

Unidade IV – Controle de Congestionamento

TP308 – Introdução às Redes de Telecomunicações

© Antônio M. Alberti 2008

Tópicos

- ✓ Introdução
- ✓ QoS
- ✓ QoS e Controle de Congestionamento
- ✓ Monitoramento de Conformidade
- ✓ Formatação de Tráfego
- ✓ Gerenciamento de Buffer
- ✓ Descarte Seletivo
- ✓ Random Early Detection (RED)
- ✓ Weighted Random Early Detection (WRED)

© Antônio M. Alberti 2008

Introdução

- ✓ O que **significa congestionamento** em uma rede de comutação de pacotes?

É a situação em que o número de pacotes nas filas dos comutadores/roteadores aumenta rapidamente, chegando até mesmo a ultrapassar a capacidade de armazenamento.

- ✓ Quais são as **conseqüências** de um congestionamento?
 - Aumento do atraso.
 - Perda de pacotes.
 - Redução da vazão.

Introdução

- ✓ O controle de congestionamento envolve diversas **ações**, dentre as seguintes:
 - **Evitar** a formação de um congestionamento.
 - **Reconhecer** que uma situação de congestionamento está se formando.
 - **Enviar** um sinal apropriado para os clientes da rede que estão causando o congestionamento.
 - **Reagir** ao congestionamento através do **gerenciamento de buffer** e **descarte de pacotes**.
 - Se necessário, **retransmitir** as informações perdidas durante o congestionamento.

Introdução

(Continua...)

- **Maximizar a eficiência** do uso dos recursos de armazenamento. Para isso é utilizado o **gerenciamento de estrutura de filas** (BM – *Buffer Management*).
- **Distribuir** os recursos de armazenamento de forma justa entre os fluxos ou conexões, de forma que a QoS de cada fluxo seja respeitada.
- **Decidir** quais pacotes devem ser descartados quando a ocupação da estrutura de filas ultrapassa um determinado valor. Para isso é utilizada função de **descarte seletivo** (SD – *Selective Discard*).

Introdução

- ✓ Dentre as técnicas existentes para **reconhecer** que uma **situação de congestionamento** está se formando estão:
 - O monitoramento de limiares de ocupação nos *buffers* da rede.
 - Quando um determinado limiar é ultrapassado, ações são tomadas.
 - A utilização de relógios no transmissor para cada pacote transmitido.
 - Quando o relógio é extrapolado (*timeout*), o pacote é retransmitido e o tamanho da janela deslizante é ajustado.

- Unidade 5 Controle de Congestionamento
- ## QoS
- ✓ QoS é o efeito coletivo de desempenho que determina o grau de satisfação do usuário de um serviço específico.
 - ✓ Qualidade de serviço pode ser definida também como o processo de transmitir informações de maneira satisfatória ou até mesmo superior à expectativa.
 - ✓ Pode ser considerada também sinônimo de previsibilidade do serviço, quando se refere a manutenção de níveis toleráveis de perda de pacotes, atraso, jitter, etc...

- Unidade 5 Controle de Congestionamento
- ## QoS e Controle de Congestionamento
- ✓ O contrato de tráfego é um acordo que define o comportamento do tráfego e o nível de serviço requerido por cada fluxo de tráfego, seja ele individual ou agregado.
 - ✓ Pode fazer parte do acordo de nível de serviço (SLA - Service Level Agreement).
 - ✓ O SLA é um acordo formal (contrato) assinado entre o provedor de serviço e o cliente deste serviço.
 - ✓ Um elemento chave do contrato de tráfego são as classes de serviço que podem ser utilizadas.

QoS e Controle de Congestionamento

- ✓ **Classes de serviço** são usadas para agrupar os pré-requisitos de **qualidade de serviço** e as **características** do fluxo de tráfego que será transportado.
- ✓ O comportamento esperado de cada fluxo de tráfego é especificado através de **descritores de tráfego** específicos para cada classe de tráfego.
- ✓ A QoS desejada por cada fluxo é definida através de **parâmetros de QoS**.
- ✓ O **contrato de tráfego** é utilizado pela **gerência** ou **sinalização** da rede para diversas funções de controle de congestionamento.

