

**Protocolos de Justiça para
Redes Ópticas em Anel**

Marcelo Mitsutoshi Uesono

Dissertação de Mestrado

Protocolos de Justiça para Redes Ópticas em Anel

Marcelo Mitsutoshi Uesono

Novembro de 2005

Banca Examinadora:

- Nelson Luis Saldanha da Fonseca
IC-UNICAMP (Orientador)
- Edmundo Roberto Mauro Madeira
IC-UNICAMP
- Alberto Paradisi
CPqD
- Helio Waldman
FEE-UNICAMP (Suplente)

UNIDADE BC
Nº CHAMADA: _____
T/UNICAMP Ue7p
V. _____ EX. _____
TOMBO BCCL 74373
PROC 16.145-09
C _____ D X
PREÇO 11,00
DATA 03/10/07
BIB-ID 415296

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Maria Júlia Milani Rodrigues – CRB8a / 2116

Uesono, Marcelo Mitsutoshi

Ue7p Protocolos de justiça para redes ópticas em anel / -- Campinas, [S.P.
:s.n.], 2006.

Orientadores : Nelson Luis Saldanha da Fonseca; Marcos Rogério
Salvador

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de
Computação.

1. Redes de computadores – Protocolos. 2. Redes de transmissão de
dados. 3. Redes em anel (Redes de computação). I. Fonseca, Nelson Luis
Saldanha da. II. Salvador, Marcos Rogério. III. Universidade Estadual de
Campinas. Instituto de Computação. IV. Título.

Título em inglês: Fairness protocols for optical ring networks.

Palavras-chave em inglês (Keywords): 1. Computer networks – Protocols. 2. Data
transmission networks. 3. Ring networks (Computing networks).

Área de concentração: Redes de Computadores

Titulação: Mestre em Ciência da Computação

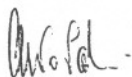
Banca examinadora: Prof. Dr. Nelson Luis Saldanha da Fonseca (IC-UNICAMP)
Prof. Dr. Edmundo Roberto Mauro Madeira (IC-UNICAMP)
Prof. Dr. Alberto Paradisi (CPqD)

Data da defesa: 07/12/2005

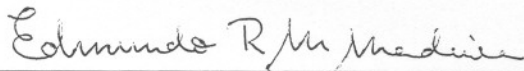
Programa de Pós-Graduação: Mestrado Acadêmico em Ciência da Computação

Protocolos de Justiça para
TERMO DE APROVAÇÃO

Tese defendida e aprovada em 07 de dezembro de 2005, pela Banca
examinadora composta pelos Professores Doutores:



Prof. Dr. Alberto Paradisi
CPqD



Prof. Dr. Edmundo Roberto Mauro Madeira
IC / UNICAMP



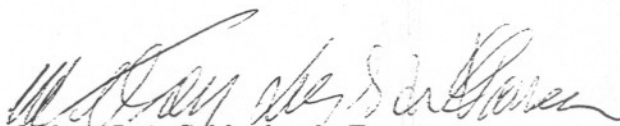
Prof. Dr. Nelson Luis Saldanha da Fonseca
IC – UNICAMP

20749066

Protocolos de Justiça para Redes Ópticas em Anel

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Marcelo Mitsutoshi Uesono e aprovada
pela Banca Examinadora.

Campinas, 18 de Novembro de 2005.



Nelson Luis Saldanha da Fonseca
IC-UNICAMP (Orientador)



Marcos Rogério Salvador
CPqD (Co-orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Resumo

Em redes ópticas em anel com *slotting*, *slots* de tamanhos fixos circulam continuamente pelo anel transportando pacotes de dados entre os nós. Em tais redes, um *slot*, ocupado com pacotes de um nó, tem seu conteúdo removido somente pelo próprio nó de origem. Entretanto, é possível remover o conteúdo do *slot* no nó de destino, técnica conhecida como remoção no destino, o que permite que um *slot* seja utilizado mais de uma vez em cada ciclo, o que é conhecido como reutilização espacial. Esta prática aumenta consideravelmente a vazão do anel.

Entretanto, se o acesso aos *slots* não for controlado, injustiças podem ocorrer devido a oportunidades desbalanceadas de acesso ao meio oferecido aos nós.

Para prevenir um acesso injusto ao meio, a ocupação dos *slots* é controlada pelos protocolos de controle de acesso ao meio (MAC), que distribuem os *slots* entre os nós, oferecendo oportunidades justas do acesso ao meio. Os protocolos MAC seguem políticas de justiça, que são regras que determinam a divisão justa dos recursos do anel.

Os protocolos MAC que oferecem justiça entre nós são comuns na literatura, entretanto, estes protocolos ignoram a justiça entre conexões TCP (*Transmission Control Protocol*). Nesta dissertação, três novos protocolos são apresentados: LCR-SD, TCP-Fair e RVQ. O protocolo LCR-SD distribui a largura de banda baseada na política de justiça entre pares origem-destino, o protocolo RVQ oferece justiça entre conexões TCP e o protocolo de TCP-Fair oferece a justiça entre as conexões TCP mantendo a justiça entre nós.

Os protocolos são comparados através de simulações realizadas no Network Simulator (NS-2). Resultados indicam que os protocolos LCR-SD, TCP-Fair e RVQ oferecem uma vazão superior ao protocolo Metaring. Além disso, apresenta-se um estudo sobre o impacto do tamanho dos *slots* no transporte de tráfego da Internet.

Abstract

In slotted ring networks, slots of fixed size continuously circulate the ring transferring data packets between nodes. In such networks, a slot occupied by packets from one node, has its content removed only by the source node. However, it is possible to remove the packet content in the destination node, technique known as destination removal, which allows a slot to be used more than once in each cycle, leading to spatial reuse, which increases considerably the throughput.

However, if the access to the slots is not regulated, unfairness may occur due to unbalanced medium access opportunities offered to the nodes.

To prevent unfair access to the medium, the occupation of the slots is regulated by a Medium Access Control protocol (MAC), which distributes the slots among the nodes, offering fair access opportunities to the medium. MAC protocols comply with fairness policies, which are rules that determine the fair distribution of the ring resources.

MAC protocols that offer fairness among nodes are common in the literature, however, these protocols ignore the fairness among TCP (Transmission Control Protocol) connections. In this dissertation, three new protocols are presented: LCR-SD, TCP-Fair and RVQ. The LCR-SD protocol distributes the bandwidth based on the source-destination node fairness policy, the RVQ protocol offers fairness among TCP connections and the TCP-Fair protocol offers fairness among TCP connections and maintains fairness among nodes.

The protocols are compared through simulations using the Network Simulator (NS-2). Results indicate that the protocols LCR-SD, TCP-Fair and RVQ offer a superior throughput compared to the Metaring protocol. Moreover, a study on the impact of the slot size on the transport of Internet traffic is presented.

Agradecimentos

Quero agradecer à minha família, que sempre me apoiou em minhas decisões e acreditaram em meu sucesso.

Ao professor Nelson pela paciência, pelos ensinamentos e pela dedicação. Ao professor Marcos pelos conhecimentos e pelas idéias.

Aos novos amigos e amigas que encontrei em meu caminho, especialmente ao Victor e à Karla, que me fizeram rir (muito:); à Ingrid, que sempre me trouxe confiança; à Teresa, ao Sérgio e à Giuliana, que me trouxeram muitas alegrias; à Ilana, que me animou ao final do meu mestrado; ao Daniel e à Larissa, com eles nunca me senti sozinho no curso; e a tantas outras pessoas que enriqueceram minha vida.

Por fim, quero agradecer ao CNPq pela ajuda financeira durante o mestrado (processo 133292/2003-7).

Sumário

Resumo	viii
Abstract	ix
Agradecimentos	x
1 Introdução	1
2 Redes de Fibra Óptica em Anel	3
2.1 Cabos de Fibras Ópticas	3
2.2 Wavelength division multiplexing	4
2.3 Add-drop Multiplexer	5
2.4 Anéis Ópticos	5
2.5 Acesso ao Meio	6
2.6 Sobrevivência em Anéis Ópticos	9
3 Dimensionamento de Slots para Transporte de Tráfego da Internet	10
3.1 Ocupação de slots com pacotes da Internet	10
3.2 Quantidade de <i>slots</i> no anel	11
3.3 Otimização	13
4 Políticas de Justiça	16
4.1 Justiça por Nó de Origem	17
4.2 Justiça por Par Origem-Destino	17
4.3 Max-Min Fairness	18
4.4 Justiça no TCP	18
5 Protocolos MAC	20
5.1 Metaring	21
5.2 Local Cyclic Reservation	22

5.2.1	Reserva cíclica	22
5.2.2	Cálculo de justiça	24
5.2.3	Aplicação da Justiça	25
5.2.4	Detalhes de Implementação	25
5.3	LCR com Justiça por par Origem-Destino	28
5.3.1	Passo 2	28
5.3.2	Cálculo da justiça	28
5.3.3	Aplicação da Justiça	30
5.4	Protocolo Ring Virtual Queue	30
5.4.1	Descrição do Protocolo	32
5.4.2	Algoritmo do Protocolo	33
5.4.3	Exemplo	34
5.4.4	Detalhes de Implementação	35
5.5	Protocolo TCP-Fair	36
5.5.1	Descrição do Protocolo	37
5.5.2	Algoritmo do Protocolo	38
5.5.3	Exemplo	39
5.6	Re-inserção do pacote de controle	41
6	Resultados Numéricos	44
6.1	Cenário de Exemplificação da Justiça Promovida pelos Protocolos	45
6.2	Simulações com Tráfego de Longa Duração	49
6.2.1	Cenário Simétrico	49
6.2.2	Cenário Assimétrico	51
6.3	Simulações com Tráfego de Curta Duração	56
6.3.1	Cenário Simétrico	56
6.3.2	Cenário Assimétrico	58
7	Conclusões	60
	Bibliografia	62
A	Implementação do Simulador	65
A.1	Implementação	65
A.2	Slotting	66
A.3	Protocolos MAC	67

Lista de Tabelas

3.1	Distribuição dos pacotes na Internet.	11
6.1	Distribuição do tamanho de pacotes UDP.	44
6.2	Distribuição da banda de cada política de justiça.	47
6.3	Distribuição da banda de cada protocolo.	47

Lista de Figuras

2.1	Multiplexação e demultiplexação de sinais ópticos.	4
2.2	Exemplo de arquitetura de um ADM.	6
2.3	Passagem de token. Somente o nó que possui o token pode transmitir. . . .	6
2.4	Acesso baseado em contenção.	7
2.5	Inserção de buffer.	8
2.6	Slotted ring.	8
3.1	Distribuição acumulada do tamanho de pacotes na Internet e a quantidade de tráfego.	12
3.2	Ocupação média em porcentagem do tamanho dos <i>slots</i>	12
3.3	Ocupação de 50 <i>slots</i> de 1500 bytes.	14
3.4	Ocupação de 50 <i>slots</i> de 1500 bytes com otimização no preenchimento. . .	14
3.5	Ocupação média em porcentagem do tamanho dos <i>slots</i> com otimização. .	15
3.6	Número médio de pacotes por <i>slot</i>	15
4.1	Anel ligando roteadores.	17
4.2	Max-min fairness – o fluxo destacado pode transferir uma carga maior sem prejudicar os outros fluxos ou nós.	18
5.1	Exemplo de protocolos de justiça global e local.	21
5.2	Nós e enlaces em um anel de 4 nós.	23
5.3	Ilustração da fase de cálculo de justiça do algoritmo LCR.	25
5.4	Conteúdo do pacote de controle em cada nó no ciclo n	26
5.5	Anel no ciclo $n+1$. O conteúdo do pacote de controle no ciclo anterior está armazenado em cada nó.	27
5.6	Carga dos fluxos sobre os enlaces. Alguns nós e enlaces estão repetidos para facilitar a visualização.	29
5.7	Cálculo da taxa justa. Enlaces analisados estão destacados.	30
5.8	Gargalo em um enlace e em um anel.	31
5.9	Descarte de pacotes de uma fila.	36
5.10	Cenário de utilização do protocolo TCP-Fair.	37

5.11	No protocolo TCP-Fair, é garantida justiça na recepção de dados dos nós do anel.	40
5.12	As conexões TCP com um mesmo destino recebem a mesma banda passante.	41
5.13	Burst de pacotes impedindo o encaminhamento do pacote de controle. . . .	42
6.1	Cenário de justiça. Cada política de justiça deve distribuir diferentemente a banda entre os fluxos.	46
6.2	Vazão dos protocolos no cenário simétrico com 4 nós variando o comprimento do anel.	50
6.3	Ciclos do pacote de controle.	51
6.4	Vazão dos protocolos no cenário simétrico com 200Km variando o número de nós.	52
6.5	Ciclos do pacote de controle.	52
6.6	Fluxos de dados no cenário assimétrico com 4 nós.	53
6.7	Vazão dos protocolos no cenário assimétrico variando o comprimento do anel.	54
6.8	Ciclos do pacote de controle.	54
6.9	Vazão dos protocolos no cenário assimétrico variando o número de nós. . .	55
6.10	Ciclos do pacote de controle.	55
6.11	ccdf da distribuição híbrida.	57
6.12	Vazão dos protocolos no cenário simétrico variando o comprimento do anel.	57
6.13	Vazão dos protocolos no cenário simétrico variando o número de nós. . . .	58
6.14	Vazão dos protocolos no cenário assimétrico variando o comprimento do anel.	58
6.15	Vazão dos protocolos no cenário assimétrico variando o número de nós do anel.	59

Capítulo 1

Introdução

As redes de fibra óptica em anel têm sido alvo de pesquisas por serem uma solução promissora para a crescente demanda por capacidade de transmissão e por oferecer capacidade de restauração no caso de falha de nó ou enlace.

Atualmente os anéis SONET/SDH são amplamente utilizados para o transporte de dados, porém, por utilizarem comutação por circuitos, não permitem que a banda ociosa de um circuito seja utilizada por outros circuitos. Redes com comutação de pacotes são mais eficientes que as redes de circuitos, pois permitem que um maior número de fluxos de dados sejam transmitidos sobre o mesmo meio, utilizando os períodos de inatividade de cada fluxo.

O transporte de dados em anéis geralmente é realizado através de *slots* de tamanho fixo, os quais circulam continuamente pelo anel e possuem um campo de dados nos quais os pacotes da rede são transportados. Um *slot* só pode ser utilizado quando estiver vazio, e nele são inseridos pacotes destinados a um mesmo nó do anel. Ao atingir o nó de destino, o *slot* é esvaziado, ficando livre para a transmissão de novos dados, técnica conhecida como remoção no destino [20].

Entretanto, em certas situações, o transporte de dados através de slots pode ocasionar injustiça de acesso no anel. Considerando que um slot só pode ser utilizado quando estiver vazio, o seguinte problema pode ocorrer: se um nó possuir muitos dados para transmissão e passar a ocupar todos os slots vazios que chegam a ele, então todos os slots que chegam ao próximo nó do anel estarão ocupados, impedindo que este nó transmita no anel. Tal cenário motiva que a utilização dos slots seja gerenciada por algum mecanismo.

A utilização dos *slots* do anel é gerenciada pelos protocolos de controle de acesso ao meio (*Medium Access Control* – MAC), que são responsáveis por oferecer oportunidade justa de transmissão no anel. Os protocolos MAC para anéis ópticos com comutação por pacotes, tais como, o Metaring [7] e o ATMR [16], controlam a transmissão dos nós no anel e realizam a distribuição da banda baseados em políticas de justiça.

As políticas de justiça são regras que determinam a divisão da banda do anel entre os usuários da rede. Existem diversos protocolos MAC voltados a oferecer diferentes tipos de justiça e que oferecem alta vazão sob diversas condições de tráfego. No entanto, existem algumas questões pouco abordadas na literatura, como a escolha do tamanho dos *slots* e o impacto da política de justiça no transporte de tráfego TCP.

Nesta tese, um estudo sobre o impacto do tamanho do *slot* na eficiência do transporte de pacotes da Internet é realizado. Demonstra-se que existem tamanhos desfavoráveis para o transporte de pacotes da Internet, o que é de suma importância para a escolha do tamanho de *slots* em futuros protocolos.

Políticas de justiça são também analisadas nesta tese. Em uma rede, dependendo da política adotada, pode-se obter desbalanceamento na divisão da banda passante. Contudo, as políticas de justiça comumente adotadas ignoram os usuários da rede oferecendo justiça entre nós do anel, proporcionando uma distribuição de banda desbalanceada sob a perspectiva dos usuários.

Nesta dissertação, são propostos três novos protocolos que oferecem justiça sob diferentes perspectivas de justiça. Os protocolos são LCR-SD, TCP-Fair e RVQ. Mostra-se que estes novos protocolos são capazes de oferecer altos níveis de vazão, sendo que cada um oferece uma diferente divisão da banda do anel.

O protocolo LCR-SD distribui a banda baseada na política de justiça por par origem-destino. O protocolo RVQ oferece justiça entre conexões TCP que cruzam o anel, sendo apropriada para redes onde predomina tráfego TCP. O protocolo TCP-Fair oferece justiça entre conexões TCP como o RVQ e mantém justiça entre nós.

Através de simulações no simulador *Network Simulator* (NS-2), os protocolos Metaring, LCR, LCR-SD, RVQ e TCP-Fair são comparados. O protocolo Metaring é comumente utilizado na literatura como referência e o protocolo LCR oferece justiça entre nós do anel e originou o protocolo LCD-SD.

Esta dissertação está organizada da seguinte maneira: no Capítulo 2 são apresentados os componentes das redes de fibra óptica em anel. No Capítulo 3 é realizada uma análise quanto à eficiência do uso de *slots* no transporte de pacotes da Internet e uma otimização é apresentada. No Capítulo 4 são apresentadas as políticas de justiça comumente utilizadas em redes de fibra óptica em anel. No Capítulo 5 são apresentados os protocolos MAC utilizados. No Capítulo 6 são apresentados resultados numéricos das simulações realizadas com os protocolos MAC e no Capítulo 7 são apresentados as conclusões e trabalhos futuros.

Capítulo 2

Redes de Fibra Óptica em Anel

Este trabalho aborda mecanismos que gerenciam a divisão da banda de um anel de comutação de pacotes. A tarefa de gerenciar a distribuição da banda de um anel é realizada pelos protocolos MAC, sendo que o oferecimento de justiça é apenas uma parte de um protocolo MAC. Outros componentes dos protocolos MAC, como a utilização de *wavelength division multiplexing* (WDM) para aumentar a capacidade de transmissão e a proteção de caminhos para oferecer resiliência não são abordados nesta dissertação.

Neste capítulo, são apresentados os componentes das redes de fibra óptica em anel. Na Seção 2.1, as fibras ópticas. Na Seção 2.2, o *wavelength division multiplexing* (WDM), para a transmissão simultânea de sinais em uma única fibra óptica. Na Seção 2.3, os *add-drop multiplexers* (ADM), responsáveis pela transmissão e recepção dos *slots*. Na Seção 2.4, os anéis ópticos, onde os ADMs transmitem e recebem informações entre si. Na Seção 2.5, os mecanismos de acesso das redes ópticas em anel. Na Seção 2.6, os mecanismos de sobrevivência de redes ópticas em anel, que garantem o funcionamento do anel no caso de falhas.

2.1 Cabos de Fibras Ópticas

Os cabos de fibras ópticas consistem de fibras de vidro (ou plástico), cada um capaz de transmitir mensagens moduladas na forma de luz. As fibras ópticas possuem várias vantagens comparadas aos meios de metal:

- Baixa atenuação, permitindo a transmissão sem repetidores por grandes distâncias;
- Grande largura de banda;
- Imunidade à interferência eletromagnética;
- Não emite ondas eletromagnéticas, dificultando "a escuta" de informações;

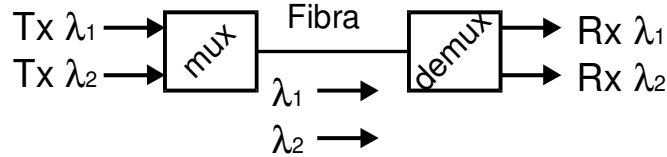


Figura 2.1: Multiplexação e demultiplexação de sinais ópticos.

- Alta resistência à eletricidade, o que a torna segura para uso próximo a equipamentos de alta-voltagem ou entre áreas com diferença de potencial;
- São mais finas e mais leves que meios de metal.

A maior desvantagem da fibra óptica é seu alto custo de instalação.

Os dados transmitidos nas fibras podem ser multiplexados em vários comprimentos de onda de luz, permitindo a transmissão simultânea de dados em uma única fibra. Isto é realizado utilizando-se *wavelength division multiplexing* (WDM).

2.2 Wavelength division multiplexing

Wavelength division multiplexing (WDM) é uma técnica que permite a transmissão simultânea de vários sinais ópticos em uma única fibra. Isto é realizado através da multiplexação dos sinais ópticos em diferentes comprimentos de onda, sendo o equivalente em fibras ópticas do *frequency division multiplexing* (FDM). Em WDM, comprimentos de onda são referidos também como canais ou lâmbdas.

Cada sinal óptico tem a capacidade de transmitir a taxas de até 10 gigabits por segundo, sendo que, normalmente, sistemas WDM utilizam dois comprimentos de onda, apesar de permitirem a utilização de mais comprimentos [4]. A Figura 2.1 ilustra dois sinais ópticos sendo multiplexados em diferentes comprimentos de onda e transmitidos na mesma fibra, e demultiplexados no receptor.

