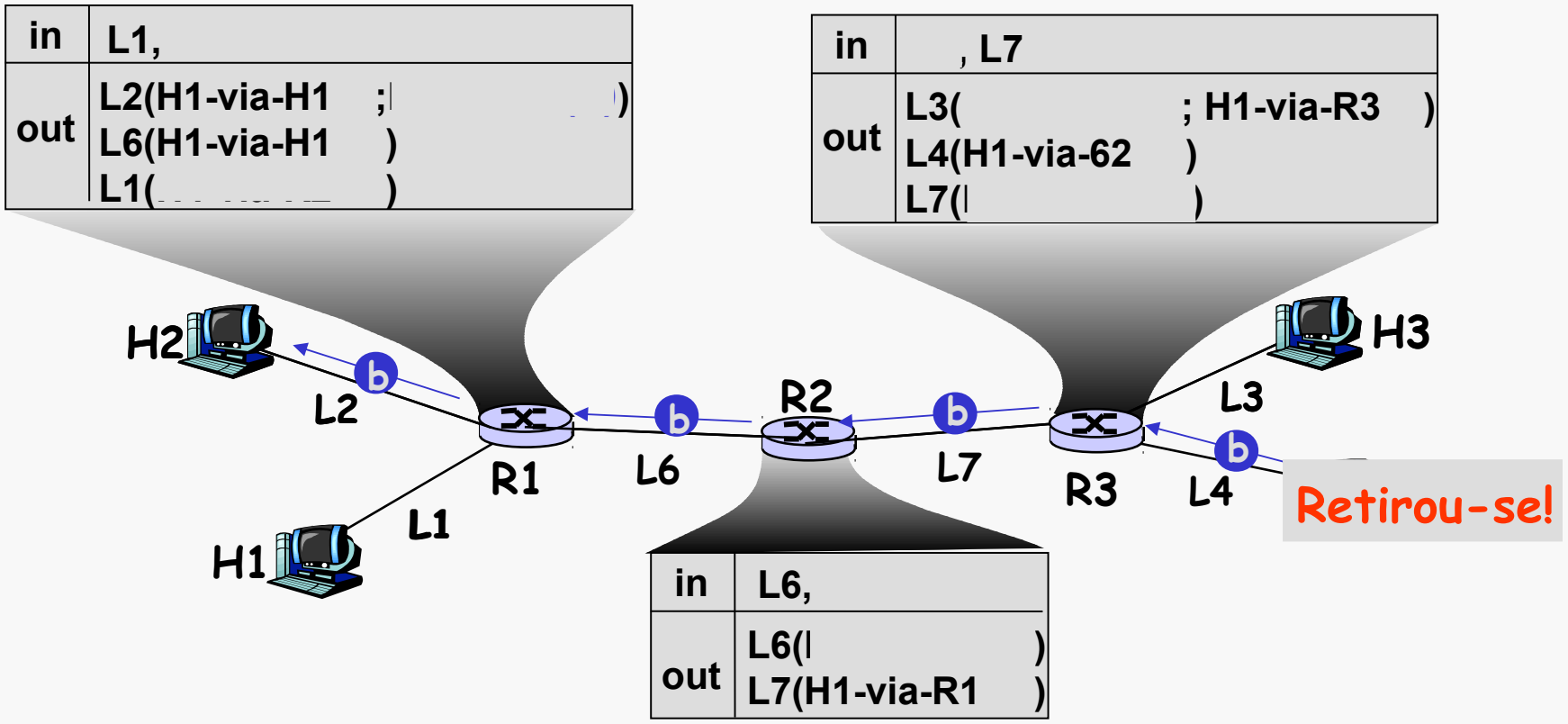
RSVP: *soft-state*

- Emissores reenviam periodicamente mensagens path para actualizar (manter) o estado
- Receptores reenviam periodicamente mensagens resv para actualizar (manter) o estado das reservas
  - As mensagens path e resv têm um campo TTL, que especifica o intervalo de actualização



RSVP: *soft-state*

- Suponham que H4 (emissor) se retira sem fazer o teardown
- Eventualmente o estado nos routers irá expirar e desaparecer!



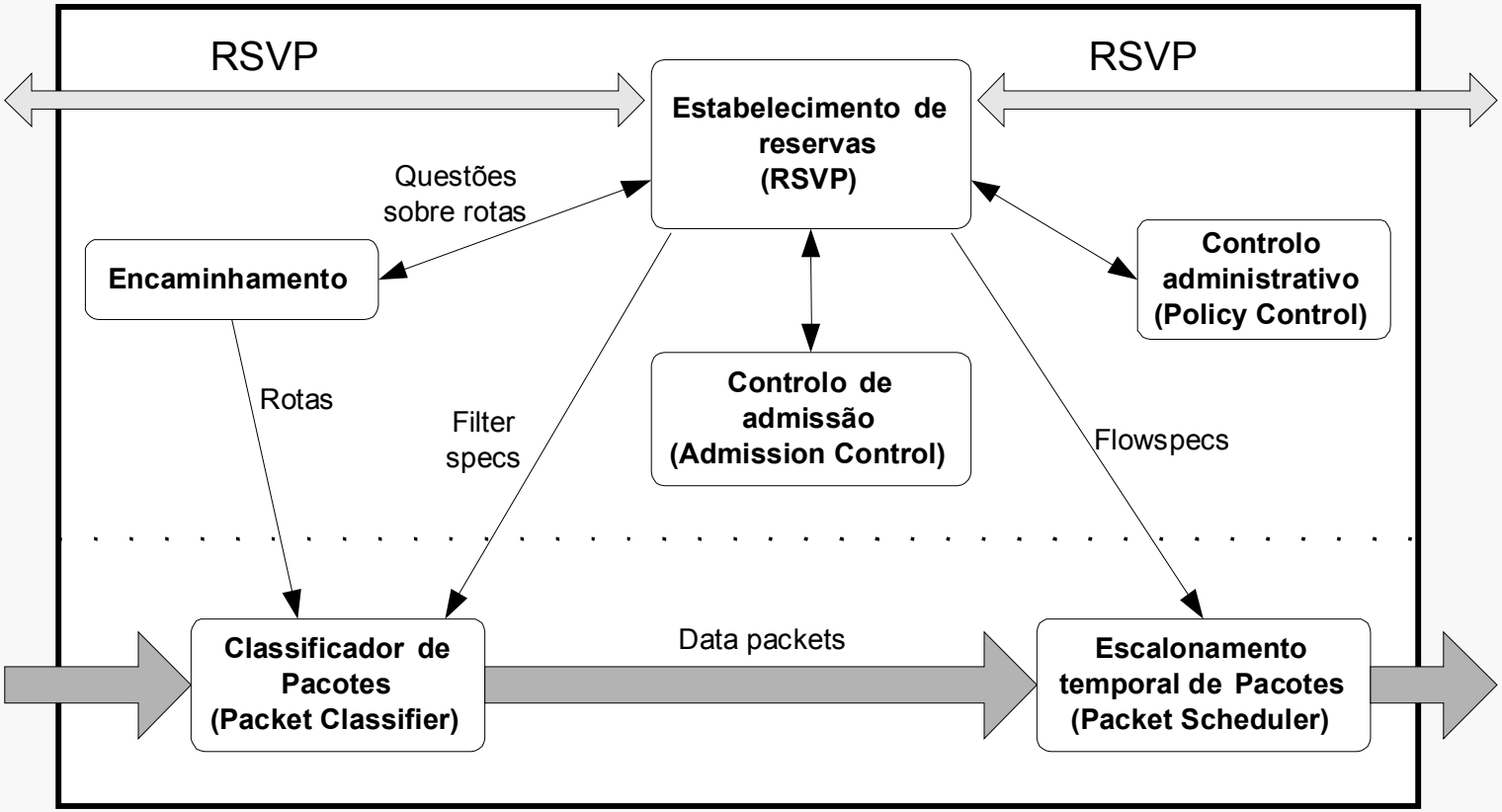
Os múltiplos usos da actualização reservation/path

