

Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Coordenação de Pós-Graduação em Ciência da Computação

Reduzindo a Duração da Fase de Partida Lenta do  
TCP Através da Inferência Colaborativa da Janela de  
Congestionamento

Giovanni de Almeida Calheiros

Dissertação submetida à Coordenação do Curso de Pós-Graduação em  
Ciência da Computação da Universidade Federal de Campina Grande -  
Campus I como parte dos requisitos necessários para obtenção do grau  
de Mestre em Ciência da Computação.

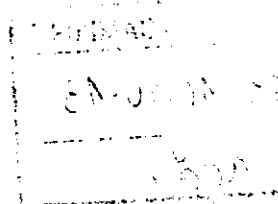
Área de Concentração: Ciência da Computação

Linha de Pesquisa: Engenharia de Software

Orientadores : Angelo Perkusich e Hyggo Almeida

Campina Grande, Paraíba, Brasil

©Giovanni de Almeida Calheiros, 30/07/2012





C152r Calheiros, Giovanni de Almeida.  
Reduzindo a duração da fase de partida lenta do TCP através da inferência colaborativa da janela de congestionamento / Giovanni de Almeida Calheiros. - Campina Grande, 2012.  
52 f.

Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2012.  
"Orientação : Prof. Dr. Angelo Perkusich, Prof. Dr. Hyggo Oliveira de Almeida".  
Referências.

1. Redes de Computadores. 2. Partida Lenta. 3. TCP. 4. Dissertação - Ciência da Computação. I. Perkusich, Angelo. II. Almeida, Hyggo Oliveira de. III. Universidade Federal de Campina Grande - Campina Grande (PB). IV. Título

CDU 004.7(043)

**"REDUZINDO A DURAÇÃO DA FASE DE PARTIDA LENTA DO TCP ATRAVÉS DA  
INFERÊNCIA COLABORATIVA DA JANELA DE CONGESTIONAMENTO"**

**GIOVANNI DE ALMEIDA CALHEIROS**

**DISSERTAÇÃO APROVADA EM 30/08/2012**

  
**HYGGO OLIVEIRA DE ALMEIDA, D.Sc**  
**Orientador(a)**

  
**ANGELO PERKUSICH, D.Sc**  
**Orientador(a)**

  
**MARCOS RICARDO ALCÂNTARA MORAIS, D.Sc**  
**Examinador(a)**

  
**LEANDRO DIAS DA SILVA, D.Sc**  
**Examinador(a)**

**CAMPINA GRANDE - PB**

## Resumo

Os algoritmos de controle de congestionamento utilizados pelo protocolo TCP (*Transmission Control Protocol*) exercem um importante papel ao evitar que a rede entre em estado de congestionamento. Todavia, em virtude desses algoritmos utilizarem uma abordagem conservadora para determinar os valores ideais para os parâmetros de cada conexão, tal abordagem prejudica os fluxos de curta duração, pois a rede é reavaliada a cada nova conexão e, por vezes, o tempo necessário para determinar os valores ideais é maior que a duração do próprio fluxo. Neste trabalho, apresenta-se o TCP-Share, um algoritmo que baseia-se no compartilhamento de informações adquiridas por conexões anteriores. Objetiva-se evitar a reavaliação do estado da rede em períodos curtos de tempo, permitindo assim, um melhor aproveitamento da largura de banda disponível. O TCP-Share foi desenvolvido com base na variante do protocolo TCP conhecida como TCP Cubic, na qual foram adicionados os componentes para compartilhamento de informações. Com o intuito de avaliar o TCP-Share realizou-se experimentos para compará-lo com o TCP Cubic original. Como resultado, constatou-se que o tempo necessário à conclusão dos fluxos de curta duração foi reduzido em até 30%.

## **Abstract**

The Congestion control algorithms for TCP protocol (*Transmission Control Protocol*) have an important role to prevent the network from a state of congestion. By the fact these algorithms adopt a conservative approach to reach a configuration state with optimal values for each connection parameters, the approach used for them is unfair to short-lived flows, since the network is reassessed at each new connection, and sometimes the time required to determine these values is greater than the flow completion time. The approach introduced in this dissertation, called TCP-Share, is based on the sharing of information acquired by previous connections in order to avoid reassessing the state of the network for short periods of time, thus allowing a better use of available bandwidth. The TCP-Share was developed based on a variant of the TCP protocol known as TCP Cubic, in which were added the components to share information. In order to evaluate the TCP-Share, experiments were performed to compare it to TCP Cubic. As a result, it was found that the time required for completion of the flows of short duration was up to 30%.

## **Agradecimentos**

Em primeiro lugar agradeço a Deus e a meus pais por terem permitido, possibilitado e auxiliado que eu galgasse o local onde estou, ensinando com seus exemplos de trabalho, caráter e respeito.

Aos meus familiares em especial a : Aline, Andréa, Cássia, Cordélia, Diogenes, Diogo, Djalma, Dona Moça, Edmar, Fátima, Giselli, Grimoaldo, Hélio, João Victor, Lenilda, Leny, Luiz Carlos, Luzinete, Luzinez, Marta, Mônica, Polyxênia, Tânia [Pombinha], Tânia [Resumida], Thiago, Rafaella, Renata, Rosamaria, Valdeci, Vegas que me apoiaram durante todo esse período no qual percebi o quanto sinto falta deles.

Aos meus amigos Caroline, Lavis, Layane, Leonardo [tio leo], Marcelo, Mayara e Michel por todo apoio ajuda que me foi dado seja ouvindo minhas queixas, reclamando daquilo que achava errado, impróprio ou injusto, seja me encorajando para que continuasse.

A todos aqueles que conheci durante o período do mestrado, que ajudaram direta ou indiretamente, em especial àqueles que moraram no edifício Vila Bela (Bami, Cristóvão, Danielle, Leandro, Leonardo [Cachorra], Raquel, Rodrigo, Sttefany), aos colegas do Laboratório de Sistemas Embarcados e Computação Pervasiva - Embedded (Caroline, Daniel, Elthon, Fred, Ivo, Junior, Kézia, Marco, Romeryto, Saulo) e aos que fizeram parte da convivência diária (Aninha, Carmelia, Laíza, Maxwell, Vera, Rebeca) por todas as conversas que tivemos.

Por fim, agradeço a CAPES pelo apoio Financeiro.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Problemática . . . . .	2
1.2	Objetivos . . . . .	5
1.3	Relevância . . . . .	5
1.4	Estrutura da dissertação . . . . .	6
<b>2</b>	<b>Fundamentação Teórica</b>	<b>7</b>
2.1	Camada de transporte . . . . .	7
2.2	Protocolo TCP . . . . .	9
2.3	Fluxos . . . . .	10
2.4	Controle de congestionamento . . . . .	11
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>14</b>
3.1	Alteração da função de crescimento . . . . .	14
3.1.1	<i>CapStart: An Adaptive TCP Slow Start for High Speed Networks</i> . .	15
3.1.2	<i>Burst TCP: an approach for benefiting mice flows</i> . . . . .	15
3.1.3	<i>TCP Startup Performance in Large Bandwidth Delay Networks</i> . .	16
3.2	Reutilização do bloco de controle . . . . .	17
3.2.1	<i>TCP Fast Start</i> . . . . .	17
3.2.2	<i>SPAND: Shared Passive Network Performance Discovery</i> . . . . .	17
3.3	Agrupamento de conexões . . . . .	19
3.4	Utilização de informações de contexto . . . . .	20
3.5	Considerações sobre os trabalhos relacionados . . . . .	21

<b>4</b>	<b>TCP-Share</b>	<b>24</b>
4.1	Arquitetura da solução . . . . .	25
4.2	Compartilhamento da informação . . . . .	26
4.3	Funcionamento . . . . .	30
4.4	Armazenamento da informação . . . . .	33
4.5	Segurança . . . . .	33
4.6	Redução da agressividade . . . . .	34
<b>5</b>	<b>Avaliação Experimental</b>	<b>35</b>
5.1	Alterações no Simulador de redes NS3 . . . . .	35
5.2	Cenário . . . . .	36
5.3	Experimentos . . . . .	37
5.4	Análise dos resultados . . . . .	38
5.4.1	Cenário 1 . . . . .	39
5.4.2	Cenário 2 . . . . .	39
5.4.3	Cenário 3 . . . . .	40
5.4.4	Cenário 4 . . . . .	41
5.4.5	Aumento da Vazão . . . . .	41
5.4.6	Conclusão dos resultados . . . . .	43
<b>6</b>	<b>Conclusão e trabalhos futuros</b>	<b>44</b>



# Glossário

ACK - *Acknowledgement*

AIMD - *Additive Increase/Multiplicative Decrease*

BDP - *Bandwidth Delay Product*

*bps* - bits por segundo

DHCP - *Dynamic Host Configuration Protocol*

DHT - *Distributed Hash Table*

ECN - *Explicit Congestion Notification*

*Gbps* - Gigabits por segundo

B-TCP - *Burst Transmission Control Protocol*

HTTP - *Hypertext Transfer Protocol*

IP - *Internet Protocol*

KB - *KiloByte*

*Kbps* - Kilobits por segundo

*Mbps* - Megabits por segundo

*ms* - milissegundo

MSS - *Maximum Segment Size*

P2P - *Peer-to-Peer*

RFC - *Request For Comments*

RTO - *Retransmission Timeout*

RTT - *Round-trip Time*

TCB - *Transmission Control Block*

TCP - *Transmission Control Protocol*

UDP - *User Datagram Protocol*

XCP - *eXplicit Control Protocol*



# Lista de Figuras

1.1	Ilustração para as fases do controle de congestionamento para o protocolo TCP.	3
1.2	Ilustração da subutilização do enlace devida a partida lenta. . . . .	4
2.1	Ilustração arquitetura hierárquica do Modelo TCP/IP. . . . .	8
2.2	Ilustração comunicação lógica entre as aplicações em execução nos computadores (adaptada de [12]). . . . .	9
2.3	Ilustração do comportamento de crescimento da janela na fase de partida lenta (adaptada de [20]). . . . .	12
2.4	Ilustração do comportamento de crescimento da janela na fase de Prevenção de Congestionamento. . . . .	13
3.1	Ilustração do relatório de desempenho (adaptada de [35]). . . . .	18
3.2	Ilustração da arquitetura empregada pelo SPAND (adaptada de [35]). . . . .	18
3.3	Ilustração do relatório de requisição (adaptada de [35]). . . . .	19
3.4	Ilustração do relatório de resposta (adaptada de [35]). . . . .	19
4.1	Ilustração da arquitetura da solução proposta. . . . .	26
4.2	Ilustração da arquitetura interna do TCP-Share e comunicação com a pilha TCP/IP. . . . .	27
4.3	Ilustração do compartilhamento da informação utilizando o envio por difusão.	28
4.4	Ilustração do compartilhamento da informação utilizando o envio por difusão limitada. . . . .	29
4.5	Ilustração do compartilhamento da informação utilizando um servidor centralizado. . . . .	30
4.6	Fluxograma representando o funcionamento da solução proposta. . . . .	31

---

5.1	Ilustração da topologia empregada na simulação. . . . .	36
5.2	Gráfico do comportamento do fluxo para o cenário 1. . . . .	39
5.3	Gráfico do comportamento do fluxo para o cenário 2. . . . .	40
5.4	Gráfico do comportamento do fluxo para o cenário 3. . . . .	40
5.5	Gráfico do comportamento do fluxo para o cenário 4. . . . .	41
5.6	Gráfico comparativo entre os tempos de conclusão para os 4 cenários. . . .	42

# Lista de Tabelas

3.1	Comparativo entre as soluções propostas . . . . .	23
4.1	Resumo das características de cada método de compartilhamento. . . . .	27
5.1	Tabela contendo as variáveis independentes . . . . .	38
5.2	Tabela contendo as variáveis dependentes . . . . .	38
5.3	Comparativo entre o aumento da vazão e aumento da janela compartilhada .	42

## Lista de Códigos Fonte

4.1	Diretiva utilizada na configuração automática via DHCP . . . . .	29
4.2	Pseudo código do procedimento de limpeza do valor armazenado . . . . .	30
4.3	Pseudo código do procedimento de obtenção da informação . . . . .	32
4.4	Pseudo código do procedimento de armazenamento da informação . . . . .	32

# Capítulo 1

## Introdução

A *Internet* teve seu início quando os pesquisadores interligaram as diversas redes existentes nos anos 70 [24]. Para permitir essa interligação foi proposto o conjunto de protocolos TCP/IP (*Transmission Control Protocol / Internet Protocol*). Este conjunto de protocolos permite a comunicação entre redes com diferentes tecnologias para controle de enlace de dados existentes.

Desde seus primórdios, a forma como as redes vem sendo utilizadas tem se diversificado. Nos anos 70, quando a ARPANET foi criada, haviam disponíveis apenas alguns *kbits* de velocidade nos enlaces e a principal função das redes era a troca de mensagens textuais [37]. Atualmente a *Internet* é um meio eficiente para a troca de informações, seja para leitura de notícias, comércio eletrônico, compartilhamento de músicas ou conteúdo multimídia por demanda.

Juntamente com a diversificação dos usos da rede houve a evolução das tecnologias de interconexão. Essa evolução disponibilizou enlaces com velocidades que atingiram vários *gbits* e também permitiu que o acesso à *Internet* ocorresse quase em qualquer lugar. Estudos indicam que a média de velocidade das conexões com a *Internet* no mundo em 2010 foi de 1,8 Mbps, sendo que mais de 50% dos usuários dispunham de conexões de banda larga acima de 2 Mbps [38].

Apesar de todas as alterações ocorridas, tanto nas tecnologias de interconexão quanto nas formas de uso, essencialmente as comunicações na *Internet* não sofreram alterações drásticas. Estas comunicações ocorrem por meio da troca de cadeias de pacotes, as quais são nomeadas fluxos. É possível classificar os fluxos quanto ao seu comprimento, medido

em número de segmentos ou em *bytes*, como fluxos de curta duração (*short lived flows*) e fluxos de longa duração (*long lived flows*) comparados por alguns autores a ratos e elefantes [5, 18, 29, 40]. Vários autores argumentam que o tráfego *Web*, em sua maioria, é composto por fluxos de curta duração e o tamanho de cada fluxo em média é de 384KB [5, 7, 13, 18, 32].