QoS e Controle de Congestionamento

- ✓ Para garantir que a QoS seja mantida para todos os fluxos da rede, esta **deve verificar** se o **tráfego submetido** está de acordo com o **negociado** no contrato de tráfego.
- ✓ Esta verificação é feita por **funções de monitoramento de conformidade**.
- ✓ O tráfego do fluxo que **satisfaz** as funções de monitoramento de conformidade é então enviado pela rede, onde deverá passar por vários pontos de multiplexação.

QoS e Controle de Congestionamento

- ✓ Nestes pontos de multiplexação, a comutação de pacotes é utilizada, e **estruturas de filas** (ou *buffers*) são utilizados para armazenar temporariamente os pacotes que aguardam por transmissão.
- ✓ As estruturas de filas são servidas de acordo com um **algoritmo de escalonamento**, que deve ser projetado para **atender** aos pré-requisitos de QoS de todos fluxos.
- ✓ Funções de **controle de congestionamento** são utilizadas para **evitar** e **gerenciar** situações de congestionamento na rede.

QoS e Controle de Congestionamento

- ✓ O controle de **congestionamento/fluxo** **atua reduzindo o tráfego das fontes** a medida que o nível de congestionamento está aumentando.
- ✓ Uma das formas de efetivar este mecanismo é enviando pacotes especiais (*choke packet*) para a fonte **reduzir** sua taxa de transmissão.
- ✓ Outra forma é fazer este **controle** via **piggyback**, usando o cabeçalho de pacotes que trafegam na direção oposta ao congestionamento.

QoS e Controle de Congestionamento

- ✓ O uso de **notificações diretas** avisando da situação de congestionamento também é usada. Um exemplo é a extensão **ECN** (*Explicit Congestion Notification*) existente em redes IP.
- ✓ Para **evitar/reagir** ao congestionamento, são também usadas as funções:
 - **Gerenciamento de Buffer**.
 - **Descarte Seletivo**.

QoS e Controle de Congestionamento

- ✓ O **Gerenciamento de Buffer distribui** os recursos de armazenamento **de forma justa** entre os fluxos (ou conexões), de forma a melhorar a QoS na rede.
- ✓ Decide quando pacotes **devem ser descartados** para **evitar** ou **sair** de uma situação de congestionamento.
- ✓ **A escolha** de quais pacotes devem ser removidos **é feita** pelo mecanismo de **Descarte Seletivo**, em função das orientações providas pelo gerenciamento de *buffer*.

Classes de Serviço

- ✓ A definição das classes de serviço depende da tecnologia utilizada.
- ✓ Tecnologias como ATM, Diff-Serv, Ethernet definem classes de serviço.
- ✓ Exemplos de Classes de Serviço:
 - **Taxa de *Bits* Constante**
 - Utilizada para tráfegos interativos em tempo real bastante sensíveis ao atraso, ao *jitter* e a perda de pacotes. Ex. emulação de circuitos.
 - **Taxa de *Bits* Variável em Tempo Real**
 - Utilizada para tráfegos interativos em tempo real sensíveis ao atraso, ao *jitter* e perda de pacotes. Ex. tráfego MPEG 2.

Classes de Serviço

- **Taxa de *Bits* Variável Não em Tempo Real**
 - Utilizada para tráfegos não interativos. Em geral, não são sensíveis ao atraso e ao *jitter*, mas requerem controle da perda de pacotes. Ex. tráfego de banco de dados, tráfego de aplicativos.
- **Taxa de *Bits* Disponível ou Melhor Esforço (*Best Effort*)**
 - Utilizado para tráfegos pouco sensíveis aos parâmetros de QoS em geral Ex. tráfego de Internet.

Parâmetros de QoS

- ✓ A QoS requerida por um fluxo é caracterizada através de um conjunto de parâmetros de QoS.
- ✓ Também dependem da tecnologia. No geral tem-se:
 - **Máximo Atraso de Transferência Fim a Fim**
 - Determina o máximo atraso de transferência fim a fim tolerável.
 - **Variação de Atraso Pico a Pico**
 - É a diferença entre o melhor e o pior caso de variação de atraso experimentado pelos pacotes na rede.
 - **Taxa de Perda de Pacotes**
 - É a razão aceitável da soma dos pacotes perdidos sobre o total de pacotes transmitidos.