- **Recuperar da perda de uma mensagem anterior de actualização**
  - **O tempo esperado até à recepção de uma nova actualização deve ser maior do que o *timeout interval***
- **Suportar receptor/emissor que desaparece sem efectuar o *teardown***
  - **O estado do emissor/receptor irá expirar e desaparecer**
- **As actualizações das reservas irão fazer com que novas reservas possam ser feitas para um receptor que pretenda receber de um emissor que entretanto se tenha juntado desde a última actualização de reservas**

Outras Mensagens RSVP

- **PATH ERR (*Type* = 0x03):**
  - **Enviada pelos routers em situações de erro**
- **RESV ERR (*Type* = 0x04):**
  - **Enviada pelos routers quando uma reserva não pode ser suportada**
- **PATH TEAR (*Type* = 0x05):**
  - **Enviada pelos emissores quando termina o envio de informação**
- **RESV TEAR (*Type* = 0x06):**
  - **Enviada pelos receptores quando não querem mais uma reserva**
- **RESV CONFIRMATION (*Type* = 0x07):**
  - **Enviada pelos routers para confirmar que uma reserva foi estabelecida**

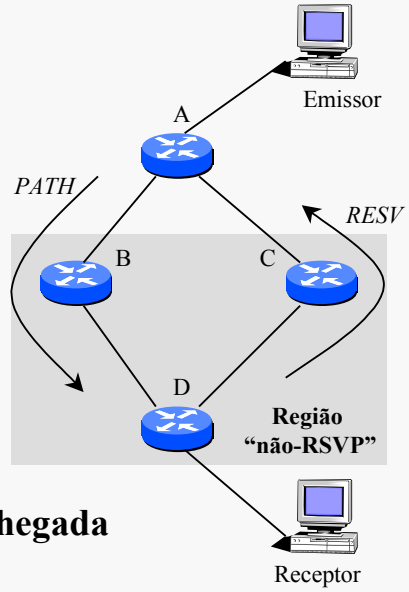
Arquitectura dos Routers



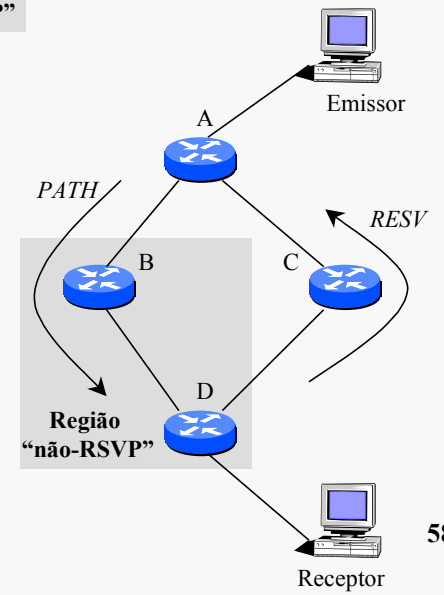
Regiões sem suporte RSVP

- **O objecto RSVP_HOP enviado nas mensagens PATH permite que as mensagens RESV sejam correctamente encaminhadas**

- **O *Logical Interface Handler* (LIH) do objecto RSVP_HOP permite resolver o problema quando uma mensagem RESV chega a um router por uma interface diferente da que foi utilizada para enviar a mensagem PATH**
  - **LIH especifica a interface de saída correcta, a que deve receber o pedido, independente da interface de chegada**



- **Um router RSVP reencaminha uma mensagem RESV sem a processar se não tiver recebido qualquer mensagem PATH**



Características do RSVP

- **Modelo multiponto-multiponto**
- **Reservas iniciadas pelos receptores**
- **Reservas temporizadas (*soft state*)**
- **Separação entre reserva e encaminhamento**
- **Separação entre reserva e filtragem de pacotes**
- **Suporte de diferentes estilos de reservas**
- **Agregação de reservas**

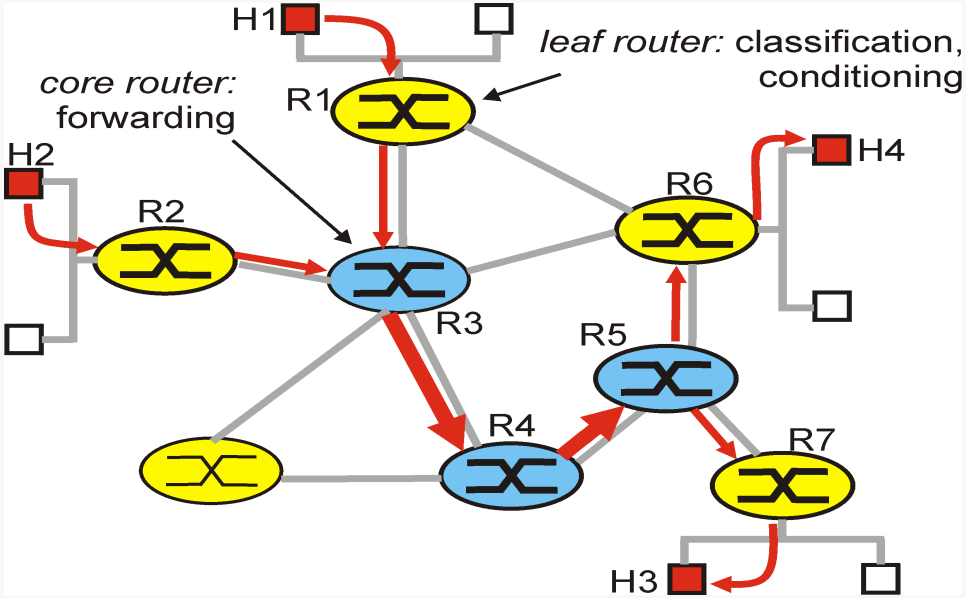
Arquitectura “*Differentiated Services*”

Arquitectura *Differentiated Services* (*DiffServ*)

- **Problemas da arquitectura *Integrated Services*:**
  - **Os *Routers* mantêm a informação de estado das reservas extremo-a-extremo**
    - **pouca escalabilidade**
  - **Os *Routers* determinam o atendimento com base em múltiplos campos (endereço origem e destino, protocolo, porto origem e destino)**
    - **penaliza o desempenho**
  - **Suporta apenas duas classes de serviço "*controlled load*" e "*guaranteed service*"**
    - **pouca flexibilidade**
  - **Exige sinalização extremo-a-extremo RSVP**
    - **tempo de estabelecimento das reservas elevados**
- **Arquitectura DiffServ:**
  - **Por contrato, o fluxo de tráfego de cada cliente é classificado como pertencente a uma classe particular**
  - **Trata classes de fluxos que exijam a mesma Qualidade de Serviço**
  - **À entrada da rede, os pacotes são marcados como pertencentes à classe contratada e o escalonamento dos pacotes é baseado na marca do pacote**

