

Teste06

Qualidade de Serviço (QoS)

Best Effort

IntServ

Problemas

DiffServ

Funcionamento

Classes de Serviço

Arquitetura

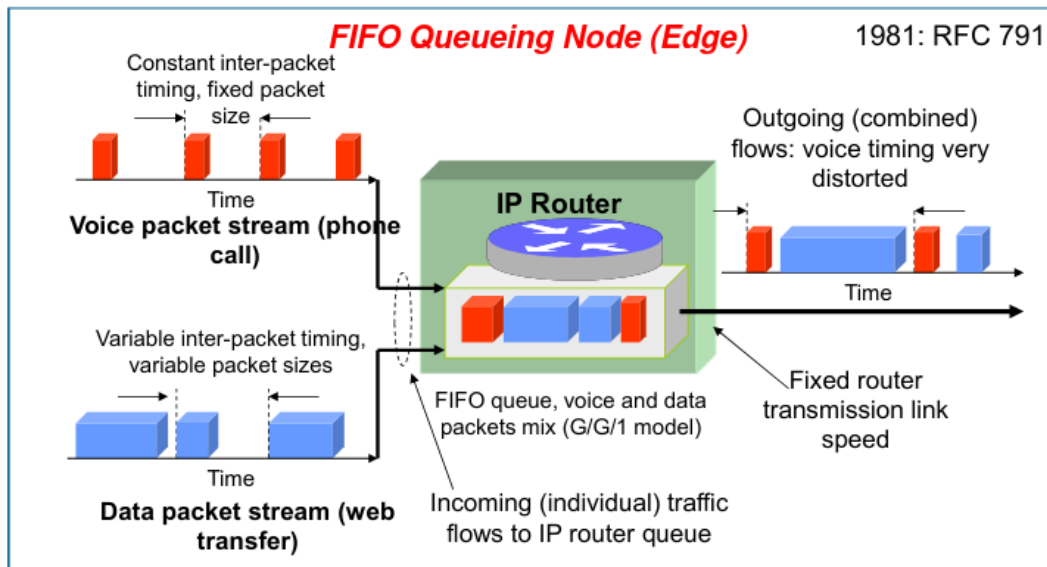
Qualidade de Serviço (QoS)

- Existem 3 serviços que queremos disponibilizar na rede
 - voz
 - video
 - dados
- O caminho da indústria foi usar todos estes serviços no mesmo canal, no entanto para isso foi preciso distinguir o tipo de dados que circula na rede para depois saber o que fazer com os mesmos

Então o problema é, como é que consigo usar estes 3 serviços diferentes na mesma rede????

Best Effort

Modelo Best Effort



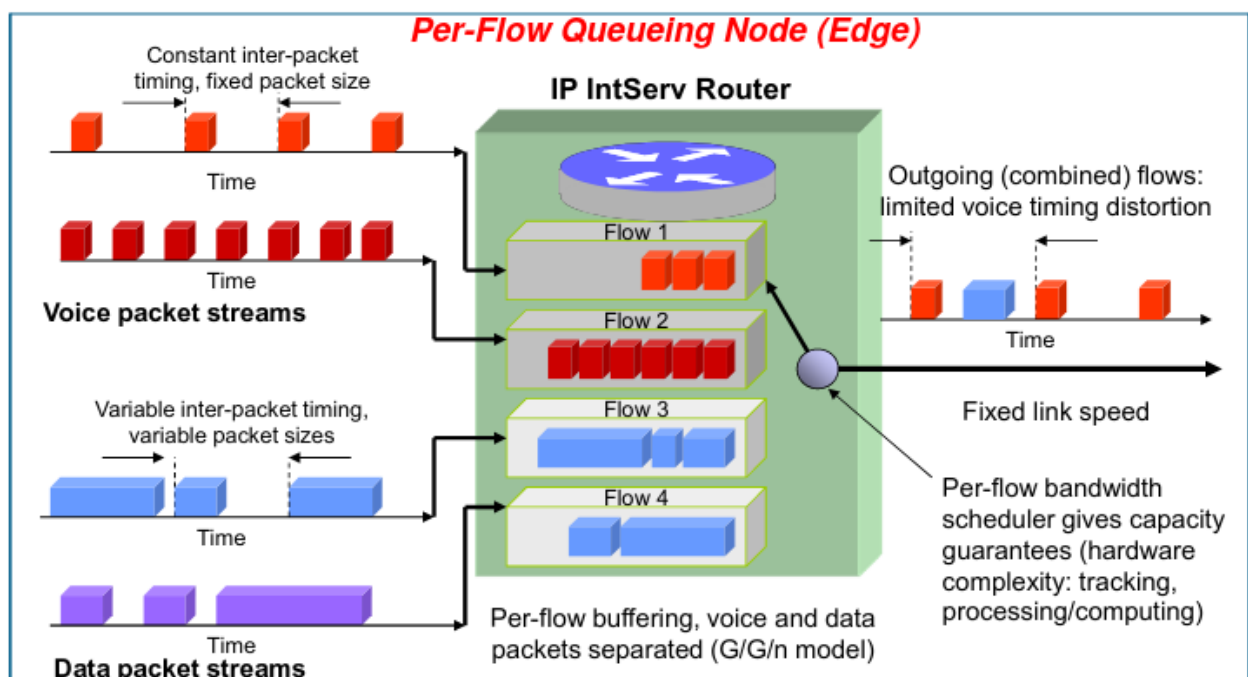
- A internet sempre trabalhou com o *best effort* no entanto esta maneira não permite a integração dos serviços todos
- No *best effort*, a rede IP tenta encaminhar o tráfego o mais rápido possível sem assegurar qualquer garantia quantitativa ao mesmo
- **Por exemplo**
 - Tenho um pacote de voz e depois tenho um pacote com um ficheiro .iso. Os pacotes de voz são então afetados caso o .iso seja demasiado grande
- Uma solução tentada foi a chamada *over-provisioning* em que basicamente tenta-se manter a oferta de largura de banda acima da procura
- Então o IETF criou o IntServ e o DiffServ
 - IntServ → Um fluxo é admitido na rede se esta o puder suportar
 - DiffServ → Um fluxo é marcado para tratamento diferenciado

- É comum o tráfego ser dividido em quatro classes de serviço
 - Gold
 - Tráfego crítico: transações, VoIP, ...
 - Silver
 - Largura de banda garantida: Streaming, Messaging, intranet
 - Bronze
 - Best-Effort // Default class : Internet browsing, E-Mail
 - Less-than-Best-Effort
 - Opcional, *packet drop* muito provável: MySpace, YouTube, Kazaa, FTP, ...

IntServ

- O **IntServ** não consegue ser implementado como foi pensado, ou seja, só é bom no papel uma vez que é restrito

Modelo IntServ: Gestão das filas de espera internas



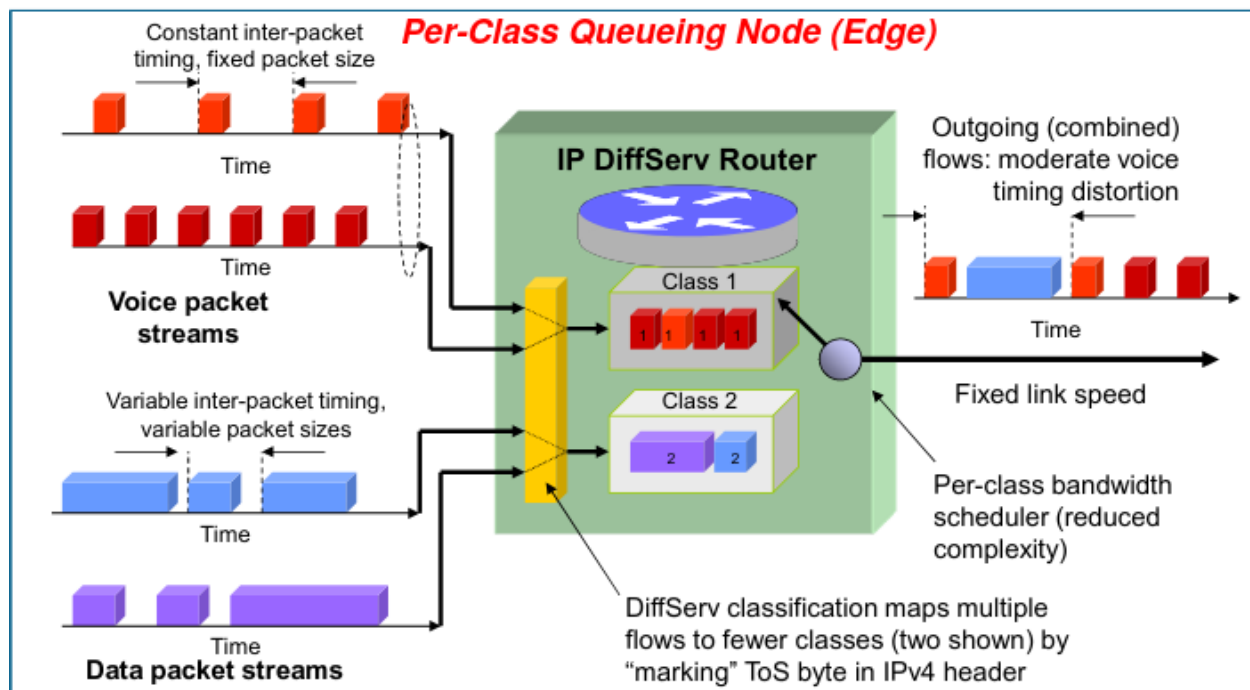
Problemas

- **Problemas e Fragilidades**
 - Necessidade de guardar (RAM) e gerir (CPU) estado em cada router do caminho por cada fluxo que atravessa a rede
 - Complexidade de implementação ao nível do hardware
 - Complexidade ao nível do software (RSVP)
 - Latência adicional na inicialização de cada sessão
- **Estado actual**
 - Modelo praticamente abandonado.
 - Protocolo RSVP resuscitado e adaptado para apoiar o *Traffic Engineering* do modelo DiffServ e do MPLS

DiffServ

- O **DiffServ** na teórica e na prática funciona uma vez que é menos restrito. Em que basicamente em vez de se focar nos fluxos, a atenção é posta nas classes de tráfego suportadas

Modelo DiffServ: Gestão das filas de espera internas

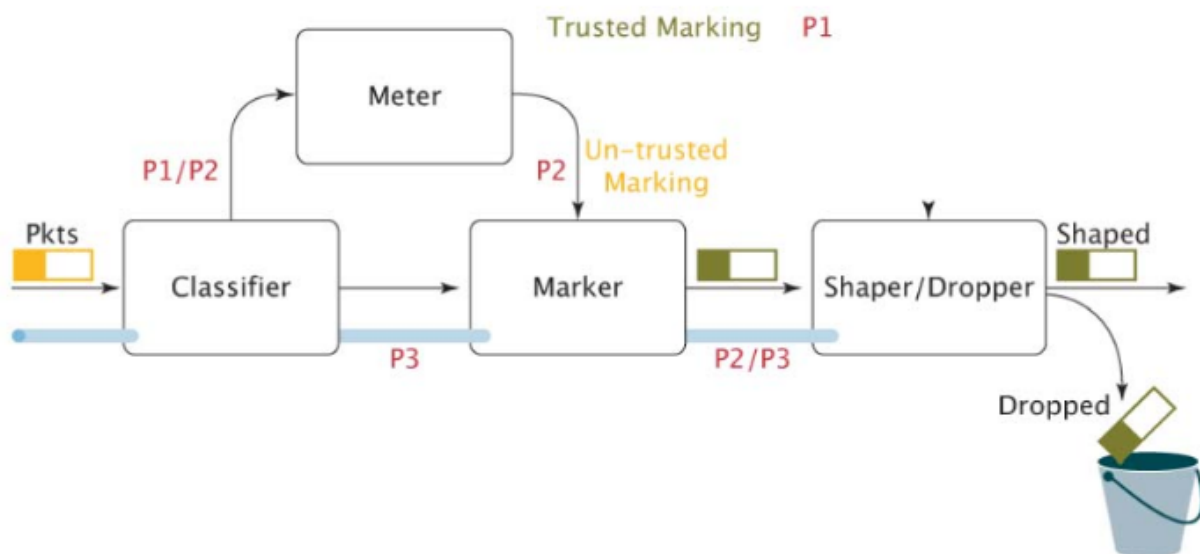


Funcionamento

- Basicamente classifica o pacote
 - "Este pacote é uma chamada de voz" (**Class of Service**)
- Dentro da rede não existe nenhum equipamento a dizer que já está sobrecarregado e não consegue lidar com um certo pacote, mas sim equipamentos configurados para sabere o que conseguem fazer, ou seja, se conseguem lidar com voz, dados ou video.
- Então, o segredo não é não deixar entrar na rede, mas sim marcar trafego com aquilo que a rede tolera
- Para isto temos de configurar a rede, testamos, se não houver problema está tudo bem.
 - Se algum cliente se queixar, vemos onde está o problema, se precisarmos de mais acesso para voz, configuramos então o equipamento responsável pela VOZ.

- **Per Hop Behaviour** → a complexidade passa a ser indexada ao número de classes e não ao número de fluxos que atravessam a mesma
- Existem então filas na entrada, em que cada fila representa um dos 3 serviços
- Depois existe um router/switch que classifica o serviço do pacote que quer entrar e depois o equipamento responsável por certo serviço, recebe e trabalha-o
- Vai haver também uma fila de "urgências" que serve o tráfego mais prioritário

- *Per-Hop Behaviour (PHB)*

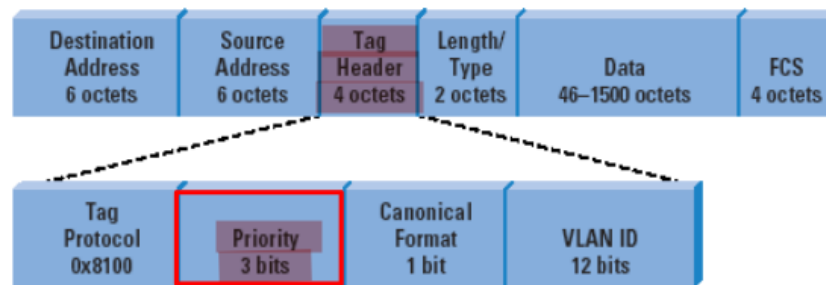


- **Classificador**
 - Recebemos um pacote e temos de classificar em que situação estamos
 - Se o pacote já vem marcado e confiamos nessa marcação
 - Se não confiamos na marcação com que vem e queremos que seja marcado segundo a política implementada no equipamento
 - Se não vem marcado e temos de marcar
- **Meter**

- Temos um contador que diz qual é o ritmo de entrada de tráfego para sabermos se já entrou mais do que aquilo que é previsto na rede
- É aqui que nós podemos aumentar ou diminuir a quantidade de tráfego que queremos na nossa rede
- **Marker**
 - Só depois dos passos anteriores é que vamos marcar segundo a classificação atribuída
- **Shaper/Droper**
 - Aqui vemos o que fazer com o pacote quando é para o mesmo ir para a fila
 - Podemos aceitar o tráfego metendo-o na fila de espera respetiva
 - Se a fila estiver cheia
 - Podemos escolher para o mesmo ir para o lixo
 - Ou mandamos um pacote dessa fila para o lixo e metemos o novo no fim da fila
- **Normalmente os switches da camada de acesso é que têm o trabalho da marcação do tráfego, todos os outros equipamentos da rede confiam neste tráfego uma vez que é marcado por um switch importante**
- Esta marcação é feita no quadro/pacote
- **Funciona em L2 (3 bits de marcação) e L3 (6 bits de marcação)**

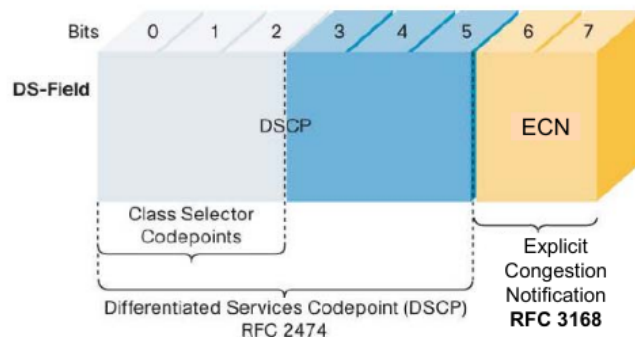
- **Nível 2**

- **Campo Priority Code Point (PCP)** de 3 bits integrado na VLAN Tag
- Proposto em 1998 pelo grupo de trabalho 802.1p para QoS L2
- Utilização especificada na norma 802.1D-1998 (hoje 802.1D-2004)



- **Nível 3**

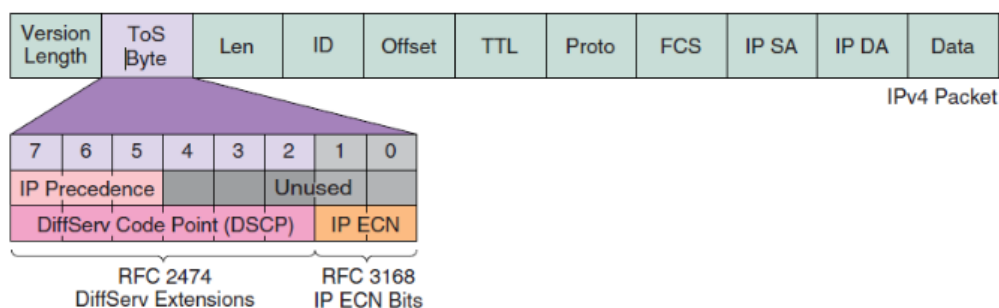
- Actualmente: *Differentiated Services Code Point (DSCP)*



Qualquer rede pode usar $2^6=64$ combinações diferenciadoras livremente mas é boa prática aderir às sugestões do IETF para uma utilização mais consensual.

- **Nível 3**

- Actualmente: *Differentiated Services Code Point (DSCP)*



Classes de Serviço

PCP	Network priority	Acronym	Traffic characteristics
1	0 (lowest)	BK	Background
0	1	BE	Best Effort
2	2	EE	Excellent Effort
3	3	CA	Critical Applications
4	4	VI	Video, < 100 ms latency
5	5	VO	Voice, < 10 ms latency
6	6	IC	Internetwork Control
7	7 (highest)	NC	Network Control

Os tais 3 bits da prioridade em L2 permitem classificar 8 coisas

Arquitetura

- A rede é organizada em domínios DiffServ
- Um domínio DS é composto por

- **DS Boundary Nodes**

- DS Ingress Nodes, DS Egress Nodes
- Constituem Ponto de Demarcação
- Podem interligar dois ou mais domínios DS
- Efectuam Acondicionamento de Tráfego:
 - Marcação/Remarcação, Modelação (*Shape*), e Rejeição (Drop) em caso de congestão

Bandwidth Broker:
 Agente que conhece as prioridades e políticas empresariais de tratamento de tráfego ([RFC 2638](#))



- **DS Interior Nodes**

- Constituem o núcleo do domínio DS
 - Escalonamento (*Schedule*) e Política de Rejeição (*Drop*): Assegurar SLAs
- Operam sobre agregados de tráfego



Service Level Agreements