Descritores de Tráfego

- ✓ Servem para descrever o tráfego que um determinado cliente está enviando para a rede.
- ✓ Também dependem da tecnologia. No geral tem-se:
 - **Taxa de Pico**
 - Taxa máxima da fonte em pacotes por segundo.
 - **Taxa Sustentável**
 - Taxa média sustentada pela fonte em pacotes por segundo.
 - **Tamanho Máximo de Surto**
 - Quantidade máxima de pacotes que pode ser transmitida na taxa de pico sem violar a taxa sustentável.
 - **Tolerância de Surto**
 - Tempo em que a fonte pode transmitir a uma taxa igual a de pico.
 - **Taxa Mínima**

- Unidade 5 Controle de Congestionamento
- ## Monitoramento de Conformidade
- ✓ Verifica se o tráfego **submetido** para a rede está de acordo com o contrato de tráfego **negociado**.
 - ✓ Para isso, a rede **atua** sobre os pacotes **não conformes** por meio da função de **policimento de tráfego** (TP – *Traffic Policing*).
 - ✓ O policiamento é geralmente realizado na interface usuário rede, embora também possa ser realizado entre duas redes.

© Antônio M. Alberti 2008

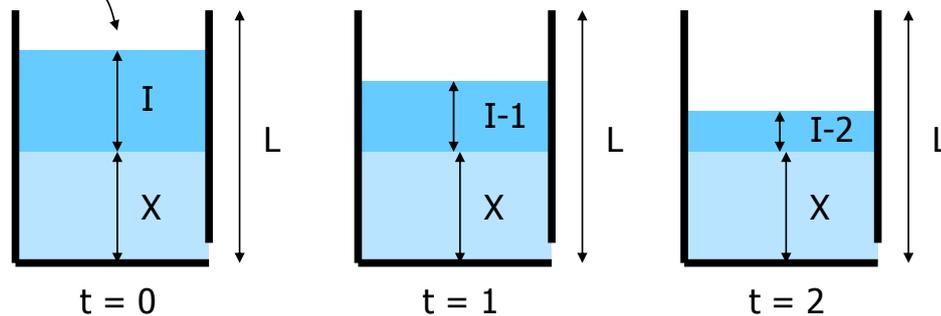
- Unidade 5 Controle de Congestionamento
- ## Monitoramento de Conformidade
- ✓ O policiamento de tráfego é uma função **não intrusiva**:
 - Não atrasa ou modifica as características de um determinado fluxo, a não ser pela **remoção** de pacotes não conformes e pela **mudança** da prioridade dos pacotes.
 - ✓ No contexto das redes ATM, o policiamento de tráfego é implementado através de **algoritmos genéricos de controle de taxa** (GCRA – *Generic Cell Rate Algorithm*).
 - ✓ Dois tipos de GCRA foram definidos:
 - **Leaky Bucket**
 - **Virtual Scheduling**

© Antônio M. Alberti 2008

Algoritmo Leaky Bucket

- ✓ O algoritmo do “**balde furado**” ou **leaky bucket** é baseado na analogia com um balde que é enchido com **I** unidades toda vez que um pacote de tamanho fixo (célula) é **recebido**, e que perde constantemente uma unidade a cada **intervalo de tempo**. **L** é a capacidade do balde.

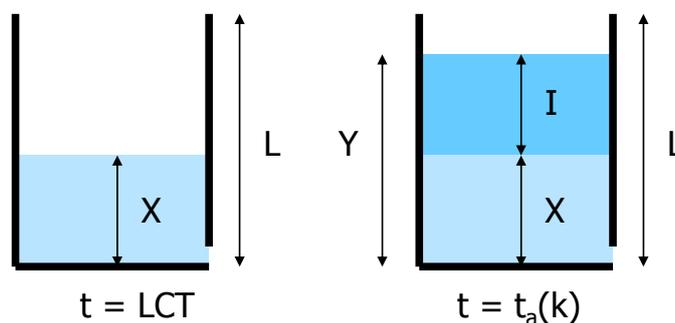
Chegada de um novo pacote
Incrementa de I



© Antônio M. Alberti 2008

Algoritmo Leaky Bucket

- ✓ Variáveis:
 - LCT – Tempo da chegada da última célula conforme.
 - $t_a(k)$ – Tempo de chegada da célula k.
 - X – Ocupação do balde no instante LCT.
 - Y – Ocupação do balde no instante $t_a(k)$.
 - L – Capacidade do balde.