Ideias Base

- **Implementar operações simples de encaminhamento nos *routers* interiores (*core routers*) da rede e deixar as operações complexas para os *routers* fronteira (*edge routers*) da rede.**
- **Definem-se apenas elementos funcionais que permitem suportar qualquer classe de serviço.**

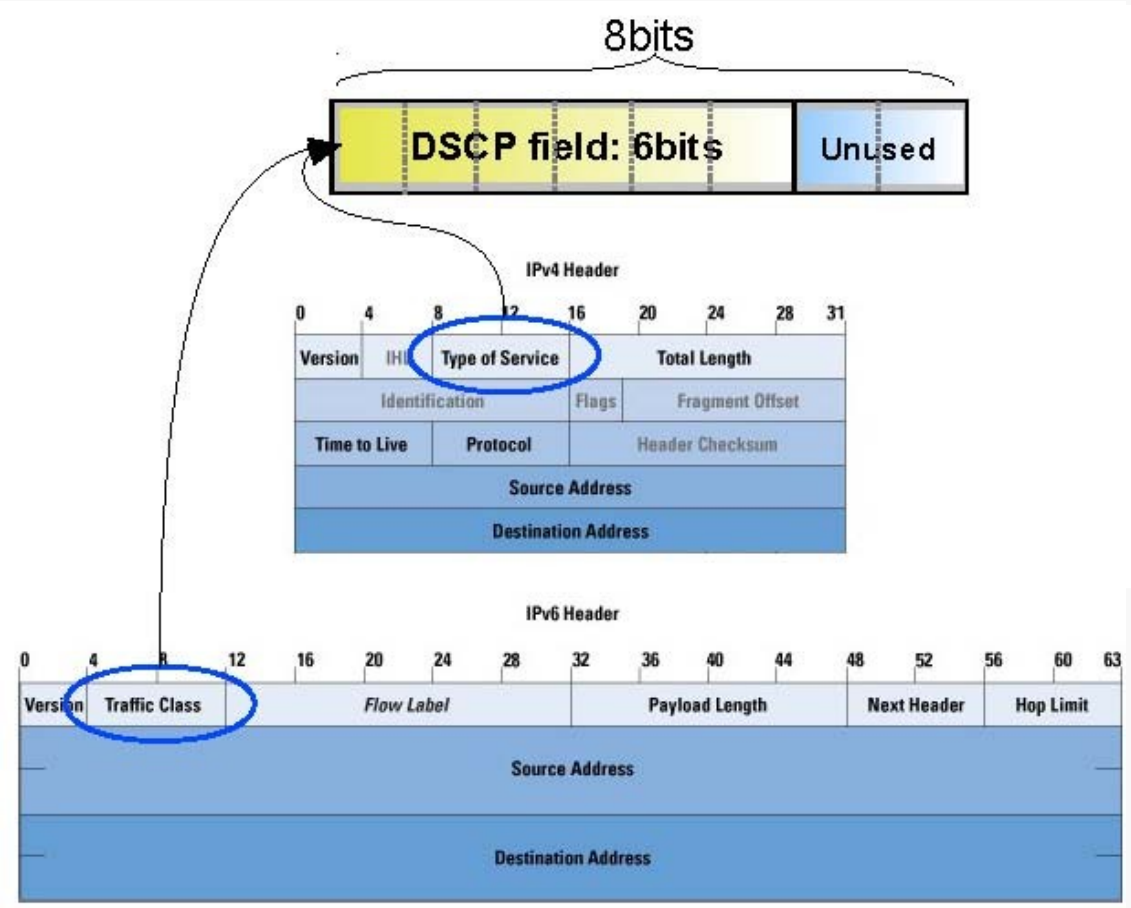


• Com base na marcação ele prioritiza a passagem ...

Elementos funcionais

- ***Edge Routers*:**
  - **Classificam os pacotes: marcam cada pacote no campo *Type of Service* do cabeçalho IP**
  - **Condicionam o tráfego: por exemplo, usam um "*Token Bucket*" para verificar se o tráfego de entrada é o contratado e**
    - **atrasam o tráfego em excesso ou**
    - **descartam o tráfego em excesso**
- ***Core Routers*:**
  - **Identificam o tratamento a dar aos pacotes com base na marca e de acordo com um *Per-Hop-Behavior* (PHB)**

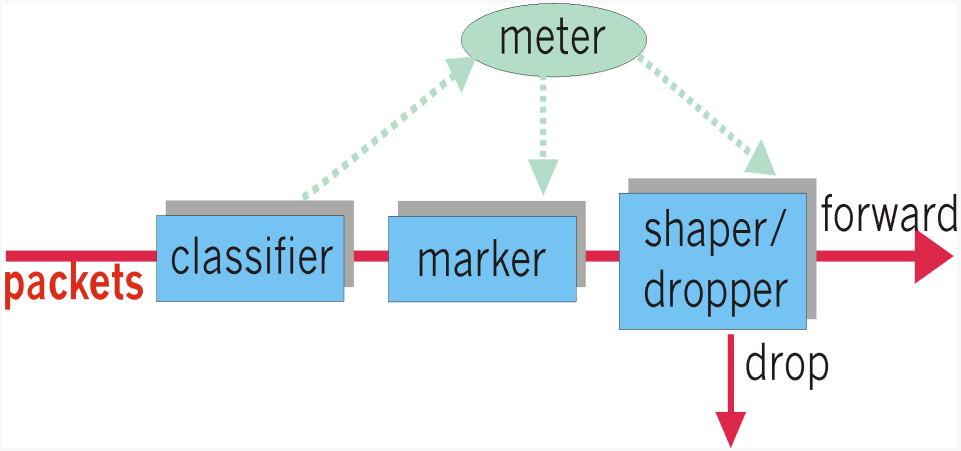
Edge Routers: marcação dos pacotes



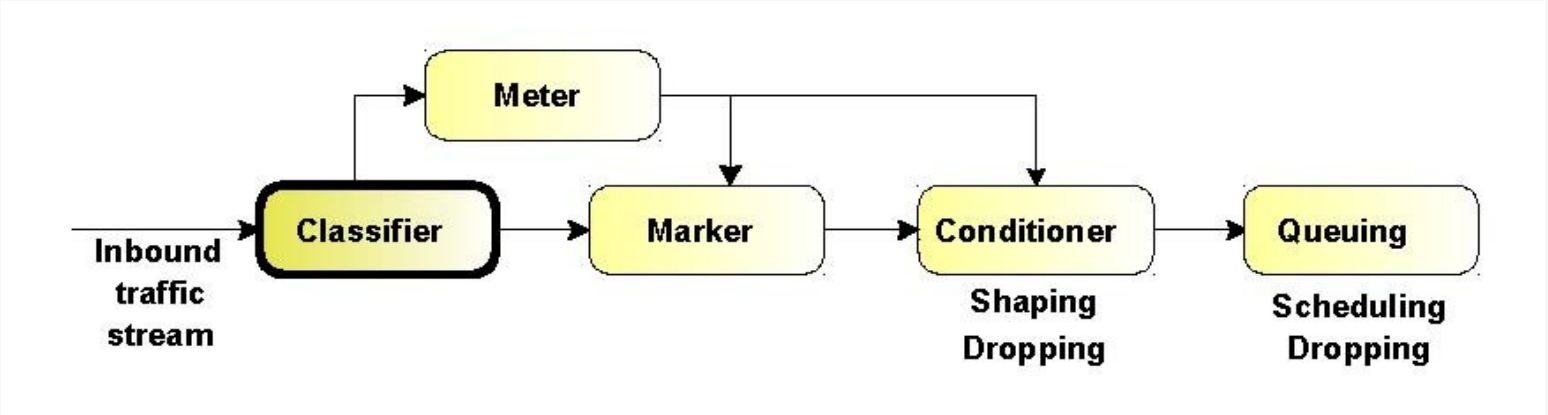
- **Os pacotes são marcados no campo *Type of Service* (TOS) do cabeçalho IPv4 ou *Traffic Class* do cabeçalho IPv6**
  - **DSCP – *Differentiated Service Code Point***
  - **CU – *Currently Unused***

