



**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CURITIBA**

GERÊNCIA DE PESQUISA E PÓS-GRADUAÇÃO

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL - CPGEI**

RODRIGO ANDREOLA

**ANÁLISE DE UM MODELO TCP MAX-PLUS
APLICADO A UMA REDE SEM FIO**

DISSERTAÇÃO DE MESTRADO

**CURITIBA
ABRIL - 2007**

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

DISSERTAÇÃO
apresentada à UTFPR
para obtenção do grau de

MESTRE EM CIÊNCIAS

por

RODRIGO ANDREOLA

**ANÁLISE DE UM MODELO TCP MAX-PLUS APLICADO A
UMA REDE SEM FIO**

Banca Examinadora:

Presidente e Orientador:

PROF. DR. RICARDO LÜDERS

CPGEI/UTFPR

Examinadores:

PROF^a. DR^a. CRISTINA DUARTE MURTA

INF/UFPR

PROF^a. DR^a. ANELISE MUNARETTO FONSECA

CPGEI/UTFPR

PROF^a. DR^a. KEIKO VERÔNICA ONO FONSECA

CPGEI/UTFPR

Curitiba, abril de 2007.

RODRIGO ANDREOLA

ANÁLISE DE UM MODELO TCP MAX-PLUS APLICADO A UMA REDE SEM FIO

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do grau de “Mestre em Ciências” – Área de Concentração: Telemática.

Orientador: Prof. Dr. Ricardo Lüders

Curitiba

2007

Ficha catalográfica elaborada pela Biblioteca da UTFPR – Campus Curitiba

A555a Andreola, Rodrigo

“Análise de um modelo TCP Max-Plus aplicado a uma rede sem fio” /
Rodrigo Andreola. Curitiba. UTFPR, 2007
XI, 74 f. : il. : 30 cm

Orientador: Prof. Dr. Ricardo Lüders
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná.
Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Cu-
ritiba, 2007
Bibliografia f. 69-70

1. Redes de longa distância. 2. Redes sem fio. 3. Teleinformática. I. Lüders,
Ricardo, orient. II. Universidade Tecnológica Federal do Paraná. Curso de
Pós-Graduação em Engenharia Elétrica e Informática Industrial. III. Título.

CDD: 004.67

AGRADECIMENTOS

Agradeço especialmente à minha esposa e sempre companheira, Lu, por compreender meus muitos momentos de ausência e principalmente pelos atos e palavras de incentivo durante toda a elaboração deste trabalho.

A meus pais, que não somente me ensinaram a importância do estudo, deram-me todo o suporte necessário durante cada passo da vida acadêmica.

Agradeço ao professor e orientador Ricardo Lüders pela dedicação e por ter me motivado nos vários momentos de dúvida. Este trabalho não teria sido concluído sem seu apoio.

Gostaria ainda de agradecer a todos os professores do mestrado pelos novos conhecimentos oferecidos. Ao amigo Alcides Troller e ao professor Roberto Hexsel pela indicação ao programa. Finalmente, agradeço a todos os amigos que de alguma forma me incentivaram neste trabalho.

SUMÁRIO

LISTA DE FIGURAS	vii
LISTA DE TABELAS	viii
LISTA DE ABREVIATURAS E SIGLAS	ix
RESUMO	x
ABSTRACT	xi
1 INTRODUÇÃO	1
1.1 MOTIVAÇÃO	1
1.2 OBJETIVO	2
1.3 ORGANIZAÇÃO DO TRABALHO	3
2 MODELO MAX-PLUS PARA TRÁFEGO TCP	5
2.1 INTRODUÇÃO	5
2.2 PROTOCOLO TCP	5
2.3 ÁLGEBRA MAX-PLUS	8
2.4 MODELO TCP MAX-PLUS	10
2.4.1 Topologia da rede	11
2.4.2 Modelo max-plus	12
2.5 EXEMPLO NUMÉRICO	18
2.6 FERRAMENTAS COMPUTACIONAIS	22
2.7 DETERMINAÇÃO DA VELOCIDADE DA REDE USANDO MAX-PLUS	27
2.8 COMPARAÇÃO DE MODELOS: MAX-PLUS x SIMULAÇÃO	29
2.9 CONCLUSÃO	31
3 ASPECTOS DE DESEMPENHO EM REDES SEM FIO	33
3.1 INTRODUÇÃO	33
3.2 FATORES QUE AFETAM O DESEMPENHO DE REDES SEM FIO	33
3.2.1 Controle de acesso ao meio	33
3.2.2 <i>Overheads</i> impostos pelo MAC/DCF e pelo protocolo TCP	37
3.2.3 Comunicação <i>multi-hop</i>	38
3.3 LIMITAÇÕES DO TCP EM REDES SEM FIO	41
3.4 CONCLUSÃO	43
4 ANÁLISE DO MODELO TCP MAX-PLUS PARA UMA REDE SEM FIO <i>AD HOC</i>	45
4.1 INTRODUÇÃO	45

4.2 METODOLOGIA	45
4.3 CENÁRIO DE ESTUDO I	46
4.3.1 Modelo TCP max-plus.....	47
4.3.2 Resultados obtidos com o modelo max-plus	51
4.3.3 Simulação da rede usando <i>ns-2</i>	52
4.3.4 Resultados obtidos na simulação <i>ns-2</i>	55
4.3.5 Análise do cenário I.....	56
4.4 CENÁRIO DE ESTUDO II	57
4.4.1 Modelo TCP max-plus modificado	58
4.4.2 Resultados obtidos no modelo max-plus modificado.....	59
4.4.3 Análise do cenário II.....	60
4.5 CENÁRIO DE ESTUDO III.....	61
4.5.1 Simulação da rede modificada.....	61
4.5.2 Resultados obtidos na simulação <i>ns-2</i> modificada	64
4.5.3 Análise do cenário III	64
4.6 CONCLUSÃO	65
5 DISCUSSÃO E CONCLUSÕES	67
5.1 CONCLUSÕES	67
5.2 TRABALHOS FUTUROS	68
REFERÊNCIAS BIBLIOGRÁFICAS	69
ANEXO 1 - CÓDIGOS-FONTE DAS FUNÇÕES <i>MATLAB</i>.....	71

LISTA DE FIGURAS

Figura 1: Modelo de rede.....	12
Figura 2: Exemplo de programa formulado a partir das Equações 1 e 2.....	25
Figura 3: Exemplo de programa formulado a partir da Equação 5.....	26
Figura 4: Seqüência de eventos para transmissão usando DCF.....	36
Figura 5: Exemplo de rede sem fio <i>ad hoc</i> com topologia em linha.....	39
Figura 6: Código <i>Matlab</i> para o cenário I.....	50
Figura 7: Topologia da rede para a simulação (cenário I).	52
Figura 8: Código gerador do cenário I no simulador <i>ns-2</i>	54
Figura 9: Topologia da rede para a simulação (cenário III).....	62
Figura 10: Código gerador do cenário III no simulador <i>ns-2</i>	63

LISTA DE TABELAS

Tabela 1: Dimensões das matrizes nas Equações 5 e 6.....	16
Tabela 2: Parâmetros da rede utilizada no exemplo.....	18
Tabela 3: Exemplo de tempos de saída	19
Tabela 4: Funções desenvolvidas no <i>Matlab</i> para modelagem TCP max-plus	23
Tabela 5: Exemplo de taxa média de envio	28
Tabela 6: Função desenvolvida no <i>Matlab</i> para <i>Rate Based Loss Detection</i>	29
Tabela 7: Parâmetros da rede simulada.....	30
Tabela 8: Taxa de transferência esperada para o TCP em redes sem fio	42
Tabela 9: Parâmetros utilizados no modelo max-plus (cenário I).....	48
Tabela 10: Tempo de chegada de ACKs para o modelo max-plus (cenário I).	51
Tabela 11: Parâmetros utilizados na simulação (cenário I)	53
Tabela 12: Tempo de chegada de ACKs com a simulação (cenário I).	56
Tabela 13: Comparação entre as taxas de transmissão (cenário I).....	57
Tabela 14: Tempo de chegada de ACKs com o modelo max-plus modificado (cenário II)...	59
Tabela 15: Comparação entre as taxas de transmissão (cenário II).	60
Tabela 16: Tempo de chegada de ACKs com a simulação modificada (cenário III).	64
Tabela 17: Comparação entre as taxas de transmissão (cenário III).....	65