Região Não Conforme:
 $Y > L$

Região Conforme:
 $0 < Y < L$

© Antônio M. Alberti 2008

Algoritmo Leaky Bucket

- ✓ O algoritmo é executado toda vez que uma novo pacote de tamanho fixo ou célula é recebida. Portanto, o enchimento e a drenagem do “balde” são feitos de acordo com o instante de tempo da **chegada da última célula conforme** (LCT).
- ✓ Ou seja, quando uma célula é recebida no instante $t_a(k)$, o “balde” esvazia de $(t_a(k) - LCT)$, o que é equivalente a continuamente esvaziar o “balde” uma unidade a cada unidade de tempo.

Algoritmo Leaky Bucket

- ✓ Uma ocupação **negativa** do “balde” é resultado da chegada atrasada de uma célula. Neste caso, é necessário zerar o “balde” a fim de prevenir o acúmulo de créditos, que acarreta a geração de longos surtos de células (*bursts*).
- ✓ Se a célula recebida é considerada conforme, o “balde” é enchido de I unidades.
- ✓ O algoritmo *Leaky Bucket* também é conhecido pela notação **GCRA (I,L)**, onde I é o incremento e L é capacidade do balde.

Algoritmo Leaky Bucket

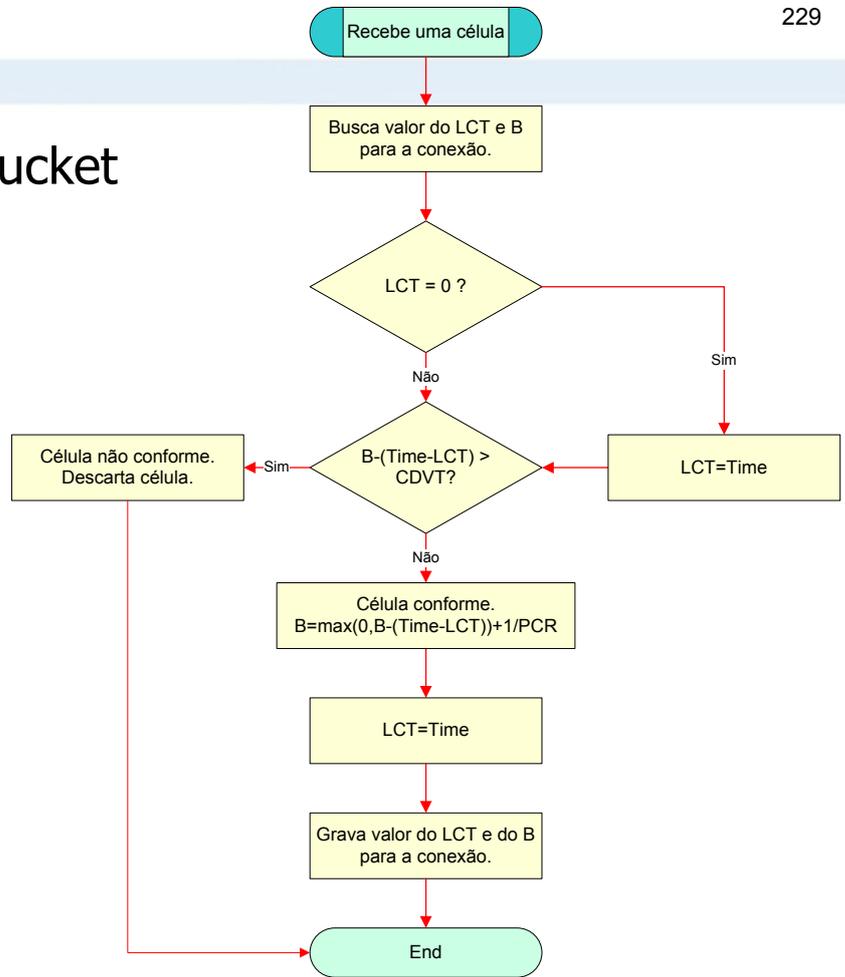
- ✓ Nas redes ATM, os policiadores *Leaky Bucket* podem ser classificados como: **simples** e **duplos**.
 - **Simples**
 - Somente espaça as células de acordo com os descritores de tráfego **PCR** (*Peak Cell Rate*) e **CDVT** (*Cell Delay Variation Tolerance*).
 - Possuem a notação: **GCRA(1/PCR,CDVT)**, onde o incremento I é igual a 1/PCR e o tamanho do “balde” L é igual ao CDVT.