Egde Routers: Classificação e condicionamento de tráfego

- ***Classifier***: **identifica a classe de tráfego a que pertencem os pacotes**
- ***Marker***: **preenche o campo ToS dos pacotes com o DSCP apropriado**
- ***Shaper***: **atrasa os pacotes dos fluxos por forma a estarem de acordo com os SLAs (*Service Level Agreement*) respectivos**
- ***Dropper***: **elimina alguns pacotes dos fluxos por forma a estarem de acordo com os SLAs respectivos**
- ***Meter***: **mede as características dos fluxos de pacotes para determinar se estão de acordo com os SLAs respectivos**

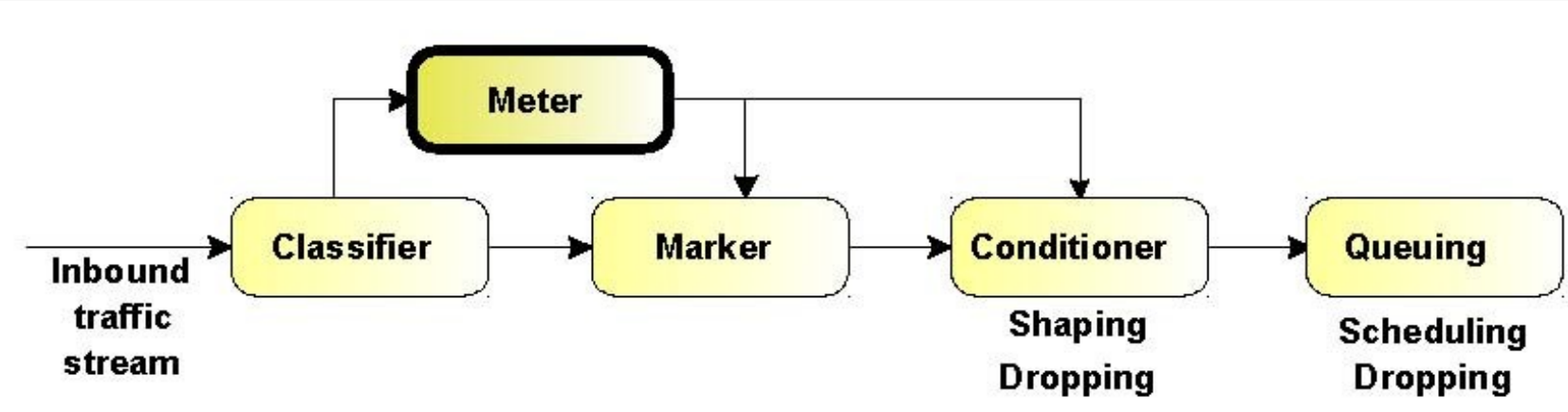


Classificação



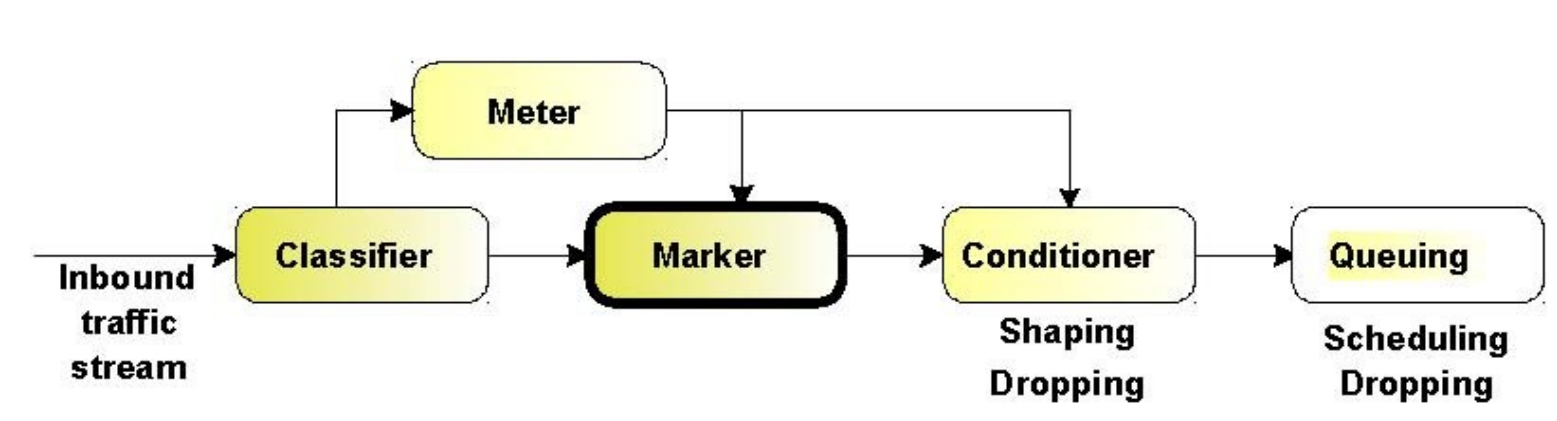
- **Muitos mecanismos de QoS tradicionais incluem classificadores intrínsecos**
  - **Committed Access Rate (CAR)**
  - **Propagação de políticas de QoS via BGP (QPPB)**
  - **Route-maps**
  - **Mecanismos de Queuing**
  - **...**

Monitorização



- **O modelo Token Bucket é usado na monitorização**
  - **Committed Access Rate (CAR)**
  - **Generic Traffic Shaping (GTS)**
  - **Frame Relay Traffic Shaping (FRTS)**
  - **Class-based Weighted Fair Queuing (CB-WFQ)**
  - **Class-based Low Latency Queuing (CB-LLQ) (Cisco-based – inclui prioridades)**
  - **Class-based Policing**
  - **Class-based Shaping**
  - **IP RTP Prioritization**