Sistemas mais avançados e capazes de multiplexar ainda mais comprimentos de onda em uma única fibra foram criados, denominados *dense wavelength division multiplexing* (DWDM). Sistemas DWDM permitem a multiplexação de dezenas ou até centenas de comprimentos de onda em uma única fibra através da transmissão de sinais em comprimentos de onda de forma mais precisa, permitindo que mais sinais sejam multiplexados na mesma faixa de comprimento de onda. Sistemas DWDM são capazes de atingir taxas de transmissão na ordem de terabits por segundo [2], porém, uma desvantagem do DWDM é seu custo, pois requer lasers precisos e controle de temperatura.

Devido ao alto custo do DWDM, uma solução mais barata, intermediária ao WDM e DWDM, foi então criada. O *coarse wavelength division multiplexing* (CWDM) normalmente utiliza 8 comprimentos de onda, mas até 18 comprimentos são possíveis [1]. A maior vantagem do CWDM em relação ao DWDM é o custo por possuir um maior espaço entre comprimentos de onda, o que permite a utilização de lasers de menor precisão e dispensa também o controle de temperatura.

Em anéis ópticos, a multiplexação dos sinais é realizada pelos *Add-Drop Multiplexers*.

2.3 Add-drop Multiplexer

Add-drop multiplexer (ADM) é o equipamento que realiza a demultiplexação e a multiplexação dos sinais ópticos. Existem diversas arquiteturas de ADM. A Figura 2.2 exibe uma arquitetura utilizada em anéis com *slots*. Os ADMs são utilizados em anéis ópticos e realizam a inserção e a retirada dos dados em um anel óptico.

A seguir são descritas as operações que são realizadas ao se receber um *slot* do anel:

- O *splitter* retira uma fração do sinal de entrada para leitura e o resto do sinal é colocado em um *fiber delay*, para esperar que o sistema se prepare para tomar uma ação.
- A fração retirada pelo *splitter* é analisada e decide-se se os dados devem continuar no anel ou se devem ser retirados;
- Baseado na decisão tomada, o sistema pode realizar a retirada dos dados do *slot* assim que o sinal sair do *fiber delay*. A retirada é realizada pelo *circulator*;
- Em seguida, se o *slot* estiver livre, o sistema pode inserir novos dados no anel pelo *coupler*.

2.4 Anéis Ópticos

No transporte de pacotes em redes ópticas, a arquitetura em anel tem sido preferida às redes em malha pelo fato de que os ADMs possuem um mecanismo de comutação mais simples que os de um roteador de pacotes óptico e conseqüentemente em equipamentos mais econômicos. Outros fatores são a resiliência oferecida no caso de falhas de enlace ou de nó, e a capacidade de transmitir dados por longas distâncias, dado que cada nó do anel age como um repetidor. Por estas razões, anéis *synchronous optical network* (SONET) e *synchronous digital hierarchy* (SDH) são amplamente utilizados.

Para transportar dados no anel, mecanismos de controle de acesso ao meio são utilizados.

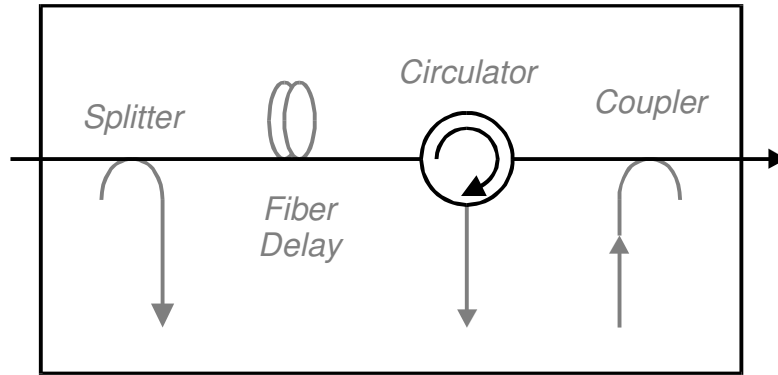


Figura 2.2: Exemplo de arquitetura de um ADM.

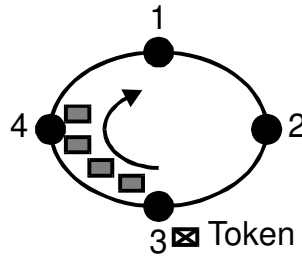


Figura 2.3: Passagem de token. Somente o nó que possui o token pode transmitir.

2.5 Acesso ao Meio

Para transportar dados no anel, é necessário um mecanismo de acesso ao meio a fim de coordenar a transmissão de dados nas fibras, evitando ou tratando colisões. Existem diversas formas de se transmitir dados em um anel:

- Passagem de *token*: utiliza um pacote especial de controle chamado *token*, o qual circula pelo anel. Um nó só pode transmitir dados se possuir o *token* (e.g. FDDI). Na Figura 2.3, o *token* está no nó 3, portanto só este nó transmite no anel. O protocolo do anel eventualmente encaminha o *token* para o próximo nó para que este possa transmitir;
- Contenção (*contention*): O pacote é enviado quando o meio estiver ocioso. Se for detectado que ocorrerá uma colisão, então o pacote sendo enviado tem sua transmissão cancelada momentaneamente e é enviado em uma próxima oportunidade (e.g. CSMA/RN). No exemplo da Figura 2.4 o canal está transportando sete pacotes

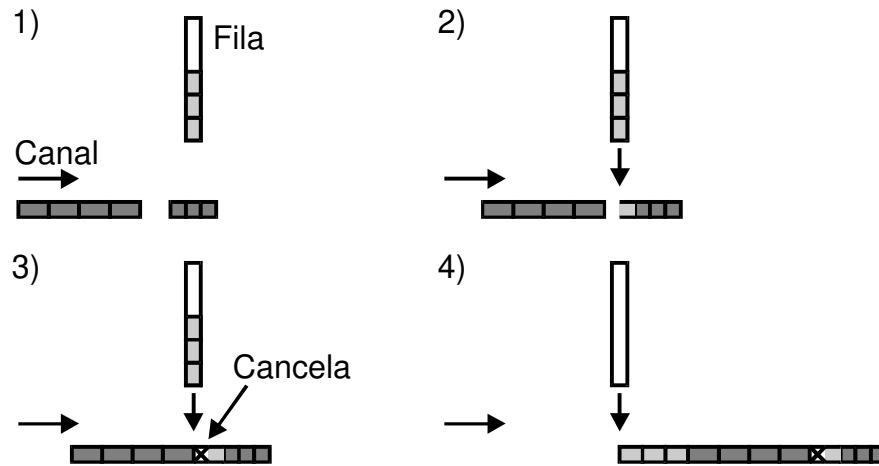


Figura 2.4: Acesso baseado em contenção.

sendo que existe um espaço entre eles (1). Ao detectar que existe espaço livre, o nó começa a transmitir o primeiro pacote da fila (2). Ao detectar a chegada de pacotes no canal, o nó cancela a transmissão do pacote da fila e insere uma marcação de cancela do pacote parcialmente enviado (3). Assim que existe espaço livre no canal, o nó tenta novamente enviar o primeiro pacote da fila e continua transmitindo os outros pacotes enquanto o canal estiver livre (4);

- **Inserção de buffer (*buffer insertion*):** se o buffer de transição e o canal estiverem vazios, então se utiliza o meio para transmissão dos pacotes da fila. Se um pacote vier pelo canal enquanto um pacote da fila estiver sendo enviado, então os dados são armazenados no buffer de transição para que sejam enviados após o envio do pacote da fila (ex: *asynchronous MetaRing*). Na Figura 2.5, o canal está transportando sete pacotes sendo que existe um espaço entre eles (1). Ao detectar que existe espaço livre, o nó começa a transmitir o primeiro pacote da fila até o fim, e se for detectada a chegada de pacotes no canal, os dados do canal são armazenados em um buffer (2). Assim que o pacote da fila for transmitido, são transmitidos os dados do buffer (3); Se o buffer estiver vazio e existir espaço livre no canal, os outros pacotes da fila são transmitidos (4);
- **Slotted ring (ou *time slotting*):** a capacidade do anel é dividida em *slots* geralmente de tamanhos iguais (e.g. *slotted mode MetaRing*). Os pacotes são transmitidos em um campo de dados dentro dos *slots*. Na Figura 2.6, o anel possui *slots* que são utilizados para transporte, sendo que o protocolo MAC gerencia sua utilização;

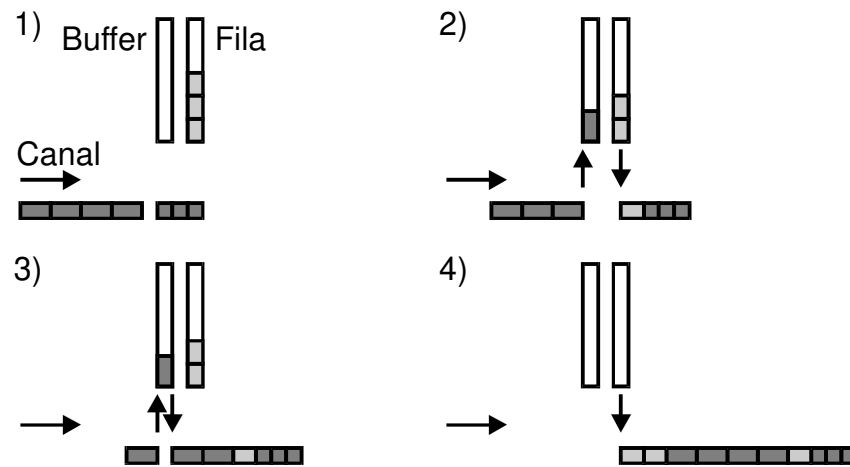


Figura 2.5: Inserção de buffer.

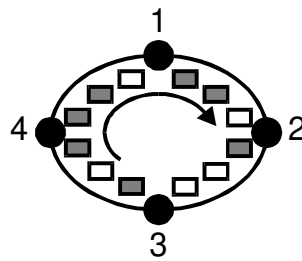


Figura 2.6: Slotted ring.

Os mecanismos de inserção de buffer e *slotted ring* apresentam uma alta vazão em anéis ópticos. Os protocolos introduzidos nesta dissertação utilizam mecanismo de *slots*.

2.6 Sobrevivência em Anéis Ópticos

A sobrevivência em redes de fibra óptica é um assunto de grande importância, dado que a interrupção de um enlace com capacidade de vários gigabits por segundo acarreta em uma grande perda de dados. Desta forma, é necessário que sistemas automáticos restaurem a rede da forma mais rápida e eficiente possível.

Pode-se permitir que camadas superiores de rede realizem a restauração da rede. Protocolos de roteamento podem desviar o tráfego de enlaces defeituosos e protocolos de transporte podem se recuperar de erros e perdas de dados.

Entretanto, pode ser necessário realizar a restauração da rede na camada óptica. Isso ocorre quando o tempo da restauração realizada por camadas superiores é inaceitavelmente grande e viola os requisitos de qualidade e confiabilidade da rede, ou quando as camadas superiores simplesmente não possuem um sistema de restauração adequado.

Em redes de fibras ópticas, interrupções podem ocorrer devido às falhas de enlace ou de equipamento. O primeiro caso é mais comum, pois normalmente os cabos de fibra óptica passam por dutos terrestres juntos a cabos de eletricidade, água ou gás. Falhas de equipamento devem-se principalmente à degradação ou à ineficiência do gerenciamento.

Em [19] e [13], uma introdução à sobrevivência de redes WDM, tanto para redes ópticas em anel quanto em malha é apresentada. Apesar de muito importante, nesta dissertação a sobrevivência em anéis ópticos não é abordada.

Capítulo 3

Dimensionamento de Slots para Transporte de Tráfego da Internet

Neste capítulo a eficiência do transporte de pacotes da Internet em *slots* é estudada. Apresenta-se uma forma de alocação de *slots* que permite um alto grau de ocupação com pacotes da Internet.

3.1 Ocupação de slots com pacotes da Internet

Dado que os *slots* têm tamanhos fixos e os pacotes têm tamanhos variados, ao se transportar pacotes de dados em um *slot*, geralmente ocorre um desperdício, um espaço do *slot* não utilizado. Por exemplo, se em uma fila existem os pacotes de tamanho 1500, 370, 100 e 1500 bytes a serem transportados em um *slot* com espaço para 2000 bytes, então os pacotes de 1500, 370 e 100 bytes são inseridos no *slot*, totalizando 1970 bytes, e deixando 30 bytes do *slot* não utilizados.

Investigações a respeito da distribuição do tamanho de pacotes na Internet mostram que certos tamanhos de pacotes são mais frequentes que outros. A distribuição de pacotes apresentada em [9] é utilizada para análise. A Tabela 3.1 mostra os tamanhos de pacotes mais comuns na Internet. Os pacotes de 40-44 bytes incluem pacotes TCP de controle como ACK, SYN, FIN, RST e pacotes Telnet com um caractere. Muitas implementações de TCP que não utilizam *Path MTU Discovery* utilizam um *Maximum Segment Size* (MSS) de 512 ou 536 para conexões TCP entre redes diferentes, resultando em pacotes IP de 552 e 576 bytes. Um *Maximum Transmission Unit* (MTU) de 1500 bytes é característico de redes Ethernet.

A Figura 3.1 mostra a distribuição acumulada do tamanho dos pacotes na Internet. A curva “Pacotes” representa a distribuição acumulada do tamanho dos pacotes. A curva “Bytes” representa a quantidade de dados que cada tamanho de pacote representa, ou seja,

Tamanho do Pacote (bytes)	Pacotes (%)
40	33,5
44	6,4
552	4,7
576	5,3
1500	16,5

Tabela 3.1: Distribuição dos pacotes na Internet.

apesar de 16,5% dos pacotes terem 1500 bytes, devido ao seu tamanho, estes representam 54,6% dos dados.

Baseado nesta distribuição de tamanho de pacotes, simulações foram realizadas testando a ocupação média de 500 *slots*. As simulações foram realizadas com diferentes tamanhos de *slots*, utilizando *slots* com tamanho de 1500 bytes até 15000 bytes, em passos de 10 bytes. Os resultados são apresentados na Figura 3.2.

A Figura 3.2 mostra que, quanto maior o tamanho do *slot*, mais eficiente é a ocupação. É importante notar que, a partir dos 3000 bytes, a quantidade de bytes desperdiçados é praticamente constante, aproximadamente 556 bytes. Em *slots* maiores, este desperdício é menos significativo, por isso a ocupação se torna mais eficiente. Por exemplo, 556 bytes em um *slot* de 3000 bytes representam 18,5% do *slot*, enquanto em um *slot* de 15000 bytes representam 3,7% do *slot*.

Entretanto, apesar da ocupação ser mais eficiente em *slots* maiores, quanto maior o tamanho do *slot*, menor a quantidade de *slots* no anel, o que pode acarretar em um impacto negativo na justiça, dado que um número menor de *slots* torna menos precisa a distribuição da banda. Por exemplo, se dois nós devem utilizar o mesmo número de *slots* mas existem 5 *slots* disponíveis, então a distribuição entre os nós é de 3 e 2 *slots*, resultando em uma distribuição de banda de 60% e 40% entre os nós, por outro lado, se existem 51 *slots* disponíveis, então a distribuição entre os nós é de 26 e 25 *slots*, resultando em uma distribuição de banda de 51% e 49% entre os nós.

3.2 Quantidade de *slots* no anel

A quantidade de *slots* no anel depende de diversos fatores como o comprimento do anel, a velocidade da propagação da luz na fibra, a largura de banda e o tamanho dos *slots*.

O número de *slots* em um anel é fixo, e para se contar esse número, vamos inicialmente analisar o anel em um dado instante de tempo. A quantidade de *slots* no anel é igual à quantidade de bytes existentes no anel dividido pelo tamanho dos *slots* em bytes. Sejam:

ring_slots – quantidade de *slots* existentes no anel.

ring_bytes – quantidade de bytes existentes no anel.

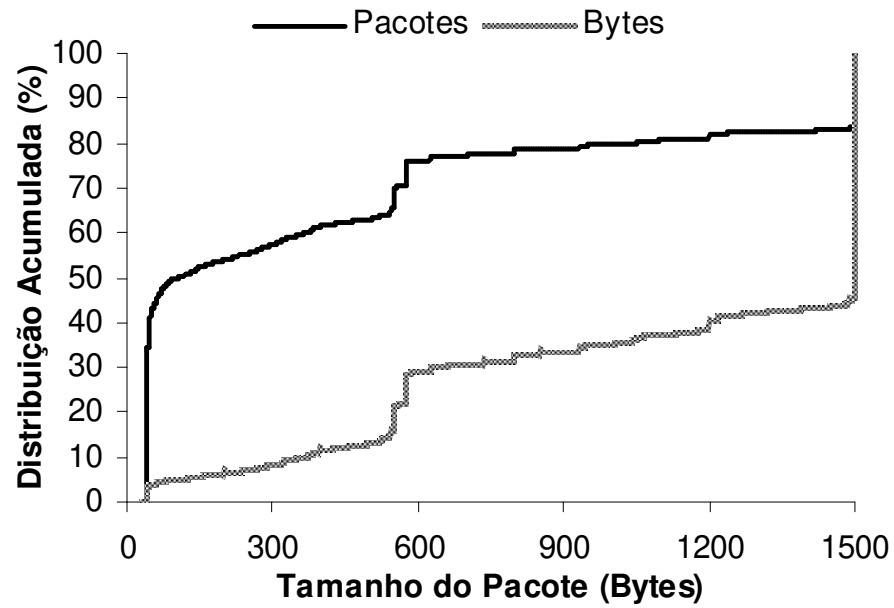


Figura 3.1: Distribuição acumulada do tamanho de pacotes na Internet e a quantidade de tráfego.

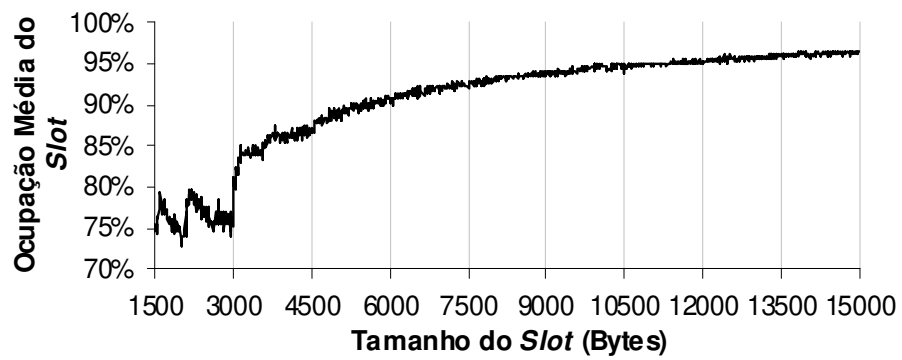


Figura 3.2: Ocupação média em porcentagem do tamanho dos slots.

slot_size – tamanho dos *slots* em bytes.

Então: $ring_slots = ring_bytes / slot_size$ (1).

A quantidade de bytes existentes no anel pode ser calculada pela quantidade de bytes que passam por um nó do anel em um ciclo, então, *ring_bytes* é igual ao tempo de um ciclo do anel vezes a banda da fibra. Seja:

cycle_time – tempo que os *slots* levam para circular o anel.

Então: $ring_bytes = cycle_time * bandwidth$ (2).

cycle_time é igual ao tempo que os *slots* levam cruzando as fibras mais o tempo que os *slots* levam cruzando os nós do anel. Sejam:

fiber_time – tempo que os *slots* levam cruzando as fibras.

node_time – tempo que os *slots* levam cruzando os nós do anel.

Então: $cycle_time = fiber_time + node_time$ (3).

O tempo que os *slots* levam cruzando as fibras é igual ao comprimento do anel dividido pela velocidade de propagação da luz nas fibras. Sejam:

ring_size – comprimento do anel.

fiber_speed – velocidade de propagação da luz nas fibras.

Então: $fiber_time = ring_size / fiber_speed$ (4).

Unindo (1) e (2): $ring_slots = cycle_time * bandwidth / slot_size$

Unindo com (3): $ring_slots = (fiber_time + node_time) * bandwidth / slot_size$

Unindo com (4):

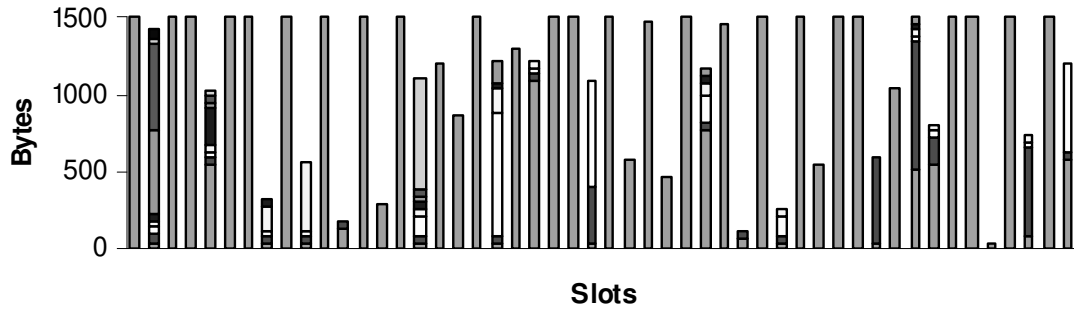
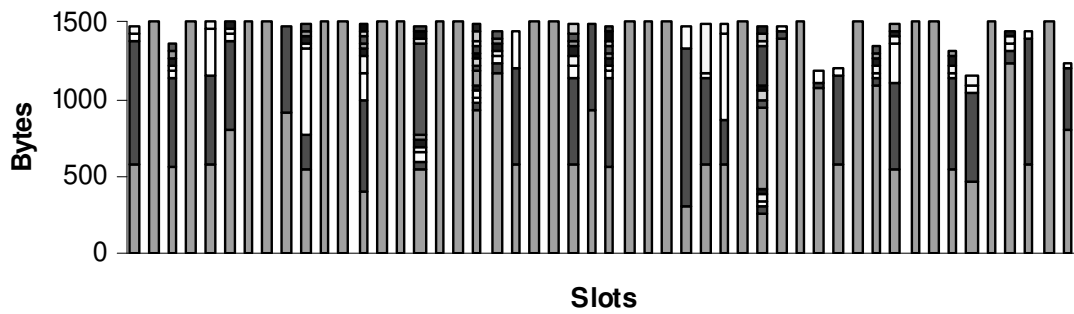
$ring_slots = ((ring_size / fiber_speed) + node_time) * bandwidth / slot_size$

Então por exemplo: em um anel de 100 km, com velocidade de propagação da luz 200.000 km/segundo, com uma latência total dos nós 0,2 milissegundos, com largura de banda 10 Gigabits/segundo e com *slots* de 3000 bytes, existem 313 *slots*. Outro exemplo: em um anel de 50 km, com velocidade de propagação da luz 200.000 km/segundo, com uma latência total dos nós 0,2 milissegundos, com largura de banda 2,5 Gigabits/segundo e com *slots* de 3000 bytes, existem 50 *slots*.

3.3 Otimização

É possível implementar uma otimização a fim de se obter um maior grau de ocupação dos *slots*. Na simulação anterior, que resultou na Figura 3.2, ao preencher os *slots*, se o próximo pacote da fila é maior que o espaço remanescente do *slot*, então o *slot* é encaminhado. A Figura 3.3 ilustra a ocupação de 50 *slots*. Na Figura 3.3 existem 50 colunas, representando cada uma a ocupação de um *slot*. Cada barra em uma coluna representa o espaço ocupado por um pacote de dados.

A política de ocupação de *slots* proposta consiste em, ao preencher um *slot*, se o próximo pacote da fila é maior que o espaço remanescente no *slot*, então é procurado na

Figura 3.3: Ocupação de 50 *slots* de 1500 bytes.Figura 3.4: Ocupação de 50 *slots* de 1500 bytes com otimização no preenchimento.

fila se existem pacotes menores que podem ser inseridos neste espaço livre remanescente, a fim de aproveitar o espaço do *slot* que seria desperdiçado. A Figura 3.4 ilustra a ocupação de 50 *slots* utilizando esta otimização.

Nesta simulação foi utilizada uma fila de comprimento 10 pacotes e, comparando as Figuras 3.3 e 3.4, é possível notar que ocorre significativa melhora na eficiência da ocupação dos *slots*. Nas simulações com *slots* de 1500 bytes, sem a otimização a ocupação média dos *slots* foi de 75,7%, enquanto com a otimização a ocupação foi de 97,0%.

A Figura 3.5 mostra a ocupação média do *slot* em porcentagem do tamanho do *slot* utilizando a otimização e com uma fila de 100 pacotes.

Com a otimização, em *slots* maiores que 3000 bytes a ocupação é maior que 99%. Comparando as Figuras 3.2 e 3.5, é notável a melhoria obtida com a otimização.

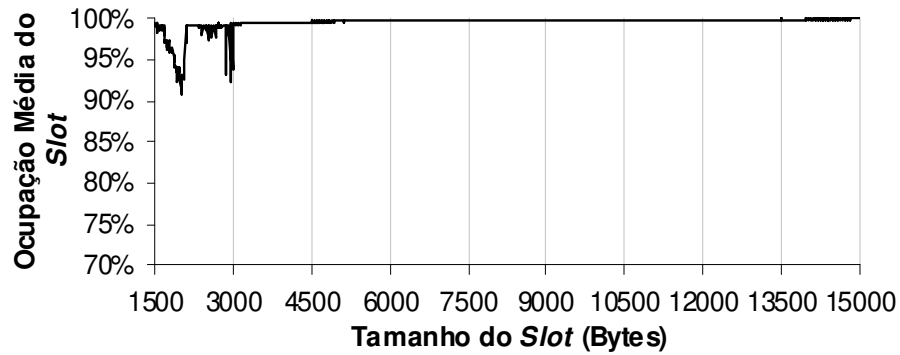


Figura 3.5: Ocupação média em porcentagem do tamanho dos *slots* com otimização.

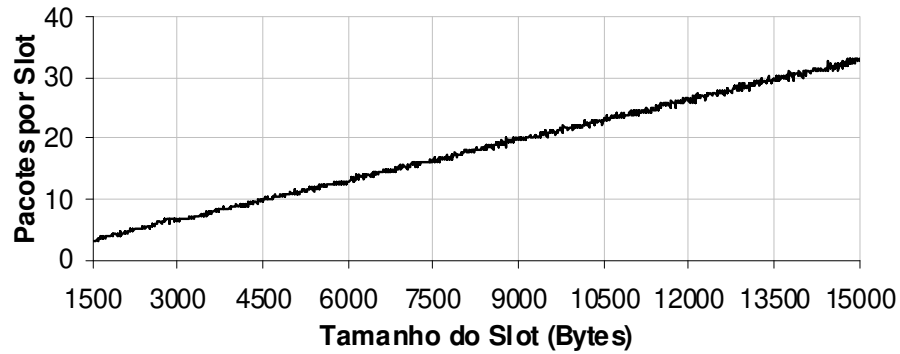


Figura 3.6: Número médio de pacotes por *slot*.

Note que estas simulações não levam em conta o espaço ocupado por cabeçalhos do *slot*. Em diversos protocolos, o tamanho ocupado pelo cabeçalho varia de acordo com o número de pacotes no *slot*. A Figura 3.6 apresenta o número médio de pacotes por *slot* para as simulações com a otimização.

Neste trabalho são utilizados *slots* de 3100 bytes em todas as simulações, pois como mostra a Figura 3.6, *slots* maiores que 3000 bytes oferecem uma boa ocupação, e informações de cabeçalho não chegam a ocupar 100 bytes.

Capítulo 4

Políticas de Justiça

Neste capítulo, são apresentados conceitos de justiça necessários para a discussão dos protocolos MAC de justiça. São apresentados as políticas de justiça mais conhecidas e o conceito de justiça oferecido pelo TCP.

Protocolos MAC desenvolvidos para essas redes geralmente suportam remoção de pacotes no destino. Com esta funcionalidade, consegue-se a reutilização espacial e, conseqüentemente, melhora-se consideravelmente a vazão do anel. No entanto, a capacidade extra, obtida com a reutilização espacial, se não controlada, pode gerar sensíveis injustiças de acesso entre os nós da rede e torna necessária a utilização de um protocolo de justiça para garantir o acesso justo ao meio. No entanto, existem diferentes noções de justiça em anéis ópticos, expressos pelas políticas de justiça.

O termo “fluxo” é utilizado para se referir ao agregado de tráfego transportado entre um par origem-destino; por exemplo, dados transmitidos no anel pelo nó 1 que têm como destino o nó 2 compõem o fluxo 1-2.

Existem diversas políticas de justiça para redes em anel óptico, sendo que a maioria segue os modelos de justiça por nó de origem e justiça por par origem-destino. No entanto, existem outras políticas de justiça, como o *Ring Ingress-Aggregated with Spatial Reuse fairness* (RIAS) [12], que oferece justiça entre os nós do anel e entre fluxos partindo de um mesmo nó, e *proportional fairness* (Ethernet Ring), que é um anel de roteadores. Cada política promove justiça sob diferente perspectiva e favorecem diferentes tipos de aplicação.

As políticas de justiça apresentadas são a justiça por nó de origem e a justiça por par origem-destino, por serem amplamente utilizadas. É também apresentado o conceito do *Max-min Fairness* no contexto das redes ópticas em anel, que propõe uma utilização mais eficiente da banda e a justiça do protocolo TCP, por ser o principal protocolo de transporte utilizado na Internet.

A Figura 4.1 é utilizada como referência, a figura da esquerda mostra o anel com os

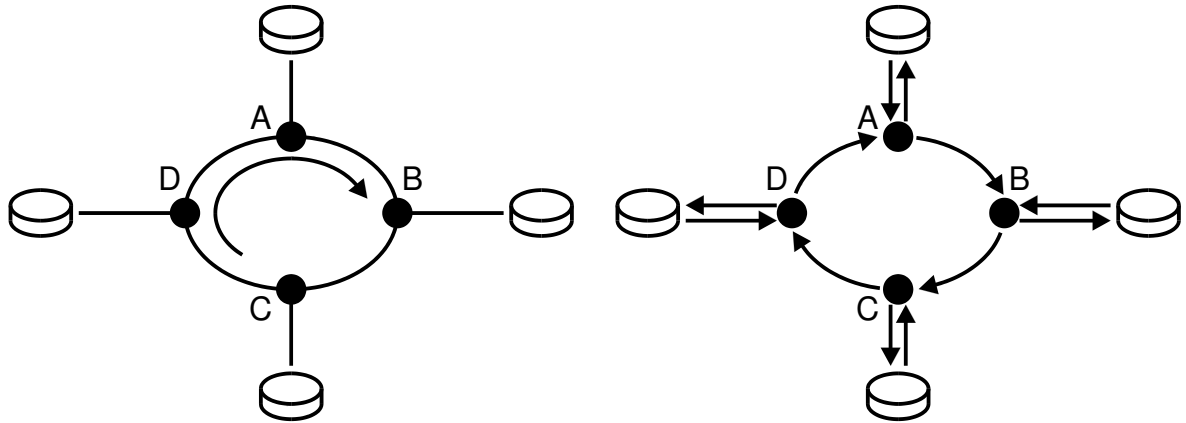


Figura 4.1: Anel ligando roteadores.

nós A, B, C e D ligando roteadores. Os enlaces entre os roteadores e o anel são *full-duplex*. Para facilitar a discussão, cada enlace *full-duplex* será tratado como dois enlaces independentes como na figura à direita. Também são mencionadas as filas que dão acesso aos enlaces, a fila que armazena pacotes do fluxo A-B é chamada de fila A-B.

4.1 Justiça por Nó de Origem

Na política de justiça por nó de origem (*Source Node Fairness*), a banda do anel é dividida igualmente entre os nós do anel. Por exemplo, se por um anel (Figura 4.1) passam os fluxos A-D, B-C e B-D, estes geram um gargalo no enlace B-C. Os nós A e B, que disputam o mesmo gargalo, recebem cada um 50% da capacidade de transmissão do anel.

Entretanto, é necessário levar em consideração que, ao oferecer igualdade entre nós, pode ocorrer desigualdade entre fluxos. Neste exemplo, enquanto o fluxo A-D recebe 50% da capacidade do anel, os fluxos B-C e B-D recebem em média 25% da capacidade do anel. Em certas situações, isso pode acarretar em assimetria de caminhos [5], onde um fluxo A-B recebe uma banda muito diferente à do fluxo B-A, o que tem efeitos negativos sobre o TCP, como degradação do desempenho da rede. Este assunto é abordado em mais detalhes na RFC 3449.

4.2 Justiça por Par Origem-Destino

Na política de justiça por par origem-destino (*Source-Destination Node Fairness*), a banda do anel é dividida igualmente entre os fluxos do anel. Assumindo o mesmo cenário anterior,

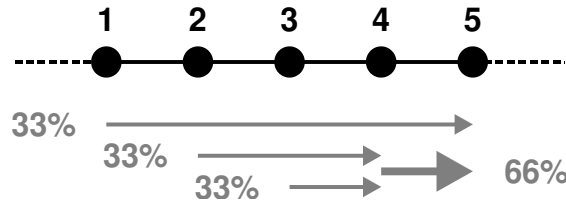


Figura 4.2: Max-min fairness – o fluxo destacado pode transferir uma carga maior sem prejudicar os outros fluxos ou nós.

com os fluxos A-D, B-C, B-D, nesta política é oferecida a mesma banda para todos os fluxos, ou seja, 33,33% da capacidade de transmissão do anel.

Esta política, por procurar manter a mesma vazão entre os fluxos, tende a diminuir a assimetria de caminhos.

Um ponto negativo é que, por oferecer a mesma banda entre fluxos, pode ocorrer uma distribuição de banda injusta entre os nós, pois um nó que transmite menos fluxos recebe menos banda que um nó que transmite mais fluxos.

4.3 Max-Min Fairness

Foi apontado em [21] que a política de “mesma oportunidade de acesso” pode ser considerada injusta, dado que alguns nós podem aumentar sua capacidade de transmissão sem afetar outros nós (*max-min fairness*). Na Figura 4.2, o fluxo 4-5 pode ocupar 66% da capacidade do anel sem afetar os outros fluxos.

4.4 Justiça no TCP

O *Transmission Control Protocol* (TCP) fornece um serviço confiável de transmissão de dados, e é o principal protocolo de transporte em uso hoje, responsável por de mais de 90% dos pacotes na Internet [9].

Para o leitor que não possui familiaridade com o funcionamento do TCP, em [15] e [14] o mecanismo de funcionamento do TCP é brevemente explicado. Em [11], são comparados diferentes tipos de TCP. Aqui será discutido somente o aspecto da justiça entre conexões TCP.

O TCP é um protocolo que adapta a taxa de transferência de dados às condições de carga da rede e à capacidade processamento do receptor. Se a rede e o receptor possuírem capacidade disponível adicional, o emissor aumentará a taxa de transmissão tentando

ocupar esta capacidade ociosa. Por outro lado, se houver congestionamento, o emissor reduzirá sua taxa de transmissão para permitir que a rede se recupere. Esta adaptação tenta obter a maior taxa de transferência possível sem provocar perda contínua de dados. Em [17], são descritos algoritmos para implementação de protocolos de transporte baseados em janelas.

Os protocolos que diminuem as taxas de transmissão de fluxos que ocupam muita banda e permitem que os fluxos com taxas menores ocupem mais banda, são referidos como protocolos *TCP-Friendly*.

Capítulo 5

Protocolos MAC

Neste capítulo, como referências são apresentados os protocolos MAC Metaring [7] e *Local Cyclic Reservation* (LCR) [22]. O Metaring é um protocolo que oferece justiça por nó de origem e é utilizado como referência na literatura. O LCR é um protocolo que oferece justiça por nó de origem.

Também são introduzidos novos protocolos, o LCR-SD, o *Ring Virtual Queue* (RVQ) e o TCP-Fair. O LCR-SD é uma modificação do protocolo LCR e oferece a justiça por par origem-destino, o RVQ e o TCP-Fair oferecem justiça entre conexões TCP.

Os protocolos de acesso podem ser locais ou globais. Protocolos globais consideram a rede um único recurso de comunicação e, como resultado, todos os nós recebem as mesmas limitações de transmissão. Protocolos locais consideram cada enlace um recurso de comunicação e somente os nós competindo pelo mesmo recurso recebem limitações.

A Figura 5.1 ilustra a diferença entre protocolos locais e globais. Observa-se a representação de um anel unidirecional com oito nós. O tráfego do anel pode ser dividido em duas partições, permitindo uma capacidade de transmissão de duas vezes a capacidade dos enlaces. Na primeira partição, os nós 1 e 2 requerem 100% da capacidade do anel, esta demanda excede a capacidade dos enlaces, gerando um gargalo. Na segunda partição, o nó 5 requer 30% da capacidade do anel e o nó 6 requer 70%. Ambos os protocolos atribuem 50% de capacidade aos nós disputando o enlace gargalo 2-3. Entretanto, nos protocolos globais, devido à limitação de transmissão de 50% imposta aos nós 1 e 2, o nó 6 recebe a mesma limitação de transmissão de 50%, quando poderia utilizar 70%, como acontece em protocolos locais.

A seguir são apresentados os protocolos MAC analisados.

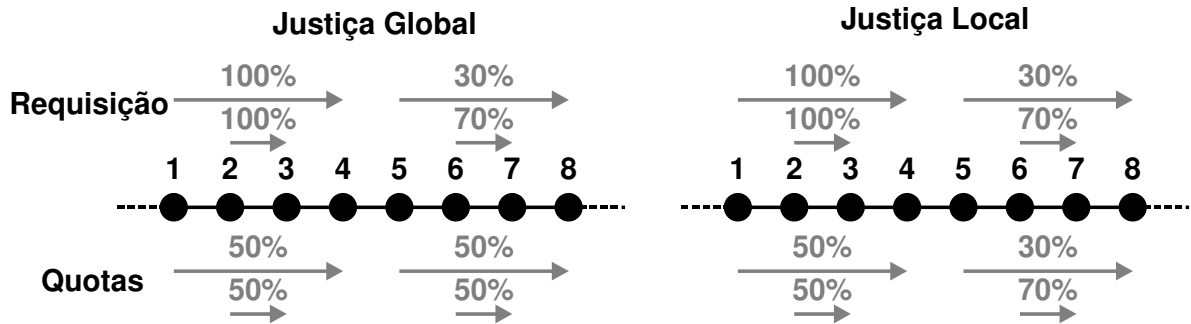


Figura 5.1: Exemplo de protocolos de justiça global e local.

5.1 Metaring

O MetaRing [7] é um protocolo global que utiliza a política de justiça por nó de origem. Este protocolo utiliza tanto o acesso por *slots* como por inserção de buffer.

No acesso por *slots*, o anel é dividido em fatias de tamanhos iguais (*slots*). Cada *slot* carrega um bit que informa se este está ocupado ou vazio (ex. HORNET [24]). Ao atingir o nó de destino, os dados são removidos do *slot* liberando-o para ser utilizado novamente, técnica conhecida como remoção no destino.

Neste protocolo, um sinal de controle denominado “SAT” (*Satisfied*) circula pelos nós do anel atribuindo quotas de transmissão. O pacote de controle SAT pode se propagar na mesma direção dos dados ou, no caso de anéis bidirecionais, no sentido oposto aos dados.

O protocolo funciona, da seguinte forma: se o nó que receber o pacote de controle SAT possuir pacotes para transmitir e ainda não tiver enviado pelo menos 1 pacote, então este nó será dito “*starved*” e irá liberar o SAT somente quando não houver mais pacotes para transmitir ou quando transmitir 1 pacote, tornando-se, então, satisfeito (*SATisfied*).

A não liberação do pacote de controle SAT impede que os outros nós tenham suas quotas renovadas, fazendo com que em algum momento parem de transmitir, permitindo que o nó que possui o SAT tenha oportunidade para utilizar sua quota. Note que tal nó somente irá renovar sua quota quando liberar o SAT.

O algoritmo é eficiente na prevenção de *starvation* e garante a divisão justa da banda do anel.

Entretanto, o protocolo possui alguns problemas relativos à capacidade de vazão em cenários desbalanceados, e são abordados em [8], onde são introduzidos mecanismos para tratar destes problemas.

O primeiro ocorre quando o anel está muito carregado, fazendo com que um nó leve um grande intervalo de tempo para conseguir acessar o anel. Isto se deve principalmente

pelo fato da quota ser atribuída de forma estática. No caso em que somente um nó possui dados para transmitir, então para que o anel seja maximamente utilizado, é necessário que sua quota seja maior ou igual à capacidade de um ciclo do anel. No entanto, no caso de alta carga, o tempo de renovação da quota será muito grande, aumentando o atraso dos pacotes.

O segundo problema é a duração do ciclo, determinada pelo enlace mais carregado, provocando perdas desnecessárias de vazão em cenários assimétricos.

Os nós que transmitem dados somente por enlaces com pouca carga esgotarão sua quota enquanto o SAT está sendo retido por nós que utilizam enlaces muito carregados.

Nas simulações, é utilizada a versão unidirecional para anéis em *slot* com o pacote de controle deslocando no mesmo sentido dos dados e incluindo a otimização apresentada em [8] algoritmo B.

5.2 Local Cyclic Reservation

O protocolo Local Cyclic Reservation (LCR) [22] foi concebido para anéis unidirecionais. É um protocolo local e adaptativo que trabalha com o modelo de justiça de vazão por nó de origem e utiliza os seguintes mecanismos:

- Reserva cíclica de recursos: coleta e compartilha informações de requisição;
- Cálculo de justiça: determina a quantidade justa de banda que cada nó pode transmitir sobre cada enlace durante um ciclo. Um ciclo é o espaço de tempo entre passagens do pacote de controle;
- Aplicação da justiça: permite que cada nó transmita somente o que lhe foi permitido.

Estes mecanismos estão descritos mais detalhadamente a seguir.

5.2.1 Reserva cíclica

A reserva cíclica utiliza um pacote de controle que circula pelo anel na mesma direção dos dados para coletar e distribuir reservas por recursos.

A reserva por recursos corresponde à demanda do nó sobre os enlaces em um ciclo, e pode ser expresso em forma absoluta ou em forma de fração da capacidade do enlace no ciclo. Para facilitar a compreensão será adotada a partir deste ponto a forma fracionária.

Seja N o número de nós no anel e n_i o i -ésimo nó do anel para $i = 0, \dots, N - 1$. Seja o número de enlaces no anel $L = N$ e l_k o k -ésimo enlace do anel de forma que l_k une os nós n_{k-1} e n_k .

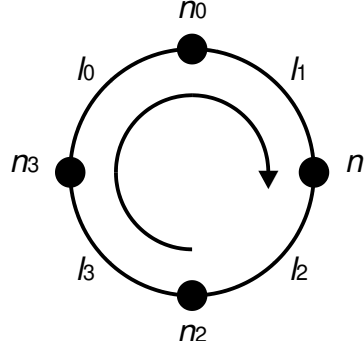


Figura 5.2: Nós e enlaces em um anel de 4 nós.

Cada nó possui $N - 1$ filas, uma para cada destino possível. Assim, a demanda do nó n_i sobre o enlace l_k é dada por:

$$D_i^k = \sum_{j=k}^{|i-1+L|_L} d_i^j,$$

sendo d_i^j , para $j = 0, \dots, N - 1$, a carga do nó n_i na fila destinada ao nó n_j e $|\cdot|_L$ a operação de módulo.

Para gerar a fração da demanda, o protocolo também verifica se a demanda sobre um enlace não excede sua capacidade no ciclo. Seja C a capacidade do anel em um ciclo, seja E_i^k a fração da demanda do nó n_i sobre o enlace l_k . Se a carga total do nó $D_i^{|i+1|_L}$ for menor ou igual a C , então $E_i^k = D_i^k / C$; senão, $E_i^k = (D_i^k / C) \times f$, onde o fator de correção da demanda $f = C / D_i^{|i+1|_L}$.

O pacote de controle carrega a matriz $r = [r_i^j]_{i=0, \dots, N-1, j=0, \dots, L-1}$, onde r_i^j é a requisição do nó n_i sobre o enlace l_j . Conforme o pacote circula pelo anel, cada nó insere sua requisição sobre cada enlace atualizando os elementos correspondentes na matriz r . O LCR requer que os nós conheçam os pedidos de requisição de todos os outros enlaces e, conforme o pacote de controle circula pelo anel, os nós atualizam a matriz local $\alpha = [\alpha_i^j]_{i=0, \dots, N-1, j=0, \dots, L-1}$ com o valor correspondente da matriz r .

Na chegada do pacote de controle, o nó n_j realiza os seguintes passos.

Passo 1: Obtenção das requisições realizadas pelos nós posteriores, ou seja, $\alpha_i^k = r_i^k$ para $k = 0, \dots, L - 1$ e $i = j + 1, \dots, N - 1$;

Passo 2: Inserção da própria requisição no pacote de controle, ou seja, o cálculo do valor de E_j^k e a atribuição $r_j^k = E_j^k$, para $k = 0, \dots, L - 1$;

Passo 3: Obtenção das requisições realizadas pelos nós anteriores, ou seja, $\alpha_i^k = r_i^k$ para $k = 0, \dots, L - 1$ e $i = 0, \dots, j - 1$;

Passo 4: Encaminhamento do pacote de controle para o próximo nó.

5.2.2 Cálculo de justiça

Um nó calcula sua própria taxa justa por enlace entre os Passos 1 e 2. A execução do algoritmo gera a matriz de justiça $\beta = [\beta_i^k]_{i=0,\dots,N-1,k=0,\dots,L-1}$ onde β_i^k é a taxa justa permitida ao nó n_i sobre o enlace l_k . A matriz de justiça determina quanto tráfego o nó n_i pode transmitir sobre o enlace l_k durante o ciclo de justiça, e pode ser expresso em bits, bytes ou *slots*.

Inicialmente a matriz $\beta = \alpha$. O algoritmo para o cálculo de justiça começa eliminando o gargalo mais pesado, pois a eliminação de um gargalo pesado pode eliminar um gargalo menor, e ocorre da seguinte forma:

```

[ Encontre o gargalo mais pesado
  Enquanto existir gargalo
    Elimine o gargalo
    Atualize as requisições dos enlaces posteriores
    Encontre o gargalo mais pesado
  Fim do laço

```

Para eliminar um gargalo em um enlace l_k , o LCR avalia as requisições dos nós sobre o enlace em ordem crescente de demanda.

Dada a demanda β_i^k do nó n_i sobre o enlace l_k , se $\beta_i^k > r_{fair}$, então $\beta_i^k = r_{fair}$, onde a taxa justa $r_{fair} = c/x$, sendo inicialmente $x = N$ e $c = 1$, onde x é o número de nós competindo pelo enlace e c a fração da capacidade restante no enlace.

Os valores de capacidade restante c e número de nós competindo pelo enlace x são atualizados conforme o algoritmo analisa as requisições. Portanto, após calcular β_i^k , o algoritmo atualiza a capacidade restante do enlace subtraindo a taxa justa calculada, ou seja, $c \leftarrow c - \beta_i^k$, e em seguida atualiza o número de nós restantes disputando pelo gargalo, ou seja, $x \leftarrow x - 1$.

Após eliminar um gargalo, são atualizadas as requisições dos enlaces posteriores, já que é possível que a quota de um nó sobre um enlace l_k seja maior que a quota sobre o enlace l_{k+1} , o que é ineficiente dado que os dados de um nó que passam pelo enlace l_{k+1} passam pelo enlace l_k . Para prevenir esta situação, não se permite que um nó tenha, sobre os enlaces posteriores, uma quota maior que sobre enlaces anteriores.

O processo se repete até que todos os gargalos tenham sido eliminados.

A Figura 5.3 mostra a evolução do algoritmo através da resolução sucessiva de gargalos. Na Figura 5.3, modificações do passo anterior estão sublinhadas. As somas das requisições sobre cada enlace estão em negrito.

No primeiro passo, o gargalo na coluna 1 é eliminado. A banda é dividida igualmente entre os nós e certifica-se que as requisições subseqüentes são iguais ou menores aos da coluna 1.

No segundo passo, o gargalo na coluna 3 é eliminado. Novamente a banda é dividida

				Início				Primeiro passo			
-	β_0^1	β_0^2	β_0^3	-	1,00	0,67	0,33	-	1,00	0,67	0,33
β_1^0	-	β_1^2	β_1^3	0,33	-	1,00	0,67	0,33	-	1,00	0,67
β_2^0	β_2^1	-	β_2^3	0,67	0,33	-	1,00	<u>0,33</u>	0,33	-	1,00
β_3^0	β_3^1	β_3^2	-	1,00	0,67	0,33	-	<u>0,33</u>	<u>0,33</u>	0,33	-
l_0	l_1	l_2	l_3	2,00	2,00	2,00	2,00	1,00	1,67	2,00	2,00

Segundo passo				Terceiro passo				Quarto passo			
-	1,00	<u>0,33</u>	0,33	-	<u>0,33</u>	0,33	0,33	-	0,33	0,33	0,33
0,33	-	<u>0,33</u>	<u>0,33</u>	0,33	-	0,33	0,33	0,33	-	0,33	0,33
0,33	0,33	-	1,00	0,33	0,33	-	1,00	0,33	0,33	-	<u>0,33</u>
0,33	0,33	0,33	-	0,33	0,33	0,33	-	0,33	0,33	0,33	-
1,00	1,67	1,00	1,67	1,00	1,00	1,00	1,67	1,00	1,00	1,00	1,00

Figura 5.3: Ilustração da fase de cálculo de justiça do algoritmo LCR.

igualmente entre os nós e as requisições subsequentes são iguais ou menores às colunas anteriores, e assim sucessivamente.

5.2.3 Aplicação da Justiça

O LCR atribui quotas aos nós. No início do ciclo de justiça, o nó n_i atualiza sua quota sobre os enlaces com a taxa justa correspondente, ou seja, $tx_k = \beta_i^k \times C$ para $k = 0, \dots, L - 1$.

Dado um pacote de tamanho s destinado ao nó n_k , o nó n_i tem a permissão de transmitir o dado pacote somente se $tx_{i+1}, tx_{i+2}, \dots, tx_k$ forem maiores ou iguais a s . Se a transmissão for permitida, então o mecanismo de aplicação de justiça subtrai s de $tx_{i+1}, tx_{i+2}, \dots, tx_k$ para registrar a transmissão.

5.2.4 Detalhes de Implementação

A seguir são descritos detalhes de implementação referentes ao protocolo LCR.

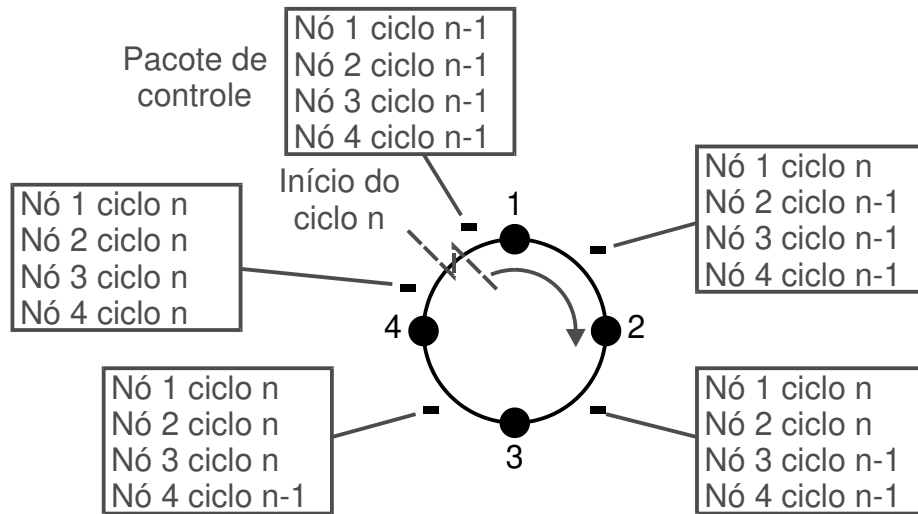


Figura 5.4: Conteúdo do pacote de controle em cada nó no ciclo n .

Pacote de Controle

O protocolo LCR utiliza um pacote de controle que circula pelo anel para distribuir os pedidos de requisição entre os nós. À medida que o pacote de controle circula pelo anel, este é atualizado pelos nós (Figura 5.4).

No Início do ciclo n , o nó 1 recebe o pacote de controle com informações de todos os nós no ciclo $n - 1$. O nó 1 atualiza o pacote de controle e o encaminha. O nó 2 recebe o pacote de controle com informações de todos os nós no ciclo $n - 1$ e do nó 1 no ciclo n , e assim por diante.

Entretanto, se todo nó simplesmente utilizasse a informação contida no pacote de controle, cada nó trabalharia com informações diferentes sobre o anel. Dessa forma, a distribuição não seria precisa.

Para que todos os nós trabalhem com a mesma informação, simplesmente é armazenado o conteúdo do pacote de controle no ciclo anterior. A Figura 5.5 ilustra o ciclo $n + 1$, com cada nó tendo armazenado localmente uma cópia do pacote de controle que recebeu no ciclo n .

Dessa forma, utilizando as informações armazenadas localmente e as informações do pacote de controle, é possível, para todos os nós, utilizar a mesma informação. Na Figura 5.5, todo nó pode utilizar as informações dos nós no ciclo n .

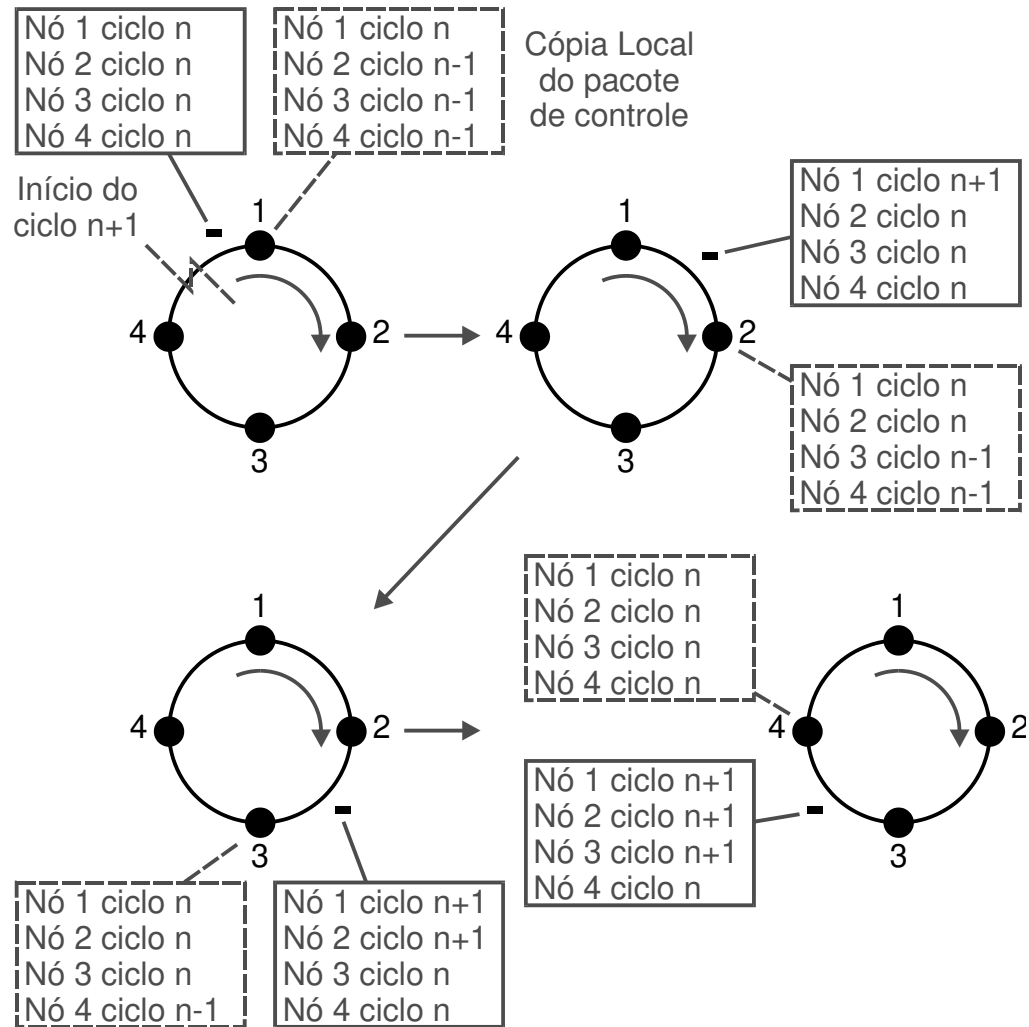


Figura 5.5: Anel no ciclo $n+1$. O conteúdo do pacote de controle no ciclo anterior está armazenado em cada nó.

Requisição por Banda

O nó, ao registrar no pacote de controle a requisição por banda, arredonda para cima a requisição de forma que a requisição seja múltipla ao tamanho do *slot*. Isso se deve ao fato de que, no caso da requisição ser menor que um *slot*, o nó não sofra *starvation*.

Dessa forma, um nó que possui dados para transmitir, sempre requisitará pelo menos 1 *slot*.

Distribuição da Banda de um Enlace Gargalo

Ao distribuir a banda de um enlace gargalo, a quota atribuída é sempre múltipla ao tamanho de um *slot*. Não há razão para atribuir a um nó uma fração de *slot* pois um *slot* não pode ser simplesmente parcialmente ocupado.

5.3 LCR com Justiça por par Origem-Destino

O protocolo LCR com Justiça por par Origem-Destino (LCR-SD) [25] é uma variação do protocolo LCR e implementa a política de justiça por par origem-destino, ao invés da justiça por nó de origem como no LCR original.

Em relação ao LCR, o LCR-SD implementa modificações no Passo 2, no cálculo da justiça e na aplicação da justiça.

5.3.1 Passo 2

A matriz de requisição transmitida pelo pacote de controle carrega a matriz r ,

$$r = [r_i^j]_{i=0,\dots,N-1,j=0,\dots,L-1},$$

onde r_i^j é a carga do nó n_i direcionado ao nó n_j , ou seja, a carga do fluxo $i - j$, sendo que a carga de um fluxo não pode ser maior que a capacidade do anel.

5.3.2 Cálculo da justiça

Inicialmente a matriz $\beta = \alpha$. Para eliminar um gargalo, a capacidade do enlace é dividida igualmente entre os fluxos que o utilizam.

O algoritmo para a computação das taxas parte do maior gargalo para o menor:

```

[ Seleciona o maior gargalo
  Enquanto houver gargalo
    Solucione o gargalo
    Seleciona maior gargalo
  Fim do laço
```

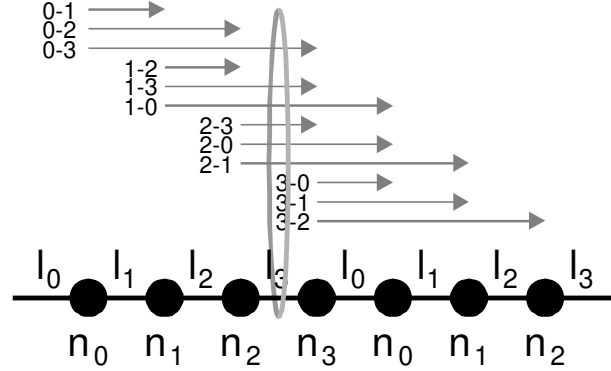


Figura 5.6: Carga dos fluxos sobre os enlaces. Alguns nós e enlaces estão repetidos para facilitar a visualização.

Para encontrar o maior gargalo, o algoritmo calcula a carga sobre todos os enlaces D_k . A carga sobre o enlace l_k é igual à soma da carga de todos os fluxos que passam por ele, e é dada por:

$$D_k = \sum_{i=0}^{N-1} \left(\sum_{j=k}^{|i-1+N|_N} \beta_i^j \right), \text{ para } i \neq k,$$

onde β_i^j é a taxa justa permitida ao fluxo $i - j$ e o símbolo $|\cdot|_N$ denota o operador módulo N . Para ilustrar esta função, observe a Figura 5.6, ela mostra a carga dos fluxos sobre o enlace l_3 .

Para eliminar um gargalo em um enlace l_k , o LCR-SD avalia as requisições dos fluxos sobre o enlace em ordem crescente de demanda.

Dada a demanda β_i^j do fluxo $i - j$, se $\beta_i^j > r_{fair}$, então $\beta_i^j = r_{fair}$, onde a taxa justa $r_{fair} = c/x$, sendo inicialmente $x = N \times (N - 1)/2$ e $c = 1$, onde x é o número de fluxos competindo pelo enlace e c a capacidade fracional restante no enlace.

Os valores de capacidade restante c e número de fluxos competindo pelo enlace x são atualizados conforme o algoritmo analisa as requisições. Portanto, após calcular β_i^j , o algoritmo atualiza a capacidade restante do enlace subtraindo a taxa justa calculada, ou seja, $c \leftarrow c - \beta_i^j$, e em seguida atualiza o número de nós restantes disputando pelo gargalo, ou seja, $x \leftarrow x - 1$.

O processo se repete até que todos os gargalos tenham sido eliminados.

Para ilustrar o funcionamento do protocolo, é apresentado a seguir um exemplo para um anel de 4 nós. Na Figura 5.7, a linha inferior em negrito indica a carga sobre cada enlace, por exemplo, D_0 representa a carga dos fluxos $\beta_1^0, \beta_2^0, \beta_3^0, \beta_3^1, \beta_3^2$ sobre o enlace l_0 .

				Início				Primeiro passo			
-	β_0^1	β_0^2	β_0^3	-	1,00	1,00	1,00	-	1,00	1,00	1,00
β_1^0	-	β_1^2	β_1^3	1,00	-	1,00	1,00	<u>0,17</u>	-	1,00	1,00
β_2^0	β_2^1	-	β_2^3	1,00	1,00	-	1,00	<u>0,17</u>	<u>0,17</u>	-	1,00
β_3^0	β_3^1	β_3^2	-	1,00	1,00	1,00	-	<u>0,17</u>	<u>0,17</u>	<u>0,17</u>	-
D_0	D_1	D_2	D_3	6,00	6,00	6,00	6,00	1,00	3,50	4,33	3,50

Segundo passo				Terceiro passo				Quarto passo			
-	1,00	<u>0,17</u>	<u>0,17</u>	-	<u>0,17</u>	0,17	0,17	-	0,17	0,17	0,17
<u>0,17</u>	-	<u>0,17</u>	<u>0,17</u>	0,17	-	0,17	0,17	0,17	-	0,17	0,17
0,17	0,17	-	1,00	0,17	0,17	-	1,00	0,17	0,17	-	<u>0,17</u>
0,17	0,17	0,17	-	0,17	0,17	0,17	-	0,17	0,17	0,17	-
1,00	1,83	1,00	1,83	1,00	1,00	1,00	1,83	1,00	1,00	1,00	1,00

Figura 5.7: Cálculo da taxa justa. Enlaces analisados estão destacados.

No primeiro passo, o gargalo no enlace l_0 é eliminado e a banda é dividida igualmente entre os seis fluxos que passam pelo enlace l_0 .

No segundo passo, o gargalo no enlace l_2 é eliminado. Novamente a banda é dividida igualmente entre os fluxos que passam pelo enlace, e assim sucessivamente.

5.3.3 Aplicação da Justiça

O LCR-SD atribui quotas aos fluxos. No início do ciclo de justiça, o nó n_i atualiza a quota dos fluxos com a taxa justa correspondente, ou seja, $tx_j = \beta_i^j \times C$ para $j = 0, \dots, L - 1$.

Dado um pacote de tamanho s destinado ao nó n_j , o nó n_i tem a permissão de transmitir o dado pacote somente se tx_j for maior ou igual a s . Se a transmissão for permitida, então o mecanismo de aplicação de justiça subtrai s de tx_j para registrar a transmissão.

5.4 Protocolo Ring Virtual Queue

O protocolo *Ring Virtual Queue* (RVQ) é um protocolo global que tem como objetivo oferecer justiça entre conexões TCP. Para obter tal resultado, o protocolo reproduz nas

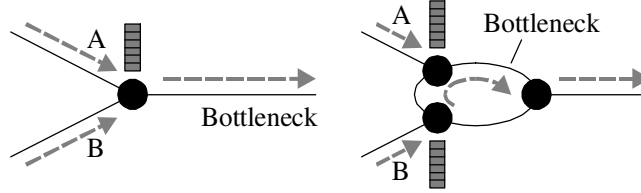


Figura 5.8: Gargalo em um enlace e em um anel.

filas do anel um comportamento tal como se todos os fluxos compartilhassem a mesma fila, pois quando fluxos compartilham a mesma fila (Figura 5.8 esquerda), as conexões TCP controlam suas taxas de transmissão promovendo justiça entre si.

Em roteadores, são amplamente adotadas filas FCFS (*first come first served*) com *Tail-Drop*. Neste tipo de fila, quando a carga dos fluxos A e B (Figura 5.8 esquerda) superam a capacidade do enlace, o comprimento da fila aumenta e ocorre perda de pacotes. A perda de pacotes em ambos os fluxos tende a ser proporcional às suas respectivas taxas de transmissão. Além disso, a divisão da banda do enlace gargalo entre os fluxos também tende a ser proporcional às suas taxas de transmissão.

Em um anel os fluxos não compartilham a mesma fila (direita da Figura 5.8), de fato, o protocolo utiliza uma fila para cada fluxo. O protocolo RVQ emula uma fila virtual, que é a união de todas as filas do anel, a fim de reproduzir o comportamento da esquerda da Figura 5.8. Para isso, o protocolo utiliza um nó líder, responsável por calcular a perda de pacotes e as taxas de transmissão. Informações são disseminadas entre os nós através de um pacote de controle que circula pelo anel. O protocolo trabalha em ciclos, cada passagem do pacote de controle pelo nó líder inicia um novo ciclo.

Os nós do anel enviam o tamanho de suas filas ao nó líder que, baseado no tamanho das filas e na carga sobre os enlaces do anel, estabelece como deve ocorrer a perda de pacotes e a vazão das filas a fim de emular a fila virtual.

A porcentagem de dados a descartar em uma fila é igual à razão entre a quantidade de dados que excede o tamanho máximo da fila virtual pelo tamanho máximo da fila virtual. A porcentagem de dados das filas a encaminhar pelo anel é igual à razão entre a quantidade de dados que podem ser transportados pelos enlaces do anel pela carga do enlace mais carregado.

Portanto, o protocolo RVQ executa dois procedimentos simultaneamente: controlar o tamanho e a vazão das filas.

Controle do tamanho das filas:

- a) O pacote de controle circula pelo anel coletando o tamanho de cada fila. Quando um nó recebe o pacote de controle, o tamanho de suas filas é adicionado a um campo

do pacote de controle. O nó líder utiliza este campo para recuperar o tamanho da fila virtual.

- b) Quando o pacote de controle chega ao nó líder, é determinado se o tamanho da fila virtual excedeu o seu tamanho limite. Caso tenha excedido, o nó líder determina o percentual da fila virtual que deve ser descartado. O nó líder então insere esta informação no pacote de controle e o envia pelo anel.
- c) Quando um nó do anel recebe o pacote de controle, ele realiza o descarte de pacotes de suas filas de acordo com o percentual informado pelo pacote de controle.

Controle de vazão das filas

- a) Cada fila informa no pacote de controle sua carga de transmissão sobre os enlaces do anel e armazena localmente no nó o tamanho das filas.
- b) Quando o pacote de controle chega ao nó líder, é determinado se a carga total da fila virtual sobre os enlaces do anel excede sua capacidade de transmissão. Caso tenha excedido, o nó líder determina o percentual da carga da fila virtual que pode ser transmitido em um ciclo. O nó líder então insere esta informação no pacote de controle e o envia pelo anel.
- c) Quando um nó recebe o pacote de controle, é atribuída às filas sua capacidade de transmissão, baseada no percentual informado pacote de controle e no tamanho das filas no ciclo anterior.

5.4.1 Descrição do Protocolo

Aqui é descrito o funcionamento do algoritmo, que é apresentado a seguir.

Nos Passos 1 e 2 do algoritmo são realizados o descarte da pacotes e a atribuição de banda relativos ao ciclo anterior. Os Passos 3 e 4 realizam a marcação de informações no pacote e controle destinados ao nó líder. Somente o nó líder realiza o Passo 5, que realiza os cálculos de descarte e atribuição de banda. Cada passo é explicado a seguir.

Passo 1 – ao receber o pacote de controle, o nó calcula a quantidade de dados a ser descartado. A quantidade de dados que pode ser transmitidos é armazenada em $queue_quota[i]$ e é igual à fração $c_queue_fraction$ da requisição realizada no ciclo anterior $queue_request[i]$. A fração $c_queue_fraction$ é determinada pelo nó líder.

Passo 2 – O nó determina o novo tamanho das filas baseado em informação do nó líder. O novo tamanho das filas $queue_size[i]$ é a porcentagem $c_queue_correction$ do tamanho das filas, determinada pelo nó líder.

Passo 3 – A soma do tamanho das filas é marcada no pacote de controle para ser utilizada pelo nó líder. $c_virtual_queue_size$ é a soma do tamanho de todas as filas no anel.

Passo 4 – A quantidade de dados das filas para se transmitir sobre cada enlace é marcada no pacote de controle e o tamanho das filas é armazenado no nó para ser utilizado no próximo ciclo. $c_link_load[i]$ é a quantidade total de dados para se transmitir sobre cada enlace e é atualizada baseada no tamanho das filas. Em $queue_request[i]$ é armazenado o tamanho das filas para ser utilizado no próximo ciclo no Passo 1.

Passo 5 – Este passo só é executado pelo nó líder.

- a) O percentual das filas que tem que ser descartado é calculado usando o comprimento da fila virtual e seu tamanho limite. $virtual_queue_limit$ é o tamanho máximo da fila virtual e $c_queue_correction$ a porcentagem das filas que devem ser descartadas.
- b) A soma do tamanho das filas é reinicializada. No próximo ciclo, esta informação é coletada novamente no próximo ciclo.
- c) O enlace mais carregado do anel é detectado e é calculada a fração dessa carga que pode ser transmitida sem exceder a capacidade do enlace. $c_queue_fraction$ é a fração das filas que podem ser transmitidas em um ciclo sem exceder a capacidade dos enlaces.
- d) A carga de transmissão sobre os enlaces é reinicializada. Esta informação é coletada novamente no próximo ciclo.

Passo 6 – Encaminha o pacote de controle.

5.4.2 Algoritmo do Protocolo

Na descrição do protocolo, o símbolo $[]$ indica que a variável é um vetor, e o prefixo $c_$ indica que a variável pertence ao pacote de controle. Ao receber o pacote de controle, os nós executam os seguintes passos:

Passo 1: cálculo da quantidade de dados a transmitir

Para todas as filas deste nó

$$queue_quota[i] = queue_request[i] \times c_queue_fraction$$

Passo 2: controle do comprimento das filas

Se $c_queue_correction > 0$ então

Para todas as filas deste nó

Descarte pacotes até que

$$queue_size[i] = queue_size[i] \times (1 - c_queue_correction)$$

Passo 3: Marcação do tamanho das filas

Para todas as filas deste nó

$$c_virtual_queue_size = c_virtual_queue_size + queue_size[i]$$

Passo 4: Marcação da quantidade de dados para se transmitir nos enlaces do anel

Para todos os enlaces do anel

Para todas as filas que passam pelo enlace

$$c_link_load[i] = c_link_load[i] + queue_size[i]$$

Para todas as filas deste nó

$$queue_request[i] = queue_size[i]$$

Passo 5: Cálculos do nó líder

Se este é o nó líder então

$$a) c_queue_correction = 1 - (virtual_queue_limit / c_virtual_queue_size)$$

$$b) c_virtual_queue_size = 0$$

$$c) \text{ Para o valor mais alto de } c_link_load[i]$$

$$c_queue_fraction = link_capacity / c_link_load[i]$$

$$d) \text{ Para todos os enlaces no anel}$$

$$c_link_load[i] = 0$$

Passo 6: Encaminha o pacote de controle

5.4.3 Exemplo

Para ilustrar o funcionamento do protocolo, é apresentado a seguir um exemplo para um anel de 4 nós.

Um nó, ao receber o pacote de controle, calcula no Passo 1 a quantidade de dados a transmitir para cada fila do nó baseado na requisição realizada no ciclo anterior. A variável $c_queue_fraction$, transmitida pelo pacote de controle, indica a fração da requisição que o anel é capaz de transportar. Quando o anel é capaz de transmitir todas as requisições realizadas, $c_queue_fraction$ é igual a 1.

Sendo a requisição realizada $queue_request[] = \{10000, 13000, 9000\}$ e a fração a ser transmitida $c_queue_fraction = 0,5$. A quantidade de dados que cada fila pode transmitir $queue_quota[] = queue_request[] * c_queue_fraction = \{5000, 6500, 4500\}$.

No Passo 2, o comprimento das filas é controlado. A variável $c_queue_correction$ do pacote de controle indica a fração das filas que deve ser descartada. Caso o valor de $c_queue_correction$ seja igual a 0,2, então 20% dos dados de cada uma das filas são descartados.

No Passo 3, é atualizada a variável do pacote de controle $c_virtual_queue_size$, que o nó líder utiliza para determinar o comprimento da fila virtual. Simplesmente, o nó soma

nesta variável o comprimento de todas as suas filas. Quando o pacote de controle circula pelo anel e retorna ao nó líder, a variável $c_virtual_queue_size$ possui a soma de todas as filas do anel, que é o tamanho da fila virtual.

Sendo $queue_size[] = \{13000, 12000, 11000\}$, então

$$c_virtual_queue_size = c_virtual_queue_size + 13000 + 12000 + 11000.$$

No Passo 4, é somada no pacote de controle a requisição dos fluxos de dados sobre os enlaces do anel, para que o nó líder possa determinar se existe gargalo no anel. Sendo o tamanho das filas $queue_size[] = \{13000, 12000, 11000\}$, o primeiro enlace que liga o nó ao próximo nó do anel, transporta dados de todos os fluxos, sendo a requisição sobre este primeiro enlace $13000 + 12000 + 11000 = 36000$, a requisição sobre o segundo enlace é $12000 + 11000 = 23000$ e a requisição sobre o terceiro enlace é 11000 . Essas requisições são somadas no vetor $c_link_load[]$ do pacote de controle. Quando o pacote de controle circula pelo anel e retorna ao nó líder, o vetor $c_link_load[]$ possui a soma de todas as requisições sobre os enlaces do anel.

Em seguida, o valor das requisições realizadas é armazenado no nó em um vetor $queue_request[]$ para uso no próximo ciclo. Então $queue_request[] \leftarrow queue_size[] = \{13000, 12000, 11000\}$.

O Passo 5, é executado somente pelo nó líder. São realizados cálculos baseados na variável $c_virtual_queue_size$ e no vetor $queue_request[]$.

$c_queue_correction$ é a fração das filas que deve ser descartada. As filas do anel devem ter pacotes descartados caso o tamanho da fila virtual exceda o tamanho máximo da fila virtual. Então

$$c_queue_correction = 1 - (virtual_queue_limit / c_virtual_queue_size).$$

$c_queue_fraction$ é a fração da requisição dos fluxos que o anel é capaz de transportar. Para calcular $c_queue_fraction$ é necessário localizar qual o enlace que recebeu a maior requisição, sendo $c_link_load[i]$ este enlace, e $link_capacity$ a capacidade de transmissão de um enlace em um ciclo do pacote de controle, então $c_queue_fraction = link_capacity / c_link_load[i]$.

A variável $c_virtual_queue_size$ e o vetor $queue_request[]$ são reiniciados.

No Passo 6, o pacote de controle é encaminhado ao próximo nó do anel.

5.4.4 Detalhes de Implementação

A seguir são descritos detalhes de implementação referentes ao protocolo RVQ.

Distribuição da Banda de um Enlace Gargalo

No Passo 1 do algoritmo, o valor atribuído a $queue_quota[i]$ é múltiplo do tamanho do *slot* arredondado para cima. Isso é devido a dois motivos: o valor é múltiplo do tamanho do *slot* porque um *slot* não pode ser parcialmente utilizado e não há sentido em atribuir

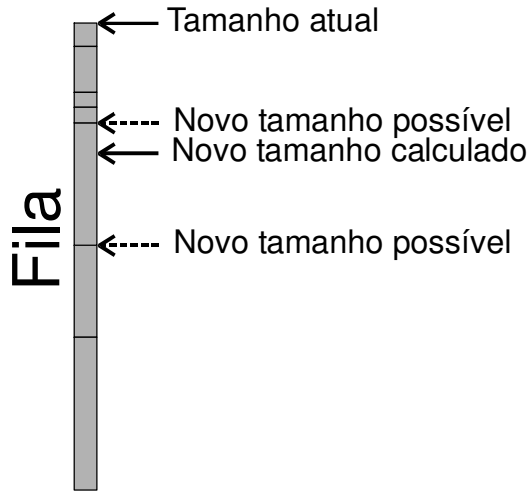


Figura 5.9: Descarte de pacotes de uma fila.

uma quota que é uma fração de *slot*, e o valor é arredondado para cima para que, no caso de pequenas requisições, o valor da quota seja pelo menos 1 *slot*, para que não ocorra *starvation*.

Descarte de pacotes

Considere a Figura 5.9, no Passo 2 o protocolo RVQ calcula o novo tamanho da fila, entretanto, geralmente não é possível descartar pacotes da fila de forma que o novo tamanho da fila seja exatamente o valor calculado. Normalmente existem dois tamanhos de fila possíveis, um maior e outro menor. É escolhido dos novos tamanhos possíveis o mais próximo ao valor calculado.

5.5 Protocolo TCP-Fair

Em certas situações, deseja-se obter justiça entre conexões TCP, como oferecido pelo protocolo RVQ, e ao mesmo tempo justiça de recepção de dados entre os nós. Considere o cenário apresentado na Figura 5.10. Neste cenário, usuários dos nós 1, 2 e 3 estão ligados à Internet pelo anel. Se estes nós são pontos de acesso de diferentes organizações, e têm igual direito sobre o anel, então, independentemente do número de usuários de cada organização, deseja-se que cada uma receba a mesma oportunidade de acesso aos recursos oferecidos no anel.

A política de justiça por nó de origem oferece aos nós do anel igualdade de envio de dados, entretanto, dependendo da situação, pode ser desejável obter igualdade de recepção de dados e não de envio. Um exemplo é acesso de usuários à Internet, onde os usuários

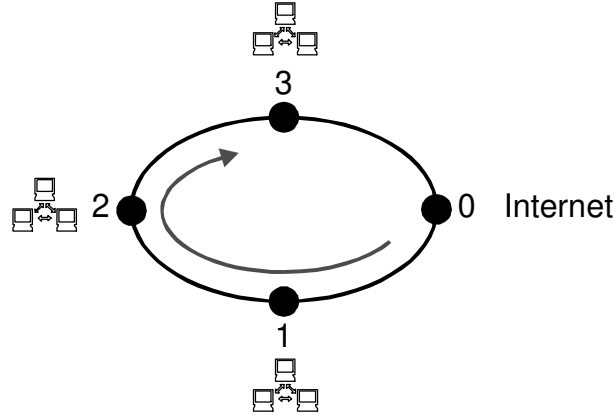


Figura 5.10: Cenário de utilização do protocolo TCP-Fair.

predominantemente realizam requisições de informações e recebem dados da Internet.

O protocolo TCP-Fair [26] é um protocolo global que oferece justiça de recepção entre os nós do anel e justiça entre as conexões TCP com um mesmo nó de destino. Para isso, o protocolo utiliza o mecanismo de fila virtual do protocolo RVQ, porém, cada nó possui uma fila virtual para recepção de dados, sendo que uma fila virtual é composta pelas filas com mesmo nó de destino. Isto permite justiça entre conexões TCP com um mesmo nó de destino.

Em comparação com o protocolo RVQ, no protocolo TCP-Fair cada nó é líder de sua própria fila virtual.

5.5.1 Descrição do Protocolo

A função principal do protocolo é gerenciar o tamanho e a vazão das filas. O protocolo trabalha com um pacote de controle que dissemina as informações entre os nós do anel. Os nós possuem uma fila para cada destino possível.

Nos Passos 1, 2 e 3 é controlada a vazão das filas. Nos Passos 4 e 5 é controlado o comprimento das filas e no Passo 6 é encaminhado o pacote de controle. Os passos estão descritos detalhadamente a seguir. Seja i o nó atual:

Passo 1 – O pacote de controle carrega o vetor $c_virtual_queue_request[j]$, que indica o total em bytes das requisições direcionados ao nó j . O pacote de controle carrega também o vetor $c_virtual_queue_quota[j]$, que indica a quantidade de dados em bytes que pode ser enviada ao nó j em um ciclo. No nó i , para toda fila virtual $j \neq i$, se a fila virtual j é capaz de dar vazão a todos os dados direcionados a ela então toda a banda requisitada pela fila $i - j$ é atribuída à fila, senão, é diminuída a vazão da fila $i - j$

proporcionalmente à capacidade de recepção da fila virtual j .

Passo 2 – O pacote de controle carrega o vetor $c_link_load[]$, que registra o total das requisições das filas sobre os enlaces. O nó i atualiza este vetor de acordo com os comprimentos de suas filas.

Passo 3 – Se o enlace mais carregado, $c_link_load[k]$, é maior que a capacidade de transmissão dos enlaces do anel, $link_capacity$, então a vazão de todas as filas é diminuída de forma proporcional diminuindo a carga sobre o enlace $c_link_load[k]$ até eliminar o gargalo, evitando injustiça na transmissão de dados no anel.

Passo 4 – a) Atualiza a carga deste nó, i , direcionada aos outros nós, $j \neq i$, do anel no vetor $c_virtual_queue_size[j]$.

b) Atualiza a carga direcionada a este nó $c_virtual_queue_request[i]$ no ciclo que acabou.

c) Atualiza a carga que deverá ser encaminhada a este nó no próximo ciclo na variável $c_virtual_queue_quota[i]$.

Passo 5 – Caso uma fila virtual exceda seu tamanho limite, então o comprimento da fila virtual é ajustado, de forma a realizar um descarte proporcional de pacotes para que se mantenha a justiça entre as conexões TCP. Esta proporcionalidade no descarte dos pacotes é um ponto crucial para se manter a justiça entre as conexões TCP, pois o descarte sinaliza o congestionamento às várias conexões TCP de forma a manter a justiça entre elas.

Passo 6 – Encaminha o pacote de controle para o próximo nó do anel.

5.5.2 Algoritmo do Protocolo

O símbolo $[]$ indica que a variável é um vetor, e o prefixo $c_$ indica que a variável pertence ao pacote de controle. Ao receber o pacote de controle, os nós executam os seguintes passos:

Passo 1: Calcula a vazão das filas

No nó i , para toda fila virtual j

Se $c_virtual_queue_request[j] \leq c_virtual_queue_quota[j]$ então

$queue_quota[i, j] = queue_request[i, j]$

senão

$queue_quota[i, j] = queue_request[i, j] \times \frac{c_virtual_queue_quota[j]}{c_virtual_queue_request[j]}$

Passo 2: Atualiza no pacote de controle a carga sobre os enlaces

No nó i , para todo enlace k

Para toda fila $i - j$ que utiliza o enlace k

$c_link_load[k] = c_link_load[k] - queue_request[i, j] + queue_size[i, j]$

Passo 3: Diminui a vazão das filas caso exista gargalo no anel

Se $c_link_load[k] > link_capacity$ então

$$queue_quota[i, j] = queue_quota[i, j] \times \frac{link_capacity}{c_link_load[k]}$$

Passo 4: Atualiza no pacote de controle as informações das filas

a) Atualiza no pacote de controle as informações das filas virtuais

No nó i , para todo nó de destino $j \neq i$

$$c_virtual_queue_size[j] =$$

$$c_virtual_queue_size[j] - queue_request[i, j] + queue_size[i, j]$$

$$queue_request[i, j] = queue_size[i, j]$$

b) Atualiza no pacote de controle a carga direcionada a este nó no ciclo que acabou

$$c_virtual_queue_request[i] = c_virtual_queue_size[i]$$

c) Atualiza a carga que deverá ser encaminhada a este nó no próximo ciclo

Se $c_virtual_queue_request[i] < virtual_queue_capacity[i]$ então

$$c_virtual_queue_quota[i] = c_virtual_queue_request[i]$$

senão

$$c_virtual_queue_quota[i] = virtual_queue_capacity[i]$$

Passo 5: Controla o comprimento das filas

No nó i , para todo nó de destino $j \neq i$

$$queue_size[i, j] = queue_size[i, j] \times \frac{virtual_queue_limit[j]}{virtual_queue_request[j]}$$

Passo 6: Encaminha o pacote de controle para o próximo nó do anel.

5.5.3 Exemplo

Para ilustrar o funcionamento do protocolo, é apresentado a seguir um exemplo para um anel de 4 nós. Neste exemplo da Figura 5.11, os 4 nós são campus onde seu usuários realizam, em um típico ambiente HTTP, a requisição de informações em outros campus.

O protocolo TCP-Fair garante a mesma capacidade de recepção de dados a todos os nós, ideal para este ambiente, isto é realizado no Passo 4-c, onde se limita a quantidade de dados que pode ser transmitida aos nós. Neste passo, o valor da variável $c_virtual_queue_quota[j]$ corresponde à carga direcionada ao nó j . Para todos os nós, é sempre menor ou igual a $virtual_queue_capacity[j]$, que é o limite de dados que um nó pode receber em um ciclo.

O Passo 4-c realiza o cálculo da quantidade de banda que pode ser transmitida a um nó, enquanto o Passo 1 realiza a atribuição da banda às filas. No Passo 1, é realizada a atribuição de quotas baseado em $c_virtual_queue_quota[]$,

$$queue_quota[i, j] = queue_request[i, j] \times \frac{c_virtual_queue_quota[j]}{c_virtual_queue_request[j]}.$$

O Passo 1 distribui entre as filas com dados direcionados ao nó j , a banda que o nó j pode receber $c_virtual_queue_quota[j]$.

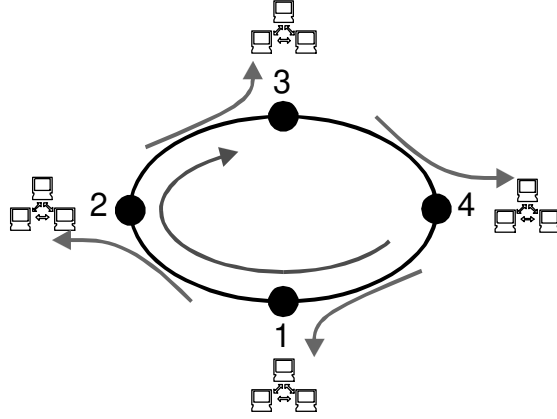


Figura 5.11: No protocolo TCP-Fair, é garantida justiça na recepção de dados dos nós do anel.

Prova: dado que $c_virtual_queue_request[j]$ é a soma de todas as requisições destinadas ao nó j ,

$$c_virtual_queue_request[j] = \sum_{i=0}^{N-1} queue_request[i, j],$$

a soma de todas quotas distribuídas no Passo 1 direcionadas ao nó j , é igual a:

$$\sum_{i=0}^{N-1} queue_quota[i, j] = \sum_{i=0}^{N-1} queue_request[i, j] \times \frac{c_virtual_queue_quota[j]}{c_virtual_queue_request[j]},$$

$$\sum_{i=0}^{N-1} queue_quota[i, j] = \frac{c_virtual_queue_quota[j]}{c_virtual_queue_request[j]} \times \sum_{i=0}^{N-1} queue_request[i, j],$$

$$\sum_{i=0}^{N-1} queue_quota[i, j] = \frac{c_virtual_queue_quota[j]}{c_virtual_queue_request[j]} \times c_virtual_queue_request[j],$$

$$\sum_{i=0}^{N-1} queue_quota[i, j] = c_virtual_queue_quota[j].$$

A outra característica do protocolo TCP-Fair é a justiça entre conexões TCP. As conexões TCP com mesmo nó de destino recebem a mesma banda (Figura 5.12).

A distribuição de banda entre as conexões TCP não é realizada pelo protocolo, o controle da distribuição da banda é realizado pelos *end-systems*. O princípio utilizado é o mesmo do protocolo RVQ, o protocolo TCP-Fair realiza o descarte de pacotes nas filas de forma a sinalizar congestionamento às conexões TCP, as quais controlam sua taxa de transmissão e dividem a banda passante.

As filas com um mesmo nó de destino devem realizar o descarte de pacotes conjuntamente. Isto é realizado nos Passos 4-b e 5. No Passo 4-b, o valor da variável

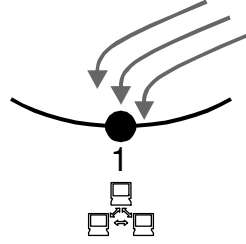


Figura 5.12: As conexões TCP com um mesmo destino recebem a mesma banda passante.

$c_virtual_queue_request[i]$ é fixado no pacote de controle e todas as filas com dados destinado ao nó i utilizarão o mesmo valor. No passo 5,

$$queue_size[i, j] = queue_size[i, j] \times \frac{virtual_queue_limit[j]}{c_virtual_queue_request[j]},$$

O valor $virtual_queue_limit[j]$ é o comprimento máximo da fila virtual e é pré-definido e fixo. Dessa forma, a fração

$$virtual_queue_limit[j]/c_virtual_queue_request[j]$$

é a mesma para todas as filas com dados destinado ao nó i . Assim, a mesma fração das filas é descartada.

Os Passos 2 e 4-a do protocolo realizam a atualização de variáveis. O Passo 3 controla a vazão das filas de forma a evitar gargalos no anel. O Passo 6 realiza o encaminhamento do pacote de controle.

5.6 Re-inserção do pacote de controle

Os protocolos apresentados requerem a recepção de um pacote de controle, seu processamento e em seguida o envio para o próximo nó do anel; portanto, o pacote de controle não é enviado no mesmo *slot* em que foi recebido.

Desta forma, o seguinte problema ocorre: no momento que um nó envia o pacote de controle, este tem sua quota de transmissão renovada e passa a utilizar todos os *slots* livres seguintes. Este comportamento afeta o nó seguinte do anel, pois no momento em que o pacote de controle é recebido pelo nó seguinte, uma rajada de *slots* ocupados é recebida, impedindo que o pacote de controle seja encaminhado aumentando consideravelmente o tempo de um ciclo e prejudicando a eficiência dos protocolos (Figura 5.13).

Uma forma de diminuir este problema é através da não utilização de um *slot* vazio após um determinado intervalo de tempo após o envio do pacote de controle. Caso se saiba que um nó leva 50 microssegundos para processar o pacote de controle, então, após 50 microssegundos do envio do pacote de controle, deixa-se de utilizar o primeiro *slot* vazio, permitindo que este *slot* vazio seja utilizado pelo nó seguinte, para envio do pacote

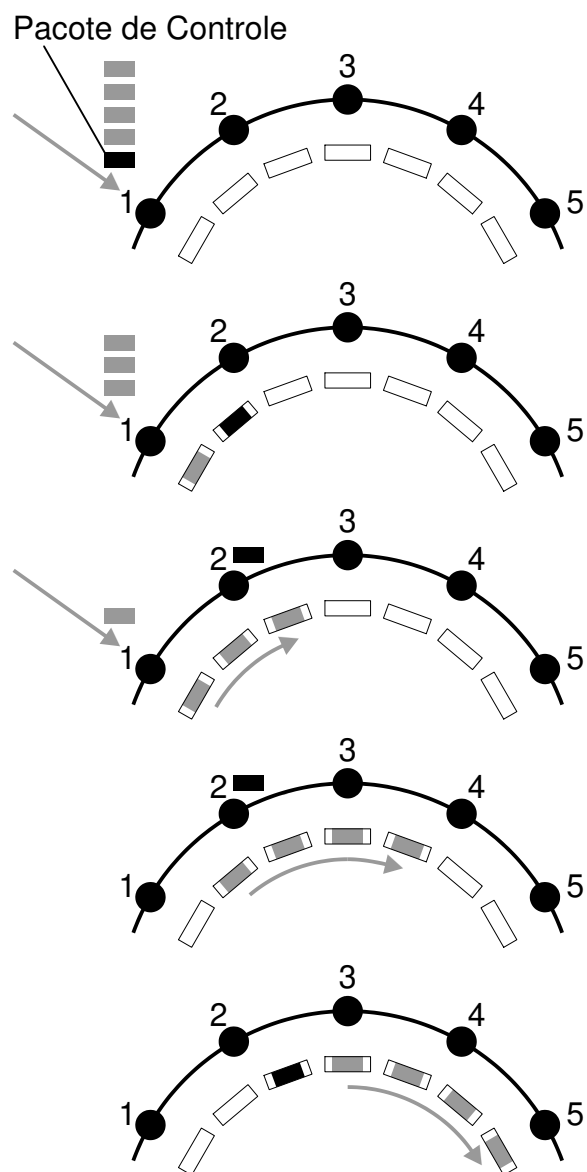


Figura 5.13: Burst de pacotes impedindo o encaminhamento do pacote de controle.

de controle. Esta pequena implementação provê um aumento significativo na vazão destes protocolos.

Capítulo 6

Resultados Numéricos

Neste capítulo, são apresentados resultados de simulações. As simulações foram realizadas utilizando o Network Simulator (NS-2) e foram implementados módulos adicionais que simulam as características de uma rede óptica em anel. O Apêndice 1 descreve as características destes módulos.

As simulações foram realizadas com tráfego UDP e TCP. Para simular o tráfego UDP, foi utilizado um gerador de tráfego com a chegada de pacotes em intervalos exponenciais e tamanho de pacotes seguindo a distribuição apresentada em [22] (Tabela 6.1). Simulações com vários percentuais de carga UDP foram realizados, mas não mostraram alteração significativa nos resultados das simulações. Os resultados apresentados utilizam uma carga UDP de 10% da capacidade de vazão do anel. Para o tráfego TCP, foi utilizado o TCP Reno com a opção `rfc2988_`, uma fonte de tráfego FTP infinita e um receptor do tipo *Delayed ACK*.

As simulações consideram um anel com capacidade de transmissão de 2,5Gbps e *slots* de 3100 bytes. A velocidade de propagação da luz na fibra é de 200.000 Km/s, comum em fibras comerciais. Os *hosts* que geram o tráfego estão fora do anel, ligados a este por enlaces de 25Gb/s, a velocidade elevada destes enlaces serve para evitar que estes não se tornem gargalos e influenciem no comportamento do tráfego no anel.

Tamanho do Pacote (Bytes)	Probabilidade (%)
64	60
596	15
700	5
800	5
1100	7
1500	8

Tabela 6.1: Distribuição do tamanho de pacotes UDP.

Os *slots* são *Jumbo Datagrams*, ou seja, cada *slot* transporta vários datagramas IP e seguem o modelo do protocolo *Packet Aggregate Transmission* (PAT) [23].

Para o protocolo MetaRing foi utilizado $k = l = 2 \times \text{número_de_slots_no_anel}$, sendo l a quantidade mínima de slots que o protocolo garante de transmissão a cada nó a cada passagem do sinal SAT, e k a quantidade máxima de pacotes que cada nó pode transmitir a cada ciclo do sinal SAT. O valor $k = 2 \times \text{número_de_slots_no_anel}$ permite que, caso somente um nó tenha dados para transmitir, o nó seja capaz de utilizar toda a banda do anel. A igualdade $k = l$ ajuda a manter a justiça entre os nós do anel.

Foram realizadas simulações em três cenários de tráfego para analisar diferentes características dos protocolos. O primeiro grupo de simulações analisa a capacidade dos protocolos de oferecer justiça, o cenário utilizado apresenta diferentes números de conexões TCP em cada fluxo. O segundo grupo de simulações analisa a capacidade de vazão dos protocolos no transporte de tráfego TCP de longa duração e tráfego UDP. O tráfego TCP de longa duração é gerado por fontes de dados FTP com uma quantidade infinita de dados para transmitir, enquanto o tráfego UDP ocupa uma banda igual a 10% da capacidade do anel. O terceiro grupo de simulações analisa a capacidade de vazão do anel em presença de tráfego TCP de curta duração e tráfego UDP. O tráfego TCP de curta duração é gerado por fontes de dados que realizam a transmissão de dados com tamanho seguindo uma distribuição híbrida de cauda longa lognormal-pareto, enquanto o tráfego UDP ocupa uma banda igual a 10% da capacidade do anel. O tráfego TCP de curta duração utilizada é apresentada em detalhes junto aos resultados.

6.1 Cenário de Exemplificação da Justiça Promovida pelos Protocolos

Cada política de justiça apresenta uma concepção diferente de justiça. Neste primeiro cenário de simulação é utilizada uma configuração de tráfego que visa expor os diferentes tipos de justiça oferecidos pelos protocolos.

Nesta simulação, é utilizado um anel de 50 Km, onde os fluxos são compostos por conexões TCP que formam um gargalo no enlace 3-4 como ilustrado na Figura 6.1. Os fluxos 1-6, 3-4 e 3-5 possuem 100 conexões TCP, o fluxo 2-6 possui 300 conexões TCP e o fluxo 3-6 possui 200 conexões TCP.

Como são utilizadas conexões TCP, são produzidos fluxos de ACK, porém, a banda ocupada por estes fluxos é pequena e nenhum destes fluxos cruza o enlace gargalo 3-4.

A política de justiça por nó de origem (Metaring e LCR) divide a banda do enlace gargalo igualmente entre os nós do anel. No cenário, os nós 1, 2 e 3 dividem igualmente entre si a banda do enlace gargalo 3-4. O nó 3 origina três fluxos, porém, nesta política de

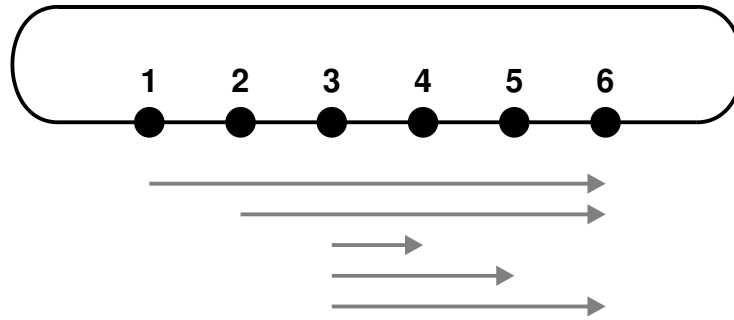


Figura 6.1: Cenário de justiça. Cada política de justiça deve distribuir diferentemente a banda entre os fluxos.

justiça, não existe regra para a divisão da banda entre fluxos originados com um mesmo nó de origem.

A política de justiça por par origem-destino (LCR-SD) divide a banda do enlace gargalo igualmente entre os fluxos de dados. No cenário, a mesma banda é oferecida a cada fluxo, ou seja, 20% da banda do enlace gargalo.

O protocolo TCP-Fair utiliza um novo conceito de justiça. O protocolo oferece justiça de recepção de dados entre os nós, esta característica é interessante quando os nós do anel priorizam a recepção de dados, por exemplo, nós requisitando dados da Internet. Neste cenário, os fluxos que disputam por banda no enlace gargalo são destinados a três nós distintos, e cada nó de destino recebe $1/3$ da banda. O nó 6 recebe três fluxos com diferentes quantidades de conexões em cada um. O protocolo, em conjunto com o mecanismo de justiça do TCP, permite a divisão da banda de forma justa entre as conexões que possuem um mesmo destino. Os fluxos 1-6, 2-6 e 3-6 são compostos respectivamente por 100, 300 e 200 conexões TCP, totalizando 600 conexões destinadas ao nó 6. Assim, a vazão de 33% do nó 6 é distribuída de forma justa entre as conexões, atribuindo aos fluxos 1-6, 2-6 e 3-6 respectivamente 5,56%, 16,67% e 11,11% da banda.

O protocolo RVQ oferece justiça de vazão entre conexões TCP. Neste cenário, existem 800 conexões TCP no anel que disputam por banda no enlace gargalo. Para oferecer justiça entre as conexões TCP, cada um dos cinco fluxos de dados recebe banda proporcionalmente ao número de conexões TCP em cada uma. Os fluxos 1-6, 2-6, 3-4, 3-5 e 3-6 são compostos respectivamente por 100, 300, 100, 100 e 200 conexões TCP, assim, a vazão de cada fluxo é respectivamente 12,5%, 37,5%, 12,5%, 12,5% e 25% da banda do anel.

Como esta simulação tem a intenção de medir o impacto dos protocolos MAC sobre a justiça do protocolo TCP, neste cenário não são utilizados fluxos de dados UDP, pois o tráfego UDP prejudica a justiça promovida pelo protocolo TCP, por ser um tipo de

Fluxo	Conexões	Nó de Origem	Par Origem-Destino	TCP-Fair	RVQ
1 – 6	100	33, 33%	20, 00%	5, 56%	12, 50%
2 – 6	300	33, 33%	20, 00%	16, 67%	37, 50%
3 – 4	100		20, 00%	33, 33%	12, 50%
3 – 5	100	33, 33%	20, 00%	33, 33%	12, 50%
3 – 6	200		20, 00%	11, 11%	25, 00%

Tabela 6.2: Distribuição da banda de cada política de justiça.

Fluxo	Metaring	LCR	LCR-SD
1 – 6	33, 37% $\pm$ 0, 36%	33, 03% $\pm$ 0, 46%	19, 96% $\pm$ 0, 05%
2 – 6	33, 30% $\pm$ 0, 23%	33, 05% $\pm$ 0, 06%	19, 87% $\pm$ 0, 10%
3 – 4	10, 90% $\pm$ 0, 62%	11, 17% $\pm$ 0, 51%	20, 08% $\pm$ 0, 08%
3 – 5	10, 57% $\pm$ 0, 23%	10, 81% $\pm$ 0, 33%	20, 01% $\pm$ 0, 07%
3 – 6	11, 85% $\pm$ 0, 46%	11, 94% $\pm$ 0, 40%	20, 08% $\pm$ 0, 09%

Fluxo	Conexões	TCP-Fair	RVQ
1 – 6	100	5, 91% $\pm$ 0, 54%	12, 62% $\pm$ 0, 16%
2 – 6	300	18, 32% $\pm$ 0, 41%	37, 58% $\pm$ 0, 77%
3 – 4	100	32, 77% $\pm$ 0, 12%	12, 69% $\pm$ 0, 19%
3 – 5	100	32, 74% $\pm$ 0, 13%	12, 53% $\pm$ 0, 14%
3 – 6	200	10, 26% $\pm$ 0, 11%	24, 58% $\pm$ 0, 51%

Tabela 6.3: Distribuição da banda de cada protocolo.

tráfego que não responde a congestionamentos da rede.

A Tabela 6.2 resume a distribuição de banda de cada política de justiça e é a distribuição ideal de banda de acordo com cada perspectiva de justiça.

A Tabela 6.3 apresenta o resultado das simulações e mostra que a divisão da banda realizada pelos protocolos está próxima ao ideal de acordo com a Tabela 6.2.

O intervalo de confiança foi obtido através de 10 simulações e da utilização da tabela t-student.

Analisando a Tabela 6.3, o Metaring e o LCR oferecem justiça entre nós de origem, atribuindo aproximadamente 33,33% da banda entre os nós, entretanto, o nó 6 recebe muito mais dados que os outros nós (mais de 78%), portanto, não existe justiça quanto à recepção de dados.

O protocolo Metaring obteve, para os fluxos 1-6 2-6 3-4 3-5 e 3-6, respectivamente os índices de justiça de 0,76 0,72 0,63 0,56 e 0,57. O protocolo obteve, como menor índice de justiça, o valor 0,56, que é 25,4% menor que o valor de referência. O maior índice de

justiça foi 0,76, 1,2% superior ao valor de referência.

O protocolo LCR obteve, para os fluxos 1-6 2-6 3-4 3-5 e 3-6, respectivamente os índices de justiça de 0,76 0,75 0,60 0,59 e 0,60. O menor índice de justiça foi 0,59, 21,4% menor que o índice de referência, e o maior índice foi 0,76, 1,2% superior ao valor de referência.

O LCR-SD oferece justiça entre fluxos, aproximadamente 20% da banda a cada fluxo, característica interessante ao TCP, por minimizar a assimetria de caminhos. O nó 3 possui maior banda no envio de dados, pois é o nó que origina mais fluxos (60,17%). O nó 6 é o que recebe mais dados, pois é o nó que recebe mais fluxos (59,91%).

O TCP-Fair oferece justiça de recepção de dados, dividindo a banda do enlace gargalo de forma a manter igualdade de recepção de dados entre os nós 4, 5 e 6. A Tabela 6.3 mostra que o protocolo oferece aproximadamente 1/3 da banda de recepção entre os nós de destino. Os fluxos 1-6, 2-6 e 3-6 compartilham a banda de acordo com o número de conexões TCP, permitindo justiça entre as conexões TCP com destino ao nó 6.

O RVQ permite que as conexões TCP controlem a ocupação da banda, atingindo uma ocupação justa e próxima à ideal. Cada conexão TCP nos fluxos 1-6, 2-6, 3-4, 3-5 e 3-6 recebeu em média respectivamente 1,26%, 1,25%, 1,27%, 1,25% e 1,23% da banda do anel.

Este cenário mostra a importância da escolha da política de justiça, pois a distribuição de banda é radicalmente diferente dependendo da política adotada.

Foi medido também o grau de justiça entre conexões TCP utilizando o índice de justiça de Jain [18]. Neste índice, um valor de 1 indica uma rede totalmente justa, enquanto um valor 0 uma rede totalmente injusta. O índice de Jain é calculado através da seguinte fórmula:

$$f(x) = \frac{\left[\sum_{x=1}^n x_i \right]^2}{\sum_{x=1}^n x_i^2} \quad x_i \geq 0$$

Este índice mede a igualdade na alocação de banda x dos usuários. Se todos os usuários recebem a mesma banda, ou seja, x é igual para todos os usuários, então o índice de justiça é igual a 1. Conforme a desigualdade aumenta, a justiça diminui e o índice se aproxima a 0.

Como referência, foi realizada uma simulação de comunicação de 200 conexões TCP entre dois nós ligados através de um roteador por enlaces de 0,5 gigabits por segundo e com um *round trip time* (RTT) médio de 46 milissegundos. O índice de Jain resultante foi 0,7507 com um intervalo de confiança de 90% de $\pm 0,0182$. O valor 0,7507 é utilizado a seguir como referência.

Na simulação do protocolo TCP-Fair, o índice foi aplicado às conexões TCP com mesmo nó de destino para se analisar o nível de justiça oferecido pelo protocolo às conexões TCP com o mesmo nó de destino, e demonstrar que o protocolo oferece a justiça que

propõe. Nas simulações, o índice de justiça, com intervalo de confiança de 90%, foi de $0,68 \pm 0,16$, $0,76 \pm 0,10$, $0,73 \pm 0,04$ respectivamente entre as conexões TCP com destino aos nós 4, 5 e 6. Os índices de justiça atingidos estão próximos ao índice de referência 0,7507. O valor 0,68 é 9,4% inferior ao valor de referência e o valor 0,76 é 1,2% superior ao valor de referência.

Na simulação do protocolo RVQ, o índice foi aplicado englobando todas as conexões TCP a fim de se analisar a justiça oferecida pelo protocolo entre as conexões, resultando em um índice de justiça, com intervalo de confiança de 90%, de $0,82 \pm 0,02$. Este valor é 9,2% superior ao valor de referência, indicando que o protocolo RVQ proporcionou um maior grau de justiça entre as conexões TCP quando comparado ao cenário de referência.

Os protocolos Metaring, LCR e LCR-SD não têm como objetivo oferecer justiça entre conexões TCP em diferentes fluxos. Nestes protocolos, existe justiça apenas entre conexões TCP de um mesmo fluxo. O índice de justiça das conexões TCP de cada fluxo é apresentado a seguir.

O protocolo LCR-SD obteve, para os fluxos 1-6 2-6 3-4 3-5 e 3-6, respectivamente os índices de justiça de 0,76 0,78 0,76 0,78 0,78. Todos os índices de justiça são superiores ao valor de referência, chegando a um máximo de 0,78, 3,9% superior ao índice de referência, indicando que o protocolo LCR-SD proporciona um maior grau de justiça entre as conexões TCP.

6.2 Simulações com Tráfego de Longa Duração

As simulações desta seção foram realizadas com tráfego TCP e fonte de dados FTP com uma quantidade infinita de dados para transmissão e tráfego UDP ocupando 10% da capacidade do anel. Foram realizadas simulações em dois cenários, um simétrico, onde a carga dos nós é igualmente distribuída e uma simulação em cenário assimétrico, onde um nó especial recebe dados de todos os outros nós e também transmite para todos os outros nós.

6.2.1 Cenário Simétrico

Este cenário analisa o desempenho dos protocolos quanto à vazão. Neste cenário, todos os nós transmitem para todos os outros nós, sendo que 10% da capacidade do anel são ocupados por tráfego UDP e o restante da capacidade é ocupado por 1000 conexões TCP com uma quantidade infinita de dados para transmissão.

Simulações com outros percentuais de carga UDP foram realizados sem demonstrar alteração significativa na vazão.

As simulações foram realizadas variando o número de nós e o comprimento do anel.

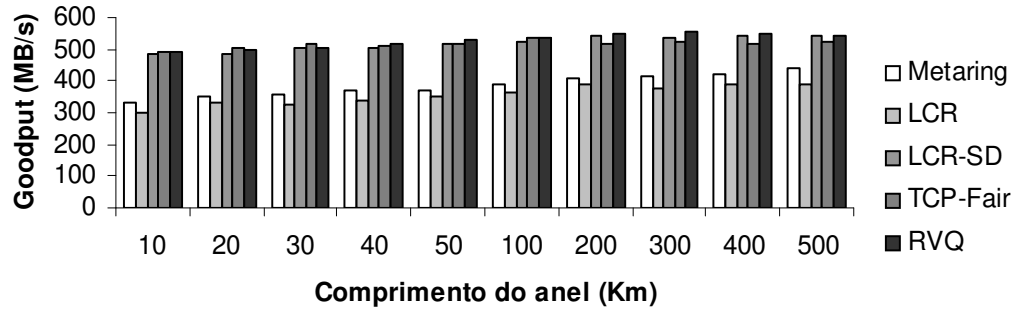


Figura 6.2: Vazão dos protocolos no cenário simétrico com 4 nós variando o comprimento do anel.

A Figura 6.2 mostra que a vazão dos protocolos é maior com um anel de maior comprimento. Isso é explicado pela quantidade de *slots* no anel. O número maior de *slots* torna a distribuição de *slots* entre fluxos mais precisa. Quando existem poucos *slots*, os protocolos LCR, LCR-SD e TCP-Fair dependem de aproximações para distribuir os *slots*.

Outro fator é a ocupação da banda com informações de controle. Se um anel de 10 Km possui 10 *slots* e um *slot* é ocupado pelo pacote de controle, então 10% da banda do anel é ocupado pelo pacote de controle. Se um anel de 100 Km possui 100 *slots*, então 1% da capacidade é ocupada com o pacote de controle. A Figura 6.3 mostra como o comprimento do anel influencia no tempo do ciclo do pacote de controle.

Variando o comprimento do anel, os protocolos Metaring, LCR, LCR-SD, TCP-Fair e RVQ obtiveram um *goodput* médio de 385, 356, 518, 515, 527 MB/s, respectivamente.

O protocolo Metaring utiliza um valor fixo para a atribuição de quotas, ou seja, independente da carga sobre o anel, a quota atribuída aos nós é a mesma. Dessa forma, neste cenário, onde há saturação do tráfego no anel, as quotas atribuídas são maiores que a capacidade de vazão do anel, provocando um atraso no encaminhamento do pacote de controle, pois os nós não encaminham o pacote de controle enquanto não se tornam satisfeitos. Isto se reflete em uma menor frequência de ciclos do pacote de controle, observado na Figura 6.3, e conseqüentemente em uma menor frequência na renovação de quotas e em uma menor vazão, observado na Figura 6.2.

O protocolo LCR, devido ao cálculo da matriz de justiça, resulta em uma distribuição de quotas menos eficiente que os protocolos LCR-SD, TCP-Fair e RVQ, visível na Figura 6.2.

Os mecanismos de atribuição de banda dos protocolos LCR-SD, TCP-Fair e RVQ mostram-se mais eficientes quanto à vazão que oferecem comparados aos protocolos Metar-

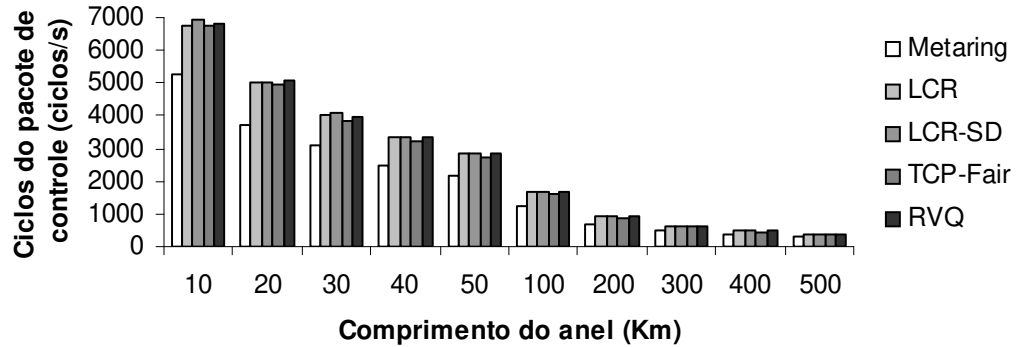


Figura 6.3: Ciclos do pacote de controle.

ing e LCR (Figura 6.2).

A Figura 6.4 mostra que, em geral, a vazão dos protocolos é maior com um número menor de nós. Isso é devido ao pacote de controle que, por ser transmitido na mesma direção que os dados, sofre com o problema apresentado na seção de re-inserção do pacote de controle. Assim, à medida que o número de nós aumenta, torna-se maior o tempo médio do encaminhamento do pacote de controle (Figura 6.5).

Outro fator é a distribuição das quotas entre os fluxos do anel. Com 18 nós, existem 306 fluxos. O número elevado de fluxos fraciona significativamente a banda do anel, prejudicando a vazão.

Variando o número de nós do anel, os protocolos Metaring, LCR, LCR-SD, TCP-Fair e RVQ obtiveram um *goodput* médio de 356 MB/s, 267 MB/s, 469 MB/s, 508 MB/s e 490 MB/s respectivamente.

Os protocolos LCR-SD, TCP-Fair e RVQ apresentam uma maior vazão comparados aos protocolos Metaring e LCR.

6.2.2 Cenário Assimétrico

No cenário assimétrico, também, é avaliada a vazão, porém a carga no anel não é balanceada como no cenário simétrico. Existe um nó especial que transmite e recebe de todos os outros nós (e.g. nó 4 na Figura 6.6).

Comparando ao cenário simétrico, os protocolos Metaring e TCP-Fair apresentam uma vazão menor (Figura 6.7), o LCR apresenta uma vazão maior e os protocolos LCR-SD e RVQ não apresentaram alteração significativa na vazão (aproximadamente 5%).

A vazão tende a ser maior com anéis de maior comprimento e, assim como no cenário

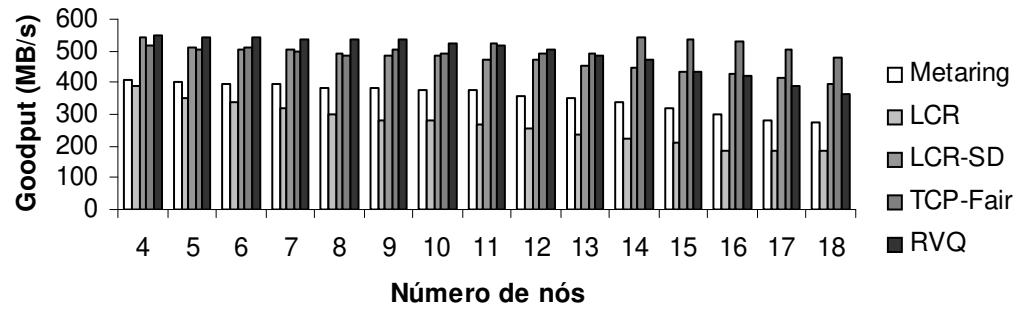


Figura 6.4: Vazão dos protocolos no cenário simétrico com 200Km variando o número de nós.

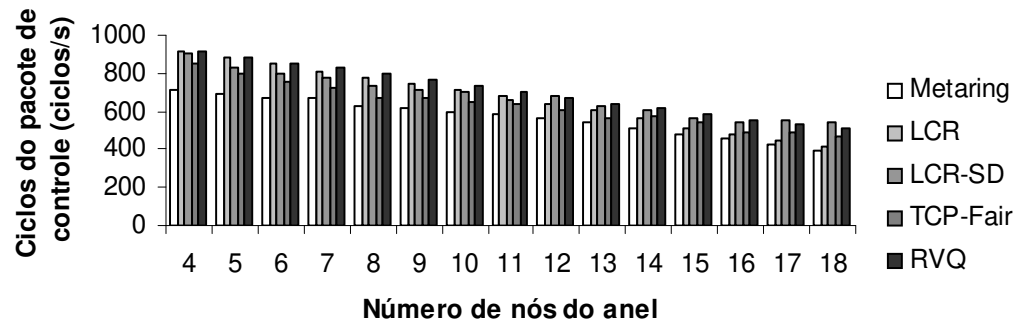


Figura 6.5: Ciclos do pacote de controle.

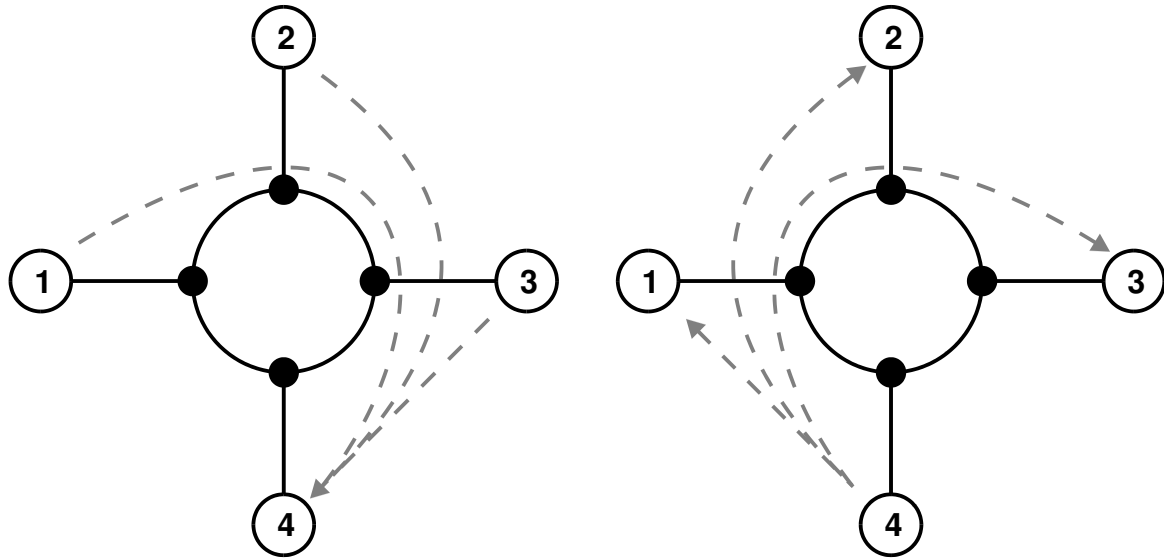


Figura 6.6: Fluxos de dados no cenário assimétrico com 4 nós.

simétrico, o número de ciclos do pacote de controle muda com o comprimento do anel (Figura 6.8).

Variando o comprimento do anel, os protocolos Metaring, LCR, LCR-SD, TCP-Fair e RVQ obtiveram respectivamente um *goodput* médio de 296, 483, 495, 334, 498 MB/s.

Os protocolos Metaring e TCP-Fair procuram manter igualdade de vazão entre os nós, dessa forma, quando um nó possui uma carga muito maior em relação aos outros nós, sua vazão é reduzida procurando manter igualdade entre os nós. Neste cenário, o nó especial possui uma vazão muito maior, tanto na recepção quanto no envio, quando comparado aos outros nós do anel.

O protocolo LCR apresenta uma vazão maior no cenário assimétrico que no cenário simétrico. Isto se deve ao cálculo da matriz de justiça que, no cenário assimétrico, resulta em uma distribuição de quotas mais eficiente que no cenário simétrico.

A vazão dos protocolos é maior com um número menor de nós da mesma forma como no cenário simétrico (Figura 6.9).

De forma geral, a vazão dos protocolos é menor do que no caso simétrico com exceção do protocolo LCR.

Variando o número de nós do anel, os protocolos Metaring, LCR, LCR-SD e TCP-Fair obtiveram respectivamente um *goodput* médio de 202, 283, 322, 262, 342 MB/s.

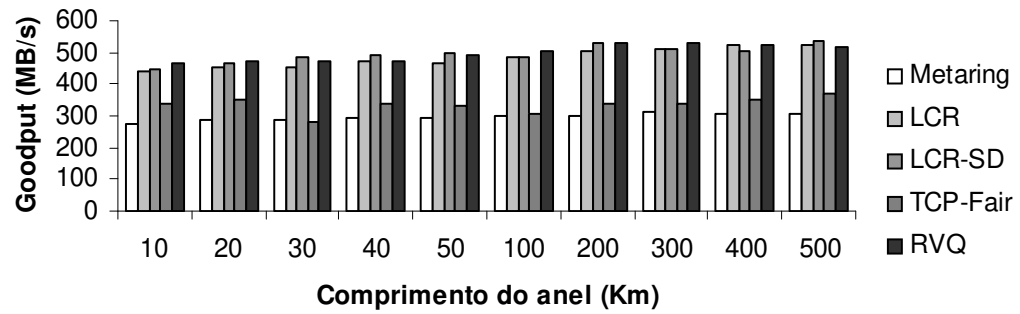


Figura 6.7: Vazão dos protocolos no cenário assimétrico variando o comprimento do anel.

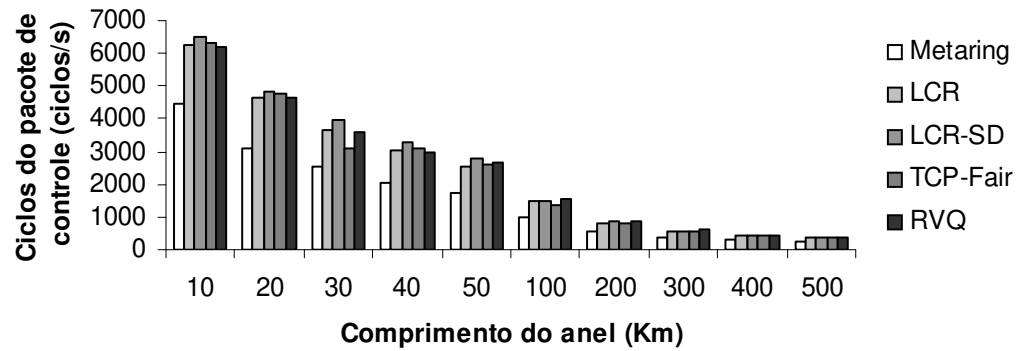


Figura 6.8: Ciclos do pacote de controle.

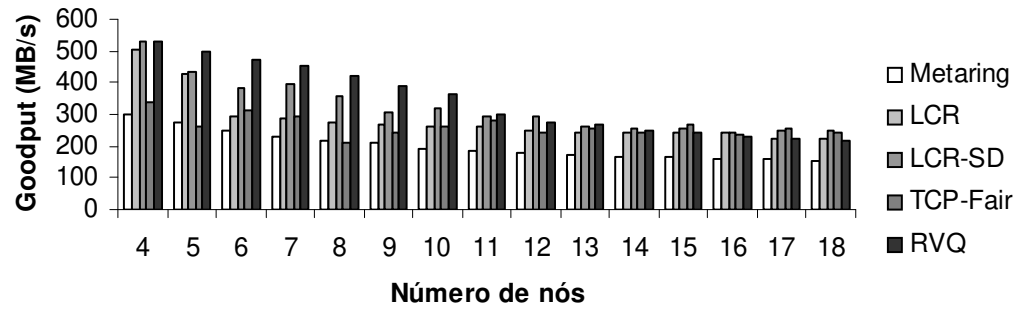


Figura 6.9: Vazão dos protocolos no cenário assimétrico variando o número de nós.

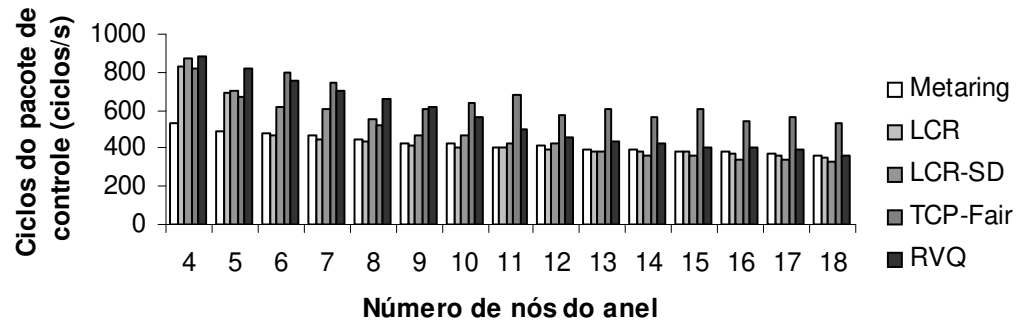


Figura 6.10: Ciclos do pacote de controle.

6.3 Simulações com Tráfego de Curta Duração

Simulações com tráfego TCP de curta duração também foram realizadas.

Diversas distribuições na Internet apresentam cauda longa [10]:

- tamanho dos arquivos requisitados por usuários;
- arquivos transmitidos na rede;
- duração das transmissões de arquivos;
- tamanho dos arquivos armazenados em servidores.

Na simulação, as conexões TCP transmitem arquivos com tamanho seguindo a distribuição híbrida de cauda longa apresentada em [6], onde o corpo é uma distribuição lognormal e a cauda é uma distribuição pareto.

A cauda pareto tem os parâmetros $\beta = 133000$ e $\alpha = 1, 1$. A média da pareto é dada por

$$\mu = \alpha * \beta / (\alpha - 1) = 1463000$$

O corpo lognormal representa 93% da distribuição, tem o valor máximo de corte (máximo) de 133000 e tem os parâmetros $M = 9,357$ e $S = 1,318$. A média e desvio padrão da lognormal são dados por:

$$\mu = \exp(M + S^2/2) = 27599,75017$$

$$\sigma^2 = \exp(S^2 + 2M) * (\exp(S^2) - 1) = 3565719841$$

$$\sigma = 59713,6487$$

A Figura 6.11 mostra o *Complementary Cumulative Distribution Function* (ccdf) do gerador híbrido implementado.

6.3.1 Cenário Simétrico

Foram utilizados novamente os cenários simétrico e assimétrico. Foi utilizada uma carga UDP de 10% e 4000 conexões TCP, sendo que a quantidade de dados transmitida pelas conexões TCP possui a distribuição híbrida lognormal/pareto.

Variando o comprimento do anel, os protocolos Metaring, LCR, LCR-SD, TCP-Fair e RVQ obtiveram respectivamente um *goodput* médio de 378, 349, 495, 467, 472 MB/s (Figura 6.12).

Variando o número de nós do anel, os protocolos Metaring, LCR, LCR-SD, TCP-Fair e RVQ obtiveram respectivamente um *goodput* médio de 359, 262, 443, 471, 415 MB/s (Figura 6.13).

No cenário simétrico, as simulações com tráfego de curta duração não provocaram alterações consideráveis quanto à vazão quando comparadas às simulações com tráfego TCP com carga infinita.

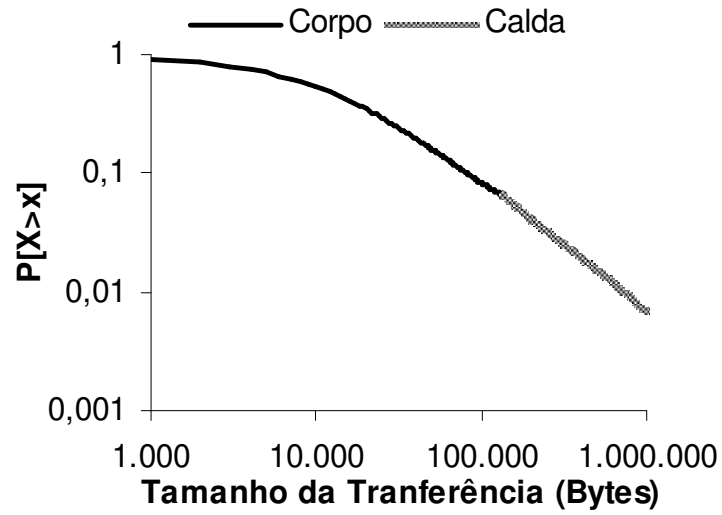


Figura 6.11: ccdf da distribuição híbrida.

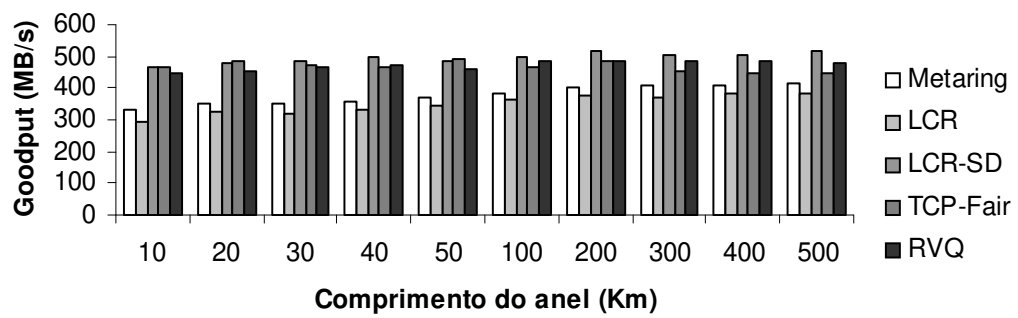


Figura 6.12: Vazão dos protocolos no cenário simétrico variando o comprimento do anel.

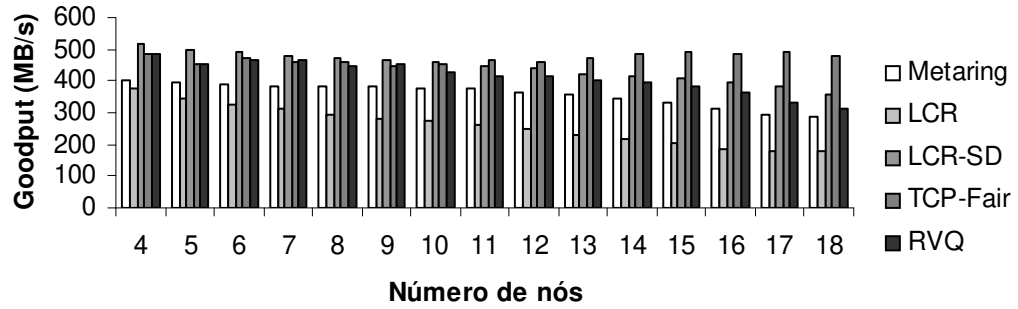


Figura 6.13: Vazão dos protocolos no cenário simétrico variando o número de nós.

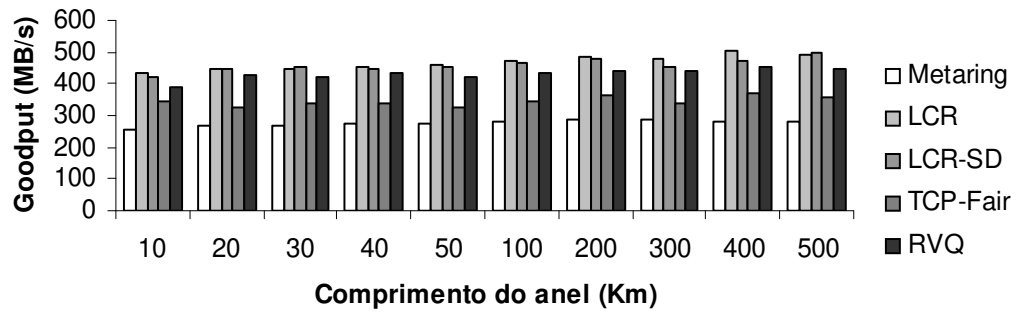


Figura 6.14: Vazão dos protocolos no cenário assimétrico variando o comprimento do anel.

6.3.2 Cenário Assimétrico

Variando o comprimento do anel, os protocolos Metaring, LCR, LCR-SD, TCP-Fair e RVQ obtiveram respectivamente um *goodput* médio de 276, 467, 459, 345, 430 MB/s (Figura 6.14).

Variando o número de nós do anel, os protocolos Metaring, LCR, LCR-SD, TCP-Fair e RVQ obtiveram respectivamente um *goodput* médio de 194, 253, 293, 271, 311 MB/s (Figura 6.15).

Novamente, as simulações com tráfego de curta duração provocaram somente pequenas alterações quando comparadas às simulações com tráfego TCP de carga infinita.

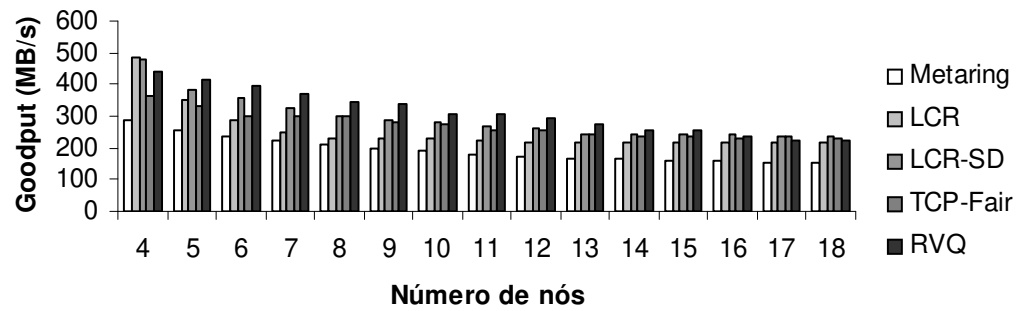


Figura 6.15: Vazão dos protocolos no cenário assimétrico variando o número de nós do anel.

Capítulo 7

Conclusões

Esta dissertação analisou o problema da distribuição de banda em redes ópticas em anel. O controle da distribuição da banda é realizado pelo protocolo MAC, o qual, seguindo uma política de justiça, distribui a banda entre os diversos nós ou fluxos presentes no anel.

Foram analisadas as políticas de justiça por nó de origem (source node fairness) e por par origem-destino (source-destination fairness) por serem políticas amplamente utilizadas.

Foi, também, introduzida uma nova política de justiça, do protocolo TCP-Fair, baseada na justiça de recepção de dados entre nós, oposto à justiça por nó de origem, a qual prioriza a justiça de envio de dados, e uma política de justiça entre conexões TCP, do protocolo RVQ. Oferecer justiça de recepção de dados é interessante em redes onde predominam usuários que realizam requisições de dados (e.g. usuários da Internet). A característica de manter a justiça entre conexões TCP também é inovadora, característica não abordada em outras políticas.

Os protocolos MAC analisados foram o Metaring, LCR, LCR-SD, TCP-Fair e RVQ. O protocolo Metaring [7] é utilizado como referência de comparação. O protocolo LCR [22] introduz um mecanismo de reserva e atribuição de recursos que previne *starvation*. O protocolo LCR-SD [25] é uma extensão do LCR implementando a justiça por par origem-destino. O protocolo TCP-Fair provê justiça de recepção entre nós e justiça entre receptores em um mesmo nó. O protocolo RVQ oferece justiça entre conexões TCP, o que o torna um protocolo muito flexível e adaptável a alterações nas condições de tráfego. Os protocolos propostos neste trabalho se adaptam dinamicamente à rede, e não requerem parâmetros de configuração como o Metaring, os quais, muitas vezes, são obtidos por tentativa e erro.

Foram realizadas simulações no NS-2 (*Network Simulator*), a fim de analisar o comportamento dos protocolos no transporte de tráfego TCP e UDP.

Quanto à justiça, todos os protocolos mostraram ser capazes de distribuir a banda do anel seguindo suas respectivas políticas de justiça. Também foi salientado que a alocação da banda muda radicalmente dependendo da política de justiça adotada, o que torna a escolha da política de justiça extremamente importante.

Quanto à eficiência na alocação da capacidade do anel, foram realizadas simulações em dois cenários: simétrico e assimétrico. Os protocolos LCR-SD, TCP-Fair e RVQ mostraram uma vazão superior em comparação ao Metaring em todas as simulações. A vazão do protocolo Metaring foi maior que a vazão do LCR no cenário simétrico, porém menor no cenário assimétrico.

O comprimento do anel não mostrou ser um fator de grande impacto na vazão dos protocolos, enquanto o número de nós mostrou um impacto significativo. O aumento do número de nós diminui as chances de encaminhamento do pacote de controle sem atrasos. Isso aumenta o tempo que o pacote de controle requer para circular pelo anel, acarretando em perda da capacidade devido ao atraso na renovação das quotas.

Como trabalho futuro sugere-se o tratamento do problema do encaminhamento do pacote de controle, pois em todos os protocolos apresentados, um atraso no encaminhamento do pacote de controle acarreta em perda da ocupação da banda, devido ao fato de que há um atraso para pacote de controle ao renovar as quotas de transmissão. Em diversas redes ópticas de comutação de pacotes, existe um canal de controle para envio do cabeçalhos dos pacotes; uma possível solução seria a utilização deste canal para o envio do pacote de controle.

Referências Bibliográficas

- [1] Coarse wavelength division multiplexing. [http:// www.fiber-optics.info/ articles/ cwdm.htm](http://www.fiber-optics.info/articles/cwdm.htm).
- [2] Dense wavelength-division multiplexing. [http:// www.fiber-optics.info/ articles/ dwdm.htm](http://www.fiber-optics.info/articles/dwdm.htm).
- [3] Tutorial for the network simulator "ns". [http:// www.isi.edu/ nsnam/ ns/ tutorial/ index.html](http://www.isi.edu/nsnam/ns/tutorial/index.html).
- [4] Wavelength-division multiplexing. [http:// www.fiber-optics.info/ articles/ wdm.htm](http://www.fiber-optics.info/articles/wdm.htm).
- [5] H. Balakrishnan, V. N. Padmanabhan, G. Fairhurst, and M. Sooriyabandara. Tcp performance implications of network path asymmetry. Request for Comments 3449, Internet Engineering Task Force, dec 2002.
- [6] Paul Barford and Mark Crovella. Generating representative web workloads for network and server performance evaluation. In *SIGMETRICS '98/PERFORMANCE '98: Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 151–160, New York, NY, USA, 1998. ACM Press.
- [7] I. Cidon and Y. Ofek. Metaring - a full-duplex ring with fairness and spatial reuse. In *Proceedings IEEE INFOCOM'90*, pages 969–981, jun 1990.
- [8] Israel Cidon, Leonidas Georgiadis, rin Roch Gué, and Yuval Shavitt. Improved fairness algorithms for rings with spatial reuse. *IEEE/ACM Transactions on Networking*, 5(2):190–204, apr 1997.
- [9] K. Claffy, G. Miller, and K. Thompson. The nature of the beast: Recent traffic measurements from an internet backbone. In *Proceedings of INET'98*. Internet Society, jul 1998.

- [10] Mark E. Crovella, Murad S. Taqqu, and Azer Bestavros. *Heavy-tailed probability distributions in the World Wide Web*. Birkhauser Boston Inc., Cambridge, MA, USA, 1998.
- [11] Kevin Fall and Sally Floyd. Simulation-based comparisons of Tahoe, Reno and SACK TCP. *ACM SIGCOMM Computer Communication Review*, 26(3):5–21, jul 1996.
- [12] Violeta Gambiroza, Ping Yuan, Laura Balzano, Yonghe Liu, Steve Sheafor, and Edward Knightly. Design, analysis, and implementation of dvsr: a fair high-performance protocol for packet rings. *IEEE/ACM Transactions on Networking*, 12(1):85–102, feb 2004.
- [13] Ornan Gerstel and Rajiv Ramaswami. Optical layer survivability: A services perspective. *IEEE Communications Magazine*, 38(3):104–113, mar 2000.
- [14] G. Huston. Future for tcp. *Internet Protocol Journal*, 3(3):2–27, jun 2000.
- [15] G. Huston. Tcp performance. *Internet Protocol Journal*, 3(2), jun 2000.
- [16] Kazuo Imai, Tadashi Ito, Hideki Kasahara, and Naotaka Morita. Atmr: Asynchronous transfer mode ring protocol. *Computer Networks and ISDN Systems*, 26(6-8):785–798, mar 1994.
- [17] Van Jacobson. Congestion avoidance and control. In *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*, pages 314–329, New York, NY, USA, 1988. ACM Press.
- [18] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. DEC Research Report TR-301, DEC, sep 1984.
- [19] G. Maier, S. De Patre, A. Pattavina, and M. Martinelli. Optical network survivability: protection techniques in the wdm layer. *Photonic Network Communications*, 4:251–269, jul 2002.
- [20] M. A. Marsan, C. Casetti, S. M. Grasso, and F. Néri. Slot reuse in mac protocols for man's. *IEEE Journal on Selected Areas of Communications*, 11(8):1290–1301, oct 1993.
- [21] Alain J. Mayer, Yoram Ofek, and Moti Yung. Approximating max-min fair rates via distributed local scheduling with partial information. In *Proceedings IEEE INFOCOM '96*, pages 928–936, mar 1996.

- [22] M. R. Salvador, M. M. Uesono, and N.L.S. da Fonseca. A local fairness protocol for optical packet-switched wdm ring networks. In *IEEE International Conference on Communications*, volume 3, pages 1524–1528. IEEE, may 2005.
- [23] Marcos Rogério Salvador, Sonia M. Heemstra de Groot, and Diptish Dey. An all-optical wdm packet-switched network architecture with support for group communication. In *ICN '01: Proceedings of the First International Conference on Networking-Part 1*, pages 326–335, London, UK, 2001. Springer-Verlag.
- [24] Kapil V. Shrikhande, Ian M. White, Duang rudee Wonglumsom, Steven M. Gemelos, Matthew S. Rogge, Yasuyuki Fukushima, Moritz Avenarius, and Leonid G. Kazovsky. Hornet: A packet-over-wdm multiple access metropolitan area ring network. *IEEE Journal on Selected Areas in Communications*, 18(10):2004–2016, oct 2000.
- [25] M. M. Uesono, M. R. Salvador, and N.L.S. da Fonseca. A packet ring fairness protocol and its impact on tcp fairness. In *Proceedings of IEEE Globecom*, volume 2, pages 1044–1048. IEEE, nov 2005.
- [26] M. M. Uesono, M. R. Salvador, and N.L.S. da Fonseca. Protocolo mac para redes Ópticas de pacotes em anel para promoção de justiça entre conexões tcp. In *XXII Simpósio Brasileiro de Telecomunicações*, pages 634–637. SBC, sep 2005.

Apêndice A

Implementação do Simulador

Neste apêndice, é descrito, brevemente, a implementação do simulador.

O simulador utilizado nesta tese foi o NS-2 (*Network Simulator*), desenvolvido na Universidade de Berkeley em C++ e OTcl (extensão orientada a objeto do TCL). O NS-2 utiliza o C++ para a implementação do simulador e o OTcl, como interpretador de scripts de usuário.

O NS-2 não possui mecanismos para a simulação de redes ópticas em anel. Dessa forma, foi necessário implementar estes mecanismos. As implementações incluem sistemas de filas e enlaces que simulam as características de redes de fibra óptica em anel e os protocolos MAC utilizados nas simulações.

A.1 Implementação

Conhecer a forma como o simulador funciona é essencial para se implementar extensões no NS.

A seguir é apresentado, de forma simplificada, como o NS realiza uma simulação:

Para se criar uma simulação, ou seja, para se informar ao NS como é o cenário a ser simulado, é criado um script TCL/TK. O script de simulação deve realizar as seguintes tarefas:

O primeiro passo é criar uma instância do simulador. Nesta instância, são registrados os nós (que são os roteadores), os enlaces (conexões entre os nós), os protocolos de transporte (como o TCP ou o UDP) e as fontes de tráfego (como, por exemplo, tráfego on/off, FTP, chegadas de pacotes em intervalos com distribuição exponencial ou pareto).

Além disso, é necessário configurar cada um desses componentes: é necessário configurar, nos enlaces, sua banda, sua latência e os nós aos quais ele está ligado; em uma comunicação TCP, é necessário ligar uma fonte TCP a um receptor e associar cada um destes a um nó; as fontes de tráfego (como o FTP) devem ser associadas a um canal de

comunicação (como o TCP) e devem ser configuradas as características do tráfego, como no caso de tráfego FTP, o momento em que o tráfego tem início e a quantidade de dados a ser transmitido.

O script TCL/TK, então, entrega o resto das tarefas ao NS, que simula como os pacotes de dados trafegam na rede.

O nó que origina uma transmissão de dados, como, por exemplo, uma fonte de dados FTP, cria uma carga, ou seja, uma quantidade de dados a ser transmitido; o protocolo de transporte associado a esta fonte de dados, como o TCP, cria os pacotes a partir desta carga e os direciona ao nó de destino.

O nó analisa o endereço destino do pacote e determina para qual nó o pacote deve ser enviado. O pacote é então colocado na fila correta onde aguarda até que o enlace da interface esteja pronto para transmiti-lo. O enlace retira o pacote da fila e calcula o tempo em que este deve ser entregue na outra ponta do enlace. Quando o pacote é recebido pelo próximo nó, é determinado, a partir do endereço de destino, se o pacote é destinado ao próprio nó ou se deve ser encaminhado para um outro nó. O processo se repete até que o pacote seja entregue ao nó de destino.

Para se criar extensões no NS, como por exemplo, um novo tipo de fila, é necessário criar o componente, normalmente em C ou C++, implementar formas para que o script TCL/TK possa instanciar este componente e fazer com que o componente interaja de maneira correta com os outros componentes do NS.

Um tutorial sobre como implementar extensões no NS foi publicado em [3].

A.2 Slotting

Para implementar o *slotting* no anel, foram criados novos mecanismos de filas e enlaces.

Como em um anel todos os *slots* possuem o mesmo tamanho, então a chegada e o envio de *slots* ocorre em intervalos constantes de tempo. Para simular este comportamento, a fila verifica, em intervalos constantes de tempo, se existem pacotes a serem transmitidos no anel.

Outra característica é que, quando um nó recebe um *slot* pela fibra, este é colocado em um fiber delay. Se o *slot* não é destinado ao nó, então o *slot* continua na fibra e segue em direção ao próximo nó do anel. No NS-2, não existe um único enlace que liga todos os nós como no anel, os nós estão ligados por enlaces distintos. Dessa forma, no momento em que um nó recebe um *slot*, este é recolocado no anel e transmitido em direção ao próximo nó. Para que o *slot* não sofra um atraso na fila, existe um sistema de prioridades de filas. Existem três filas principais em um nó, uma de alta prioridade destinada aos pacotes dos *slots* vindos do anel, outra de média prioridade para o pacote de controle e outra de baixa prioridade para pacotes vindos de fora do anel.

Devido ao sistema de pacotes utilizado no NS-2, e para economia de memória e processamento, não são instanciados *slots* reais nas simulações, ao invés, são realizados cálculos para determinar os momentos em que os pacotes devem sair e chegar aos nós como se estivessem em *slots*.

A.3 Protocolos MAC

Os protocolos MAC são compostos por dois mecanismos, um responsável por receber o pacote de controle e realizar seu processamento, e outro mecanismo responsável por controlar a vazão das filas obedecendo às quotas atribuídas aos fluxos ou ao próprio nó.

Ao receber o pacote de controle, cada protocolo MAC realiza os cálculos de acordo com seu próprio algoritmo e no final encaminha o pacote de controle ao próximo nó. O pacote de controle é colocado na fila de média prioridade citada anteriormente e é encaminhado ao próximo nó assim que existir um *slot* vazio.

O mecanismo de controle de quotas funciona continuamente gerenciando a utilização dos *slots* do anel. Este mecanismo controla a vazão da fila de baixa prioridade, também, citada anteriormente.