Algoritmo Leaky Bucket

- **Duplos**
 - Policiam as células de acordo com os descritores de tráfego **PCR**, **CVDT**, **SCR** (*Sustainable Cell Rate*) e **MBS** (*Maximum Burst Size*).
 - Consistem de dois policiadores GCRA(I,L) em série.
 - O primeiro possui a notação: **GCRA(1/PCR, CDVT)**.
 - O segundo possui a notação **GCRA(1/SCR,BT-CDVT)**.

Algoritmo Leaky Bucket

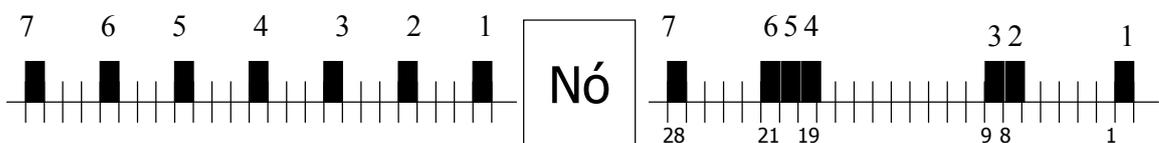
LCT - Last Conformance Time
B - Bucket Occupation
CDVT - Cell Delay Variation Tolerance
PCR - Peak Cell Rate



Algoritmo Leaky Bucket

✓ Exemplo:

- Considere um fluxo de tráfego que possui uma taxa de 0.25 células/segundo (C/S) de taxa de pico em um enlace com taxa de 1 C/S. Ou seja, esta fonte ocupa 1 slot a cada quatro slots do enlace.
- Este enlace leva a um nó de rede, que altera o fluxo de tráfego conforme a figura abaixo.



- Considere agora que o tráfego após o nó seja policiado por um GCRA(4,2), ou seja $1/PCR = 4$ C/S e $CDVT = 2$ S.

Algoritmo Leaky Bucket

✓ Exemplo:

Célula	Time	B	LCT	B-(Time-LCT)	Conforme?
1	1	0	1	0	Sim
2	8	4	1	-3	Sim
3	9	4	8	3	Não
4	19	4	8	-7	Sim
5	20	4	19	3	Não
6	21	4	19	2	Sim

Algoritmo Token Bucket

- ✓ Este regulador de tráfego é uma variação do *Leaky Bucket* e a tradução mais aproximada seria “balde de fichas”.
- ✓ Neste algoritmo, o balde furado retém **fichas**, geradas a uma taxa de r fichas/segundo.
- ✓ Para que um pacote possa ser transmitido, ele deve capturar e destruir uma ficha.
- ✓ Assim, o *Token Bucket* permite que créditos de banda sejam guardados até o tamanho máximo do balde, que vale B_T fichas (ou pacotes).

Algoritmo Token Bucket

- ✓ Isto significa que **rajadas** de até B_T pacotes podem ser enviadas serialmente, permitindo um maior escoamento do tráfego.
- ✓ Entretanto, mesmo havendo fichas disponíveis a taxa máxima de saída é P pacotes por segundo.
- ✓ Segundo **Giroux**, o tamanho máximo do balde pode ser relacionado com os demais parâmetros do contrato da seguinte forma:

$$B_T = \left[\frac{\frac{MBS}{PCR}}{PCR - SCR} \right]$$

© Antônio M. Alberti 2008

Algoritmo Token Bucket

- ✓ Além disto, a taxa r pode ser assumida como sendo igual ao descritor de tráfego SCR e a taxa de pico P como sendo igual ao descritor de tráfego PCR .
- ✓ Isto significa que a longo prazo o regulador produzirá na sua saída um tráfego com taxa média igual ao SCR e taxa de pico igual ao PCR .