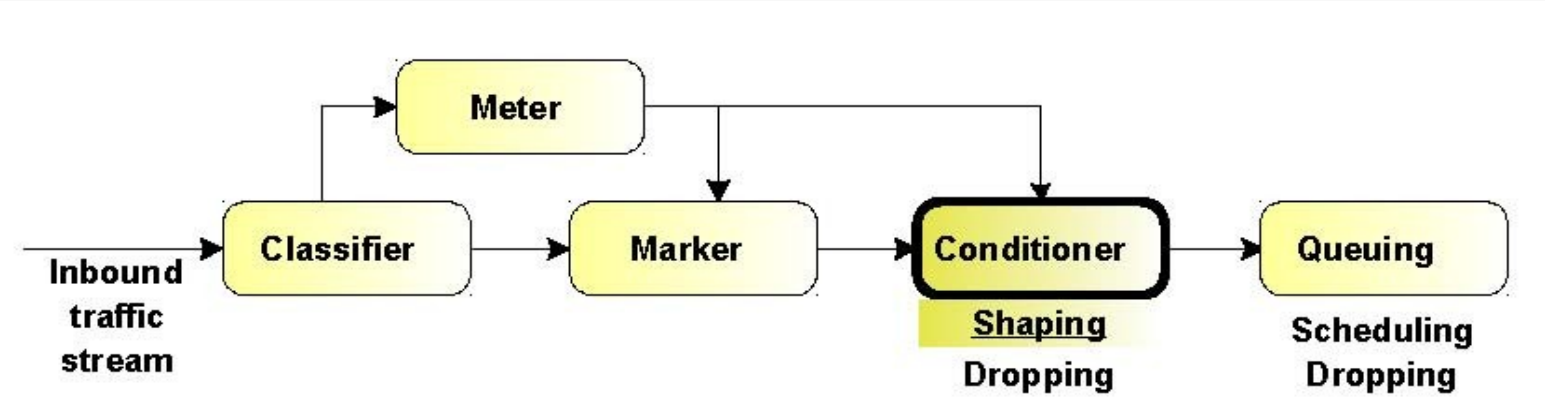
Marcação



- **A marcação é usada para estabelecer:**
  - **IP precedence**
  - **DSCP**
  - **QoS group**
  - **MPLS experimental bits**
  - **Frame Relay DE bit**
  - **ATM CLP bit**
  - **IEEE 802.1Q or ISL CoS**

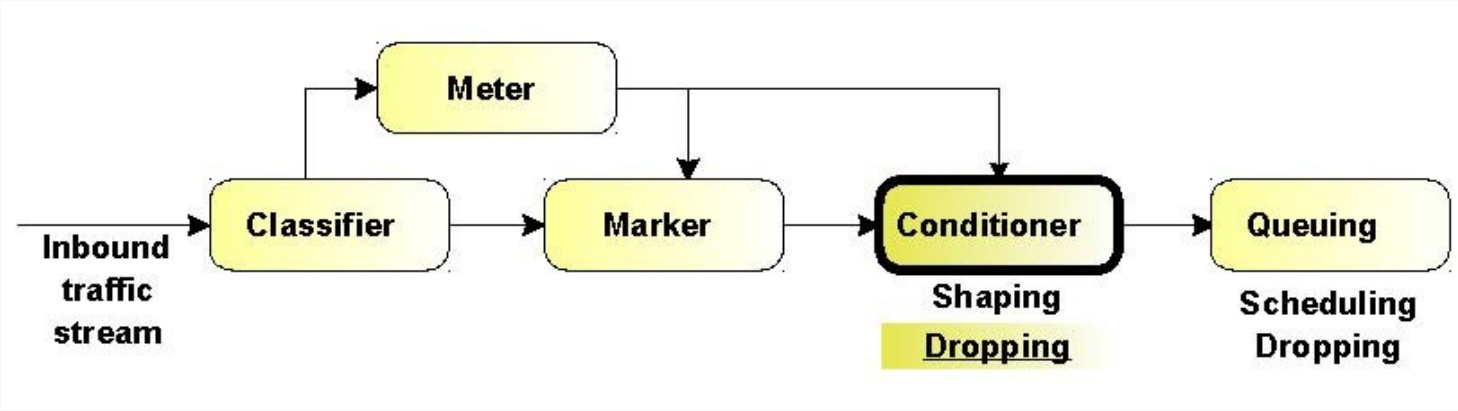
- **Mecanismos de marcação:**
  - **Comitted Access Rate (CAR)**
  - **QoS Policy Propagation**
  - **through BGP (QPPB)**
  - **Policy-based Routing (PBR)**
  - **Class-based Marking**

Condicionamento



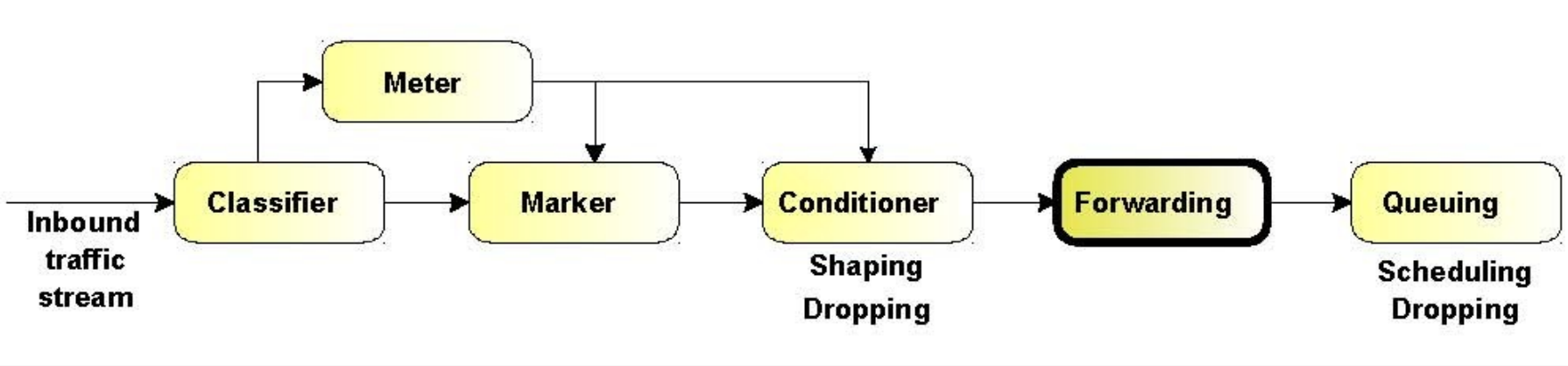
- **Mecanismos de *Traffic Shaping*:**
  - **Generic Traffic Shaping (GTS)**
  - **Frame Relay Traffic Shaping (FRTS)**
  - **Class-based Shaping**
  - **Hardware shaping on ATM VC**

Condicionamento



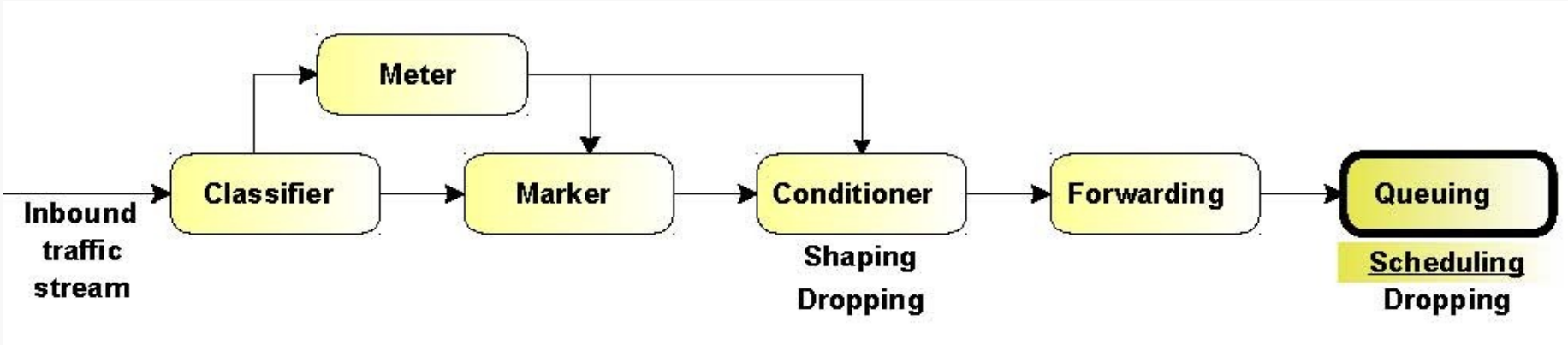
- **Mecanismos de *Dropping*:**
  - **O Committed Access Rate (CAR) e Class-based Policing podem descartar os pacotes que excedem a taxa contratualizada**
  - **O Weighted Random Early Detection (WRED) pode descartar pacotes aleatoriamente quando um interface se aproxima do nível de congestionamento**

Encaminhamento



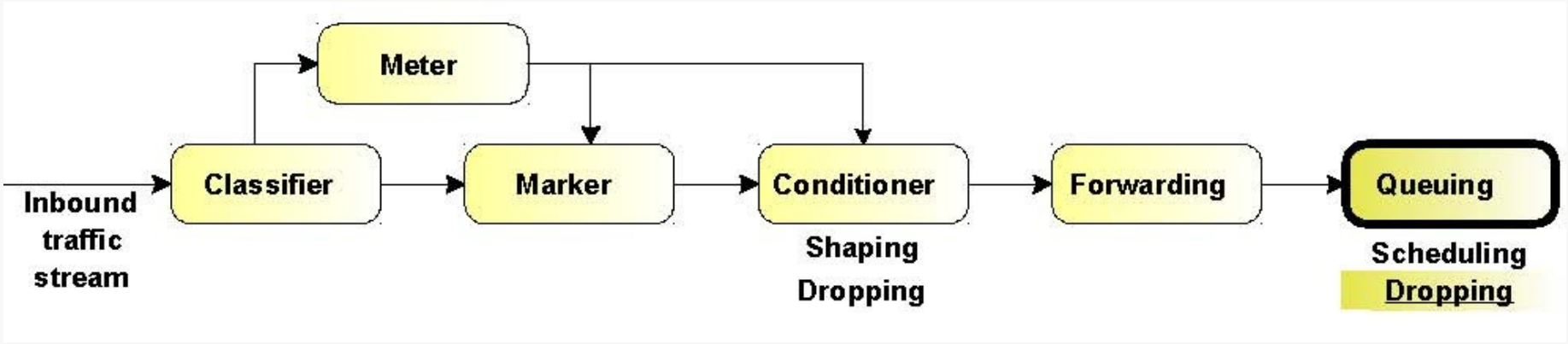
- **Mecanismos de encaminhamento**
  - **Routing**
  - **e.g. Cisco Express Forwarding (CEF)**

Queuing



- **Mecanismos de queuing tradicionais**
  - **FIFO, Priority Queuing (PQ), Custom Queuing (CQ)**
- **Família Weighted Fair Queuing (WFQ)**
  - **WFQ, dWFQ (distributed, implementação de um WFQ mas num Versatile Interface Processor (VIP)), CoS-based dWFQ, QoS-group dWFQ**
- **Mecanismos de queuing avançados**
  - **Class-based WFQ, Class-based LLQ**

Queuing



- **Mecanismos de *dropping***
  - ***Tail drop* quando ocorre congestionamento da fila de espera**
  - **WFQ possui um esquema melhorado de *tail-drop***
  - **WRED descarta os pacotes aleatoriamente quando a fila de espera se aproxima do congestionamento**

Encaminhamento nos *Core Routers*

↳ Os Edge Routers também fazem isto...

- **Diferentes PHBs (*Per-Hop-Behaviors*) resultam em diferentes desempenhos da rede que podem ser observáveis**
- **Os PHBs não especificam que mecanismos de atendimento às filas de espera devem ser usados**
- **Exemplos de PHBs:**
  - **Aos pacotes da Classe A é atribuído x% da largura de banda da ligação física durante qualquer intervalo de tempo de uma duração especificada**
  - **Os pacotes da Classe A são sempre servidos primeiro que os pacotes da classe B**
  - **Os pacotes da Classe A são servidos com o dobro da largura de banda de serviço dos pacotes da Classe B**

Definir para 1 serviço

Per-hop-behaviors (PHBs)

- **PHB por omissão**
  - **Serviço *best effort* tradicional**
    - **Valor de DSCP (recomendado) é de "000000"**
- **Class-Selector PHBs**
  - **Para preservar a compatibilidade com o esquema de precedência do IP (IP precedence)**
    - **Valores DSCP da forma "xxx000"**
  - **Estes PHBs asseguram que nós compatíveis com DiffServ podem coexistir com nós que apenas suportam a precedência IP**
- **Expedited Forwarding (EF) PHB**
  - **Proporcionam um serviço de baixas perdas, baixa latência, baixo *jitter* e largura de banda assegurada**
    - **Valor de DSCP recomendado é de "101110"**
- **Assured Forwarding (AF) PHB**
  - **Pode oferecer diferentes garantias: por exemplo, o tráfego pode ser dividido nas classes ouro, prata e bronze, sendo-lhes atribuídas as percentagens de largura de banda de 50%, 30% e 20%, respectivamente**
  - **O PHB AFxy define quatro classes AFx: AF1, AF2, AF3, and AF4**

Expedited Forwarding

- **O PHB *Expedited Forwarding* (EF):**
  - **Assegura uma taxa mínima de transmissão**
  - **Garante largura de banda – a largura de banda é assegurada através de encaminhamento prioritário**
  - **Efectua o policiamento da largura de banda – a classe não pode ultrapassar a largura de banda garantida (o tráfego em excesso é descartado)**
- **Valor de DSCP: "101110"; idêntico a um valor *IP precedence* de 5 para dispositivos não compatíveis com DiffServ**

Implementações EF PHB

- **Priority Queuing**
- **Prioritização do IP RTP**
- **Class-based Low-latency Queuing (CB-LLQ)**
- **Strict Priority queuing com Modified Deficit Round Robin (MDRR)**

Assured Forwarding

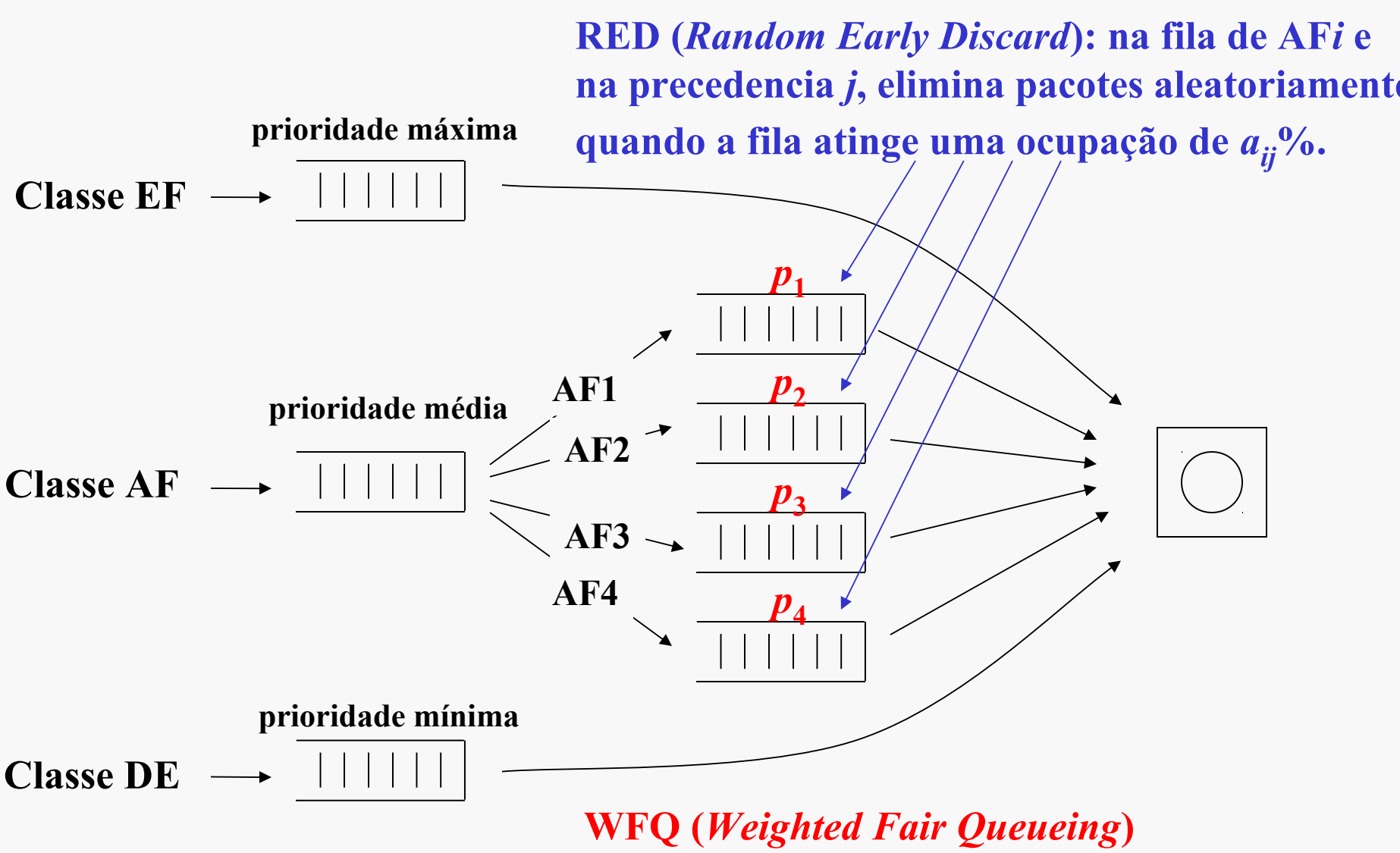
- **O PHB *Assured Forwarding* (AF):**
  - **Garante largura de banda**
  - **Permite acesso à largura de banda extra, se estiver disponível**
- **Quatro classes standard (AF1, AF2, AF3 e AF4)**
- **Valores de DSCP na seguinte gama: “aaadd0”, em que “aaa” é um valor binário da classe e “dd” é a probabilidade de descarte**

Classes de serviço *DiffServ*

- ***Default*** **(DE) → DSCP = 000000**
  - **serviço *best-effort* com uma única fila de espera do tipo FIFO**
- ***Expedited Forwarding*** **(EF) → DSCP = 101110**
  - **serviço tipo "linha alugada virtual"**
  - **disponibiliza controle de perdas, do atraso e da variância do atraso dentro de uma determinada largura de banda máxima**
- ***Assured Forwarding*** **(AF)**
  - **fornece uma Qualidade de Serviço relativa (AF*i* é servido com mais largura de banda que AF*j* para *i*<*j*)**
  - **em cada classe há 3 níveis de precedência para eliminação de pacotes em caso de congestionamento**

AF Codepoints	AF1	AF2	AF3	AF4
Low drop precedence	001010	010010	011010	100010
Medium drop precedence	001100	010100	011100	100100
High drop precedence	001110	010110	011110	100110

Exemplo de Implementação



Definição dos PHBs AF

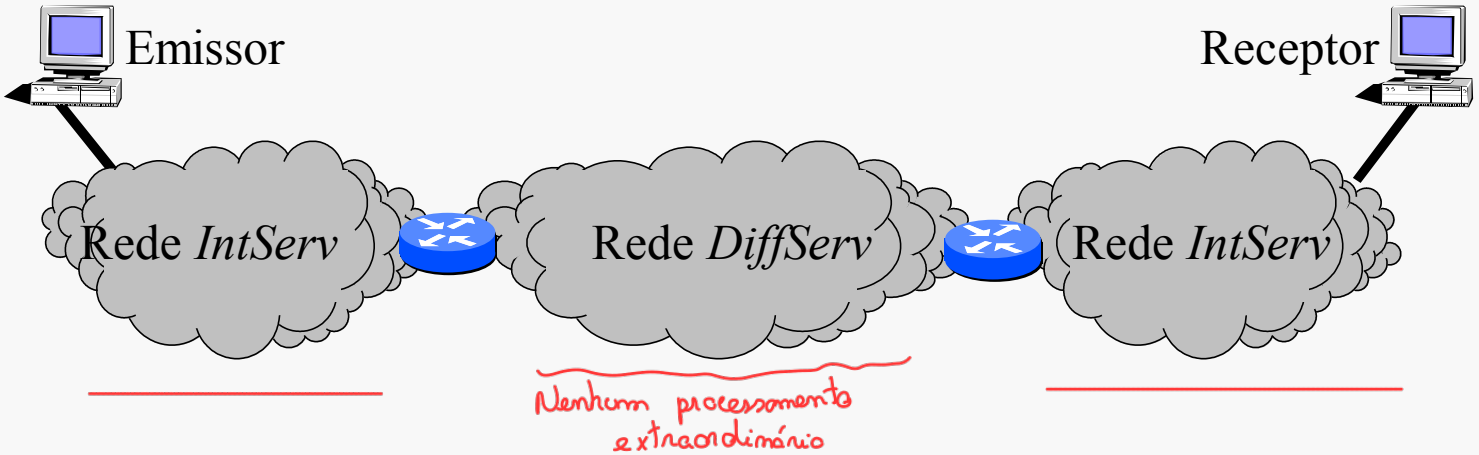
- **Um nó DiffServ deve alocar uma quantidade configurável e mínima de recursos de encaminhamento (espaço de armazenamento e largura de banda) por cada classe AF**
- **Os recursos em excesso poderão ser alocados entre as classes que estiverem activas. A forma de o fazer deve ser especificada.**
- **Não é permitida a reordenação dos pacotes IP do mesmo fluxo se eles pertencerem à mesma classe AF**

Implementação dos PHBs AF

- **CBWFQ (4 classes) com WRED dentro de cada classe**
- **(M)DRR com WRED dentro de cada classe**
- **Opcionalmente, Custom Queuing (não suporta descarte diferenciado)**

Integração *IntServ* e *DiffServ*

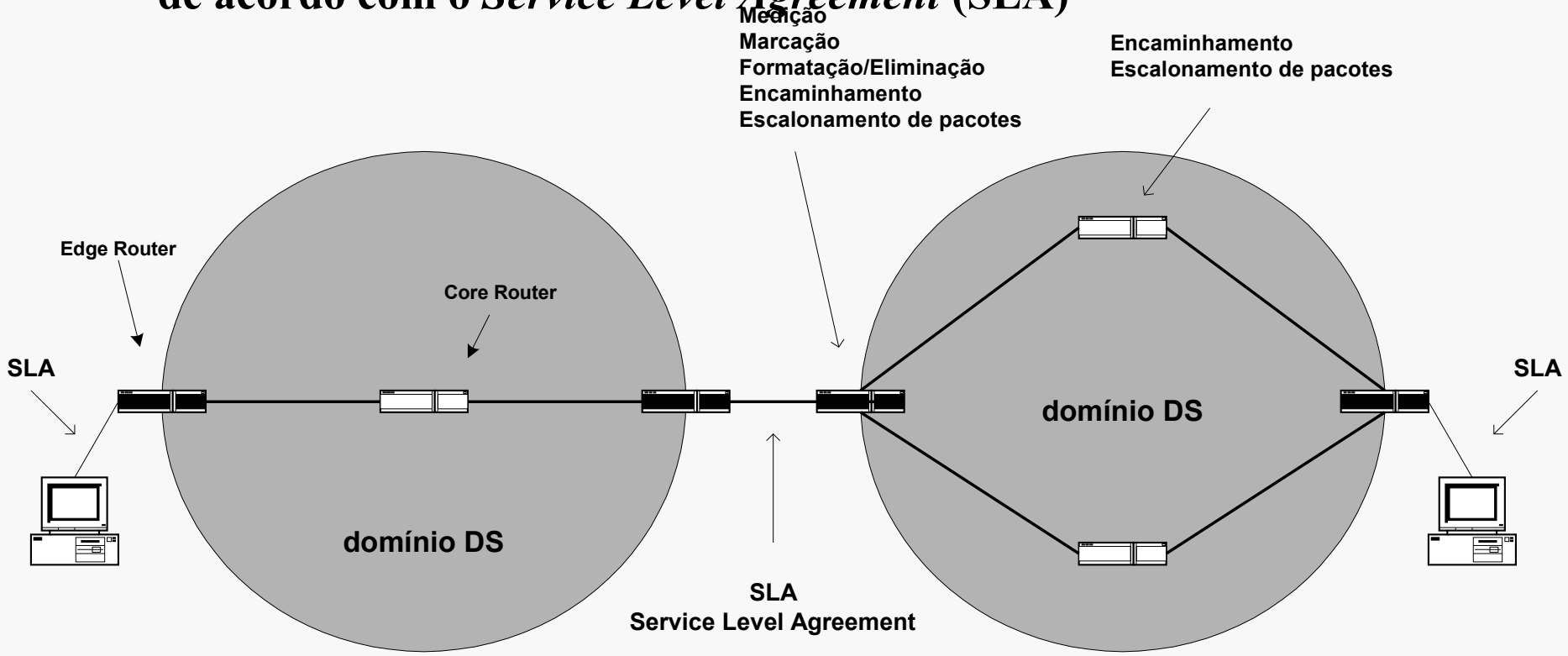
- **Usar:**
  - **a arquitectura *IntServ* (apropriada para redes pequenas) nas redes de acesso**
  - **a arquitectura *DiffServ* (apropriada para redes grandes) na rede de trânsito**
- **Os routers fronteira dos dois tipos de rede:**
  - **classificam os pedidos RSVP nas classes de serviço *DiffServ* apropriadas**
  - **se não houver recursos suficientes, recusam os pedidos de reserva RSVP**
- **Vantagens:**
  - **Proporcionar serviços *IntServ* em grandes redes**
  - **Proporcionar controlo de admissão explícito em vez de SLAs em redes *DiffServ***



Domínios *DiffServ* e SLAs

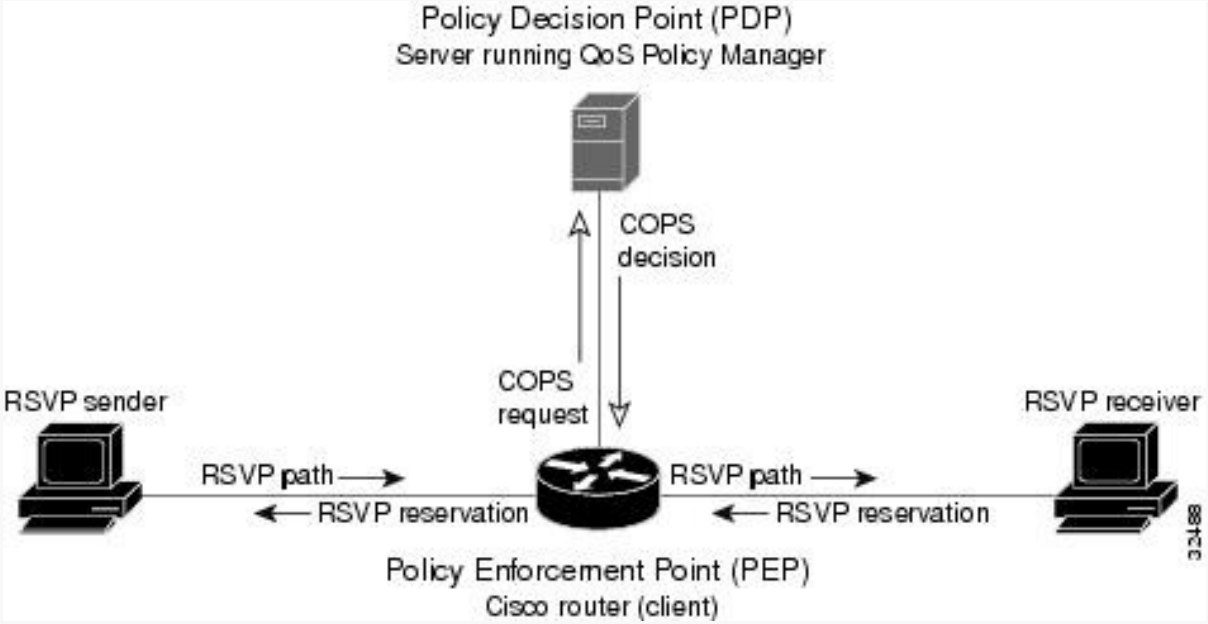
A Qualidade de Serviço proporcionada a um cliente é configurada:

- **por gestão (configuração do condicionamento de tráfego no *Edge Router* respectivo)**
- **de acordo com o *Service Level Agreement* (SLA)**



Common Open Policy Service

- **Common Open Policy Service (COPS) proporciona os seguintes benefícios quando usado com RSVP:**
  - **Gestão centralizada dos serviços**
  - **Controlo centralizado de admissão e autorização dos fluxos RSVP**
- **As soluções de QoS baseadas em RSVP tornam-se mais escaláveis**



COPS para RSVP

- **O servidor PDP contém as políticas de reserva**
- **O router é configurado para solicitar ao servidor as decisões relativas ás mensagens RSVP.**
- **Os fluxos de tráfego são processados pelo router (Policy Enforcement Point - PEP) :**
  - **Quando uma mensagem de sinalização RSVP chega ao router, este pergunta ao servidor PDP como a processar – aceitar, rejeitar, encaminhar, etc**
  - **O servidor PDP envia a decisão para o router, que então processa a mensagem**
- **Alternativamente, é possível configurar o router para tomar as decisões localmente, sem ter que consultar antes o servidor PDP server.**