Cerca de 90% dos fluxos da *Internet* são gerenciados pelo protocolo TCP [16]. Este protocolo disponibiliza serviços como entrega ordenada dos segmentos e o controle de fluxo e congestionamento através de diversas variantes [9, 17, 19, 33, 36]. Estas variantes foram desenvolvidas com o objetivo de proporcionar um uso mais eficiente dos enlaces de dados disponíveis, uma vez que a diversidade de tecnologias e da larguras de banda, em alguns casos, não permite uma utilização eficiente desses enlaces. Essa eficiência pode ser vista como a melhor utilização da largura de banda disponível no enlace de dados. É neste cenário de melhoria do uso da largura de banda disponível nos enlaces de dados que se insere este trabalho.

## 1.1 Problemática

Cada fluxo TCP é composto por vários pacotes de dados, que no contexto do TCP também são chamados de segmentos, providos pelas aplicações. Estes segmentos são criados, na camada de transporte da pilha TCP/IP e são enviados para as camadas inferiores até a chegada ao seu destino final. O controle de quantos segmentos serão transmitidos ao destino é feito pelo controle de congestionamento.

O controle de congestionamento é dividido em 4 fases: partida lenta, prevenção de congestionamento, recuperação rápida e retransmissão rápida, onde a mudança entre as fases acontece quando da ocorrência de determinados eventos na rede como a perda de pacotes ou quando determinados limiares são atingidos. Porém, neste trabalho de dissertação, serão consideradas apenas as fases de partida lenta e a de prevenção de congestionamento, pois o controle de congestionamento geralmente não atinge as demais fases durante um fluxo de curta duração. Na fase de partida lenta procura-se avaliar a largura de banda disponível para a conexão, partindo de uma transmissão mínima de poucos segmentos até o limiar de mudança de fase denominado *ssthresh* (*Slow Start Threshold*) ou até o ponto de saturação da rede, onde começam a ocorrer os primeiros eventos de perda. A fase de prevenção de



congestionamento tem a função de evitar que a rede seja sobrecarregada. Para isso, nesta fase o controle de congestionamento aumenta a velocidade do fluxo TCP mais lentamente do que na fase de partida lenta, assim é possível manter uma velocidade mais elevada por um tempo mais longo, pois na ocorrência da perda de pacotes a velocidade de transmissão será reduzida o que acarreta em um aumento na duração do fluxo.

As variantes do TCP implementam a fase de prevenção de congestionamento de maneiras distintas, porém, elas compartilham o mecanismo de funcionamento da fase de partida lenta. Nesta fase, como ilustrado na Figura 1.1, a velocidade tende a crescer exponencialmente até um limiar de troca de fase. A partir desse limiar, a fase de partida lenta é finalizada e é iniciada a fase de prevenção de congestionamento.

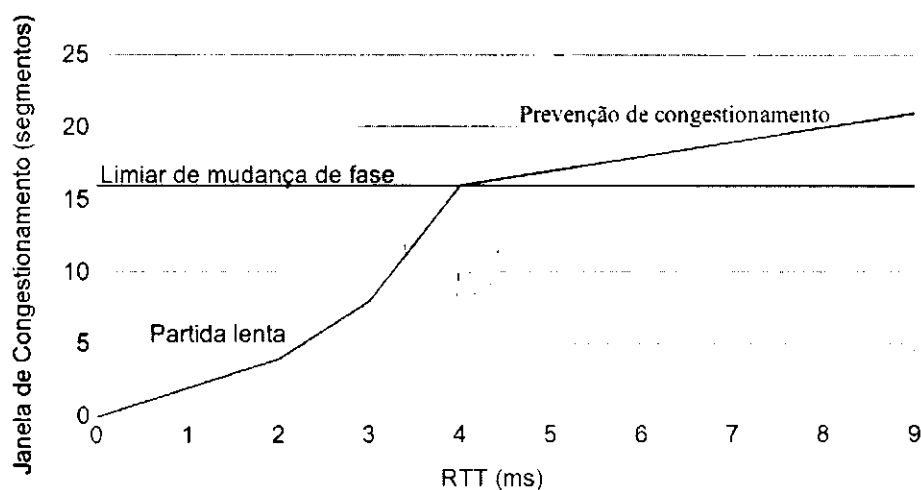


Figura 1.1: Ilustração para as fases do controle de congestionamento para o protocolo TCP.

Porém, desde a década de 90 com o surgimento das redes gigabit, que são redes com capacidade de transmissão da ordem de alguns gigabits por segundo, várias limitações relacionadas ao desempenho dos fluxos TCP ficaram mais evidentes, devido ao fato de existirem protocolos concebidos com base em limitações superadas, como canais de baixa velocidade, sendo utilizados em novos canais de alta velocidade [37].

Uma dessas limitações, como ilustrado na Figura 1.2, deve-se ao fato de que a fase de partida lenta que é considerada um dos alicerces do serviço de melhor esforço do IP para a *Internet* [28], apesar de agressiva no incremento da taxa de envio de dados, utiliza sempre o mesmo valor inicial [40] e aumenta gradativamente a quantidade de pacotes enviados até

que seja atingido o limiar para a troca de fase ou até que a perda de um pacote ocorra. Esse comportamento conservador gera uma subutilização do enlace.

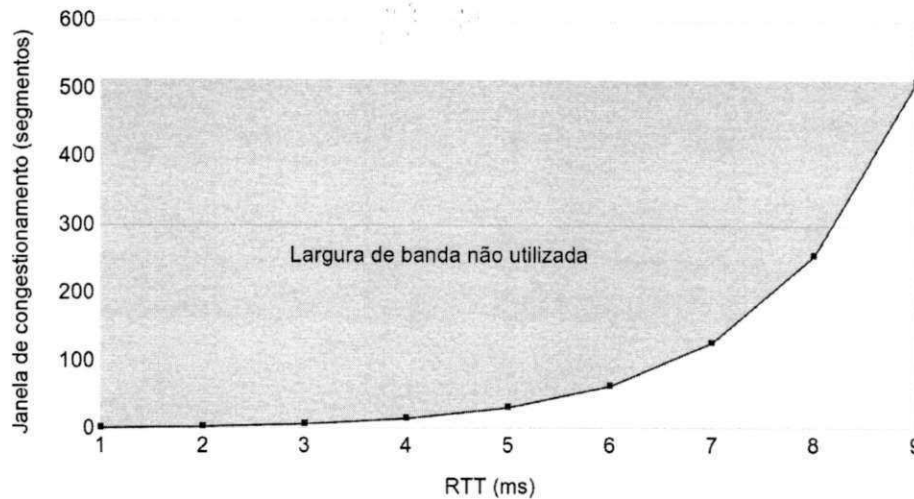


Figura 1.2: Ilustração da subutilização do enlace devida a partida lenta.

Essa limitação torna-se ainda mais severa quando fluxos de curta duração são utilizados, pois esses fluxos tendem a ser finalizados ainda da fase de partida lenta ou antes mesmo que o enlace seja utilizado de forma eficiente [40], eficiência essa que pode ser compreendida como a maior utilização da banda disponível no enlace no momento.

Há diversas propostas para redução da subutilização dos enlaces através da alteração da maneira como o controle de congestionamento calcula quantos segmentos devem ser enviados [18, 33]. Essas abordagens possibilitam reduzir a subutilização dos enlaces através da utilização de uma função de crescimento da janela de transmissão de forma mais agressiva, porém, a capacidade do enlace continua sendo avaliada, tendo como ponto de partida um valor em torno de 4 segmentos, sempre que uma nova conexão é estabelecida.

Em outras abordagens as informações de conexões anteriores são reutilizadas para evitar a reavaliação do estado da rede. Em [35] cada sistema final deve enviar para um servidor central relatórios sobre suas últimas conexões e a arquitetura, paralelamente, efetua uma medição passiva das condições do enlace. Porém a presença do servidor central apresenta-se como um ponto de falha da arquitetura e a medição passiva das condições do enlace apresenta problemas para o isolamento do tráfego dos dados. Em [29] são armazenados dados

referentes as conexões anteriores pra que as novas conexões iniciem com valores mais elevados, mas para sua utilização é necessária a alteração dos roteadores da rede o que dificulta sua implementação. Em [39] é utilizado um histórico das últimas conexões como base para a definição da nova janela de congestionamento inicial, contudo o valor da janela de congestionamento inicial só apresentará uma melhor definição após um número elevados de conexões estar disponível no histórico.

Neste contexto, identifica-se o problema abordado neste trabalho: como melhorar a utilização da largura de banda disponíveis no enlace de dados, focando-se na fase de partida lenta, mantendo o isolamento entre as camadas da pilha TCP/IP.

## 1.2 Objetivos

Nesse trabalho o objetivo é desenvolver um algoritmo, para melhorar a utilização dos recursos de rede em cada conexão, mais especificamente na fase de partida lenta.

Os objetivos específicos do trabalho são:

- analisar os protocolos existentes buscando informações como algoritmos utilizados e parâmetros da conexão que podem ser compartilhados;
- definir e implementar o algoritmo TCP-Share, com base na criação de estratégias de compartilhamento das informações da conexão com outras conexões no mesmo sistema final e com outros sistemas finais na mesma rede;
- simular o funcionamento do TCP-Share e avaliar a melhoria alcançada com seu uso, por meio da redução do tempo de conclusão dos fluxos.

## 1.3 Relevância

Segundo Floyd [2], a utilização de uma janela de congestionamento inicial maior permite aos fluxos, tanto de curta quanto de longa duração, reduzir o tempo necessário para sua conclusão em alguns *RTT*. Assim sendo, um mecanismo que possibilite o reaproveitamento da janela de congestionamento avaliada por conexões anteriores, vem a permitir uma melhor utilização da largura de banda disponível ao mesmo tempo em que diminui a necessidade de

atualizações do valor da janela inicial, uma vez que, esse valor é obtido a partir de outros sistemas finais que já avaliaram o enlace anteriormente.

## **1.4 Estrutura da dissertação**

Este trabalho está organizado como segue: no Capítulo 2, são apresentados conceitos necessários ao entendimento deste trabalho, no Capítulo 3, são apresentados alguns algumas metodologias encontradas na literatura. Nos Capítulos 4 e 5 são discutidos os detalhes de desenvolvimento e avaliação do mecanismo proposto. Por fim, no Capítulo 6 são apresentadas as considerações finais.

## Capítulo 2

# Fundamentação Teórica

Neste capítulo apresenta-se o embasamento teórico sobre os conceitos utilizados nesta dissertação. Nele serão descritos alguns aspectos da camada de transporte, necessários ao bom entendimento deste trabalho, como: o protocolo TCP, o bloco de controle e o controle de congestionamento. Também serão discutidas a caracterização dos fluxos quanto a sua duração.

### 2.1 Camada de transporte

A camada de transporte está posicionada entre as camadas de aplicação e de rede do modelo TCP/IP (Figura 2.1). Sua função básica é oferecer comunicação lógica entre as aplicações (processos, do ponto de vista do sistema operacional) que estão em execução nos computadores. A camada de transporte recebe os dados vindos da camada de aplicação, divide-os em unidades menores caso seja necessário, repassa essas unidades menores à camada de rede e assegura que todos os fragmentos chegarão corretamente à outra extremidade. Além do mais, tudo isso é feito de forma que as camadas superiores fiquem isoladas das inevitáveis mudanças na tecnologia de *hardware* [37].

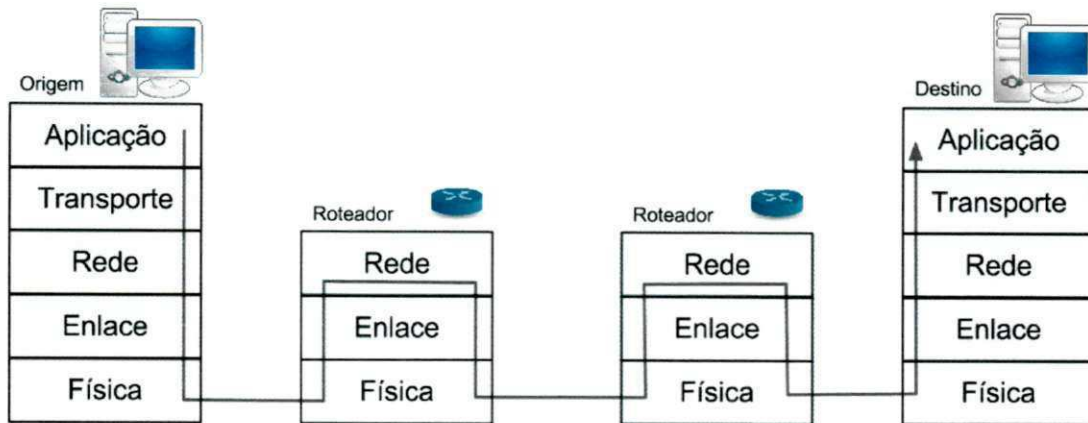


Figura 2.1: Ilustração arquitetura hierárquica do Modelo TCP/IP.

A comunicação lógica é um conceito no qual a troca de informações entre os computadores ocorre como se estes estivessem conectados diretamente. Em outras palavras, do ponto de vista das aplicações, é abstraída a complexidade sobre como será feita a entrega da mensagem, qual a melhor rota ou qual a tecnologia utilizada pela rede (essas são atribuições de outras camadas presentes no modelo TCP/IP). Um exemplo para esse conceito é o envio de cartas entre duas empresas, onde o funcionário da empresa ABC escreve suas cartas e as envia ao serviço postal que fará o possível para entregar a carta à empresa DEF. O funcionário da empresa ABC não precisa saber se a carta será transportada a pé, de carro ou de caminhão e nem qual itinerário será seguido, basta apenas encaminhar as cartas. Na Figura 2.2 é ilustrada a comunicação entre as aplicações, em execução, em computadores distintos.

Para que seja possível a comunicação entre computadores distintos, é necessária a utilização de protocolos de comunicação. Existem diversos protocolos disponíveis na camada de transporte, dentre os quais o TCP e o UDP. Basicamente, o protocolo UDP, definido na RFC [30], é uma extensão do protocolo IP, para que seja possível a multiplexação do canal de comunicação, uma vez que o protocolo IP não oferece tal serviço. O UDP é um protocolo que provê um serviço não confiável e não orientado a conexão, já o TCP, discutido com mais detalhes a seguir, provê um serviço confiável, orientado a conexão, bem como controle de fluxos e congestionamento.

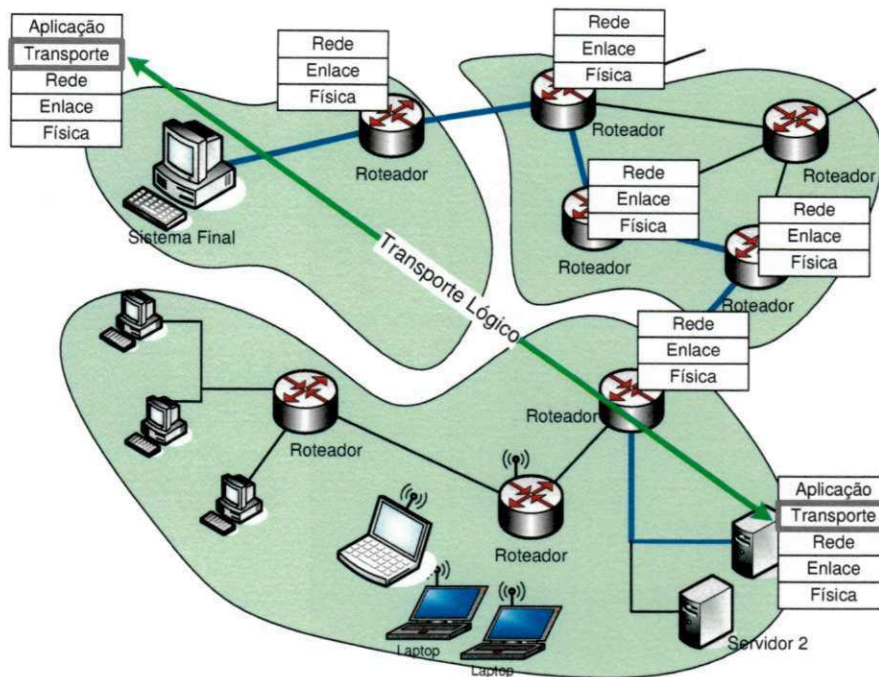


Figura 2.2: Ilustração comunicação lógica entre as aplicações em execução nos computadores (adaptada de [12]).

## 2.2 Protocolo TCP

O protocolo TCP está definido nas RFCs [3, 8, 21, 26, 31] e, como dito anteriormente, este protocolo dispõe de características como:

**Serviço Confiável** : o protocolo garante a entrega de todos os segmentos, mesmo que para isso seja necessário retransmitir alguns deles;

**Orientado à Conexão** : antes da troca de quaisquer informações é estabelecida uma conexão lógica entre os computadores envolvidos;

**Controle de Fluxo** : é o controle provido pelo TCP para evitar que um emissor mais rápido sobrecarregue um receptor mais lento. Essa sobrecarga ocorre quando o emissor envia mais segmentos do que esse receptor consegue tratar;

**Controle de Congestionamento** : segue o mesmo princípio do controle de fluxo, porém, esse controle é focado em não sobrecarregar a rede, evitando que sejam enviados mais segmentos do que a rede pode transportar.

Essas características tornam o TCP um protocolo adequado para aplicações que neces-

sitam de confiabilidade e que não apresentam restrição quanto ao tempo de conclusão dos seus fluxos. Para que essas características estejam disponíveis, o protocolo TCP mantém um conjunto de variáveis denominado Bloco de Controle de Transmissão. O Bloco de Controle de Transmissão, ou simplesmente TCB do inglês *Transmission Control Block*, é um conjunto de informações relativas à cada conexão, que permite um controle detalhado do estado no qual a conexão se encontra [31].

As informações mantidas pelo bloco de controle de transmissão são utilizadas para calcular alguns parâmetros da conexão e também para determinar em qual fase a conexão TCP se encontra. Pode-se citar como variáveis do bloco de controle de transmissão o *Round-trip Time* - RTT (tempo decorrido desde o envio de um pacote até a chegada da confirmação desse), o *Retransmission Timeout* - RTO (tempo que o emissor aguarda antes de retransmitir um pacote) e a janela de congestionamento do inglês *Congestion Window* - CWND. Essa última variável indica quantos segmentos podem ser enviados, do emissor para o receptor, sem que haja a necessidade de confirmação de qualquer segmento [24].

## 2.3 Fluxos

A troca de dados entre computadores ocorre através do envio e da recepção de cadeias de segmentos, as quais podem ser denominadas fluxos. Os fluxos são classificados levando-se em consideração o seu tamanho (medido em número de pacotes ou em bytes) ou a sua duração (medida em unidades de RTT) como fluxos de curta duração (*short lived flows*) e fluxos de longa duração (*long lived flows*), sendo nomeados por alguns autores respectivamente como ratos e elefantes [5, 18, 29, 40].

Qualitativamente, os fluxos de longa duração ou elefantes representam a maior parte do volume total de tráfego ao passo que fazem uso de uma pequena quantidade de conexões, já os fluxos de curta duração ou ratos representam uma pequena quantidade do volume de dados, porém fazem uso de uma grande quantidade de conexões. Quantitativamente, a definição de um fluxo de curta ou longa duração pode ser arbitrariamente definida por aquele que efetua a medição utilizando-se de seus próprios critérios [27].

Vários autores argumentam que o tráfego Web, em sua maioria, é composto por fluxos de curta duração e o tamanho de cada fluxo em média é de 384KB [5, 7, 13, 18, 32].



## 2.4 Controle de congestionamento

Uma vez que vários fluxos vindos de diversas partes da rede podem trafegar quase simultaneamente no enlace e que a rede disponibiliza uma quantidade limitada de recursos, faz-se necessária a criação de um mecanismo para evitar que haja excesso de dados a serem transmitidos, o que poderia levar ao colapso da rede. A esse mecanismo, proposto por Van Jacobson [20] em 1988, deu-se o nome de Controle de Congestionamento.

A criação do controle de congestionamento foi motivada pelo fato presenciado por Van Jacobson e outros pesquisadores, no qual o enlace que unia o Laboratório Nacional Lawrence Berkeley e a Universidade de Berkeley, separados por aproximadamente 400 metros, teve uma redução de sua vazão de 32 *Kbps* para 40 *bps*. Essa redução foi atribuída ao comportamento do TCP implementado no 4.3 BSD (Berkeley Unix) que precisava de aperfeiçoamentos para tratar condições extremas de congestionamento.

O controle de congestionamento do TCP pode ser dividido em fases:

**Partida Lenta** é a fase na qual as transmissões dos fluxos tem seu início e tem por objetivo permitir que cada fluxo avalie a rede e determine quanto do enlace está disponível para sua transmissão. Nessa fase a janela de envio do emissor cresce de maneira exponencial até que a perda de um pacote seja detectada ou enquanto o seu valor for inferior ao do limiar de mudança de fase. O incremento da janela de congestionamento é feito baseando-se nas confirmações recebidas pelo emissor oriundas do receptor. Em outras palavras sempre que uma confirmação é recebida pelo emissor, este aumenta a janela de congestionamento em 1 pacote. Este comportamento faz com que a janela de congestionamento tenha um crescimento exponencial, conforme pode ser visualizado na Figura 2.3.

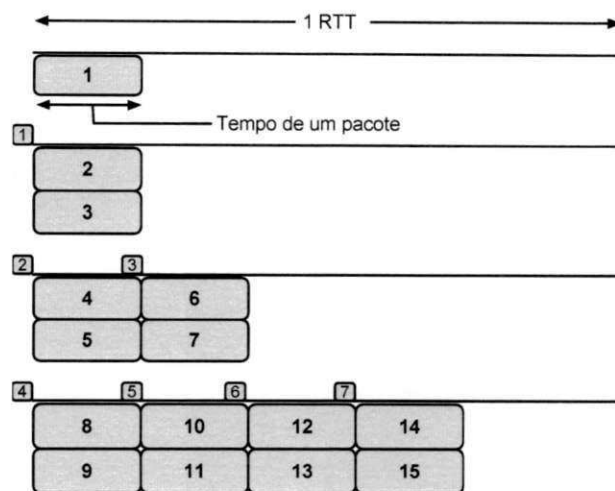


Figura 2.3: Ilustração do comportamento de crescimento da janela na fase de partida lenta (adaptada de [20]).

**Prevenção de Congestionamento** é a fase na qual o fluxo atua depois que o valor da janela de congestionamento ultrapassa o valor do limiar de mudança de fase e foi proposta inicialmente para manter a estabilidade da conexão, pois caso o ritmo de crescimento da janela de congestionamento seja mantido da mesma forma que na fase de partida lenta o limite do enlace será atingido rapidamente o que gerará uma perda e conseqüentemente uma redução da janela de congestionamento. Ao atingir essa fase o incremento da janela é regido pelo variante do TCP que está sendo utilizado. O objetivo da fase de prevenção de congestionamento é manter a janela de congestionamento com um valor elevado, já testado na fase de partida lenta, e aumentar esse valor mais lentamente que a fase anterior, evitando assim que haja uma perda decorrente do excesso de pacotes lançados na rede. O comportamento dessa fase pode ser visualizado na Figura 2.4.

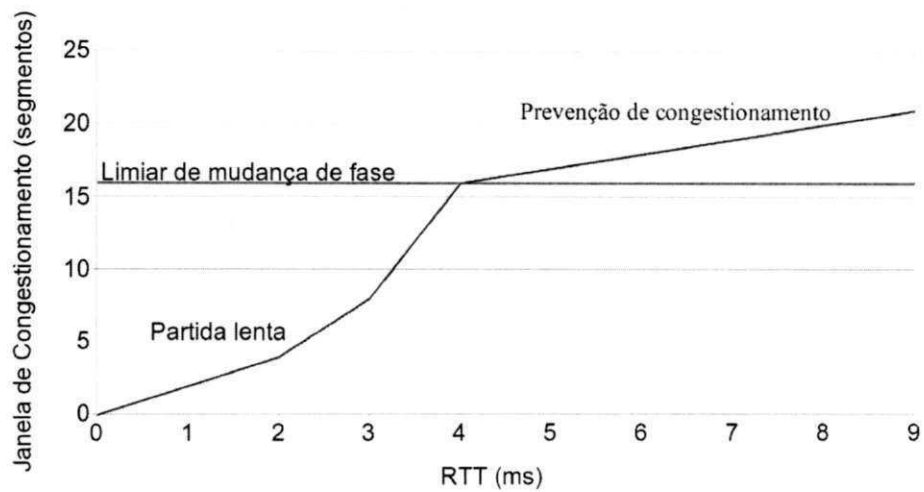


Figura 2.4: Ilustração do comportamento de crescimento da janela na fase de Prevenção de Congestionamento.

Este trabalho de dissertação será focado na fase de partida lenta, uma vez que é nessa fase que os fluxos de curta duração, em sua maioria, são finalizados.

# Capítulo 3

## Trabalhos Relacionados

Neste capítulo, serão discutidas algumas abordagens existentes na literatura relacionadas ao trabalho proposto. Dentre essas abordagens, o agrupamento de conexões e a reutilização do bloco de controle também são discutidas em Savoric [34]. As abordagens apresentadas neste capítulo tem por objetivo melhorar a utilização dos enlaces de dados e com isso permitir que o tempo de conclusão dos fluxos seja reduzido.

Com o objetivo de facilitar o entendimento, os trabalhos aqui apresentados foram agrupados da seguinte forma: alteração da função de crescimento, reutilização do bloco de controle, agrupamento de conexões e utilização de informações de contexto.

### 3.1 Alteração da função de crescimento

Dentre as abordagens cujo objetivo é melhorar a utilização do enlace de dados, pode-se dizer que a alteração da função de crescimento da janela de congestionamento é a mais empregada em virtude da quantidade de trabalhos que são encontrados na literatura.

Essa abordagem consiste na alteração da função matemática que controla o crescimento da janela de congestionamento, como por exemplo as propostas referenciadas em [9, 14, 17, 19, 23, 25, 33]. A alteração dessa função tem por objetivo melhorar a utilização do enlace de dados disponível, geralmente fazendo uso de características do mesmo.

### 3.1.1 *CapStart: An Adaptive TCP Slow Start for High Speed Networks*

No trabalho referenciado em [10] apresenta-se uma proposta de modificação do mecanismo de partida lenta do protocolo TCP voltado para ambientes com alto BDP (*Bandwidth Delay Product*, serve para estipular a quantidade de dados que pode ser enviado pelo enlace durante um determinado período de tempo). De maneira específica, discute-se a relação de compromisso existente entre o mecanismo de partida lenta tradicional, que pode levar a uma perda considerável de segmentos em virtude do seu crescimento exponencial, e o mecanismo de partida lenta limitado [15], que estipula um limite máximo, em termos de segmentos, até onde o crescimento deve ser exponencial, que pode forçar o TCP a interromper seu crescimento exponencial antes do necessário. Neste contexto, os autores fazem uso de um mecanismo que tenta detectar a real capacidade do enlace de transmissão e decidir, portanto, quando utilizar a abordagem tradicional ou a abordagem com crescimento limitado. Tal mecanismo faz uso das diferenças de *RTT* apresentadas pelos pacotes transmitidos de modo a detectar o aumento na fila do roteadores e, por consequência, um aumento no nível de utilização do canal de transmissão.

Em cenários com enlace de baixa capacidade o mecanismo de partida lenta limitada fez com que o *Capstart* conseguisse reduzir o tempo de conclusão dos fluxos uma vez que conseguiu reduzir a agressividade do crescimento exponencial e também a quantidade de segmentos perdidos. Porém, os autores ainda estão estudando maneiras de conseguir um melhor ajuste do algoritmo de partida lenta para as diversas velocidades de enlace de maneira robusta e segura e investigando se é perceptível, a partir da aplicação, a utilização do classificador de capacidade do enlace para aumentar de maneira mais rápida o crescimento da janela de congestionamento.

### 3.1.2 *Burst TCP: an approach for benefiting mice flows*

No trabalho referenciado em [18] apresenta-se uma proposta de alteração do mecanismo de crescimento da janela de congestionamento do TCP *Reno* com o objetivo de beneficiar os fluxos de curta duração. Essa alteração permite-lhes aumentar a janela de congestionamento de maneira mais agressiva com relação aos fluxos de longa duração que fazem uso da mesma abordagem.

O algoritmo utilizado no *Burst TCP* faz com que a janela de congestionamento de um fluxo cresça de forma inversamente proporcional ao seu tamanho. Todos os fluxos são inicialmente classificados como fluxos de curta duração e crescem sua janela mais agressivamente, ao passo que quando o limiar é atingido e o fluxo é reclassificado como fluxo de longa duração sua janela de congestionamento volta a ter o crescimento padrão do *TCP Reno*.

O protocolo dispõe de mecanismos que reduzem sua agressividade, no aumento da janela de congestionamento, fazendo com que seu comportamento seja o mesmo encontrado no *TCP Reno*. Esses mecanismos para redução da agressividade do protocolo são acionados quando a janela de congestionamento de um fluxo ultrapassa um determinado limiar ou quando uma perda é detectada.

Em linhas gerais o B-TCP melhorou o tempo de transferência para os fluxos de curta duração e, em alguns casos, os de longa duração também. Além do mais, os resultados mostraram que num cenário de grande congestionamento, o B-TCP tem comportamento bastante similar ao *TCP NewReno*. Todavia, mesmo fazendo uso de um crescimento diferenciado da janela de congestionamento para os fluxos de curta duração, a fase de partida lenta é repetida, desde o início, para todos os fluxos.

### 3.1.3 *TCP Startup Performance in Large Bandwidth Delay Networks*

No trabalho referenciado em [40] apresenta-se uma proposta para modificação do valor da variável *ssthresh* de modo a elevar o ponto até onde o TCP irá funcionar no modo de partida lenta e conseqüentemente com crescimento exponencial. Para tal, o mecanismo apresentado faz uso da abordagem adotada pelo *TCP Westwood* de modo a determinar a largura de banda realmente disponível fazendo uso desse valor para elevar o valor de *ssthresh* possibilitando que o TCP permaneça no regime de crescimento mais agressivo durante um tempo prolongado.

Apesar da proposta apresentada possibilitar um ajuste mais apurado do valor da variável *ssthresh*, ela foca apenas no período da transmissão posterior à primeira fase de partida lenta, visto que, durante o primeiro período de partida lenta, o valor da variável *ssthresh* é definido como o maior possível. Neste sentido, fluxos curtos que finalizam a transmissão ainda durante a primeira fase de partida lenta não conseguem obter vantagem a partir da utilização desta abordagem.

Apesar dessa abordagem possuir o potencial para proporcionar uma boa utilização dos enlaces de dados, seu desenvolvimento faz uso de características específicas dos enlaces para os quais são direcionadas. As soluções que fazem uso dessa abordagem obtêm bons resultados, em sua maioria, apenas nos cenários para os quais foram projetadas.

## 3.2 Reutilização do bloco de controle

Nesta abordagem o protocolo TCP é alterado para que as informações adquiridas por conexões anteriores sejam utilizadas como ponto de partida para o estabelecimento das novas conexões. Isso permite que as novas conexões iniciem suas transmissões sem a necessidade de avaliar completamente o enlace.

### 3.2.1 *TCP Fast Start*

No trabalho referenciado em [29] apresenta-se uma alteração para que as conexões armazenem os parâmetros avaliados recentemente e reaproveite-os quando uma nova conexão precisar ser estabelecida. As informações previamente avaliadas não são compartilhadas com qualquer outro computador.

Como a utilização de uma janela de congestionamento inicial maior pode gerar perda de pacotes em virtude da sobrecarga dos roteadores, o autor propôs a inclusão de um campo no cabeçalho do TCP denominado *Drop Priority*. Esse campo indica ao roteador que aqueles dados fazem parte de uma conexão mais recente e que podem ser descartados com menor prejuízo, uma vez que a conexão ainda está em fase de estabelecimento. A necessidade de atualização da pilha TCP/IP presente nos roteadores da rede funciona como um fator limitante à adoção do mecanismo proposto.

### 3.2.2 *SPAND: Shared Passive Network Performance Discovery*

No trabalho referenciado em [35] apresenta-se uma abordagem para descoberta e compartilhamento de dados referentes ao desempenho da rede. Para tal, cada um dos dispositivos presentes na rede local deve enviar relatórios contendo informações como a largura de banda e estatísticas sobre a perda de pacotes, utilizando um pacote com o formato representado na

Figura 3.1.

0	15	16	31
Version	Type	Transport Pr.	App. Class
Source IP Address			
Source Port		Dest Port	
Dest IP Address			
NTP Timestamp, most sig word			
NTP Timestamp, least sig word			
Length of Sample in octets			
Duration of sample in ms			
Total Packets Received			
Total Packets Lost			
Packet Size in octets			

Figura 3.1: Ilustração do relatório de desempenho (adaptada de [35]).

Empregou-se também um computador que efetua a captura e a análise do tráfego da rede e envio de relatórios baseados nessa análise para o servidor de desempenho. São atividades executadas pelo servidor de desempenho a recepção dos relatórios, consolidação das informações e o compartilhamento das informações com os dispositivos da rede. A arquitetura apresentada na Figura 3.2 ilustra os componentes envolvidos na solução e também a troca de informações entre eles.

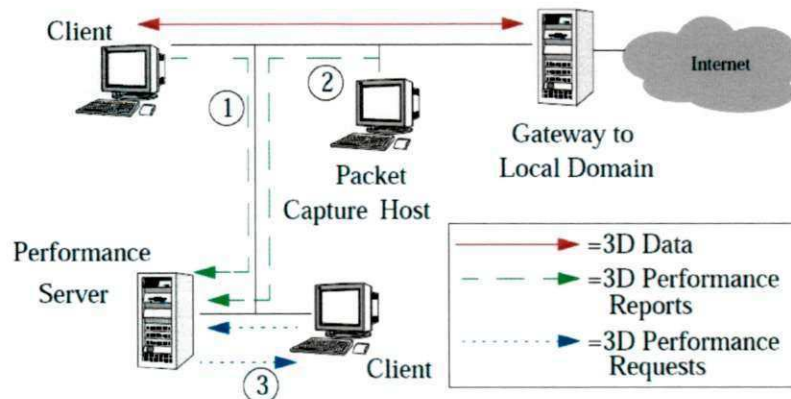


Figura 3.2: Ilustração da arquitetura empregada pelo SPAND (adaptada de [35]).



Os dispositivos presentes na rede podem consultar o servidor de desempenho para obter as informações já consolidadas utilizando os relatórios representados nas Figuras 3.3 e 3.4

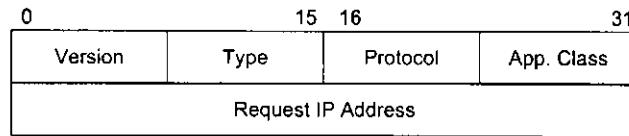


Figura 3.3: Ilustração do relatório de requisição (adaptada de [35]).

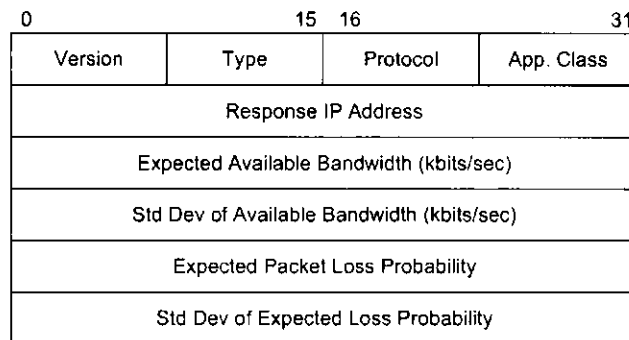


Figura 3.4: Ilustração do relatório de resposta (adaptada de [35]).

Essa abordagem apresenta dois fatores limitantes: primeiro é o fato de necessitar de um servidor central para avaliação e persistência das informações (esse servidor pode ser considerado um ponto de falha da proposta, uma vez que, sem ele não há o compartilhamento das informações); segundo que a medição passiva das informações tem dificuldades para o isolar e medir o tráfego.

### 3.3 Agrupamento de conexões

Outra abordagem encontrada na literatura é o agrupamento de conexões. Nessa abordagem as conexões são agrupadas com o objetivo de tornar o algoritmo *AIMD* mais eficiente e proporcionar um melhor aproveitamento do enlace disponível mantendo uma certa equidade entre os fluxos.

No trabalho referenciado em [22], a abordagem de agregação de fluxos foi empregada por meio do desenvolvimento de um gerenciador de conexões com 3 camadas. Cada camada

controla o incremento da janela de congestionamento de maneira diferente:

- **controle macroscópico** agrupa os fluxos utilizando um fator de compensação  $K_R$  baseado na observação do RTT de cada fluxo. A técnica de compensação do RTT modifica o algoritmo de atualização da janela de congestionamento ao receber uma confirmação (ACK), pois  $K_R$  é utilizado como base para a função de atualização da janela de congestionamento;
- **controle microscópico** possui dois modos de operação: o estado estável onde o protocolo não precisa ser agressivo ( $K_R = 1$  ou desligado) o que reduz a taxa de perda de pacotes; e o estado transiente onde o fluxo precisa ser mais agressivo ( $K_R = 0.5$  (RTT)  $1/3$  ou ligado);
- **controle intermediário** tem por função aumentar rapidamente o fator  $K_R$  de maneira a permitir um uso eficiente do enlace disponível enquanto não for detectada competição entre os fluxos, quando essa competição é detectada  $K_R$  recebe o valor 1.

A proposta mostrou-se eficiente ao permitir o aumento da janela de congestionamento para um valor ótimo de maneira mais rápida que o TCP padrão e também obteve uma redução menos drástica após a detecção de uma perda. Como resultado, a adaptação da janela de congestionamento do ALTCP é mais eficiente na utilização dos enlaces de alta velocidade. Contudo, o ALTCP ainda há a subutilização do enlace uma vez que é necessário avaliar todo o enlace quando uma nova conexão é iniciada.

### 3.4 Utilização de informações de contexto

Na literatura também são encontradas abordagens que fazem uso de informações de contexto da conexão. Essas informações permitem aos roteadores selecionar quais pacotes devem ser descartados de maneira mais acertada, pois são estudos indicam que a perda de pacotes pode aumentar quase em 8 vezes o tempo de transferência dos fluxos de curta duração [4].

Em Williamson [41] propõe-se o compartilhamento de informações entre diferentes conexões e também entre as camadas da pilha TCP/IP. São compartilhadas entre as camadas da pilha TCP/IP informações como estatísticas de perda de pacotes e também se a conexão está próxima do início ou do final.

São propostas alterações tanto na pilha TCP/IP dos emissores como dos roteadores que passam a considerar, no descarte de pacotes, se a conexão está próxima do início ou do fim. Também é sugerido que a camada de aplicação informe às camadas mais baixas da Pilha TCP/IP quantos *bytes* serão transferidos durante a conexão.

A solução proposta tem potencial para obter valores mais refinados para os parâmetros das conexões, em virtude da quantidade de informações utilizadas. Porém, o fornecimento de informações oriundas de outras camadas torna-se um complicador para sua adoção, uma vez que será necessário que o desenvolvedor tome a precaução de compartilhar as informações sob o risco de inviabilizar a utilização da solução. Outro ponto que pode ser questionado é que os roteadores devem realizar diferenciações de fluxos entre os pacotes transportados, o que aumentaria substancialmente a sobrecarga de processamento e, portanto, sua ampla adoção por fabricantes.

### 3.5 Considerações sobre os trabalhos relacionados

Existem outras abordagens relacionadas, como por exemplo: *i)* Dukkupati [13] propõe a atualização do valor da janela de congestionamento inicial, uma vez que o valor atualmente utilizado foi definido em 2002 por Floyd [2], *ii)* [6] onde propõe-se a inclusão de um protocolo na camada de aplicação e é voltado principalmente para aplicações *Web*, *iii)* Yang [42] propõe a substituição do protocolo TCP pelo XCP (*eXplicit Control Protocol*), substituição essa que apresenta benefício, pois esse novo protocolo não considera apenas se o enlace está congestionado, como é feito atualmente no TCP utilizando o ECN (*Explicit Congestion Notification*), o controle de congestionamento é feito baseando-se no grau de utilização do enlace. Essa informação é fornecida pelos roteadores, fazendo com que para utilização dessa abordagem seja necessária a alteração dos dispositivos ligados à rede.

Na avaliação experimental realizada pode-se observar uma queda no tempo de conclusão dos fluxos de curta duração de cerca de 30% com a utilização do TCP-Share, em relação aos fluxos que fizeram uso do TCP Cubic padrão. Outras abordagens da literatura também foram comparadas aos protocolos TCP nos quais foram desenvolvidas, como por exemplo o *Caps-tart* e o TCP *Reno*, o *Busrt* TCP e TCP *Fast Start* com relação ao TCP *NewReno* e obtiveram redução no tempo de conclusão dos fluxos de curta duração de cerca de 40% a 60%. Apesar

dos resultados indicarem uma maior redução no tempo de conclusão dos fluxos nas abordagens presentes na literatura, é possível dizer que estes trabalhos são fortemente acoplados ao TCP com o qual foram desenvolvidos e que os protocolos utilizados como base não apresentam bom desempenho em redes com alto BDP, como as utilizadas atualmente. Além disso, as abordagens com as melhores reduções fazem uso da comunicação direta entre diferentes camadas da pilha TCP/IP. Fato este, que obriga ao desenvolvedor de aplicações a conhecer os detalhes da abordagem utilizada para obter seus benefícios. Por outro lado, a abordagem proposta nesta dissertação apesar de obter um percentual menor de redução, foi comparada a uma variante do protocolo TCP que já faz uso de boa parte do enlace disponível e utilizou apenas informações da camada de transporte para alcançar esse resultado. Assim sendo mantém-se a independência entre as camadas da pilha TCP/IP e também a compatibilidade do TCP-Share com as aplicações atualmente em uso.

Tabela 3.1: Comparativo entre as soluções propostas

	Focada em	Reutiliza informação	Compartilha informação	Alterações necessárias
<i>Exploring Round Trip Time fairness for Adaptive Layered Transmission Control Protocol</i>	Otimização para ambientes com alto BDP	Não	Não	Borda da rede
<i>CapStart: An Adaptive TCP Slow Start for High Speed Networks</i>	Otimização para ambientes com alto BDP	Não	Não	Borda da rede
<i>Burst TCP: an approach for benefiting mice flows</i>	Beneficiar fluxos de curta duração	Não	Não	Borda da rede e Núcleo da rede
<i>TCP Startup Performance in Large Bandwidth Delay Networks</i>	Alteração do limiar de mudança de fase	Não	Não	Borda da rede
<i>A case for Context-Aware TCP/IP</i>	Melhoria de desempenho	Não	Não	Borda da rede e Núcleo da rede
<i>TCP Fast Start: A technique for speeding up web transfers</i>	Mudança da janela de congestionamento inicial	Sim	Não	Borda da rede e Núcleo da rede
<i>SPAND: Shared Passive Network Performance Discovery</i>	Melhoria de desempenho	Sim	Sim	Borda da rede e Núcleo da rede e serviços da rede
TCP-Share	Melhoria de desempenho	Sim	Sim	Borda da rede

## Capítulo 4

### TCP-Share

Como discutido anteriormente, no Capítulo 2, o algoritmo de controle de congestionamento avalia o enlace com o objetivo de atingir a maior velocidade disponível. Mas, em virtude do comportamento das atuais variantes do TCP, nas quais o valor inicial da janela de congestionamento é estático, alguns fluxos encerram-se antes de utilizarem toda a capacidade disponível do enlace ou mesmo antes de mudar da fase de partida lenta para a fase de prevenção de congestionamento.

Com o objetivo de melhorar a utilização da largura de banda disponibilizada pelos provedores de infraestrutura existem diversas propostas, como descritos no Capítulo 3. Porém, as soluções que reutilizam parâmetros de conexões anteriores apresentam problemas como a necessidade de comunicação direta entre as camadas da pilha TCP/IP, em outras palavras, as aplicações precisam informar alguns dados a camadas com as quais normalmente não possuem comunicação direta, o que aumenta a complexidade no desenvolvimento das aplicações que utilizarão a solução.

Dessa forma, o TCP-Share proporciona a redução da necessidade da avaliação completa do enlace através do compartilhamento, entre computadores localizados na mesma rede local, de parâmetros de conexão anteriores. A utilização desses parâmetros permite que as novas conexões, ao invés de iniciarem suas transmissões com um valor estático e baixo, iniciem com um valor mais elevado e adequado ao enlace no qual esse computador está instalado.

## 4.1 Arquitetura da solução

Na Figura 4.1 é apresentada uma visão geral da solução, a qual pode ser dividida em duas partes: o grupo de compartilhamento e o módulo local.

O grupo de compartilhamento é a via pela qual o TCP-Share troca os parâmetros das conexões com outros hospedeiros. A formação do grupo de compartilhamento não segue uma regra específica, pois existem diversas metodologias já consolidadas para a criação desses grupos de compartilhamento, desde as mais simples como a distribuição por difusão e a difusão seletiva, até as abordagens mais elaboradas que dispõem de controle de admissão e tolerância a falhas como a DHT (*Distributed Hash Table*). Outra abordagem possível para o grupo de compartilhamento é a centralizada, na qual um dos hospedeiros é eleito para receber as informações e redistribuí-las. Independente da abordagem utilizada, para o TCP-Share é o bastante que o grupo de compartilhamento exista e que seja possível ao módulo local a troca de mensagens através dele entre os hospedeiros, independente de topologia ou regra de formação.

O módulo local é responsável pelo monitoramento das conexões originadas no hospedeiro, salvar os parâmetros dessas conexões localmente e também mantê-los atualizados. A atualização desses parâmetros é feita em primeiro lugar utilizando os parâmetros das novas conexões que tiveram origem no hospedeiro e em segundo lugar com base nos parâmetros recebidos através do grupo de compartilhamento. Quando o módulo local é ativado, é testada a sua conectividade com o grupo de compartilhamento, caso não seja possível conectar o grupo de compartilhamento, o módulo local continua salvando os parâmetros da conexão para uso pelo próprio hospedeiro, o que permite que em redes onde haja apenas um hospedeiro utilizando a solução, este hospedeiro possa aproveitar melhor a largura de banda disponível, dentro das devidas limitações de funcionamento.

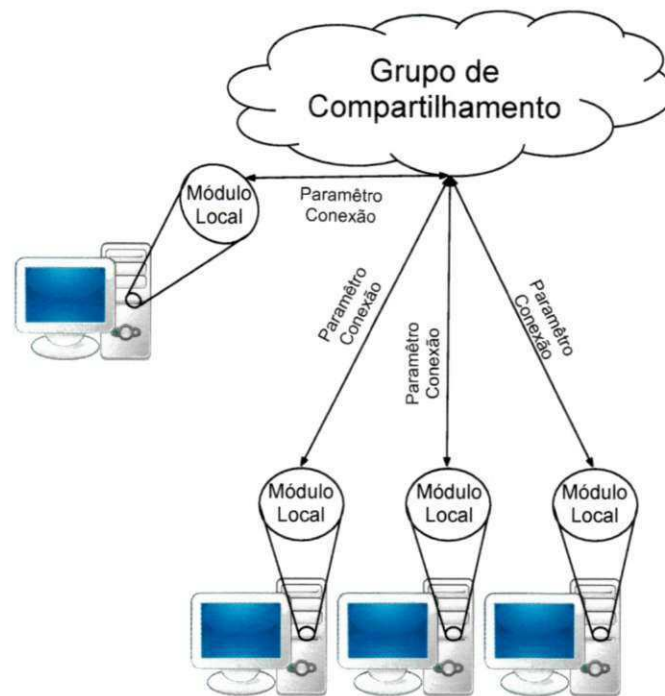


Figura 4.1: Ilustração da arquitetura da solução proposta.

O módulo local, como ilustrado na Figura 4.2, pode ser dividido em duas partes:

**Controle de congestionamento**, comunica-se diretamente com a camada de transporte da pilha TCP/IP, isso permite o acesso direto aos parâmetros de uma conexão. O controle de congestionamento tem por função controlar o aumento e a diminuição da velocidade de transferência e também salvar os parâmetros da conexão;

**Compartilhamento**, é responsável pela troca de mensagens entre o controle de congestionamento e o grupo de compartilhamento.

## 4.2 Compartilhamento da informação

Internamente, o bloco de controle pode ser compartilhado entre as conexões possibilitando que todas possam beneficiar-se desses valores. Do mesmo modo, o compartilhamento de informações do bloco de controle entre os hospedeiros do mesmo grupo de compartilhamento, pode ocorrer de maneira centralizada, onde se agrega mais um servidor/serviço à rede, ou distribuída, através da utilização de grupos difusão ou difusão limitada por exemplo. Isto per-



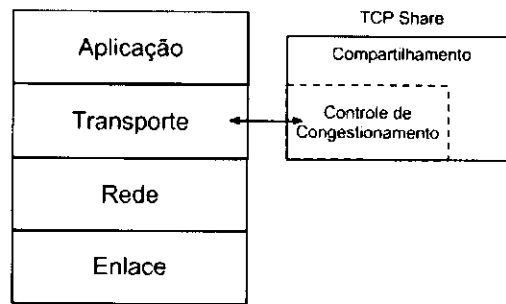


Figura 4.2: Ilustração da arquitetura interna do TCP-Share e comunicação com a pilha TCP/IP.

mite maior flexibilidade ao administrador da rede de adotar a estratégia que melhor adapta-se à sua rede.

A priori, independente do método de compartilhamento que esteja em funcionamento, o envio da informação sobre a janela de congestionamento é feito após um dos computadores enviar uma solicitação. Essa solicitação é respondida por outro computador que compartilha o valor da janela de congestionamento que mantém armazenado localmente. Na Tabela 4.1 é apresentado um resumo comparando as características de cada método de compartilhamento. As características apresentadas na tabela são: facilidade de configuração, "alta" caso não seja necessário efetuar quaisquer alterações no caso da mudança de uma rede para outra e "baixa" caso seja necessária alguma configuração; segurança, que é o quanto a fonte do compartilhamento da informação é confiável; geração de tráfego desnecessário. Entende-se por tráfego desnecessário todo o tráfego recebido por um computador que não fará uso dessas informações ao recebê-las.

Tabela 4.1: Resumo das características de cada método de compartilhamento.

Método de compartilhamento	Facilidade de configuração	Segurança	Gera tráfego desnecessário
Difusão	Alta	Baixa	Sim
Difusão limitada	Alta	Baixa	Não
Centralizado	Baixa	Alta	Não

A seguir são ilustrados e descritos os métodos de compartilhamento da informação cita-

dos na Tabela 4.1.

No compartilhamento de informação utilizando difusão todos os computadores que estão na rede local recebem a informação compartilhada, como pode ser visto na Figura 4.3. Esse método de envio é indicado para redes homogêneas nas quais todas as máquinas fazem uso da solução.

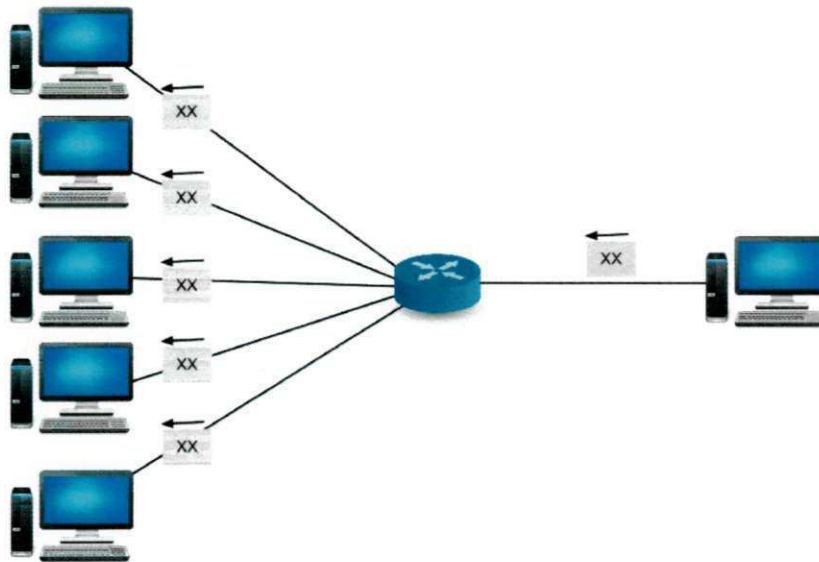


Figura 4.3: Ilustração do compartilhamento da informação utilizando o envio por difusão.

O compartilhamento de informação utilizando difusão limitada (Figura 4.4) é indicado quando nem todos os computadores presentes na rede local fazem uso da solução. A utilização desse método de compartilhamento permite que apenas os dispositivos cadastrados no grupo difusão limitada recebam as informações que estão sendo compartilhadas.

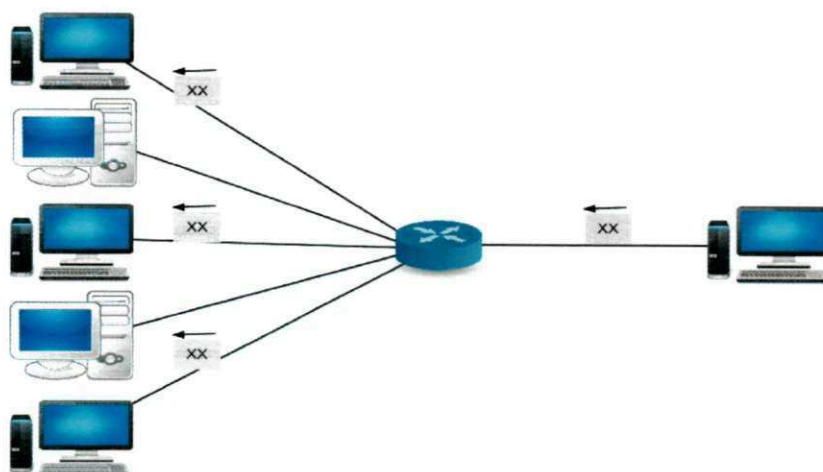


Figura 4.4: Ilustração do compartilhamento da informação utilizando o envio por difusão limitada.

O método de compartilhamento centralizado (Figura 4.5) necessita que um servidor/serviço seja disponibilizado na rede e que todas as máquinas sejam configuradas para consultá-lo. Nesse método de compartilhamento as máquinas informam ao servidor os parâmetros que foram salvos localmente. O servidor armazena os parâmetros, consolida-os e depois compartilha com as demais máquinas. Esse método de compartilhamento também provê um maior nível de segurança, uma vez que todas as máquinas compartilham sua informação de janela de congestionamento com o servidor e apenas este servidor compartilha a informação com os clientes. Porém, ele necessita de uma maior interação durante a configuração da solução, uma vez que o endereço do servidor de informação é fornecido durante a configuração, que pode ser feito tanto estaticamente quanto dinamicamente através de determinados serviços padrões numa rede, como o serviço de DHCP (*Dynamic Host Configuration Protocol*) no qual pode ser definida uma diretiva, ilustrada logo abaixo, informando qual o endereço do servidor do TCP-Share para aquela rede.

```
1 host tcpshareserver {
2   option host-name "tcpshareserver.com";
3   hardware ethernet 00:11:22:33:44:55;
4   fixed-address 192.168.1.2;
5 }
```

Código Fonte 4.1: Diretiva utilizada na configuração automática via DHCP

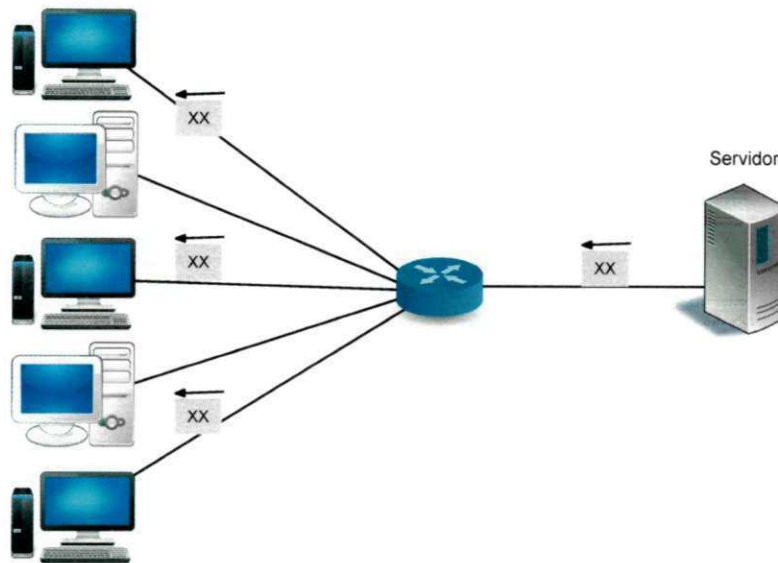


Figura 4.5: Ilustração do compartilhamento da informação utilizando um servidor centralizado.

### 4.3 Funcionamento

O funcionamento do TCP Share, ilustrado na Figura 4.6, pode ser dividido em atividades que são desempenhadas pelo módulo local e no grupo de compartilhamento.

A seguir estão descritas as atividades desempenhadas pelo módulo local.

- **Limpeza do valor armazenado:** Esse procedimento deve ser executado quando o protocolo TCP for iniciado. Ele tem a função de apagar o histórico armazenado da janela de congestionamento, evitando com isso que um dispositivo, ao mudar de rede, traga informação que possa prejudicar a sua nova rede.

```

1 inicio limpar_valor_armazenado
2     salva_valor_padrao_em_disco :
3         armazenar_informacao( 4 )
4 fim limpar_valor_armazenado

```

Código Fonte 4.2: Algoritmo para limpeza do valor armazenado

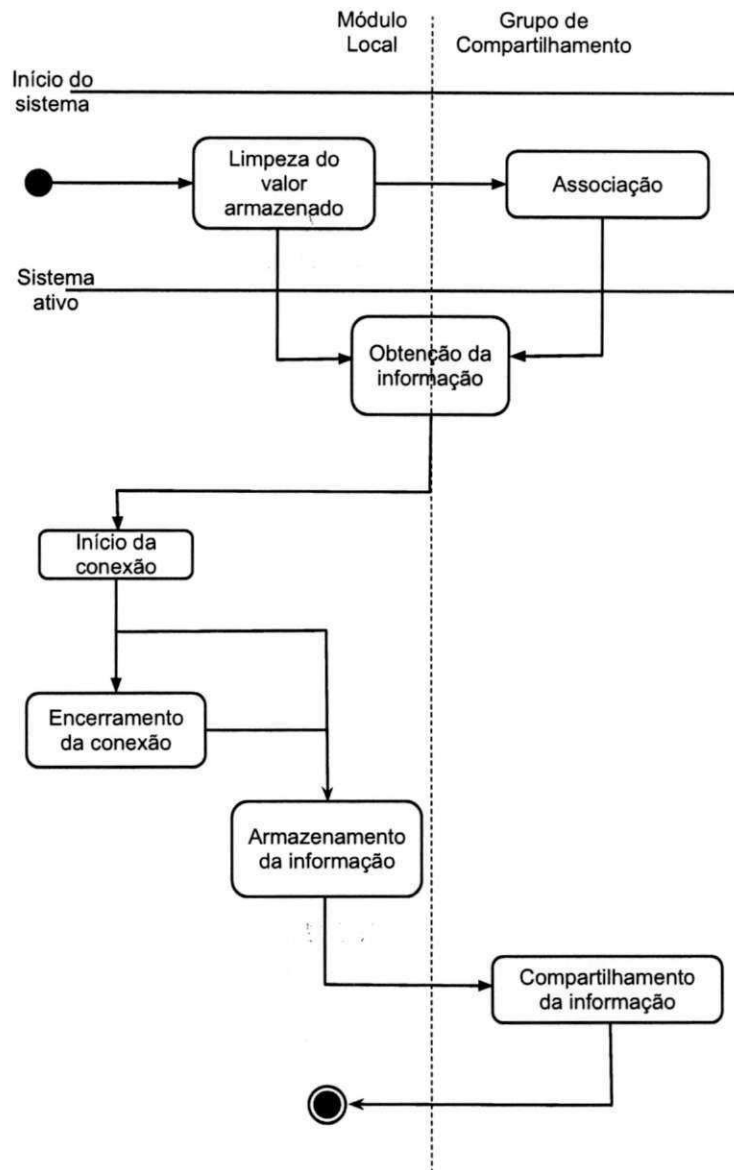


Figura 4.6: Fluxograma representando o funcionamento da solução proposta.

- **Obtenção da informação:** Nesse procedimento deve ser feita a leitura do valor da janela de congestionamento que foi armazenado. Nesse caso em especial, a informação é lida no arquivo armazenado na máquina local.

```
1 inicio obter_informacao
2
3     ler_salvo_valor_padrao_em_disco :
4         janela_de_congestionamento =
5             janela_de_congestionamento_em_disco
6 fim obter_informacao
```

Código Fonte 4.3: Algoritmo para obtenção da informação

- **Início da conexão e Encerramento da conexão:** São procedimentos relacionados à transferência de dados e transcorrem quase da mesma maneira que na implementação do padrão do TCP. A diferença entre elas é o fato da implementação proposta iniciar a conexão com um valor armazenado e não com um valor estático. Além disso, o valor da janela de congestionamento é armazenado, na máquina local, durante a conexão e após o seu término.

```
1 inicio transmitir_dados
2
3     obter_informacao ()
4
5     transmissao
6     enquanto_existir_dados_para_transmitir :
7
8         envio_dos_dados
9
10        // Evita que a cada ciclo de transmissão a janela de
11        congestionamento seja armazenada
12        se janela_avaliada <> janela_compartilhada
13            armazenar_da_informacao (janela_avaliada)
14        fim se
15    fim enquanto_existir_dados_para_transmitir
16
17    armazenar_da_informacao (janela_avaliada)
```

```
18
19     // limpeza das variáveis do bloco de controle do TCP
20     . . .
21
22     fim transmissao
23
24 fim transmitir_dados
```

Código Fonte 4.4: Algoritmo para transmissão de dados

- **Armazenamento da informação:** Esse procedimento deve ser executado durante uma conexão e também ao seu término e é responsável por salvar o valor da janela de congestionamento localmente. Este procedimento é invocado várias vezes para permitir que uma nova conexão aproveite informação de outra que ainda está em andamento.

```
1 inicio armazenar_informacao
2 \textbf{Entrada} valor_para_armazenamento
3
4     salvar_valor_em_disco :
5
6         janela_de_congestionamento_em_disco =
7             janela_de_congestionamento_avalida * 0.5
8 fim armazenar_informacao
```

Código Fonte 4.5: Algoritmo para armazenamento da informação

A seguir estão descritas as atividades desempenhadas no grupo de compartilhamento.

- **Ativação:** Esse procedimento deve acontecer logo após o procedimento de limpeza citado anteriormente. Ele permite que o módulo de compartilhamento saiba onde deve buscar e onde publicar as informações compartilhadas.

```
1 inicio ativar_compartilhamento
2
3     verifica_endereço_de_compartilhamento :
4
5         se diretiva_DHCP_recebida então
```

```
6         endereco_compartilhamento = endereço_servidor
7
8         senão obteve_resposta_broadcast então
9             endereco_compartilhamento = endereço_broadcast
10
11        senão obteve_resposta_multicast então
12            endereco_compartilhamento = endereço_broadcast
13
14        fim se
15
16    ativação :
17
18        laço
19            envia_informação_avalizada
20
21            aguarda X segundos
22
23        fim laço
24
25 fim ativar_compartilhamento
```

Código Fonte 4.6: Algoritmo para o compartilhamento da informação

- **Obtenção da informação:** Esse procedimento consulta a fonte de informação compartilhada, cadastrada anteriormente, para adquirir alguma informação. Após o recebimento de um valor, este procedimento altera o valor armazenado para o valor que acaba de ser recebido.

```
1 inicio obter_informacao
2
3     obter_informação :
4
5         valor_compartilhado = solicita_informação (
6             endereco_compartilhamento)
7
8         armazenar_informacao (valor_compartilhado)
9 fim obter_informacao
```



---

Código Fonte 4.7: Algoritmo para obtenção da informação

- **Consulta valor armazenado:** Nesse processo o módulo de compartilhamento deve acessar o arquivo armazenado localmente e recuperar o valor lá contido.
- **Compartilhamento da informação:** Esse procedimento deve ser disparado de tempos em tempos para compartilhar o valor da janela de congestionamento armazenado localmente.

## 4.4 Armazenamento da informação

Como uma forma de permitir que as máquinas tirem proveito da solução, mesmo não havendo outras máquinas compartilhando informação e também como forma de evitar possíveis ataques pelo compartilhamento de informações falsas, cada dispositivo que utiliza a solução, deve manter um registro dentro do seu sistema de arquivos. Esse registro contém os valores das janelas já avaliadas, bem como sua origem, uma vez que esses valores podem ter origem tanto na máquina local (que escreve o valor da janela de congestionamento diretamente no arquivo) quanto no compartilhamento via rede (que é escrito pelo módulo de compartilhamento da informação quando outra máquina efetua o compartilhamento).

## 4.5 Segurança

Com o compartilhamento da informação da janela de congestionamento consegue-se evitar que, a cada conexão, a rede seja totalmente reavaliada. Porém, abre-se uma brecha para que ataques sejam executados. Esses ataques podem ser baseados no compartilhamento de informações incorretas [1], fazendo com que as máquinas subutilizem o enlace (compartilhamento de um valor muito baixo para a janela de congestionamento) ou provoquem uma grande quantidade de perdas de pacotes (compartilhamento de um valor muito alto da janela). Para reduzir a ocorrência de problemas relacionados ao compartilhamento de informações erradas, pode-se adotar uma abordagem semelhante a apresentada em [11], na qual é traçada uma estratégia para o gerenciamento de confianças em redes pervasivas.

Este trabalho de dissertação não teve por objetivo a definição de políticas de validade da informação local, uma vez que tal validação é bastante complexa, demandando estudos mais aprofundados e testes mais abrangentes. Assim sendo, optou-se por considerar que uma informação é válida localmente enquanto o sistema final estiver em funcionamento.

## 4.6 Redução da agressividade

Permitir às novas conexões utilizarem a janela de congestionamento avaliada por uma conexão anterior pode fazer com que ocorram perdas de pacotes em virtude da grande quantidade de pacotes enviada. Para diminuir esse problema, o controle de congestionamento do TCP deve ser alterado de tal maneira que, quando uma nova conexão precisar ser estabelecida, a janela de congestionamento compartilhada que será utilizada por ela, representa apenas uma fração da janela de congestionamento já avaliada anteriormente. Isso permite aos novos fluxos começarem suas transmissões aproveitando uma maior quantidade do enlace que foi avaliado recentemente, bem como aos outros fluxos se acomodarem mais facilmente.

No mecanismo proposto a janela é reduzida utilizando-se um fator  $\alpha$ , que teve o valor de 0,5, pois as primeiras versões do TCP reduzem o seu limiar de mudança de fase a metade quando uma perda era detectada e partindo do pressuposto que a conexão que efetuou o compartilhamento ainda está ativa e pode disponibilizar parte da sua janela para a nova conexão.

Outra abordagem para a redução da janela de congestionamento compartilhada seria empregar o mesmo coeficiente de redução utilizado pelo controle de congestionamento adotado, assim a redução da janela de congestionamento seguiria o mesmo padrão do caso de detecção da perda de um pacote.

# Capítulo 5

## Avaliação Experimental

Neste capítulo são descritas as alterações efetuadas no código do simulador de rede NS3 <sup>1</sup>, o ambiente de simulação adotado e os resultados obtidos. Essa avaliação experimental teve por objetivo comparar o desempenho do mecanismo proposto neste trabalho de dissertação com relação ao protocolo TCP padrão.

### 5.1 Alterações no Simulador de redes NS3

Com o objetivo de avaliar a solução proposta neste trabalho foi utilizado o Network Simulator 3. Nesse simulador foi criada uma cópia alterada do controle de congestionamento TCP Cubic. Essa escolha foi feita pois os testes feitos utilizando o TCP Cubic indicam que essa variante do TCP necessita de pouco tempo para que ocorra a convergência de seus fluxos, apresenta uma boa equidade com outras transmissões que compartilham o mesmo enlace de dados e também pelo fato dela ser a variante do TCP mais utilizada nos sistemas Linux atualmente.

No TCP Cubic houveram as alterações dos procedimentos:

- *bictcp\_reset*: inicialmente este procedimento servia para atribuir os valores padrões aos parâmetros utilizados em uma conexão. Como nova atribuição esse procedimento passou a reduzir a janela de congestionamento que está sendo compartilhada pelo fator  $\alpha$  e também foi invocado um procedimento que armazena o valor dos parâmetros localmente;

---

<sup>1</sup><http://www.nsnam.org/>

- *bictcp\_cong\_avoid*: este procedimento tem como função inicial selecionar em qual fase o controle de congestionamento está atuando. Nele foi invocado um procedimento que armazena o valor dos parâmetros localmente.

No compartilhamento das informações durante a simulação, utilizou-se uma variável do NS3 que permite a todas as máquinas dentro do simulador acessar ao valor da janela de congestionamento compartilhada, simulando assim um grupo de difusão.

## 5.2 Cenário

A distribuição das máquinas durante as simulações seguiu uma topologia conhecida como *Dumbbell* que é ilustrada pela Figura 5.1. Essa topologia consiste de uma simplificação para a comunicação entre dispositivos, pois nela todo o caminho percorrido pelos segmentos é considerado como sendo apenas um salto entre os roteadores.

Para a montagem dessa topologia foram criadas duas redes compostas por 10 dispositivos cada. A rede localizada à esquerda continha os nós clientes, ou seja, os nós que enviavam as requisições. Já a rede localizada à direita continham os nós servidores, aqueles que recebiam e respondiam as requisições. Interconectando essas duas redes havia 2 nós que faziam o papel de roteadores.

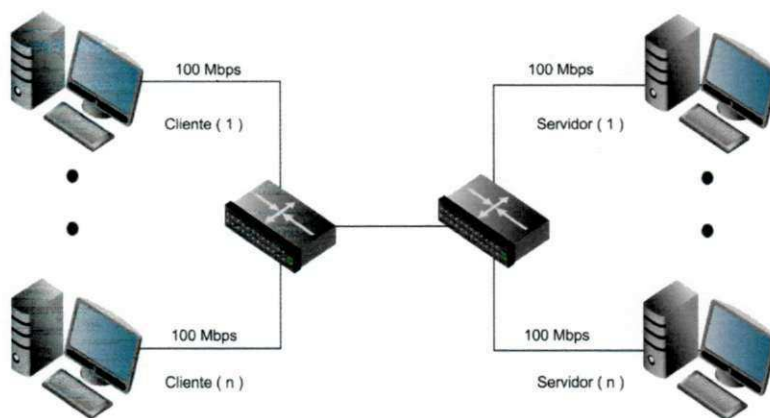


Figura 5.1: Ilustração da topologia empregada na simulação.

## 5.3 Experimentos

Para avaliar o mecanismo proposto foi realizado um teste que consistiu na simulação da interconexão entre os 10 nós clientes e os 10 nós servidores, cada conexão teve o tamanho médio de 384 KB e seguiu o procedimento convencional para o seu estabelecimento, em outras palavras, o cliente estabelece uma conexão com o servidor estes transferem os dados e após essa transferência a conexão é encerrada deixando tanto o cliente quanto o servidor inativos. O tamanho de 384 KB para cada conexão foi escolhido, pois esse valor é apresentado em [32] como sendo o tamanho médio das conexões *Web* atuais.

Para que houvesse informações salvas e compartilhadas por conexões anteriores, o início das conexões foi feito aleatoriamente pelo simulador, permitindo assim que algumas das conexões aproveitassem informações de outras.

O tempo máximo de cada simulação foi de 100 segundos. Utilizou-se esse tempo uma vez que o que estava sendo avaliado é a fase partida lenta do TCP, fase esta cuja duração é de apenas alguns poucos RTT.

O enlace entre as máquinas na rede local e o roteador tem velocidade de 100 Mbps e atraso de propagação de 10 ms. O enlace entre os roteadores teve sua velocidade e atraso variados conforme os valores apresentados na Tabela 5.1. Considerou-se desprezível o tempo de processamento por parte dos roteadores.

Durante cada uma das simulações, um nó iniciava uma transmissão que tinha duração de 100 segundos e tinha o objetivo de gerar tráfego de *background* na rede, os demais nós iniciavam seus fluxos de curta duração utilizando o mecanismo proposto ou a abordagem padrão normalmente.

Na Tabela 5.1 são exibidas as variáveis independentes, suas descrições e os valores que podem ser assumidos, sendo utilizadas como entradas para a montagem dos cenários da simulação. Na Tabela 5.2 são apresentadas as variáveis dependentes, suas descrições e unidades nas quais foram medidas.

Tabela 5.1: Tabela contendo as variáveis independentes

Variáveis	Descrição	Valores
Velocidade do enlace	Taxa de transmissão máxima que pode ser atingida pelo enlace	100 e 1000 <i>Mbps</i>
Atraso	Tempo necessário para que o pacote percorra de uma extremidade a outra do enlace	10 e 100 <i>ms</i>

Tabela 5.2: Tabela contendo as variáveis dependentes

Variáveis	Descrição	Unidade
Duração do fluxo	Tempo decorrido entre o início da conexão e sua conclusão	Segundos

Os testes foram repetidos 30 vezes para minimizar os erros de medição e permitir a comparação entre as abordagens utilizando um intervalo de confiança de 95%. Esse número de repetições foi utilizado pois ao observar os resultados das primeiras simulações constatou-se que o desvio padrão dos dados era pequeno o que permitiu o uso desse tamanho de amostra.

## 5.4 Análise dos resultados

Com o cruzamento dos valores que as variáveis independentes podem assumir, foram montados quatro cenários onde foram executados tanto o TCP Cubic padrão quanto a solução proposta. Os experimentos foram repetidos 30 vezes em cada um dos quatro cenários conforme descrito na Seção 5.3.

Observou-se que em alguns casos o tempo de conclusão dos fluxos tanto no TCP-Share quanto no TCP Cubic padrão foi o mesmo. Isso é devido ao fato de os cenários nos quais os protocolos foram testados serem semelhantes e também ao não compartilhamento de informações pelos sistemas finais, evento esse que normalmente ocorreu na primeira conexão de cada experimento.

### 5.4.1 Cenário 1

Nesse cenário a velocidade entre os roteadores foi de 100 *Mbps* e o atraso de 10 *ms*. Os tempos de conclusão medidos para os fluxos utilizando o TCP Cubic padrão foram 0,063, 0,95 e 0,162 segundos (mínimo, médio e máximo respectivamente), ao passo que para os fluxos que utilizaram a solução proposta e compartilharam informações os tempos foram de 0,042, 0,065 e 0,162 segundos (mínimo, médio e máximo respectivamente).

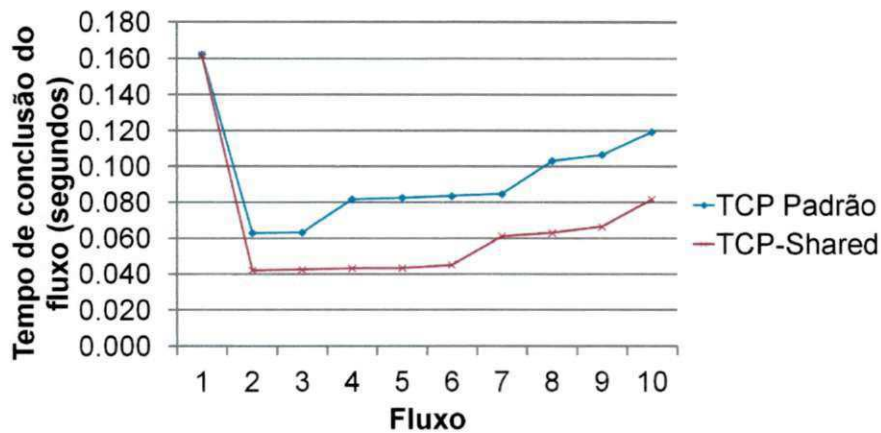


Figura 5.2: Gráfico do comportamento do fluxo para o cenário 1.

### 5.4.2 Cenário 2

Nesse cenário a velocidade entre os roteadores foi de 1000 *Mbps* e o atraso de 10 *ms*. Os tempos medidos para os fluxos utilizando o TCP Cubic padrão foram mínimo 0,060, médio 0,90 e máximo 0,160 segundos, ao passo que para os fluxos que utilizaram a solução proposta e compartilharam informações os tempos foram de 0,040, 0,060 e 0,160 segundos (mínimo, médio e máximo respectivamente).



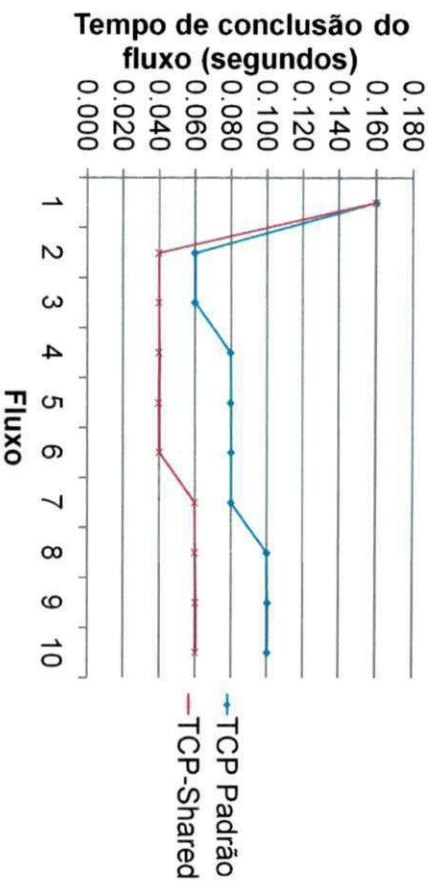


Figura 5.3: Gráfico do comportamento do fluxo para o cenário 2.

### 5.4.3 Cenário 3

Nesse cenário a velocidade entre os roteadores foi de 100 *Mbps* e o atraso de 100 *ms*. Os tempos medidos para os fluxos utilizando o TCP Cubic padrão foram mínimo 0,603, médio 0,903 e máximo 1,603 segundos, ao passo que para os fluxos que utilizaram a solução proposta e compartilharam informações os tempos foram de 0,402, 0,603 e 1,602 segundos (mínimo, médio e máximo respectivamente).

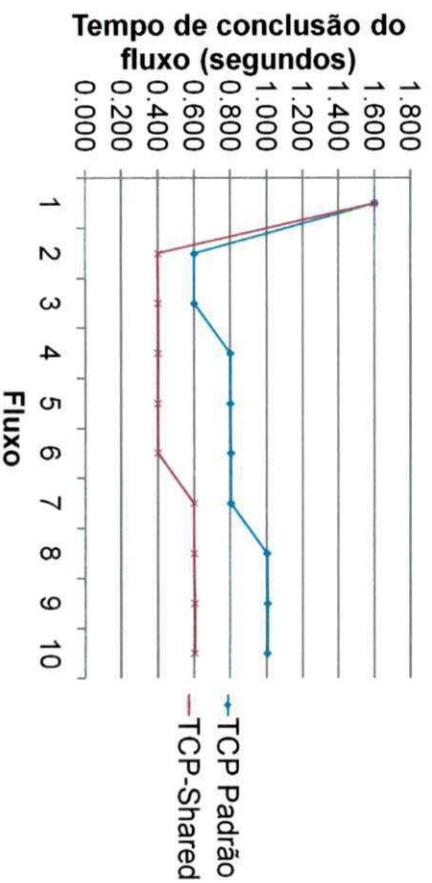


Figura 5.4: Gráfico do comportamento do fluxo para o cenário 3.



#### 5.4.4 Cenário 4

Nesse cenário a velocidade entre os roteadores foi de 1000 *Mbps* e o atraso de 100 *ms*. Os tempos medidos para os fluxos utilizando o TCP Cubic padrão foram mínimo 0,600, médio 0,900 e máximo 1,600 segundos, ao passo que para os fluxos que utilizaram a solução proposta e compartilharam informações os tempos foram de 0,400, 0,600 e 1,600 segundos (mínimo, médio e máximo respectivamente).

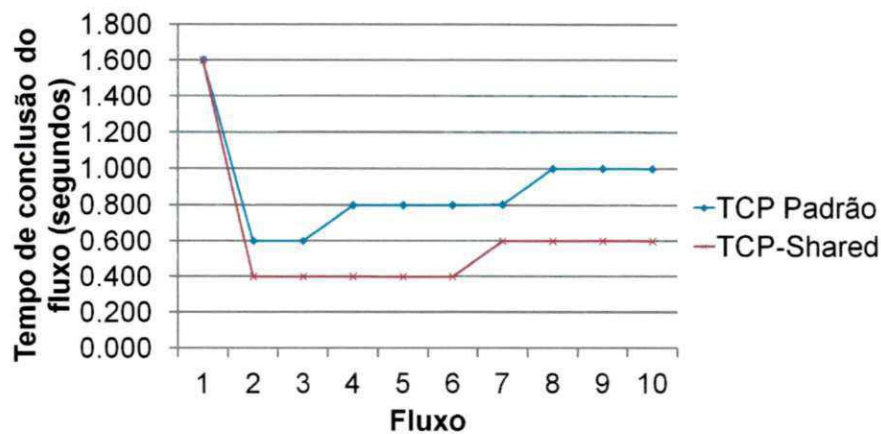


Figura 5.5: Gráfico do comportamento do fluxo para o cenário 4.

Na Figura 5.6, é perceptível que não houve interseção entre os intervalos de confiança dos tempos medidos. Isso permite dizer que os sistemas apresentam diferenças e, com os valores colhidos nos experimentos, a solução proposta apresenta uma redução média no tempo de conclusão dos fluxos de até 30%.

#### 5.4.5 Aumento da Vazão

Outra informação obtido através da análise dos dados oriundos da simulação é quanto ao aumento da vazão entre os roteadores. Na tabela 5.3 são apresentados dados referentes ao aumento da vazão a medida que o valor da janela de congestionamento compartilhada é aumentada. Na coluna 1 são apresentados os dados referentes ao TCP Cubic em sua implementação padrão, a partir da coluna 2 são apresentados os dados referentes a vazão dos fluxos que utilizam a abordagem proposta. Na coluna 2 nenhum valor foi compartilhado, na coluna 3 o valor compartilhado foi de 10 segmentos, na coluna 4 de 50 segmentos e na

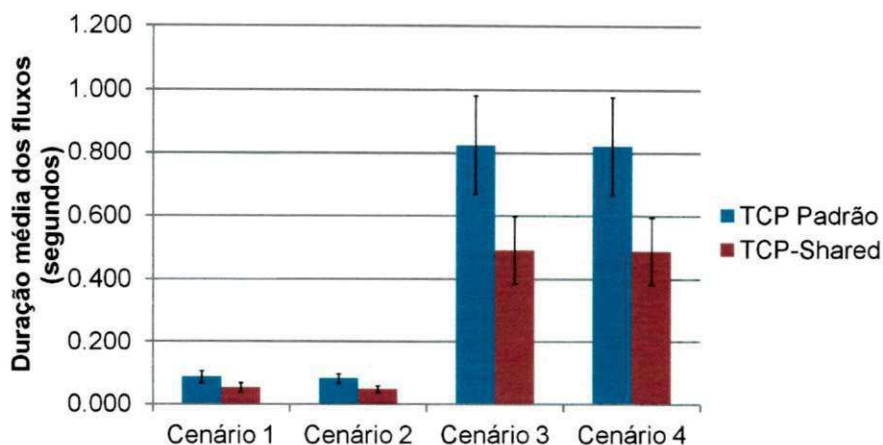


Figura 5.6: Gráfico comparativo entre os tempos de conclusão para os 4 cenários.

coluna 5 de 150 segmentos.

Tabela 5.3: Comparativo entre o aumento da vazão e aumento da janela compartilhada

Tempo em segundos	Vazão entre roteadores em <i>Mbps</i>				
	1	2	3	4	5
1	0,046	0,046	0,092	0,092	0,092
2	2,387	2,387	5,634	7,400	7,446
3	18,643	18,643	26,928	32,477	34,951
4	40,171	40,171	45,152	39,039	44,995
5	50,341	50,341	52,377	48,920	46,044
6	51,840	51,840	52,937	52,817	47,660
7	46,477	46,477	46,014	42,227	42,864
8	54,649	54,649	54,619	50,482	56,259
9	41,149	41,149	42,836	39,047	44,846
De 1 a 9	305,703	305,703	326,589	312,500	325,155
De 1 a 4	61,247	61,247	77,806	79,007	87,483

### 5.4.6 Conclusão dos resultados

Como pode ser visto nas Figuras 5.2, 5.3, 5.4 e 5.5, todas as conexões, exceto a primeira, apresentaram uma redução na duração do fluxo em relação ao TCP Cubic padrão. Esse comportamento pode ser atribuído ao fato de não haver qualquer informação armazenada sobre as condições recentes da rede na primeira conexão de cada execução, fazendo com que o processo de avaliação do enlace seja executado utilizando os valores padrões do protocolo TCP.

Observou-se que a redução no tempo de conclusão dos fluxos foi de até 30% e também que não ocorreram eventos nos quais o tempo de conclusão dos fluxos utilizando o TCP-Share fosse maior que os que utilizaram o TCP Cubic padrão.

Não foi objetivo deste trabalho testar o comportamento de fluxos de longa duração utilizando o TCP-Share, uma vez que esse tipo de fluxo consegue utilizar a maior parte da largura de banda disponível. Isto pode ser considerado como uma limitação desta avaliação.

## Capítulo 6

### Conclusão e trabalhos futuros

A *Internet* tem experimentado evoluções tanto em suas tecnologias de conexão, que disponibilizaram enlaces com velocidades que atingiram vários *gbps* e também permitiram que o acesso ocorra quase em qualquer lugar, quanto nas formas de utilização com o surgimento de novos conteúdos. Associando-se isto ao fato de que cerca de 90% dos fluxos de dados que nela trafegam serem gerenciados pelo protocolo TCP, as pesquisas para o desenvolvimento de novas variantes do controle de congestionamento do protocolo TCP têm aumentado.

Apesar da ampla diversidade de variantes deste protocolo disponíveis atualmente, várias delas não possuem mecanismos que tirem proveito das condições da rede avaliada por fluxos recentemente finalizados, o que pode reduzir o tempo de conclusão dos fluxos e melhorar a utilização dos enlaces de dados disponíveis.

Neste trabalho foi apresentado um algoritmo que proporciona um melhor aproveitamento da largura de banda disponível. Para alcançar esse objetivo, o controle de congestionamento foi alterado de maneira que as informações obtidas por conexões recentemente finalizadas sejam salvas localmente e compartilhadas com as máquinas localizadas na mesma rede local.

Como forma de avaliar o mecanismo proposto neste trabalho foram realizadas simulações, conforme pode ser visto no Capítulo 5, onde se constatou que a reutilização da janela de congestionamento em novas conexões permitiu uma redução no tempo de conclusão dos fluxos de até 30%. Como contribuições do mecanismo proposto neste trabalho pode-se citar:

- a reutilização de um dos parâmetros de uma conexão, no caso a janela de congestionamento, para fazer com que a fase de partida lenta seja iniciada com um valor mais elevado que o padrão;

- aumento do valor da janela inicial de congestionamento das conexões;
- redução do tempo de conclusão dos fluxos.

Como outras pesquisas a serem desenvolvidas, com o objetivo de aperfeiçoar o mecanismo proposto neste trabalho, podem ser citadas:

- um estudo mais aprofundado sobre abordagens mais eficientes para a redução da janela de congestionamento compartilhada, seja no caso do início de uma nova conexão ou na ocorrência de *timeout* e *reset* da conexão;
- aprimoramento do protocolo de comunicação para troca de informações, permitindo com isso a utilização de outras informações da conexão;
- definição de políticas para validar a informação compartilhada;
- utilização de outros parâmetros já avaliados, como por exemplo RTT, para a definição da janela de congestionamento inicial.

# Bibliografia

- [1] A. Akella, S. Seshan, and H. Balakrishnan. The impact of false sharing on shared congestion management. In *Network Protocols, 2003. Proceedings. 11th IEEE International Conference on*, pages 84–94. IEEE, 2002.
- [2] M. Allman, Sally Floyd, and C. Partridge. Increasing TCP’s Initial Window. RFC 3390 (Proposed Standard), 10 2002.
- [3] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581 (Proposed Standard), 04 1999. Obsoleted by RFC 5681, updated by RFC 3390.
- [4] Paul Barford and Mark Crovella. Measuring web performance in the wide area. *SIGMETRICS Perform. Eval. Rev.*, 27(2):37–48, September 1999.
- [5] Naimul Basher, Aniket Mahanti, Anirban Mahanti, Carey Williamson, and Martin Arlitt. A comparative analysis of web and peer-to-peer traffic. In *Proceeding of the 17th international conference on World Wide Web, WWW ’08*, pages 287–296, New York, NY, USA, 2008. ACM.
- [6] Mike Belshe. Spdy is an experiment with protocols for the web, 11 2011.
- [7] Y. Bhumralkar, J. Lung, and P. Varaiya. Network adaptive tcp slow start. *Infocom 2003*, 2:1221–1230, 2000.
- [8] R. Braden. Requirements for Internet Hosts - Communication Layers. RFC 1122 (Standard), 10 1989. Updated by RFCs 1349, 4379, 5884, 6093, 6298.
- [9] L.S. Brakmo, S.W. O’Malley, and L.L. Peterson. TCP Vegas: new techniques for congestion detection and avoidance. *ACM SIGCOMM Computer Communication Review*, 24(4):24–35, 1994.

- aumento do valor da janela inicial de congestionamento das conexões;
- redução do tempo de conclusão dos fluxos.

Como outras pesquisas a serem desenvolvidas, com o objetivo de aperfeiçoar o mecanismo proposto neste trabalho, podem ser citadas:

- um estudo mais aprofundado sobre abordagens mais eficientes para a redução da janela de congestionamento compartilhada, seja no caso do início de uma nova conexão ou na ocorrência de *timeout* e *reset* da conexão;
- aprimoramento do protocolo de comunicação para troca de informações, permitindo com isso a utilização de outras informações da conexão;
- definição de políticas para validar a informação compartilhada;
- utilização de outros parâmetros já avaliados, como por exemplo RTT, para a definição da janela de congestionamento inicial.

## Bibliografia

- [1] A. Akella, S. Seshan, and H. Balakrishnan. The impact of false sharing on shared congestion management. In *Network Protocols, 2003. Proceedings. 11th IEEE International Conference on*, pages 84–94. IEEE, 2002.
- [2] M. Allman, Sally Floyd, and C. Partridge. Increasing TCP's Initial Window. RFC 3390 (Proposed Standard), 10 2002.
- [3] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581 (Proposed Standard), 04 1999. Obsoleted by RFC 5681, updated by RFC 3390.
- [4] Paul Barford and Mark Crovella. Measuring web performance in the wide area. *SIGMETRICS Perform. Eval. Rev.*, 27(2):37–48, September 1999.
- [5] Naimul Basher, Aniket Mahanti, Anirban Mahanti, Carey Williamson, and Martin Arlitt. A comparative analysis of web and peer-to-peer traffic. In *Proceeding of the 17th international conference on World Wide Web, WWW '08*, pages 287–296, New York, NY, USA, 2008. ACM.
- [6] Mike Belshe. Spdy is an experiment with protocols for the web, 11 2011.
- [7] Y. Bhumralkar, J. Lung, and P. Varaiya. Network adaptive tcp slow start. *Infocom 2003*, 2:1221–1230, 2000.
- [8] R. Braden. Requirements for Internet Hosts - Communication Layers. RFC 1122 (Standard), 10 1989. Updated by RFCs 1349, 4379, 5884, 6093, 6298.
- [9] L.S. Brakmo, S.W. O'Malley, and L.L. Peterson. TCP Vegas: new techniques for congestion detection and avoidance. *ACM SIGCOMM Computer Communication Review*, 24(4):24–35, 1994.



- [10] Dirceu Cavendish, Kazumi Kumazoe, Masato Tsuru, Yuji Oie, and Mario Gerla. Caps-tart: An adaptive tcp slow start for high speed networks. In *Proceedings of the 2009 First International Conference on Evolving Internet*, pages 15–20, Washington, DC, USA, 2009. IEEE Computer Society.
- [11] Olympio Cipriano da Silva Filho. Gerenciamento de confiança em ambientes pervasivos. Dissertação de Mestrado em Engenharia Elétrica, Universidade Federal de Campina Grande, 2007.
- [12] Leandro Melo de Sales. Avaliação experimental do protocolo dccp para transmissão de conteúdos multimídia em redes sem fio 802.11g e na internet. Dissertação de Mestrado em Ciência da Computação, Universidade Federal de Campina Grande, 2008.
- [13] Nandita Dukkupati, Tiziana Refice, Yuchung Cheng, Jerry Chu, Tom Herbert, Amit Agarwal, Arvind Jain, and Natalia Sutin. An argument for increasing tcp's initial congestion window. *ACM SIGCOMM Computer Communications Review*, 40:27–33, 06 2010.
- [14] Sally Floyd. HighSpeed TCP for Large Congestion Windows. RFC 3649 (Experimental), December 2003.
- [15] Sally Floyd. Limited Slow-Start for TCP with Large Congestion Windows. RFC 3742 (Experimental), 2 2004.
- [16] Chuck Fraleigh, Sue Moon, Bryan Lyles, Chase Cotton, Mujahid Khan, Deb Moll, Rob Rockell, Ted Seely, and Christophe Diot. Packet-level traffic measurements from the sprint ip backbone. *IEEE Network*, 17(6):6–16, 2003.
- [17] M. Gerla, MY Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Mascolo. TCP Westwood: Congestion window control using bandwidth estimation. In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 3, pages 1698–1702. IEEE, 2002.
- [18] G.E. Gonçalves, S. Fernandes, D. Mariz, and D. Sadok. Burst tcp: an approach for benefiting mice flows. Dissertação de Mestrado em Ciência da Computação, Universidade Federal de Pernambuco, 2007.

- [19] L.A. Grieco and S. Mascolo. Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control. *ACM SIGCOMM Computer Communication Review*, 34(2):25–38, 2004.
- [20] V. Jacobson. Congestion avoidance and control. In *ACM SIGCOMM Computer Communication Review*, volume 18, pages 314–329. ACM, 1988.
- [21] V. Jacobson, R. Braden, and D. Borman. TCP Extensions for High Performance. RFC 1323 (Proposed Standard), 05 1992.
- [22] V. Kavidha and V. Sadasivam. Exploring Round Trip Time fairness for Adaptive Layered Transmission Control Protocol. *International Journal of Advanced Networking and Applications*, 01:353–358, 06 2010.
- [23] Tom Kelly. Scalable tcp: improving performance in highspeed wide area networks. *SIGCOMM Comput. Commun. Rev.*, 33(2):83–91, April 2003.
- [24] James F. Kurose and Keith W. Ross. *Redes de Computadores e a Internet: Uma abordagem top-down*. Addison Wesley, São Paulo, trad. 3 ed. edition, 2006.
- [25] Khaled Harfoush Lisong Xu and Injong Rhee. Binary increase congestion control (bic) for fast long-distance networks. 4:2514–2524, 2004.
- [26] M. Mathis, J. Mahdavi, Sally Floyd, and A. Romanow. TCP Selective Acknowledgment Options. RFC 2018 (Proposed Standard), 10 1996.
- [27] Tatsuya Mori, Masato Uchida, Ryoichi Kawahara, Jianping Pan, and Shigeki Goto. Identifying elephant flows through periodically sampled packets. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, IMC '04*, pages 115–120, New York, NY, USA, 2004. ACM.
- [28] K. Munir, M. Welzl, and D. Damjanovic. Linux beats windows!—or the worrying evolution of TCP in common operating systems. In *PFLDnet Workshop*, pages 43–48. Citeseer, 2007.

- [29] Venkata N. Padmanabhan and Randy H. Katz. TCP Fast Start: A Technique for Speeding up Web Transfers. In *IEEE GLOBECOM Internet Mini-Conference, Sydney, Australia*, pages 41–46, November 1998.
- [30] J. Postel. User Datagram Protocol. RFC 768 (Standard), August 1980.
- [31] J. Postel. Transmission Control Protocol. RFC 793 (Standard), 09 1981. Updated by RFCs 1122, 3168, 6093.
- [32] S. Ramachandran and A. Jain. Web page stats: size and number of resources, 2010.
- [33] I. Rhee and L. Xu. CUBIC: A new TCP-friendly high-speed TCP variant. In *Proc. PFLDnet*, volume 2005. Citeseer, 2005.
- [34] Michael Savoric, Dr.-Ing. Adam Wolisz, and Tu Berlin. Identifying and evaluating the potential of reusing network information from different flows. Technical report, Technical University Berlin, 2001.
- [35] S. Seshan, M. Stemm, and R.H. Katz. Spand: Shared passive network performance discovery. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems*, pages 13–13. Usenix Association, 1997.
- [36] K. Tan, J. Song, Q. Zhang, and M. Sridharan. A compound tcp approach for high-speed and long distance networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, 04 2006.
- [37] Andrew S. Tanenbaum. *Redes de Computadores*. Campus, quarta edition, 2003.
- [38] Akamai Technologies. The state of the internet. 2° quadrimestre. Technical Report 2, Akamai Technologies, Inc., 2010.
- [39] Joe Touch. Automating the initial window in tcp, 7 2012.
- [40] Ren Wang, G. Pau, K. Yamada, M.Y. Sanadidi, and M. Gerla. Tcp startup performance in large bandwidth networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 796–805, 03 2004.

- 
- [41] Carey Williamson and Qian Wu. A case for context-aware TCP/IP. *SIGMETRICS Perform. Eval. Rev.*, 29:11–23, 03 2002.
- [42] Yuan-Rung Yang, Chia-Tai Chan, Yaw-Chung Chen, and Yi-Cheng Chan. Smart window adjustment for xcp's congestion management. In *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, volume 1, pages 4–479, 02 2006.