© Antônio M. Alberti 2008

Gerenciamento de Estruturas de Filas

- ✓ O objetivo é **administrar** de forma eficiente o espaço disponível em uma estrutura de filas e **isolar** o tráfego destinado a diferentes filas.
- ✓ A eficiência é conseguida através do compartilhamento do espaço físico disponível no maior número possível de filas.
- ✓ Alguns dos algoritmos de gerenciamento **oferecem isolamento naturalmente**, enquanto outros precisam ser **acoplados** a algoritmos de descarte seletivo mais inteligentes de maneira a prevenir que uma fila da estrutura utilize mais recursos do que o permitido e acabe estorvando outras filas.

Gerenciamento de Estruturas de Filas

- ✓ Exemplos de algoritmos de BM:
 - **Particionamento Completo (Complete Partitioning)**
 - **Divide o espaço disponível de forma fixa para cada fila** de estrutura. Mesmo que uma fila esteja desocupada, seu espaço não pode ser utilizado por outras filas.
 - Neste algoritmo, a QoS de uma fila jamais será afetado pelo tráfego em outras filas.
 - **Compartilhamento Completo (Complete Sharing)**
 - **Todo o espaço disponível é compartilhado por todas as filas.**
 - Qualquer fila pode ocupar todo o espaço disponível.
 - Neste algoritmo, a QoS de uma fila pode ser afetado pelo tráfego em outras filas.

Gerenciamento de Estruturas de Filas

✓ Exemplos de algoritmos de BM (cont.)

▫ **Compartilhamento com Alocação Mínima**

- **Reserva um espaço mínimo para cada fila** da estrutura. O espaço remanescente pode ser ocupado totalmente por qualquer uma das filas.
- Portanto, este algoritmo prove um certo nível de isolamento entre as filas.

▫ **Compartilhamento com Tamanho Máximo de Filas**

- **Cada fila da estrutura pode ocupar um espaço máximo.**
- Quando este espaço é atingido, pacotes serão descartados mesmo que haja espaço disponível.
- Este algoritmo protege a estrutura de filas de um uso injusto do espaço disponível, porém não é eficiente pois descarta pacotes mesmo havendo espaço livre.

Gerenciamento de Estruturas de Filas

✓ Exemplos de algoritmos de BM (cont.)

▫ **Compartilhamento com Alocação Mínima e Tamanho Máximo de Filas**

- Este algoritmo é uma **combinação dos dois anteriores**, e portanto usufrui das suas vantagens.

▫ **Particionamento com Limiares Dinâmicos**

- Este algoritmo mantém limiares dinâmicos para cada conexão, que são calculados pelo Algoritmo de Controle de Admissão.
- Se uma conexão atingiu o seu limiar, o próximo pacote que chegar deste fluxo ou conexão irá acionar o Algoritmo de Descarte Seletivo para descartar este pacote ou outro menos prioritário.

Descarte Seletivo

- ✓ Descarta pacotes para evitar ou reduzir o nível de congestionamento na rede.
- ✓ Exemplos de mecanismos de descarte:
 - Descarte Baseado na Prioridade dos Pacotes
 - Descarte Baseado na Classe de Serviço
 - Descarte Baseado em Parâmetro de QoS
 - Descarte Aleatório
 - Descarte Baseado na Ocupação das Filas

Random Early Detection (RED)

- ✓ Baseia-se no princípio do **descarte antecipativo** de pacotes para evitar o congestionamento.
- ✓ Quando o **tamanho da fila** ultrapassa um determinado **limiar**, inicia-se um processo de descarte aleatório de pacotes, onde a **probabilidade de descarte** é função da taxa de ocupação da fila.
- ✓ O descarte antecipativo irá resultar na diminuição na chegada de pacotes ao destino. Conseqüentemente, tem-se a redução na janela de transmissão e uma reversão na tendência de congestionamento.
- ✓ Entretanto, este mecanismo não é eficaz se o protocolo de transporte não reage à perda de pacotes.

Weighted Random Early Detection (WRED)

- ✓ A **probabilidade** de um pacote entrante ser descartado **é definida** pela **ocupação da fila** e por um **peso** associado ao fluxo (ou classe de fluxo) ao qual o pacote pertence.
- ✓ O objetivo é que pacotes de **maior prioridade** tenham **menor probabilidade** de descarte.
- ✓ Por exemplo, uma probabilidade de descarte menor pode ser associada a fluxos de pacotes com maior prioridade ou a fluxos de pacotes que fizeram reserva de recursos.

Exercícios: