Administração de Sistemas (ASIST)

Gestão de tráfego

Novembro de 2014

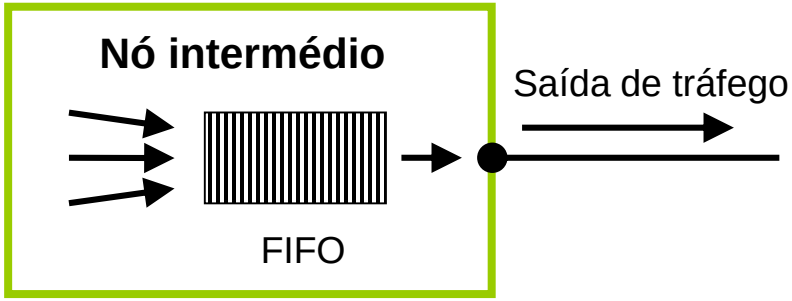
Tratamento diferenciado do tráfego de rede

As ligações de rede são recursos de capacidade limitada, partilhadas por um grande número de aplicações. O administrador deve assegurar-se de que as aplicações importantes para o negócio não são prejudicadas por outros tipos de tráfego na rede.

A saturação da rede conduz a tempos de resposta e perdas de dados nos serviços da organização fora dos parâmetros estabelecidos e constituem por isso falhas. O controlo do tráfego de rede é por isso um elemento fundamental da segurança.

O local ideal para exercer controlo sobre o tráfego é nas filas de saída dos nós intermédios, é nesse local que se produzem os congestionamentos quando a taxa de chegada de dados para emissão na interface de saída é superior à taxa de emissão dessa interface.

Sem tratamento diferenciado, todo o tráfego é tratado da mesma forma ("best-effort"), ficando a aguardar "pela sua vez" numa fila com gestão FIFO ("First In, First Out"). A fila atua como "amortecedor" acumulando dados quando existem picos de volume de entrada.



Aplicações de rede interativas – atraso na rede

Algumas aplicações são particularmente sensíveis aos atrasos na rede, normalmente designadas aplicações de tempo real. Encabeçadas pela transmissão interativa de voz e imagem, mas incluem também aplicações de acesso remoto ao ambiente de trabalho e mensagens instantâneas.

O atraso na rede ("latency") e as variações desses atrasos ("jitter") a par com as taxas de perdas de pacotes são extremamente restritivas, por exemplo numa comunicação de voz VoIP ("Voice over IP") qualquer atraso superior a 250 ms será incómodo para os utilizadores, exigindo-se tipicamente atrasos inferiores a 150 ms.

Os atrasos na rede derivam de dois factores:

- atraso de propagação: depende da distância pois a velocidade de propagação do sinal não pode ser alterada (atraso = distância / velocidade de propagação)

- atrasos nos nós intermédios: tipicamente são nós de comutação de pacotes a funcionar em modo "store & forward", como acontece em todos os "routers" e na maioria do comutadores atuais. Na melhor das hipóteses o atraso é igual ao tempo de transmissão (atraso = tamanho / taxa de transmissão). Se a saída estiver congestionada é necessário adicionar-lhe o tempo de espera na fila.

“Real-time Transport Protocol” (RTP)

O RTP foi desenvolvido como mecanismo de transporte adequado para aplicações de rede interativas, especialmente para transmissão de áudio e vídeo (“streeming”). Pode funcionar sobre TCP ou UDP, mas tipicamente usa UDP porque não necessita das funcionalidades de controlo de erros e fluxo do TCP que iriam aumentam a latência.

O RTP assegura a transferência de dados, em fluxo de tempo real entre dois nós finais. Cada pacote RTP transporta um número de sequência que permitem a sua reordenação e a deteção de perda.

Além do número de sequência transporta uma etiqueta data/hora que permite a sincronização. Ao contrário do TCP, o RTP não implementa controlo de erros, para muitas aplicações de tempo real essa funcionalidade não é relevante.

O RTP opera em conjunto com um protocolo auxiliar, o RTCP (“RTP Control Protocol”), permite a troca de informações de controlo entre os nós e vigia a qualidade do serviço. O RTCP permite ainda a sincronização entre fluxos RTP.

Redução do atraso na rede

Os protocolos de aplicações interativas têm algumas formas de reduzir os atrasos na rede, o atraso de propagação não é um parâmetro que possa ser alterado a não ser pela escolha das ligações de rede que são usadas, evitando por exemplo ligações via satélite.

Pelo contrário, o atraso de propagação nos nós intermédios pode ser melhorado através de:

- redução do tamanho dos pacotes (atraso = tamanho / taxa de transmissão)
- gestão de filas com prioridades, abandonando a gestão FIFO e permitindo que os pacotes prioritários ultrapassem na fila os menos prioritários.

A redução do volume de dados (informação útil) em cada pacote cria um problema de "overhead" (overhead = informação útil / informação total transmitida), penalizando fortemente a eficiência.

Por exemplo o cabeçalho IPv4 e o cabeçalho TCP somam um total de pelo menos 40 bytes (20+20), por isso sempre que forem enviados menos de 40 bytes de informação útil no pacote (segmento TCP), resulta um "overhead" superior a 50 %.

Compressão de cabeçalhos ("header compression")

No sentido de evitar o aumento de "overhead" com a redução do volume de dados em cada pacote foram desenvolvidos vários mecanismos de compressão de cabeçalhos. Permitem obter taxas de compressão acima de 90%, podendo reduzir um cabeçalho IP/TCP a apenas 4 bytes.

CTCP ou "VJ Compression" (RFC1144) – efectua a compressão de cabeçalhos IPv4 e TCP, desenvolvida por V. Jacobson.

IPHC (RFC2507) – "IP Header Compression" – permite a compressão de cabeçalhos IPv4, IPv6, UDP, TCP e outros. Opera normalmente ao nível de ligações entre nós intermédios.

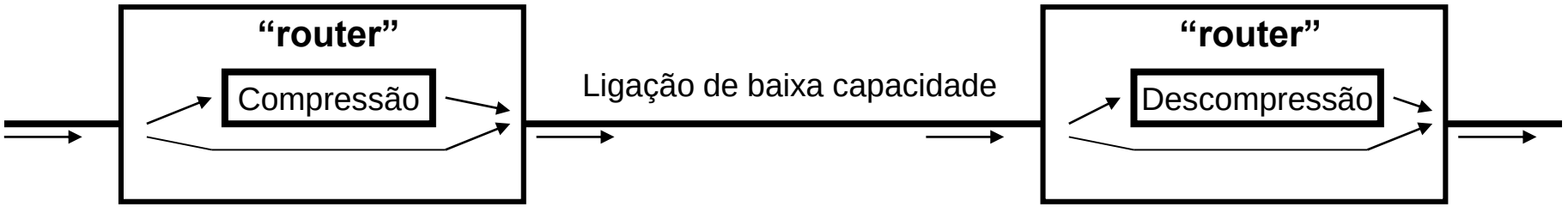
CRTP (RFC2508) – comprime cabeçalhos IP; UDP e RTP, é aplicada entre nós intermédios quando a respetiva ligação tem baixa capacidade.

ROHC (RFC3095) – "Robust Header Compression" - consegue comprimir o conjunto de cabeçalhos IP/UDP/RDP em apenas um byte. Foi desenvolvido para ambientes com elevadas taxas de erros, por exemplo redes sem fios.

Ligações de baixo débito – “header compression”

As ligações com baixa taxa de transmissão introduzem atrasos significativos, é nestas situações que se justifica a aplicação de compressão de cabeçalhos.

A compressão de cabeçalhos não é normalmente aplicada entre nós finais, mas sim entre dois nós intermédios unidos por uma ligação de baixo débito. Os nós finais são alheios a estes procedimentos que são para eles completamente transparentes.



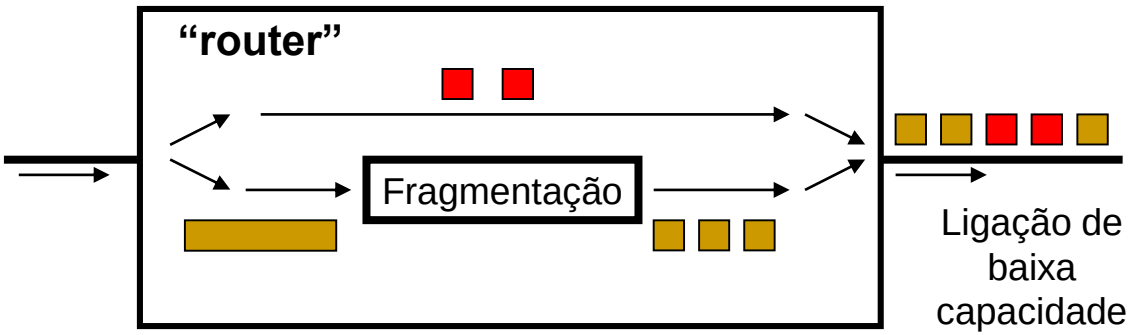
A compressão de cabeçalhos não é aplicada a todos os tipos de tráfego, normalmente apenas o tráfego correspondente a aplicações de tempo real é contemplado. A compressão pode ser unidirecional, não é obrigatório que seja aplicada sempre nos dois sentidos de circulação do tráfego.

Ligações de baixo débito – LFI

A utilização de pacotes pequenos aliada à compressão de cabeçalhos e gestão de fila com prioridade absoluta para o tráfego de tempo real não são ainda suficientes para garantir atrasos reduzidos em ligações de baixo débito.

Os problemas ocorrem quando chega ao nó intermédio um pacote prioritário, mas a interface de saída está já ocupada com a emissão de um pacote longo não prioritário. Tratando-se de uma ligação de baixo débito o tempo de espera até que essa emissão termine pode ser demasiado longo.

A técnica "Link Fragmentation and Interleaving" (LFI) resolve o problema através da fragmentação dos pacotes longos. Os fragmentos são colocados na fila onde não são obstáculo para o tráfego de tempo real.



O nó intermédio que recebe os fragmentos reconstrói o pacote original tornando o mecanismo transparente para os nós finais.

Controlo de congestionamento de rede

O congestionamento de rede ocorre nas filas de espera dos nós intermédios, como resultado a latência da rede aumenta e podem ocorrer perdas de pacotes caso a memória do nó intermédio se esgote.

A forma mais directa de se evitar o congestionamento é através do controlo de fluxo, mas essa funcionalidade apenas está disponível em protocolos de transporte como o TCP onde apenas é aplicável entre nós finais.

Os protocolos com controlo de erros por retransmissão, como por exemplo o TCP, tendem a agravar ainda mais os congestionamentos na rede.

Quando são enviados dados, o emissor espera um período de tempo pré-determinado pela chegada do respectivo “acknowledge” (ACK), esgotado esse tempo, emite novamente os mesmos dados.

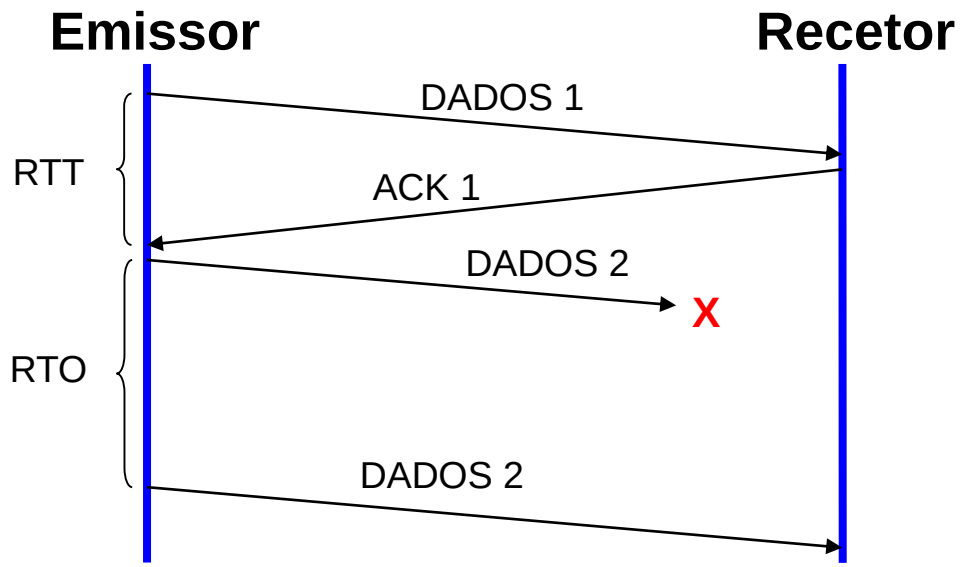
Este comportamento tem efeitos nefastos sobre um nó intermédio que está a ficar congestionado. Quando os dados ficam retidos durante algum tempo na fila, isso provoca a retransmissão dos mesmos pelo emissor, sobrecarregando ainda mais o nó. Existindo várias comunicações deste tipo através do nó intermédio o efeito é devastador.

TCP – Controlo de congestionamento

O protocolo TCP implementa alguns mecanismos com o objetivo de evitar que o congestionamento num nó intermédio seja agravado pelas retransmissões.

Quando um nó emite dados TCP, aguarda que o respetivo ACK seja devolvido. Se após algum tempo o ACK não chega, emite novamente os mesmos dados. O tempo máximo que se aguarda à espera do ACK designa-se RTO ("Retransmission TimeOut").

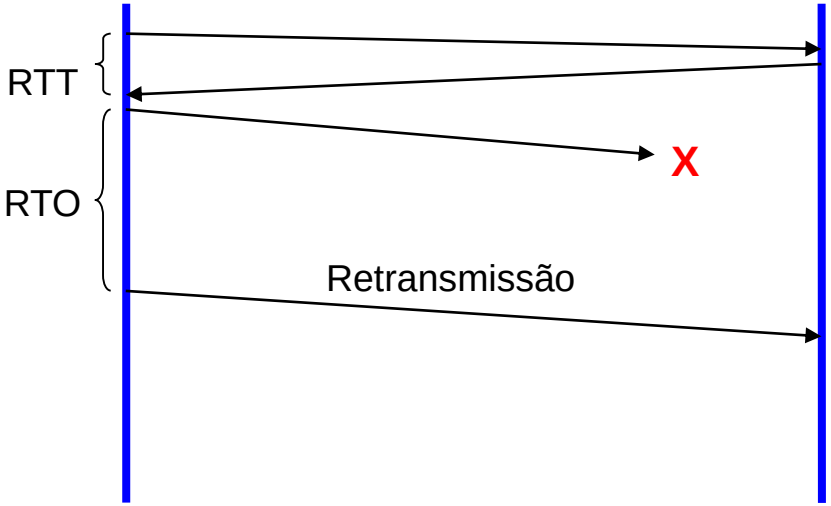
Uma vez que o nó receptor tanto pode estar na mesma rede local como no outro lado do planeta, os RTO terão de ser diferentes caso a caso.

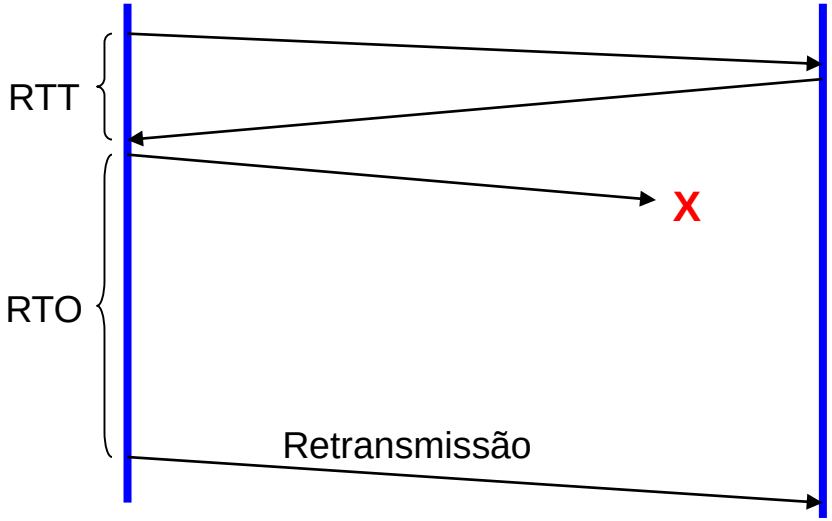


Para adaptar de forma correta o valor do RTO ao estado da rede o TCP usa os valores de RTT ("Round-Trip Time") que vai medindo, ou seja os tempos que decorrem entre a emissão dos dados e a receção do respectivo ACK.

TCP – Cálculo dinâmico do RTO

Cálculo do tempo de retransmissão (RTO) – o RTO é automaticamente adaptado às carateristicas da ligação, começa com o valor de 3 segundos, mas quando o emissor começa a receber os ACK usa o valores do RTT. Começa por calcular RTO = 3 x RTT, usando posteriormente um algoritmo que envolve os vários RTT que vão sendo obtidos, com maior peso para os últimos e tendo em consideração as variações que ele sofre. O RTO mínimo tem o valor de 1 segundo.

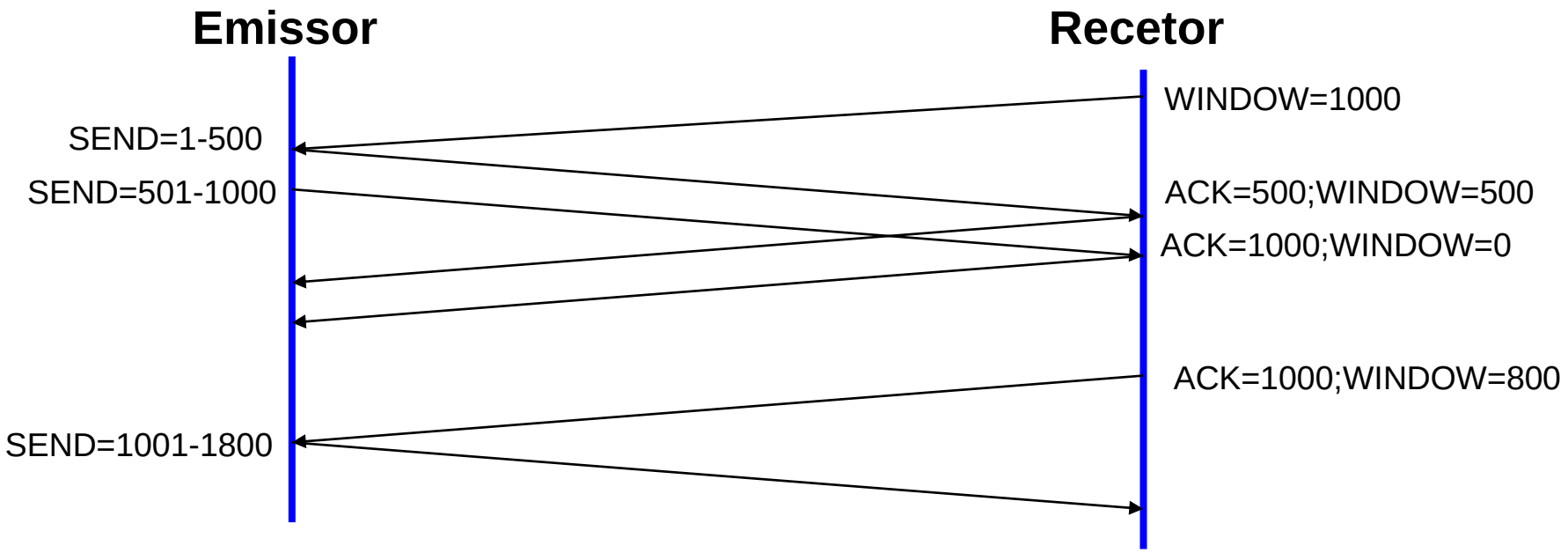




A adaptação automática do RTO evita que os pequenos atrasos na rede desencadeiem a retransmissão. Quando a rede começa a ficar congestionada o RTT aumenta e o RTO é automaticamente atualizado.

TCP – Controlo de fluxo

O TCP utiliza o protocolo da janela deslizante, o recetor anuncia ao emissor o tamanho de janela disponível em número de bytes, esse tamanho de janela de receção indica quantos bytes é possível enviar sem receber o respetivo ACK.



O tamanho de janela é controlado pelo recetor indicando quantos bytes ele está disponível para receber, no exemplo acima, após o envio de 1000 bytes o emissor é forçado a aguardar, desta forma o recetor exerce controlo sobre o fluxo dos dados.

TCP – “Congestion avoidance” e “Slow Start”

O controlo de fluxo por janela deslizante é muito mais eficiente do que o “stop & wait”, em que para cada item de informação enviado tem ser recebido o ACK antes de enviar o item seguinte. Sob o ponto de vista de congestionamento o facto de existir um maior volume de informação a circular na rede sem ACK é contudo uma desvantagem.

Os algoritmos “Congestion avoidance” e “Slow Start” são usados em conjunto pelo TCP para atenuar estes problemas.

O “Congestion avoidance” define uma janela de congestão. Para efeitos de envio de dados é sempre considerado o menor valor entre a janela de congestão, controlada pelo emissor e a janela de receção, comunicada pelo recetor.

A janela de congestão (cwnd) é iniciada com o valor correspondente a um segmento (536 ou 512 bytes), é definido ainda o valor inicial do “slow start threshold size” (ssthresh) como 65535 bytes. Sempre que cwnd < ssthresh o emissor encontra-se num modo designado “slow start”.

Por cada ACK recebido é aumentado um segmento a cwnd. Em condições normais o cwnd atinge rapidamente o valor ssthresh saindo do modo “slow start”.

TCP – Deteção de congestionamento

Para actuar com os algoritmos “Congestion avoidance” e “Slow Start” o TCP necessita de detetar as situações de congestionamento:

- não receção de ACK (esgotamento do RTO)
- receção de ACK duplicados

A receção de um ACK duplicado pode indicar que foi feita uma retransmissão devido a ter sido atingido o RTO, mas que os dados chegaram todos ao destino. Isso aponta claramente para o congestionamento num nó intermédio. Nenhuns dados se perderem, apenas demoraram muito a atravessar a rede e isso desencadeou a retransmissão.

Infelizmente o TCP usa ACK duplos para avisar o emissor de que alguns dados foram recebidos fora de ordem, por essa razão apenas quando são recebidos triplicados, podem ser usados para assinalar claramente um situação de congestionamento.

TCP – “Congestion avoidance” e “Slow Start”

Quando o congestionamento é detetado, ssthresh é redefinido com o valor correspondente a metade do tamanho de janela actual (metade do menor valor entre cwnd e a janela anunciada pelo recetor). O novo valor para ssthresh nunca poderá ser inferior ao tamanho de 2 segmentos.

Adicionalmente se o congestionamento foi detetado pela não receção de um ACK, então o valor de cwnd é redefinido com o tamanho correspondente a um segmento, entrando por isso em “slow start”.

Por cada ACK recebido, cwnd é incrementado no valor de um segmento até atingir ssthresh, nessa altura sai do modo “slow start”. O ssthresh vai sendo ajustado para metade do tamanho da janela.

Depois de sair do modo “slow start”, o valor de cwnd continua a aumentar por cada ACK recebido, mas o valor desse aumento deixa de ser um segmento e passa a ser:

$$\frac{(tamanho - de - segmento)^2}{cwnd}$$

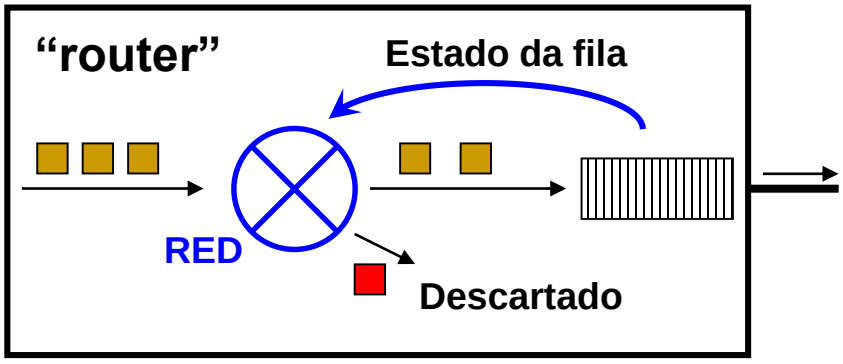
Enquanto os incrementos vão sendo cada vez menores.

RED – “Random Early Detection”

O controlo de congestionamento implementado pelo TCP pode ser aproveitado pelos nós intermédios para controlarem o fluxo e evitarem que as filas fiquem cheias.

Quando a fila de um nó intermédio fica cheia, os pacotes recebidos começam as ser descartados (“tail drop”), isso leva a que todos os emissores TCP entrem em “slow start” degradando de forma excessiva a performance da rede, este fenómeno de sincronização é conhecido por “TCP global synchronization”.

O algoritmo RED é aplicado pelos “routers” para descartar pacotes antes da fila encher completamente e desta forma evitar a sincronização. A redução de fluxo pelos emissores TCP é mais gradual e suave.



O algoritmo RED descarta os pacotes com uma certa probabilidade. Até certo grau de ocupação da fila a probabilidade é zero, pelo que nenhum pacote é descartado, depois começa a aumentar até atingir probabilidade 1 quando a fila fica cheia.

Classificação do tráfego – ACLs e NBAR

Tendo em vista um tratamento diferenciado dos diferentes tipos de tráfego, o primeiro passo é assegurar a sua identificação.

A classificação do tráfego pode ser baseada em critérios semelhantes aos usados para definir regras de filtragem numa ACL: protocolos de transporte; endereços de origem e destino; números de porto de serviço.

Normalmente estes critérios são suficientes permitindo distinguir os vários tipos de aplicação através dos respectivos número de porto de serviço.

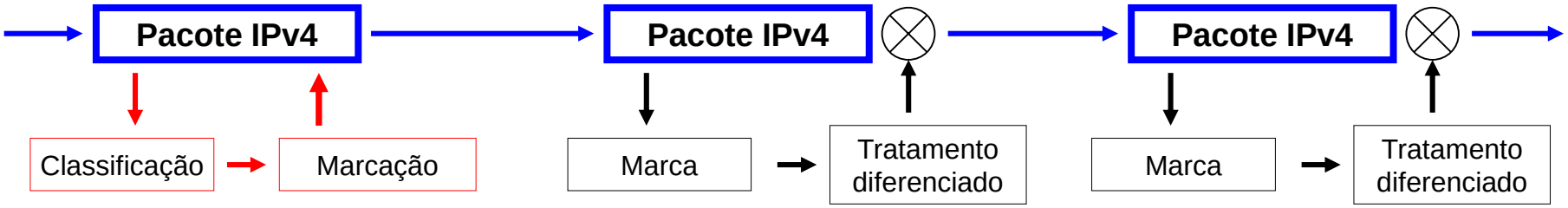
O NBAR ("Network Based Application Recognition") disponível nos "routers" CISCO permite a classificação de tráfego que não pode ser classificado apenas através de uma ACL, por exemplo aplicações que usam números de porto dinâmicos. Pode analisar os conteúdos dos protocolos de aplicação, por exemplo páginas HTML. O NBAR permite ainda interagir directamente com protocolos que implementam parâmetros de QoS, como por exemplo o RTP.

Marcação de pacotes

Nem sempre a classificação do tráfego ocorre no mesmo dispositivo onde o tratamento diferenciado é posto em prática. Frequentemente o tratamento diferenciado de um dado pacote tem lugar em vários pontos da infra-estrutura.

Se no seu percurso um pacote vai passar através de vários pontos de aplicação de tratamento diferenciado, é muito mais eficiente classificar uma única vez que do que repetir esse procedimento de classificação nos vários pontos.

Para esse efeito, após a classificação o pacote tem de ser marcado de forma que seja reconhecido nos vários locais de aplicação de tratamento diferenciado.



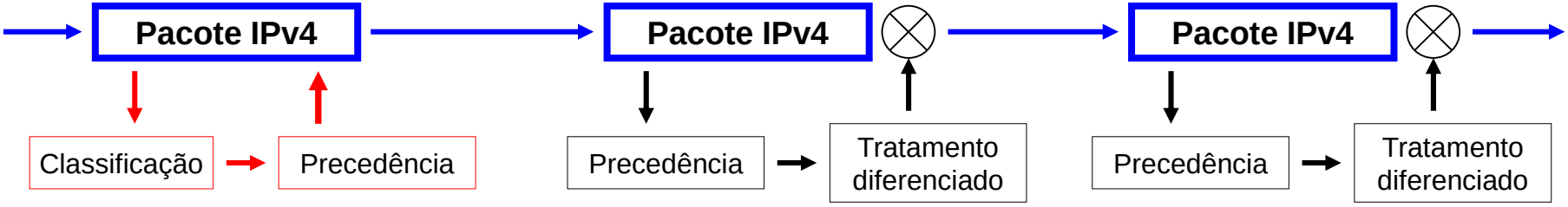
Marcação de pacotes – bits de precedência

O cabeçalho dos pacotes IPv4 possuem o campo de 8 bits designado TOS (“Type Of Service”), o cabeçalho dos pacotes IPv6 possuem o campo de 8 bits designado “Traffic Class”.

Estes campos são usados de diversas formas para marcar os pacotes de acordo com uma classificação do tipo de tráfego.

A forma de marcação mais simples designada “bits de precedência” (IP Precedence) utiliza apenas os 3 bits mais significativos do campo, permite por isso especificar valores numéricos de 0 a 7.

A utilização dos bits de precedência permite que a classificação do tráfego seja realizada num ponto da rede e o respectivo tratamento diferenciado noutros locais.



Campos “Type Of Service” e “Traffic Class”

A forma de utilizar estes campos de 8 bits, respetivamente no IPv4 e IPv6 tem sofrido diversas alterações ao longo do tempo.

Na versão mais simples (IP precedence), a utilização dos 3 bits de precedência pretende definir 7 níveis de prioridade do tráfego, desde 0 (menos prioritário) até 7 (mais prioritário), adicionalmente os 4 bits seguintes serviam para definir carateristicas desejadas no tratamento do pacote e o bit menos significativo teria obrigatoriamente o valor zero:

BIT 7 – IP Precedence (bit 2)

BIT 6 – IP Precedence (bit 1)

BIT 5 – IP Precedence (bit 0)

BIT 4 – “minimize delay” (flag)

BIT 3 – “maximize throughput” (flag)

BIT 2 – “maximize reliability” (flag)

BIT 1 – “minimize monetary cost” (flag)

BIT 0 - "MBZ" (for "must be zero")

Campos “Type Of Service” e “Traffic Class”

Numa utilização prática, os valores de precedência IP podem ter um significado meramente local. Se o administrador responsável pela classificação e marcação dos pacotes é o mesmo que é responsável pela utilização dessa marcação, para implementar o tratamento diferenciado poderá usar valores arbitrários.

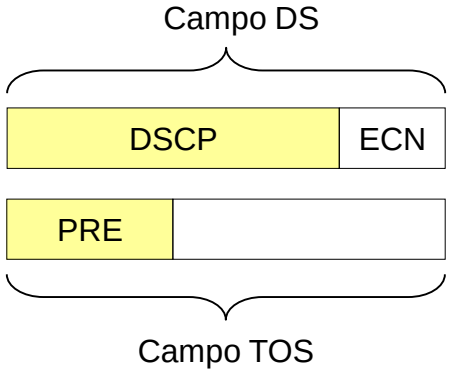
A utilização dos campos “Type Of Service” e “Traffic Class” foi posteriormente revista designando-se atualmente “Differentiated Service” ou DiffServ (DS):

Os bits 7 a 2 constituem um campo de 6 bits designado DSCP (“Differentiated Service Code Point”), os dois bits menos significativos constituem outro campo designado ECN (“Explicit Congestion Notification”).

Os 3 bits mais significativos do campo DSCP correspondem ao campo “IP Precedence”. Teoricamente os 6 bits do campo DS permitem definir 64 níveis de prioridade, contudo é aconselhada a utilização de um conjunto de valores bem definidos que entre outros aspetos garantem a compatibilidade do o campo “IP Precedence”.

DiffServ (“Differentiated Services”) - DSCP

Cada valor do campo DSCP, designado “codepoint”, representa um comportamento pretendido para os nós intermédios perante esse pacote, designado PHB (“Per Hop Behavior”).



Os “codepoints” da forma “XXX000” designam-se “Class Selector Codepoints” e permitem a compatibilidade com o campo “IP Precedence”. Os PHB do tipo “Class Selector” garante a compatibilidade com os bits de precedência IP.

O “default PHB” (codepoint 000000) deve estar sempre disponível, define um tratamento não prioritário (“best-effort”).

DSCP – PHB (“Per Hop Behavior”)

EF PHB (“Expedited Forwarding”) – “codepoint” 101110

Tráfego de alta prioridade, em tempo real: low delay, low jitter, low loss

VA PHB (“Voice Admitted”) – “codepoint” 101100

Semelhante ao EF PHB, no entanto ao contrário do anterior existe um procedimento “Call Admission Control” (CAC) que obriga à autenticação e verificação da disponibilidade dos recursos necessários.

AF PHB Group (“Assured Forwarding”) – define prioridades no encaminhamento de tráfego dividido em quatro classes. Pode por exemplo ser usado pelo RED ao descartar pacotes, nesse caso deverá atuar de forma independente para cada classe.

	Class 1	**Class 2**	**Class 3**	**Class 4**
Low Drop	001010	010010	011010	100010
Medium Drop	001100	010100	011100	100100
High Drop	001110	010110	011110	100110

ECN (“Explicit Congestion Notification”)

O campo ECN é constituído pelos dois bits menos significativos do campo DS (DiffServ), o seu objetivo é notificar os nós finais da existência de um congestionamento num nó intermédio da rede.

O campo ECN pode ter quatro valores possíveis:

00 – “Not ECN-Capable Transport”
01 – “ECN-Capable Transport” - ECT(0)
10 – “ECN-Capable Transport” - ECT(1)
11 – “Congestion Experienced” – CE

Os nós finais que suportam ECN devem colocar neste campo o valor ECT(0) ou ECT(1), nessas condições, em caso de congestionamento, os nós intermédios (“routers”) notificam os nós finais alterando o valor para CE. Por exemplo na aplicação do algoritmo RED, em lugar de descartar o pacote, este pode ser assinalado com CE e retransmitido.

ECN (“Explicit Congestion Notification”) - TCP

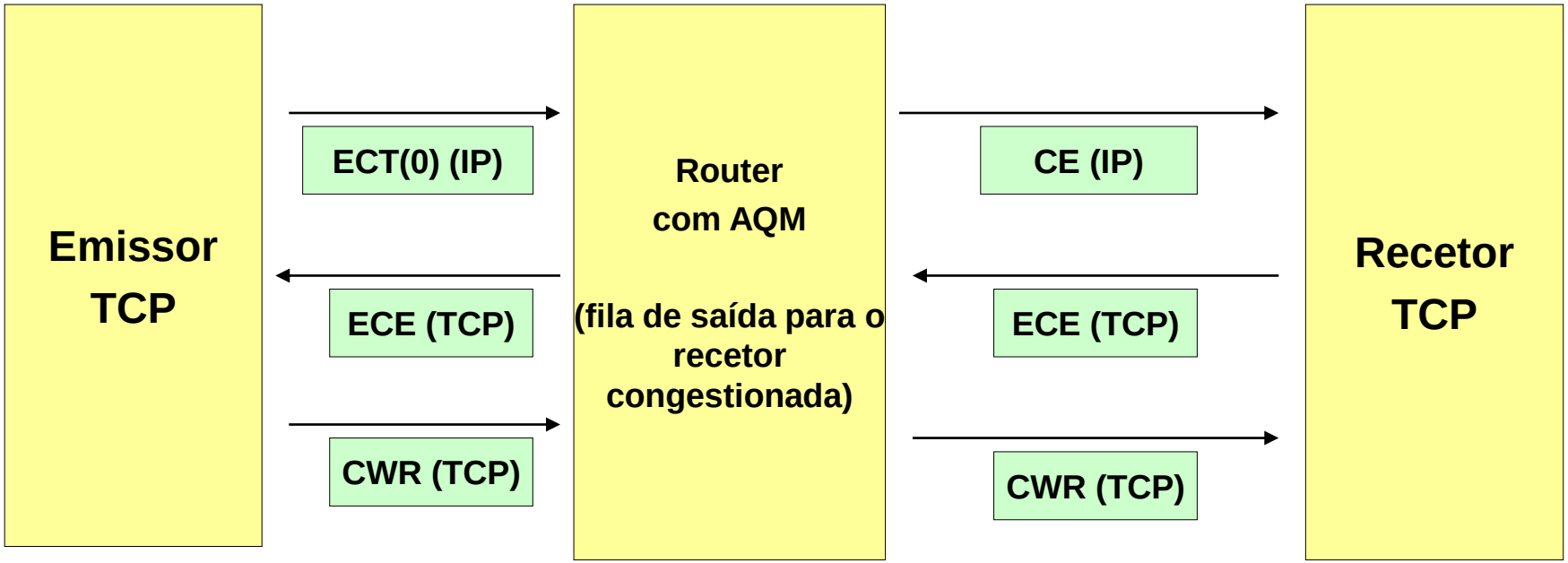
Quando um nó final recebe um pacote com o valor CE no campo ECN deve proceder sob o ponto de vista de congestionamento como se o pacote tivesse sido perdido (no entanto não tem de aguardar pelo RTO).

Nos nós intermédios os pacotes com os valores ECT(0) ou ECT(1) devem ser tratados da mesma forma, ou seja se o nó intermédio implementa um mecanismo de “Active Queue Management” (AQM), como por exemplo o RED, altera o valor para CE em caso de congestionamento.

Este mecanismo inclui o protocolo TCP. Para esse efeito são definidas duas novas “flags” no cabeçalho TCP: “ECN-Echo” (ECE) e “Congestion Window Reduced” (CWR). A “flag” ECE é ativada quando um recetor TCP recebe um pacote IP com o valor CE, informando o emissor que existe um congestionamento no percurso, o emissor posteriormente ativa a “flag” CWR para indicar que recebeu o ECE e agiu em conformidade.

Adicionalmente quando a ligação TCP é estabelecida (SYN) o nó emissor do pedido de ligação deve ativar a “flag” ECE para avisar o recetor que aceita essa funcionalidade.

ECN (“Explicit Congestion Notification”)



Referências bibliográficas

[1] Lhamas A. (2010-2011). Aulas Teóricas de ASIST.

(…)