LISTA DE ABREVIATURAS E SIGLAS

ACK	- <i>Acknowledgement</i>
CSMA/CA	- <i>Carrier Sensing Multiple Access / Collision Avoidance</i>
CTS	- <i>Clear To Send</i>
CW	- <i>Contention Window</i>
DCF	- <i>Distributed Coordination Function</i>
DIFS	- <i>Data Inter-Frame Space</i>
DSDV	- <i>Destination-Sequenced Distance-Vector</i>
FIFO	- <i>First-In, First-Out</i>
FTP	- <i>File Transfer Protocol</i>
HTTP	- <i>Hyper-Text Transfer Protocol</i>
IFS	- <i>Inter-Frame Space</i>
IP	- <i>Internet Protocol</i>
KBps	- <i>Kilobytes por segundo</i>
MAC	- <i>Media Access Control</i>
MANET	- <i>Mobile Ad hoc Network</i>
Mbps	- <i>Megabits por segundo</i>
ms	- <i>Milisegundo</i>
NAM	- <i>Network Animator</i>
ns-2	- <i>Network Simulator, version 2</i>
OSI	- <i>Open Systems Interconnection</i>
PCF	- <i>Point Coordination Function</i>
pct	- <i>Pacote</i>
POP3	- <i>Post Office Protocol 3</i>
pps	- <i>Pacotes por segundo</i>
QoS	- <i>Quality of Service</i>
RTS	- <i>Request To Send</i>
SIFS	- <i>Short Inter-Frame Space</i>
SMTP	- <i>Simple Mail Transfer Protocol</i>
TCP	- <i>Transmission Control Protocol</i>
UDP	- <i>User Datagram Protocol</i>

RESUMO

Este trabalho propõe a aplicação de um modelo para o protocolo de transporte TCP, utilizando a álgebra max-plus, em redes sem fio. A abordagem determinística e até mesmo formal do modelo TCP max-plus é pouco tradicional, sendo que na literatura é mais comum encontrar métodos estocásticos para modelagem de redes. Neste estudo, são apresentados um breve resumo do protocolo TCP e da álgebra max-plus, seguido de uma descrição do modelo TCP max-plus. Um resumo dos fatores que afetam o desempenho de redes sem fio e do protocolo TCP sobre este tipo de rede é também apresentado. Uma metodologia para verificar a aplicabilidade do modelo em redes sem fio é proposta, comparando taxas de transmissão obtidas a partir do modelo max-plus e por simulação. O modelo TCP max-plus utilizado mostrou-se capaz de reproduzir o desempenho de redes sem fio, quando seus parâmetros são corretamente definidos de forma a inserir no modelo os efeitos da rede sem fio sobre a evolução da janela do TCP e na retenção dos pacotes durante a disputa pelo uso do meio de transmissão. A alteração nos parâmetros do modelo TCP max-plus não altera sua formulação original, mantendo a linearidade do modelo. Ainda neste estudo, um conjunto de ferramentas computacionais que implementam o modelo TCP max-plus foi construído.

ABSTRACT

This study analyses the applicability of a max-plus algebra based model, for the TCP transport protocol, over wireless networks. This formal deterministic nature of the TCP max-plus model is unusual compared to many works using stochastic methods for network analysis found in the literature. This document presents a brief introduction to the TCP protocol and to the max-plus algebra, followed by a description of the TCP max-plus model. Some issues that affect wireless networks and TCP protocol performance are shown. A method to test the applicability of the model over wireless networks is presented, comparing transmission rates calculated using the max-plus model and those obtained by simulation. The TCP max-plus model was able to describe the performance of wireless networks when its parameters are adjusted to include effects of the wireless network into the model. Changing the model parameters does not change its original form, keeping its linearity. Also, a set of computational tools for the max-plus model implementation has been built.

CAPÍTULO 1

INTRODUÇÃO

1.1 MOTIVAÇÃO

Em várias áreas do conhecimento, é comum o uso de modelos para representar determinados comportamentos. Em eletrônica, por exemplo, são usados modelos para representar a propagação de ondas de rádio, os efeitos de um filtro aplicado em um sinal, dentre uma vasta gama de aplicações. Muitos modelos fazem uso de equações que especificam as variáveis e as regras que governam o comportamento modelado, permitindo prever com grande precisão, os efeitos e os resultados do comportamento do sistema sob diferentes condições.

Porém, no estudo das redes de comunicação, um grande esforço tem sido feito para obter modelos, usando as mais diferentes técnicas, que possam prever com precisão vários comportamentos observados. Por exemplo, modelos mais precisos têm sido necessários para aplicações que exigem algum tipo de qualidade de serviço. Neste sentido, a análise do comportamento de uma rede de comunicação costuma fazer uso de métodos estocásticos, adequados para obtenção de valores médios de desempenho, tempos de resposta, ou outra métrica analisada.

Até mesmo o desenvolvimento de uma tecnologia de rede ou protocolo para o qual se deseja obter um determinado comportamento costuma partir de um conhecimento empírico:

- O protocolo é construído com base em algumas premissas, e seu comportamento é avaliado por simulação ou por experimentação prática;
- Alterações são realizadas na tentativa de eliminar falhas ou comportamentos não desejados;
- Novamente o comportamento do protocolo é avaliado e eventualmente sofre novas alterações até que seu funcionamento esteja satisfatório.

Essa abordagem torna o comportamento do protocolo seja previsível dentro das condições para as quais ele foi avaliado por simulação ou experimentação. Em geral, não é possível garantir que o protocolo desenvolvido terá sempre o mesmo comportamento, nem definir formalmente quais são os limites para os quais o protocolo mantém as características desejadas.

Assim, esta dissertação pretende explorar um formalismo ainda pouco comum para analisar as redes de comunicações, com base em um modelo apresentado por Baccelli e Hong (2000), que mantém certa distância das tradicionais técnicas de modelagem de redes, utilizando a álgebra max-plus para obter um modelo linear para o protocolo de transporte TCP.

Finalmente, com a visível adoção em larga escala das redes sem fio, este tipo de rede representa um escolha importante como tecnologia de rede sobre a qual será aplicado o modelo TCP max-plus.

1.2 OBJETIVO

Partindo de uma modelagem previamente proposta por Baccelli e Hong (2000), que utiliza a álgebra max-plus para modelar o comportamento do protocolo TCP, este trabalho tem os seguintes objetivos:

- Construir o modelo TCP max-plus proposto na literatura em uma ferramenta para cálculo computacional, fazendo uso de funções pré-existentes, mas também propondo novas funções específicas para este fim;
- Reproduzir os resultados obtidos na comparação entre o modelo TCP max-plus e a simulação de rede apresentados por Baccelli e Hong (2000) para uma rede com fio, utilizando as ferramentas desenvolvidas;
- Analisar questões gerais que afetam o desempenho de redes sem fio;
- Aplicar o modelo TCP max-plus sobre uma rede sem fio, procurando identificar restrições e/ou a aplicabilidade do modelo sobre este tipo de rede.

1.3 ORGANIZAÇÃO DO TRABALHO

Esta dissertação está organizada em cinco capítulos. O capítulo 2 apresenta uma breve introdução sobre o protocolo TCP, alguns fundamentos teóricos sobre a álgebra max-plus e uma proposta de modelagem de redes TCP/IP utilizando esta álgebra. Ainda no capítulo 2 são apresentadas algumas ferramentas computacionais para uso na modelagem de redes usando a álgebra max-plus. O capítulo 3 descreve alguns aspectos sobre o desempenho de redes sem fio. No capítulo 4 é apresentada uma análise da aplicação do modelo max-plus em redes sem fio, identificando deficiências da modelagem e sugerindo alterações adequadas nos parâmetros do modelo. Finalmente, o capítulo 5 apresenta as conclusões do trabalho e as propostas de trabalhos futuros.

CAPÍTULO 2

MODELO MAX-PLUS PARA TRÁFEGO TCP

2.1 INTRODUÇÃO

Várias aplicações utilizadas na Internet de forma rotineira, entre elas o envio e recebimento de *e-mails* e a navegação por páginas da *world wide web*, dependem do transporte de dados confiável que é realizado pelo protocolo TCP (*Transmission Control Protocol*). Devido à importância do protocolo TCP na comunicação de informações entre computadores, diferentes formas de análise do comportamento adaptativo do controle de fluxo do TCP foram propostas (LAKSHMAN e MADHOW, 1997; MATHIS, SEMKE, MAHDAVI, *et al*, 1997; PADHYE, FIROIU e TOWSLEY, 1999).

Um dos modelos propostos para representar o comportamento do TCP faz uso da álgebra max-plus, resultando em um modelo que descreve o TCP como um sistema linear nessa álgebra (BACCELLI e HONG, 2000). O modelo proposto descreve a evolução dos pacotes em uma rede utilizando o protocolo TCP, permitindo calcular o desempenho desta rede com resultados comparáveis aos obtidos por simulação.

A mesma álgebra max-plus utilizada no modelo TCP é também aplicada com sucesso na modelagem de diferentes sistemas a eventos discretos (BACCELLI, COHEN, OLSDER *et al*, 1992). Quando aplicada na modelagem de redes, limitantes de desempenho podem ser obtidos, resolvendo algumas questões de qualidade de serviço (QoS), cuja teoria recebe o nome de *Network Calculus* (LE BOUDEC, THIRAN, 2001).

Neste capítulo será apresentada a álgebra max-plus e sua utilização na construção de um modelo para o TCP. Em seguida, um conjunto de ferramentas computacionais para utilização do modelo max-plus será apresentado e utilizado para obtenção de algumas medidas de desempenho da rede.

2.2 PROTOCOLO TCP

Os níveis físico e de enlace em uma rede de computadores costumam oferecer um serviço sem garantias de entrega, no qual erros de transmissão, congestionamentos ou falhas

de *hardware* comprometem a integridade dos dados (COMER, 1991). Redes com roteamento dinâmico podem entregar os pacotes no destino fora de ordem, com atrasos ou em duplicidade ou fragmentados de forma diferente da original.

Aplicações que precisam transmitir dados entre computadores demandam um serviço confiável, com garantia de entrega. Seria bastante improdutivo se cada tipo de aplicação tivesse que programar suas próprias técnicas para detectar e reparar erros, providenciar retransmissões e reorganizar os dados recebidos na ordem correta.

O protocolo TCP foi projetado para fornecer um fluxo de *bytes* confiável ponta a ponta, mesmo sobre uma rede não confiável (TANENBAUM, 1996). O TCP é capaz de prover o transporte de dados de uma forma transparente para as aplicações que estão em camadas superiores da rede.

Diferente do protocolo UDP (*User Datagram Protocol*), que não oferece um serviço com garantia de entrega, o protocolo TCP é orientado a conexão. Isto é, antes da transmissão de dados é necessário que as duas pontas envolvidas na comunicação estabeleçam uma conexão trocando três mensagens específicas, o que é chamado de *three-way handshake*. Este processo é necessário para garantir que ambas as pontas estejam preparadas para receber e enviar dados e também para que as pontas entrem em acordo em relação aos números iniciais de seqüência, que serão usados para garantir que os dados recebidos possam ser reorganizados na ordem correta.

Comer (1991) afirma que as principais características do protocolo TCP são:

- *Stream orientation* (orientação à fluxo): o fluxo de dados (*bytes*) entregue pelo TCP para a aplicação de destino é exatamente o mesmo que foi enviado pela aplicação de origem, mantendo a integridade e ordenação do fluxo de *bytes*;
- *Virtual circuit connection* (conexão com circuitos virtuais): o protocolo TCP é orientado à conexão;
- *Buffered transfer* (transferência com *buffer*): ao transmitir dados usando o TCP, uma aplicação pode enviar dados com qualquer tamanho a cada envio, inclusive com tamanho unitário. O TCP pode acumular dados para envio posterior, aumentando assim a eficiência no envio de dados, sempre procurando enviar pacotes com tamanho razoável. O TCP também pode fragmentar os dados em pacotes menores para se adequar ao tamanho máximo de pacotes que as camadas inferiores da rede são capazes de transmitir;

- *Unstructured stream* (fluxo não estruturado): o TCP não faz qualquer tipo de interpretação sobre os dados que são transmitidos, e não conhece detalhes sobre as estruturas dos dados, de forma que as aplicações que estão utilizando o TCP devem implementar mecanismos para estruturar os dados;
- *Full duplex connections* (conexões *full duplex*): uma conexão TCP entre duas pontas permite transmissão de dados nos dois sentidos. Qualquer das pontas pode operar como transmissor e receptor dentro da mesma conexão. O TCP procura, sempre que possível, enviar as confirmações de entrega de dados (*ACK*) no mesmo pacote que uma transmissão em sentido contrário, otimizando o tráfego na rede.

A confiabilidade do TCP é obtida com o uso de uma técnica conhecida como *positive acknowledgement with retransmission* (confirmação positiva com retransmissão): após receber um conjunto de dados, o receptor envia um pacote com a confirmação do recebimento dos dados, chamado de *ACK*. O transmissor controla se os pacotes enviados já foram confirmados (a confirmação ocorre quando do recebimento do *ACK*) e os retransmite após um tempo sem a confirmação de recebimento. Além de retransmitir pacotes que não foram confirmados, o transmissor também controla o número de pacotes que foram enviados à rede e que ainda não foram confirmados, evitando sobrecarregar a rede.

Para complementar a confiabilidade, os pacotes de dados são identificados com números de seqüência, permitindo que o TCP identifique situações onde os pacotes ou *ACKs* estejam duplicados.

Se por um lado o uso de confirmações positivas simples permite a confiabilidade do protocolo, seu uso pode representar uma significativa subutilização de recursos da rede, pois o envio dos próximos pacotes de dados é retido até que pacotes já enviados sejam confirmados. Esse problema fica ainda mais evidente em redes com latência (tempo de transmissão) elevada (COMER, 1991).

Para evitar este problema, o TCP implementa um conceito de confirmação positiva mais complexo, chamado de *sliding window* (janela deslizante). Este conceito obtém uma melhor ocupação dos recursos da rede, pois permite que o transmissor envie vários pacotes para a rede antes do recebimento da confirmação de entrega. Quanto maior é o tamanho da janela, maior é o volume de dados que pode ser transmitido antes mesmo da confirmação de recebimento dos pacotes de dados anteriores.

Se o tamanho da janela é igual a 1, então o protocolo se comporta como se utilizasse um sistema de confirmação positiva simples: cada dado enviado precisa ser confirmado antes do envio do próximo dado.

Um ajuste correto do tamanho da janela mantém a rede totalmente ocupada com a transmissão de dados, e por este motivo obtém uma taxa de transferência substancialmente maior que aquela obtida com um sistema de confirmação positiva simples (COMER, 1991).

Desta forma, é o sistema de *sliding window* que controla a oferta de pacotes para a rede, através do tamanho da janela utilizada.

O modelo max-plus para redes TCP, que será apresentado nas subseções a seguir, procura capturar justamente o comportamento do sistema *sliding window* do TCP, para modelar a evolução dos pacotes na rede.

2.3 ÁLGEBRA MAX-PLUS

A álgebra max-plus é uma estrutura algébrica formada por elementos e operadores diferentes dos encontrados na álgebra convencional. Enquanto a álgebra convencional pode ser definida no conjunto dos números reais com as operações de adição e multiplicação usuais, denotada por $(\mathbb{R}, +, \times)$, a álgebra max-plus pode ser definida no conjunto dos reais acrescido do elemento $-\infty$ (menos infinito), substituindo-se as operações de adição e multiplicação pelas operações de máximo (representada pelo símbolo \oplus) e soma usual (representada pelo símbolo \otimes), respectivamente, formando a estrutura $(\mathbb{R} \cup -\infty, \oplus, \otimes)$, denotada \mathbb{R}_{\max} .

Na álgebra max-plus, a operação \oplus (*max*) consiste em escolher o maior valor entre os operandos. Por exemplo, $2 \oplus 5 = 5$. A operação \otimes (*plus*) é efetuada da mesma forma que a soma na álgebra convencional. Por exemplo, $2 \otimes 2 = 4$.

O elemento neutro para a operação \oplus é representado pelo símbolo $\varepsilon = -\infty$. A identidade para a operação \otimes recebe o símbolo $e = 0$, conforme mostrado nas equações a seguir:

$$\varepsilon \oplus c = c, \forall c \in \mathbb{R}_{\max}$$

$$e \otimes c = c, \forall c \in \mathbb{R}_{\max}$$

As propriedades da álgebra max-plus permitem sua classificação como sendo um semianel idempotente (BACCELLI, COHEN, OLSDER *et al*, 1992), ou seja, satisfaz os axiomas:

1. A operação \oplus é

- Associativa:

$$\forall a, b, c \in \mathbb{R}_{\max}, (a \oplus b) \oplus c = a \oplus (b \oplus c);$$

- Comutativa:

$$\forall a, b \in \mathbb{R}_{\max}, a \oplus b = b \oplus a;$$

- Possui um elemento neutro ε :

$$\forall a \in \mathbb{R}_{\max}, a \oplus \varepsilon = \varepsilon \oplus a = a;$$

- Idempotente:

$$\forall a \in \mathbb{R}_{\max}, a \oplus a = a.$$

2. A operação \otimes é:

- Associativa:

$$\forall a, b, c \in \mathbb{R}_{\max}, (a \otimes b) \otimes c = a \otimes (b \otimes c);$$

- Distributiva em relação à operação \oplus :

$$\forall a, b, c \in \mathbb{R}_{\max}, a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$$

$$(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a);$$

- Possui um elemento identidade e :

$$\forall a \in \mathbf{R}_{\max}, e \otimes a = a \otimes e = a;$$

- Possui um elemento nulo ε :

$$\forall a \in \mathbf{R}_{\max}, a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon.$$

Além das propriedades anteriores, caso a operação \otimes seja também comutativa, ou seja, $\forall a, b \in \mathbf{R}_{\max}, a \otimes b = b \otimes a$, tem-se um semianel comutativo idempotente.

Devido ao fato da operação \otimes possuir a propriedade distributiva em relação à operação \oplus , os conceitos da álgebra linear podem ser aplicados sobre esta álgebra, permitindo seu uso juntamente com a teoria de matrizes (BACCELLI e HONG, 2000).

Esta estrutura algébrica é plenamente adaptada para modelar fenômenos que envolvam atrasos e sincronização. Um atraso é um deslocamento temporal que pode ser capturado pela soma de intervalos de tempo, enquanto a sincronização requer a maximização de instantes de tempo. Neste aspecto, a álgebra max-plus tem sido utilizada com sucesso na modelagem de sistemas a eventos discretos, quando problemas de sincronização e competição por recursos compartilhados são comuns (BACCELLI, COHEN, OLSDER *et al*, 1992). Problemas deste tipo são comumente encontrados em sistemas de computação, redes de telecomunicações, sistemas de transporte e sistemas de manufatura (COHEN, GAUBERT e QUADRAT, 1999).

Uma aplicação da álgebra max-plus, proposta para modelagem da transmissão de dados utilizando o protocolo TCP em uma rede, é apresentada a seguir.

2.4 MODELO TCP MAX-PLUS

A modelagem apresentada nesta seção foi proposta por Baccelli e Hong (2000). Neste artigo, um modelo que utiliza a álgebra max-plus é proposto para representar o comportamento do protocolo TCP, demonstrando que um sistema de comunicação utilizando TCP pode ser modelado como um sistema max-plus linear.

2.4.1 Topologia da rede

O modelo de rede utilizado é constituído por K roteadores ligados em série, onde o primeiro roteador (origem) gera um fluxo de dados (chamado de *reference flow*) que é transmitido para o último roteador (destino), utilizando o protocolo TCP para transmissão dos dados. Embora o TCP permita conexões bidirecionais entre os nós envolvidos na comunicação, foi utilizada uma conexão unidirecional no modelo.

Os roteadores são identificados como $i = 0, i = 1, i = 2$ e assim por diante até $i = K$, sendo $i = 0$ o roteador de origem, e $i = K$ o roteador destino, como mostra a Figura 1.

Cada roteador é modelado como uma fila do tipo FIFO (*First-In, First-Out*) com somente um servidor, que pode receber em sua entrada não apenas pacotes do *reference flow*, mas também pacotes de outros fluxos chamados de *cross traffic*.

Ao chegar à fila de um roteador i , um pacote n levará um tempo para ser processado e retransmitido. O tempo gasto pelo pacote no roteador, desde sua chegada à fila até o seu processamento e conseqüente saída do roteador é chamado de tempo agregado de serviço e é representado por $\sigma_i(n)$.

O tempo agregado de serviço de um pacote do *reference flow* em um roteador inclui tanto o tempo de processamento de pacotes de *cross traffic*, que eventualmente entraram na fila desde a chegada do pacote anterior do *reference flow*, bem como o tempo de processamento do próprio pacote pelo roteador.

Os roteadores estão ligados entre si por circuitos que possuem latência conhecida. A latência para transmissão de um pacote do roteador i até o roteador j é representada por $d_{i,j}$.

Na Figura 1, podem-se observar os roteadores e conexões de rede contemplados no modelo, bem como uma representação do tempo agregado de serviço e latência relacionados com a posição do pacote na rede.

O número de pacotes enviados para a rede pela origem (*flow control*) é controlado pelo mecanismo de controle dinâmico de janela do TCP.

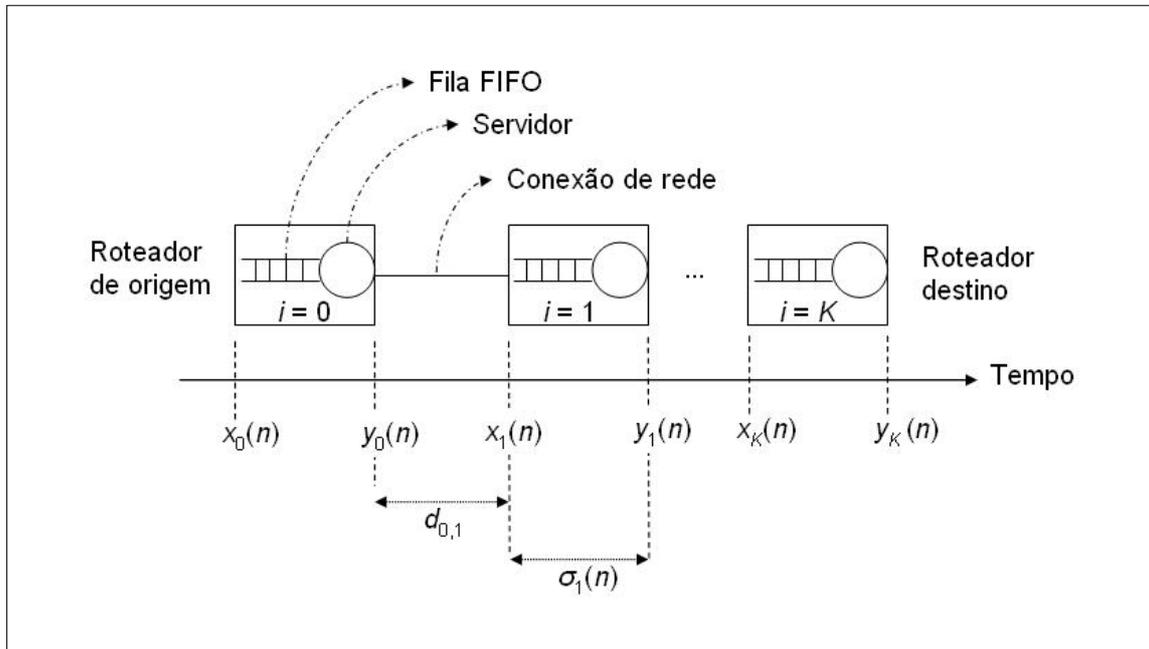


Figura 1: Modelo de rede.

O modelo apresentado considera ainda que um pacote de confirmação de entrega (*ACK*) segue um caminho direto do destino até a origem, como que se os roteadores de destino e de origem estivessem diretamente conectados por um circuito. Desta forma, os pacotes *ACK* não seguem o caminho reverso dos pacotes de dados da conexão TCP, e sim um caminho exclusivo entre o destino e a origem.

Esta consideração, embora seja uma simplificação do modelo em relação ao que realmente acontece na prática, permite representar o tempo de transmissão dos pacotes de confirmação como uma latência simples ($d_{k,0}$), sem considerar toda a complexidade da transmissão dos pacotes *ACK* (roteador a roteador) de volta à origem dos pacotes de dados.

2.4.2 Modelo max-plus

O modelo matemático utilizado é composto por um conjunto de equações que determinam os tempos de saída de cada pacote do *reference flow*, em cada um dos roteadores existentes na rede. O cálculo dos tempos de saída de cada pacote é feito considerando o

comportamento dinâmico do TCP em relação ao tamanho da janela de transmissão, e também as características da rede em uso, tais como o tempo de serviço dos roteadores e latência das conexões da rede.

O instante de tempo no qual um pacote n chega a um roteador i é representado por $x_i(n)$. Este mesmo pacote vai deixar o roteador em um tempo representado por $y_i(n)$, conforme ilustrado na Figura 1.

A diferença entre o tempo de saída e o tempo de entrada em um roteador é justamente o tempo agregado de serviço do pacote neste roteador. Assim, pode-se escrever: $y_i(n) = x_i(n) + \sigma_i(n)$.

O TCP controla dinamicamente a quantidade de pacotes oferecida para a rede, ajustando o tamanho da janela de transmissão. Assim, o TCP calcula um novo valor para o tamanho da janela sempre que a transmissão de um pacote é confirmada (no recebimento de um pacote do tipo *ACK*), ou quando é detectado *timeout* na transmissão do pacote. Assim, na transmissão de um pacote qualquer, o tamanho da janela é função do que ocorreu na transmissão do pacote anterior (confirmação ou *timeout*).

O tamanho da janela de transmissão, calculado em função do ocorrido com o pacote n é representado por v_n , que é o valor da janela de transmissão utilizado no momento da transmissão do pacote $n+1$. Ou seja, o tamanho da janela utilizada para transmissão do pacote n é v_{n-1} .

Considerando que a seqüência de valores $\{v_n\}$ é previamente conhecida, e que o tempo agregado de serviço na origem $\sigma_0(n)$ é sempre igual a 0 por definição, os tempos de saída dos pacotes em todos os roteadores podem ser obtidos com as Equações 1 e 2 na álgebra max-plus:

$$y_0(n) = y_K(n - v_{n-1}) \otimes d_{K,0} \quad (1)$$

$$y_i(n) = [y_{i-1}(n) \otimes d_{i-1,i} \oplus y_i(n-1)] \otimes \sigma_i(n) \quad , 1 \leq i \leq K \quad (2)$$

onde:

- n representa o n -ésimo pacote sendo analisado;
- $y_i(n)$ é o tempo de saída do pacote n no i -ésimo roteador. Em particular, $y_0(n)$ é o tempo de saída do pacote n no roteador de origem, $y_K(n)$ é o tempo de saída do pacote n no roteador destino (último roteador), $y_{i-1}(n)$ é o tempo de saída do pacote n no

roteador anterior àquele que se está analisando e $y_i(n-1)$ é o tempo de saída do pacote anterior àquele que se está analisando no i -ésimo roteador;

- v_{n-1} é o tamanho da janela de transmissão vigente durante a transmissão do pacote n ;
- $d_{K,0}$ é o tempo de transmissão desde o último roteador até o roteador de origem (tempo de retorno do pacote *ACK*). Em particular, $d_{i-1,i}$ é o tempo de transmissão entre o i -ésimo roteador e o roteador imediatamente anterior;
- $\sigma_i(n)$ é o tempo agregado de serviço do pacote n no i -ésimo roteador;
- \oplus representa a operação de adição na álgebra max-plus (equivalente à maximização);
- \otimes representa a operação de multiplicação da álgebra max-plus (equivalente à soma usual).

A Equação 1, em resumo, determina o instante de tempo no qual um pacote n é enviado à rede. A função desta equação é modelar o comportamento do TCP em relação ao controle da oferta de pacotes para a rede, fazendo uso do tamanho da janela de transmissão e do tempo de chegada dos pacotes de confirmação de entrega.

Uma sugestão para a interpretação da Equação 1 pode ser a seguinte: Um novo pacote pode ser enviado à rede sempre que o *ACK* (confirmação de entrega) de um determinado pacote anterior é recebida. A determinação de qual é o pacote anterior que afeta o instante de envio de um novo pacote depende do tamanho da janela de transmissão vigente v_{n-1} .

Por exemplo, se no momento da transmissão do pacote número 5 o valor vigente da janela de transmissão é 2, então o pacote 5 deve ser oferecido à rede quando da chegada da confirmação de entrega do pacote número 3. Como por definição $\sigma_0(n) \equiv 0$, o novo pacote é enviado no mesmo instante da chegada do *ACK* citado, e este instante é a soma do tempo de saída do pacote anterior (do pacote 3, neste exemplo) no último roteador com a latência do *ACK* na rede.

A Equação 2, por sua vez, tem como objetivo modelar as características da rede, determinando os tempos de saída dos pacotes em cada roteador da rede após o roteador de origem. O comportamento da rede é representado pelos tempos de transmissão $d_{i-1,i}$ entre roteadores, e também pelo tempo agregado de serviço $\sigma_i(n)$ em cada roteador.

Uma sugestão para a interpretação da Equação 2 poderia ser a seguinte. O tempo de saída de um pacote n em um roteador i é a soma entre as quantidades abaixo:

- tempo agregado de serviço $\sigma_i(n)$ do pacote neste roteador;
- o maior dos seguintes tempos:

- tempo de chegada do pacote neste roteador dado por $y_{i-1}(n) \otimes d_{i-1,i}$;
- tempo de saída do pacote imediatamente anterior deste roteador.

Como os tempos de transmissão entre roteadores são tipicamente valores constantes em uma rede, e o tempo agregado de serviço pode ser determinado em função das características de um roteador e do *cross traffic* existente, é possível obter de forma iterativa os tempos de saída de todos os pacotes do *reference flow* em todos os roteadores pertencentes à rede através da Equação 2, permitindo assim obter informações comparáveis àquelas obtidas por simulação (BACCELLI e HONG, 2000).

Na seção 2.5 será apresentado um exemplo numérico da utilização das Equações 1 e 2.

As Equações 1 e 2 podem ser reformuladas na forma de vetores, cujos componentes representam a evolução dos tempos de saída de cada pacote em cada roteador. Isso permite evidenciar a linearidade do modelo na álgebra max-plus, apresentado a seguir.

Os valores obtidos a partir da Equação 2 podem ser organizados na forma de um vetor $Y(n)$ da seguinte forma:

$$Y(n) = (y_1(n), y_2(n), \dots, y_K(n))^T \in \mathbf{R}_{\max}^K \quad (3)$$

O vetor Z , denominado *dater*, é definido como:

$$Z(n) = (Y(n) | Y(n-1) | \dots | Y(n-w^* - 1)) \in \mathbf{R}_{\max}^m \quad (4)$$

onde:

- w^* é o maior valor alcançado pela janela de transmissão (v) do TCP desde o início da transmissão até o pacote n ;
- $m = K \cdot w^*$

Enquanto o vetor Y contém informações somente sobre o pacote n , no vetor *dater* existem informações não só sobre os tempos de saída do pacote n nos roteadores, mas também informações sobre pacotes anteriores.

Neste caso, é possível obter o vetor $Z(n)$ a partir da multiplicação¹ de $Z(n-1)$ por uma matriz conveniente, que depende dos intervalos de transmissão e serviço para o pacote n ,

¹ Na álgebra max-plus o produto de matrizes é obtido de maneira análoga ao produto convencional de matrizes, substituindo-se a adição pela operação de maximização e a multiplicação pela soma usual.

assim como do tamanho da janela de transmissão vigente. Isso é mostrado na Equação 5, que é claramente linear na álgebra max-plus.

$$Z(n) = A_{v_{n-1}}(n) \otimes Z(n-1), \quad n \geq 1 \quad (5)$$

considerando:

$$Z(0) = (0, 0, \dots, 0)^t$$

A linearidade no caso da álgebra max-plus garante um certo “princípio da sobreposição” dos efeitos de atrasar um fluxo de pacotes de um valor constante (multiplicando por uma constante), assim como de impor limites superiores a um fluxo de pacotes (somando um valor limitante).

Como a matriz A é uma matriz que contém informações sobre o tempo agregado de serviço em cada roteador e também informações sobre o tempo de transmissão entre roteadores, esta matriz é formada basicamente por informações sobre a rede, que podem ser previamente determinadas antes mesmo do início da transmissão TCP.

A matriz A pode ser particionada nas matrizes M , M' , ε e D , conforme a Equação 6 abaixo:

$$\begin{aligned} A_1(n) &= (M(n) \oplus M'(n) \mid \varepsilon \mid \dots \mid \varepsilon) \oplus D, \\ A_2(n) &= (M(n) \mid M'(n) \mid \varepsilon \mid \dots \mid \varepsilon) \oplus D, \dots \\ A_w^*(n) &= (M(n) \mid \varepsilon \mid \dots \mid \varepsilon \mid M'(n)) \oplus D. \end{aligned} \quad (6)$$

Para uma melhor compreensão destas matrizes, suas dimensões são apresentadas na Tabela 1:

Tabela 1: Dimensões das matrizes nas Equações 5 e 6

Matriz	Formato
Z	\mathbf{R}_{\max}^m
A	$\mathbf{R}_{\max}^{m \times m}$
M	$\mathbf{R}_{\max}^{K \times K}$
M'	$\mathbf{R}_{\max}^{K \times K}$
ε	$\mathbf{R}_{\max}^{K \times K}$
D	$\mathbf{R}_{\max}^{m \times m}$

A matriz \mathcal{E} é formada por todos os elementos iguais a $-\infty$, ou seja, é a matriz nula na álgebra max-plus. As matrizes M e M' são montadas da seguinte forma:

$$(M(n))_{ij} = \begin{cases} \sum_{k=j}^i \sigma_k(n) + \sum_{k=j}^{i-1} d_{k,k+1}, & \text{se } i \geq j, \\ -\infty, & \text{se } i < j. \end{cases} \quad (7)$$

$$(M'(n))_{ij} = \begin{cases} \sum_{k=1}^i (d_{k-1,k} + \sigma_k(n)) + d_{K,0}, & \text{se } j = K \\ -\infty, & \text{se } j < K. \end{cases} \quad (8)$$

A matriz D tem o seguinte padrão:

$$D = \begin{bmatrix} \mathcal{E} & \mathcal{E} & \mathcal{E} & \mathcal{E} & \dots \\ I & \mathcal{E} & \mathcal{E} & \mathcal{E} & \dots \\ \mathcal{E} & I & \mathcal{E} & \mathcal{E} & \dots \\ \mathcal{E} & \mathcal{E} & I & \mathcal{E} & \dots \\ \dots & \dots & \dots & I & \dots \end{bmatrix}$$

onde:

- I é a matriz identidade², de dimensões $\mathbb{R}_{\max}^{K,K}$ e;
- \mathcal{E} trata-se da matriz anteriormente apresentada.

Uma definição mais formal para montagem da matriz D , pode ser encontrada em Baccelli e Hong, (2000).

Alguns exemplos de matriz A , poderiam ser (considerando $w^* = 3$):

$$A_2(n) = \begin{bmatrix} M(n) & M'(n) & \mathcal{E} \\ I & \mathcal{E} & \mathcal{E} \\ \mathcal{E} & I & \mathcal{E} \end{bmatrix} \quad A_3(n) = \begin{bmatrix} M(n) & \mathcal{E} & M'(n) \\ I & \mathcal{E} & \mathcal{E} \\ \mathcal{E} & I & \mathcal{E} \end{bmatrix}$$

² A matriz identidade na álgebra max-plus tem todos os elementos iguais a $-\infty$, exceto na diagonal $i=j$, onde os elementos são iguais a 0.

Observando o padrão da matriz A , é possível perceber que a maior parte dos seus elementos tem origem da matriz D .

Estes elementos da matriz A que são idênticos aos da matriz D têm por objetivo deslocar elementos de um *dater* para o novo *dater* que está sendo calculado. Este comportamento fica mais evidente no exemplo numérico que será apresentado na seção a seguir.

2.5 EXEMPLO NUMÉRICO

Para melhor entendimento do funcionamento do modelo TCP max-plus, será apresentado nesta seção um exemplo da evolução de uma conexão TCP em uma rede composta de dois roteadores entre a origem e o destino.

Os parâmetros da rede utilizada no exemplo podem ser observados na Tabela 2.

Tabela 2: Parâmetros da rede utilizada no exemplo

Parâmetro da rede	Valor (ms)
Tempo de transmissão entre a origem e o primeiro roteador	2
Tempo de transmissão entre o primeiro e o segundo roteador	3
Tempo de transmissão entre o segundo roteador e o destino	2
Tempo de retorno das confirmações (ACK)	7
Tempo agregado de serviço nos roteadores 1, 2 e destino	1

Fazendo uso das Equações 1 e 2, e também dos parâmetros da rede disponíveis na Tabela 2, é possível então calcular os tempos de saída nos vários roteadores para cada pacote que percorre a rede.

Na Tabela 3 é apresentado o tempo de saída para alguns pacotes da rede utilizada neste exemplo.

Tabela 3: Exemplo de tempos de saída

Pacote	Janela de transmissão ³	$y_o(n)$ (ms)	$y_1(n)$ (ms)	$y_2(n)$ (ms)	$y_K(n)$ (ms)
1	1	0	3	7	10
2	1	17	20	24	27
3	2	17	21	25	28
4	2	34	37	41	44
5	2	35	38	42	45
6	3	35	39	43	46
7	3	51	54	58	61
8	3	52	55	59	62
9	3	53	56	60	63
10	4	53	57	61	64
11	4	68	71	75	78
12	4	69	72	76	79
13	4	70	73	77	80
14	4	71	74	78	81
15	5	71	75	79	82

Continuando com o exemplo, pode-se ainda calcular os valores para o vetor Y e para o vetor Z (*dater*). Assim, alguns exemplos de valores para o vetor Y poderiam ser:

- $Y(1) = (3, 7, 10)$;
- $Y(4) = (37, 41, 44)$;
- $Y(15) = (75, 79, 82)$.

Ainda utilizando os mesmos dados, alguns exemplos de valores para o vetor Z poderiam ser:

³ O valor da janela de transmissão neste exemplo considerou o uso de TCP *Tahoe* sem *slow-start*

$$Z(4) = \begin{bmatrix} 37 \\ 41 \\ 44 \\ 21 \\ 25 \\ 28 \end{bmatrix} \quad Z(8) = \begin{bmatrix} 55 \\ 59 \\ 62 \\ 54 \\ 58 \\ 61 \\ 39 \\ 43 \\ 46 \end{bmatrix}$$

Para que seja possível apresentar exemplos detalhados da matriz A , com todos os seus elementos representados ao invés de apenas blocos de matrizes, é necessário montar as matrizes M, M', I e \mathcal{E} .

Por exemplo, na montagem da matriz A quando $n = 8$ e $v_{n-1} = 3$, ou seja, $A_3(8)$, as matrizes M e M' (calculadas de acordo com as Equações 7 e 8) são:

$$M(8) = \begin{bmatrix} 1 & -\infty & -\infty \\ 5 & 1 & -\infty \\ 8 & 4 & 1 \end{bmatrix} \quad M'(8) = \begin{bmatrix} -\infty & -\infty & 10 \\ -\infty & -\infty & 14 \\ -\infty & -\infty & 17 \end{bmatrix}$$

e as matrizes I e \mathcal{E} são, neste caso:

$$I = \begin{bmatrix} 0 & -\infty & -\infty \\ -\infty & 0 & -\infty \\ -\infty & -\infty & 0 \end{bmatrix} \quad \mathcal{E} = \begin{bmatrix} -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty \end{bmatrix}$$

finalmente, a matriz $A_3(8)$ neste exemplo é:

$$A_3(8) = \begin{bmatrix} 1 & -\infty & 10 \\ 5 & 1 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & 14 \\ 8 & 4 & 1 & -\infty & -\infty & -\infty & -\infty & -\infty & 17 \\ 0 & -\infty \\ -\infty & 0 & -\infty \\ -\infty & -\infty & 0 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & 0 & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & 0 & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & 0 & -\infty & -\infty & -\infty \end{bmatrix}$$

Utilizando a Equação 5, pode-se obter o valor de $Z(8)$, que é *dater* para o oitavo pacote da rede, à partir do *dater* do sétimo pacote e da matriz $A_3(8)$:

$$Z(8) = A_3(8) \otimes Z(7)$$

$$\begin{bmatrix} 1 & -\infty & 10 \\ 5 & 1 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & 14 \\ 8 & 4 & 1 & -\infty & -\infty & -\infty & -\infty & -\infty & 17 \\ 0 & -\infty \\ -\infty & 0 & -\infty \\ -\infty & -\infty & 0 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & 0 & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & 0 & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & 0 & -\infty & -\infty & -\infty \end{bmatrix} \otimes \begin{bmatrix} 54 \\ 58 \\ 61 \\ 39 \\ 43 \\ 46 \\ 38 \\ 42 \\ 45 \end{bmatrix} = \begin{bmatrix} 55 \\ 59 \\ 62 \\ 54 \\ 58 \\ 61 \\ 39 \\ 43 \\ 46 \end{bmatrix}$$

Usando a mesma lógica, pode-se obter o *dater* do pacote 9 à partir do *dater* do pacote 8 (sempre utilizando a matriz A) e assim por diante.

No exemplo anterior, os elementos das linhas 1 a 6 de $Z(7)$ tornam-se, respectivamente, os elementos das linhas 4 a 9 de $Z(8)$, por efeito da multiplicação max-plus. Já os elementos das linhas 7 a 9 de $Z(7)$ não participam mais de $Z(8)$, uma vez que, devido ao tamanho da janela de transmissão TCP vigente, estes valores não exercem mais efeito sobre o pacote atual.

Nesta seção, os cálculos foram realizados utilizando a formulação apresentada na Equação 5. Esta formulação evidencia o aspecto linear do modelo TCP max-plus, porém não é a forma computacionalmente mais eficiente para realização dos cálculos.

As matrizes utilizadas na Equação 5 podem conter um grande número de elementos, muitos deles repetidos, cujo único objetivo é deslocar elementos de um *dater* para outro. Estes elementos ocupam muita memória durante o cálculo e demandam muitas operações desnecessárias, com resultados previsíveis.

Para realizar estes cálculos de uma forma computacionalmente mais eficiente, é possível, por exemplo, escrever um programa que copia os elementos que ficam repetidos de um *dater* para outro, e somente calcular valores novos para o *dater*.

2.6 FERRAMENTAS COMPUTACIONAIS

O modelo sugerido por (BACCELLI e HONG, 2000) pode ser utilizado em um programa de computador para facilitar o cálculo das várias matrizes e vetores necessários e assim automatizar a obtenção de resultados para uma transmissão em TCP modelada usando a álgebra max-plus.

Um aplicativo que é comumente usado para cálculos matriciais e vetoriais em computadores é o *Matlab*⁴. Para executar operações na álgebra max-plus usando o *Matlab*, existe uma ferramenta chamada *Max-plus Algebra Toolbox* (STAŃCZYK, 2005), que é um conjunto de funções que pode ser agregado ao *Matlab*. O *Max-plus Algebra Toolbox* possui funções para computar tanto as operações básicas de soma e multiplicação na álgebra max-plus (*mp_sum* e *mp_multi*, respectivamente), como também funções mais avançadas para obtenção de autovalores, utilização em grafos e outras.

O que torna o *Max-plus Algebra Toolbox* interessante é a possibilidade de utilizar as funções de forma transparente tanto sobre valores escalares como sobre matrizes.

O conjunto completo de funções suportadas pela ferramenta e também instruções para seu uso podem ser encontradas em (STAŃCZYK, 2005).

Para montar no *Matlab* o modelo TCP max-plus, foram desenvolvidas funções, a partir do *Max-plus Algebra Toolbox*, para montagem do vetor Z e das matrizes M , M' , D e A . Detalhes sobre o formato para utilização das funções estão disponíveis na Tabela 4.

⁴ Matlab é uma marca registrada da empresa The MathWorks, Inc. (<http://www.mathworks.com>)

Tabela 4: Funções desenvolvidas no *Matlab* para modelagem TCP max-plus

Função	Descrição	Parâmetros
tcp_Z(y, n, v)	Retorna o vetor Z para um determinado pacote, a partir de uma matriz y com os tempos de saída dos pacotes nos roteadores.	y: matriz com $n \times i$ elementos contendo os tempos de saída de um pacote n em um roteador i ; n: número do pacote; v: vetor com a evolução da janela TCP.
tcp_D(n, K, v)	Retorna a matriz D usada na construção da matriz A .	n: número do pacote; K: número de roteadores após a origem (exclusive), até o roteador destino (inclusive); v: vetor com a evolução da janela TCP.
tcp_M(n, K, ts, d)	Retorna a matriz M usada na construção da matriz A , a partir de matrizes contendo o tempo agregado de serviço nos roteadores e o tempo de transmissão entre roteadores.	n: número do pacote; K: número de roteadores após a origem (exclusive), até o roteador destino (inclusive); ts: matriz com $n \times i$ elementos contendo o tempo agregado de serviço de um pacote n em um roteador i ; d: matriz com $n \times i$ elementos contendo o tempo de transmissão de um pacote n até o roteador i , desde o roteador imediatamente anterior.
tcp_Ml(n, K, ts, d, d_ack)	Retorna a matriz M' usada na construção da matriz A , a partir de matrizes contendo o tempo agregado de serviço nos roteadores,	n: número do pacote; K: número de roteadores após a origem (exclusive), até o roteador destino (inclusive); ts: matriz com $n \times i$ elementos contendo o tempo agregado de

	<p>tempo de transmissão entre roteadores e o tempo de retorno dos pacotes <i>ACK</i>.</p>	<p>serviço de um pacote <i>n</i> em um roteador <i>i</i>;</p> <p>d: matriz com $n \times i$ elementos contendo o tempo de transmissão de um pacote <i>n</i> até o roteador <i>i</i>, desde o roteador imediatamente anterior;</p> <p>d_ack: vetor coluna contendo o tempo de retorno da confirmação de entrega (<i>ACK</i>) do pacote <i>n</i>, desde o destino até a origem.</p>
<p>tcp_A(a, n, v, K, ts, d, d_ack)</p>	<p>Retorna a matriz <i>A</i>, a partir de todos os dados necessários para montagem das matrizes intermediárias <i>M</i>, <i>M'</i> e <i>D</i>.</p>	<p>a: índice da matriz <i>A</i> desejada. Por exemplo, A_2 ou A_3;</p> <p>n: número do pacote;</p> <p>v: vetor com a evolução da janela TCP;</p> <p>K: número de roteadores após a origem (exclusive), até o roteador destino (inclusive);</p> <p>ts: matriz com $n \times i$ elementos contendo o tempo agregado de serviço de um pacote <i>n</i> em um roteador <i>i</i>;</p> <p>d: matriz com $n \times i$ elementos contendo o tempo de transmissão de um pacote <i>n</i> até o roteador <i>i</i>, desde o roteador imediatamente anterior;</p> <p>d_ack: vetor coluna contendo o tempo de retorno da confirmação de entrega (<i>ACK</i>) do pacote <i>n</i>, desde o destino até a origem.</p>

Uma listagem dos arquivos *Matlab* que implementam as funções da Tabela 4 pode ser encontrada no apêndice.

A Figura 2 é um exemplo de programa em *Matlab*, que utiliza as funções do Max-plus *Algebra Toolbox* para simular uma transmissão em TCP com os mesmos parâmetros do exemplo apresentado na seção 2.5, fazendo uso das Equações 1 e 2 previamente apresentadas.

Ao final da execução do programa, a matriz representada pela variável *y* do *Matlab* tem valores idênticos aos valores apresentados na Tabela 3.

```
clear;
K=3; % Número de roteadores
N=21; % Número total de pacotes

% Montagem da matriz com o tempo agregado de serviço
ts = ones(N,K);

% Montagem da matriz com os tempos de transmissão
for n = 1:N;
    d(n,:) = [2,3,2];
end;

% Montagem do vetor com os tempos de retorno do ACK
d_ack = ones(N,1);
d_ack(:,1) = 7;

% Montagem do vetor com a janela de transmissão do TCP
% considerando TCP Tahoe sem slow-start
% OBS: primeiro elemento do vetor já representa Vn-1
v = [1,1,2,2,2,3,3,3,3,4,4,4,4,4,5,5,5,5,5,5,6];

% Calculando passo a passo os tempos de saída:
y = zeros(N,K+1);
for n = 1:N;
    if n == 1
        y(1,1) = 0;
    else
        delta = n - v(1,n);
        if delta < 1 delta = 1; end;

        % Equação 1
        y(n,1) = mp_multi( y(delta,K+1),d_ack(delta,1) );

    end;
    for i = 2:K+1;
        if n == 1
            saida_pct_ant = 0;
        else
            saida_pct_ant = y(n-1,i);
        end;

        % Equação 2
        y(n,i) = mp_multi( mp_sum( mp_multi( y(n,i-1),d(n,i-1) ),saida_pct_ant ), ts(n,i-1));

    end;
end;
end;
```

Figura 2: Exemplo de programa formulado a partir das Equações 1 e 2

Outra forma de se montar um programa *Matlab*, que obtém o mesmo resultado do programa anterior, será apresentada a seguir usando a função de recorrência da Equação 5.

O programa apresentado na Figura 3, além de fazer uso do Max-plus *Algebra Toolbox*, desta vez utiliza também as funções para montagem das matrizes e vetores previamente apresentados na Tabela 4, desenvolvidas especialmente para auxiliar na montagem da função de recorrência.

Mais uma vez, ao final da execução do programa, a matriz representada pela variável *y* do *Matlab* tem valores similares aos valores apresentados na Tabela 3, exceto no que diz respeito aos valores de tempo de saída para cada pacote na origem (primeira coluna da matriz *y*), valores estes que não são utilizados nas matrizes *Z* do modelo max-plus.

```
clear;
K=3; % Número de roteadores
N=21; % Número total de pacotes

% Montagem da matriz com o tempo agregado de serviço
ts = ones(N,K);

% Montagem da matriz com os tempos de transmissão
for n = 1:N;
    d(n,:) = [2,3,2];
end;

% Montagem do vetor com os tempos de retorno do ACK
d_ack = ones(N,1);
d_ack(:,1) = 7;

% Montagem do vetor com a janela de transmissão do TCP
% considerando TCP Tahoe sem slow-start
% OBS: primeiro elemento do vetor já representa Vn-1
v = [1,1,2,2,2,3,3,3,3,4,4,4,4,4,5,5,5,5,5,5,6];

% Inicialização da matriz y
y = zeros(N,K+1);

% Atribuição dos tempos do primeiro pacote
y(1,:)=[0,3,7,10];

% Cálculo dos tempos de todos os pacotes usando função de recorrência
for n=2:N
    Z = tcp_Z(y,n-1,v);
    A = tcp_A(v(1,n),n,v,K,ts,d,d_ack);

    % Equação 5
    Z = mp_multi(A,Z);

    y(n,2:K+1) = Z(1:K,1)';
end;
```

Figura 3: Exemplo de programa formulado a partir da Equação 5

É importante ressaltar que a função *tcp_A* utiliza, internamente, as funções *tcp_M*, *tcp_Ml* e *tcp_D*.

2.7 DETERMINAÇÃO DA VELOCIDADE DA REDE USANDO MAX-PLUS

A partir do modelo TCP max-plus é possível determinar, algebricamente, qual é a capacidade de transmissão de uma rede. Em (BACCELLI e HONG, 2000), diferentes técnicas para obter as capacidades de transmissão são apresentadas. Uma das técnicas, em especial, foi a técnica escolhida para comparar o desempenho da rede determinado algebricamente com resultados obtidos por simulação orientada a eventos discretos.

Esta técnica, chamada de *Rate Based Loss Detection* (detecção de perda baseada em taxa), procura identificar qual é o valor máximo de transmissão do TCP comparando a taxa média de envio de pacotes para a rede com a frequência de pacotes processados pelo roteador mais congestionado (taxa de serviço).

A taxa de serviço é $1/\sigma^*(n)$, onde $\sigma^*(n)$ é maior tempo agregado de serviço encontrado na rede até o momento, que certamente será encontrada no roteador mais congestionado.

A taxa média de envio de pacotes é $v_{n-1}/S(n)$ onde v_{n-1} é o valor vigente da janela de transmissão TCP e $S(n)$ é o valor da latência da rede (o tempo que um pacote gasta desde a entrada na rede até o recebimento da confirmação de entrega deste pacote). O valor de $S(n)$ pode ser calculado a partir do tempo de saída dos pacotes no roteador destino mais o tempo de retorno dos pacotes *ACK*.

A técnica detecta que o TCP passa a sofrer congestionamento no instante no qual a taxa média de envio de pacotes alcança a taxa de serviço, ou seja, quando:

$$\frac{v_{n-1}}{S(n)} > \frac{1}{\sigma^*(n)} \quad (9)$$

Assim, pode-se então determinar a capacidade máxima de uma rede para tráfego TCP, que é a taxa de envio vigente imediatamente antes da detecção de congestionamento pela Equação 9.

Para ilustrar essa medida, lançando mão do exemplo de rede apresentado na seção 2.5, verifica-se que a taxa de serviço é $1/\sigma^*(n) = 1$, pois o tempo agregado de serviço é constante e igual a 1 ms em todos os roteadores.

Já a taxa média de envio de pacotes neste exemplo pode ser observada na Tabela 5.

Tabela 5: Exemplo de taxa média de envio

Pacote	Janela vigente	Latência média (ms)	Taxa de envio (pct/ms)
n	v_{n-1}	$S(n)$	$v_{n-1}/S(n)$
1	1	17	0,0588
2	1	17	0,0588
3	2	18	0,1111
4	2	17	0,1176
...
40	8	17	0,4706
...
100	13	17	0,7647
...
150	16	17	0,9412
151	16	17	0,9412
152	16	17	0,9412
153	16	18	0,9444
154	17	17	1

Analisando a Tabela 5, pode-se notar que o desempenho máximo da rede utilizada no exemplo é alcançado durante a transmissão do pacote de número 153, quando a janela de transmissão do TCP está com valor 17. Isto porque no próximo pacote, de número 154, a taxa de envio atinge o valor 1, que é a taxa de serviço máxima dessa rede, evidenciando que a partir desse momento haverá congestionamento.

Conclui-se então que a velocidade máxima dessa rede é 944 pacotes por segundo, justamente o valor da taxa de envio durante a transmissão do pacote 153.

Além da velocidade máxima da rede, é possível ainda determinar a velocidade média de transmissão da rede até o momento no qual o congestionamento é detectado. O desempenho médio da rede desde o início da transmissão foi de 513 pacotes por segundo, devido ao fato de que o TCP iniciou sua transmissão com uma janela pequena, que foi ajustada gradualmente durante a transmissão. Somente quando a janela atingiu o valor 16 é que o potencial máximo da rede pode ser explorado.

O valor da velocidade média foi obtido dividindo o número de pacotes transmitidos (até o momento de detecção de congestionamento) pelo tempo de chegada ao destino do último pacote enviado antes do congestionamento.

Para permitir o uso do *Rate Based Loss Detection* no *Matlab*, foi desenvolvida uma função chamada de `tcp_G1`, complementando o conjunto de funções apresentado na seção 2.6.

Tabela 6: Função desenvolvida no *Matlab* para *Rate Based Loss Detection*

Função	Descrição	Parâmetros
<code>tcp_G1(y, v, ts, d_ack)</code>	Retorna o vetor com a diferença de tempo entre a taxa média de envio de pacotes e a taxa de serviço: $\frac{v_{n-1}}{S(n)} - \frac{1}{\sigma^*(n)}$	y: matriz com $n \times i$ elementos contendo os tempos de saída de um pacote n em um roteador i ; v: vetor com a evolução da janela TCP; ts: matriz com $n \times i$ elementos contendo o tempo agregado de serviço de um pacote n em um roteador i ; d_ack: vetor coluna contendo o tempo de retorno da confirmação de entrega (<i>ACK</i>) do pacote n , desde o destino até a origem.

Na seção a seguir, é utilizada a técnica *Rate Based Loss Detection* para obter a taxa de transmissão de uma rede modelada usando max-plus, e o resultado é comparado com os valores obtidos por simulação. No Capítulo 4, novamente esta técnica será utilizada para determinar a taxa de transmissão de uma rede usando o modelo max-plus.

2.8 COMPARAÇÃO DE MODELOS: MAX-PLUS x SIMULAÇÃO

Baccelli e Hong (2000) afirmam que as taxas médias de transferência calculadas de forma determinística pelo modelo max-plus são comparáveis àquelas obtidas com o uso do simulador *ns*.

Para validar o funcionamento das ferramentas desenvolvidas para no *Matlab*, e também para reproduzir o resultado apresentado por Baccelli e Hong (2000), um exemplo de

rede semelhante ao utilizado pelos autores citados foi montado em laboratório usando o simulador *ns-2* e também no *Matlab* usando as ferramentas max-plus apresentadas na seção 2.6.

A rede utilizada possui transmissão em TCP com 10 roteadores entre a origem e o destino, não incluindo a origem. Os roteadores são ligados por circuitos com taxas de transmissão e latência conforme a Tabela 7:

Tabela 7: Parâmetros da rede simulada

Circuito	Latência (ms)	Taxa de transmissão (Mbps)
Entre a origem e o roteador 1	0,1	10
Entre roteador 1 e roteador 2	0,1	5
Entre roteador 2 e roteador 3	0,1	4
Entre roteador 3 e roteador 4	0,1	2
Entre roteador 4 e roteador 5	0,1	5
Entre roteador 5 e roteador 6	0,1	4
Entre roteador 6 e roteador 7	0,1	5
Entre roteador 7 e roteador 8	0,1	5
Entre roteador 8 e roteador 9	0,1	4
Entre roteador 9 e o destino	0,1	5
Entre o destino e a origem (ACKs)	1	5

O tamanho dos pacotes de dados foi fixado em 1250 *bytes*, com pacotes ACK de 40 *bytes*.

Para representar a rede acima usando o modelo max-plus, no *Matlab*, a taxa de transmissão dos circuitos foi transformada em uma indicação do tempo agregado de serviço nos roteadores. Por exemplo, o circuito com menor taxa de transmissão, com 2 Mbps, indica um tempo agregado de serviço de 5 ms (o maior tempo agregado de serviço entre todos os roteadores). Este valor é obtido transformando a taxa de transmissão de *bits* por segundo para *bytes* por segundo (dividindo por 8), e em seguida para pacotes por segundo (dividindo por 1250, o tamanho de cada pacote). O inverso desse valor é o tempo agregado de serviço.

Além disso, a técnica para controle da janela TCP utilizada no exemplo do artigo não é citada, motivo pelo qual se optou pelo uso do TCP *Tahoe* sem *slow-start*, técnica que já havia sido usada previamente pelos autores em outro exemplo.

Simulando o modelo no *Matlab*, foi encontrada a taxa de transmissão média de aproximadamente 115 pacotes por segundo (1,15 Mbps), na chegada do pacote de número 21 (o último pacote antes do início do congestionamento). Este resultado é inferior ao valor apurado por Baccelli e Hong (2000), pois a simulação realizada no Matlab considera somente o número de pacotes transmitidos até o momento de detecção de congestionamento para cálculo da taxa de transmissão média, sem que a rede tenha oportunidade de continuar sua transmissão por mais algum tempo em velocidade máxima ou próxima à máxima.

A mesma rede foi simulada em laboratório usando o simulador *ns-2*. A rede foi modelada usando circuitos entre os nós com velocidades de acordo com a Tabela 7, filas com tamanho igual a 2 pacotes e finalmente pacotes com tamanho igual ao utilizado na simulação em *Matlab*.

A taxa de transmissão obtida na simulação em *ns-2*, apurada até o momento da entrega do pacote de número 21, foi de 112,7 pacotes por segundo.

Estes resultados mostram que os valores obtidos por simulação *ns-2* são de fato comparáveis aos resultados calculados usando o modelo max-plus. Segundo Baccelli e Hong (2000), as diferenças existentes devem-se ao fato de que enquanto no modelo max-plus a detecção de congestionamento é instantânea, o que é uma simplificação em relação ao que de fato ocorre em uma transmissão TCP, em uma simulação *ns-2* a detecção de congestionamento não é instantânea.

2.9 CONCLUSÃO

Uma importante propriedade do modelo de rede obtido por Baccelli e Hong (2000) é a linearidade. Isto permite modelar transmissões em TCP, ainda que trafegando sobre diferentes redes, como sistemas individuais. E, devido ao fato dos sistemas serem lineares, é possível prever o resultado de operações como, por exemplo, o encadeamento destes sistemas.

É importante salientar que o modelo é linear, mas possivelmente variante no tempo, pois a matriz A utilizada na multiplicação dos *daters* não é constante.

O modelo max-plus considera que os pacotes de confirmação de entrega (*ACKs*) seguem um caminho na rede diferente do caminho seguido pelos pacotes de dados. Essa premissa é uma simplificação em relação a uma rede real, onde tipicamente os pacotes *ACK* retornam à origem dos dados pelo mesmo caminho utilizado pelos pacotes de dados, porém em direção contrária.

Em uma rede que utiliza circuitos com fio para conexão entre os roteadores, esta simplificação não traz prejuízo ao modelo, pois tipicamente circuitos com fio utilizam transmissão *full-duplex* e, portanto, os pacotes *ACK* não interferem nos pacotes de dados que passam em sentido contrário pelos circuitos. Com os dois tipos de pacotes podendo ser transmitidos ao mesmo tempo em sentidos opostos em um circuito, o modelo permanece próximo à realidade.

No modelo TCP max-plus, alguns parâmetros procuram modelar a rede sobre a qual o TCP trafega. Os parâmetros utilizados para modelar a rede, que são o tempo agregado de serviço nos roteadores e a latência na transmissão entre os roteadores, podem ser especificados para diferentes tecnologias de rede, sem que haja necessidade de alterações na estrutura do modelo.

Nos próximos capítulos, a aplicabilidade do modelo TCP max-plus em redes sem fio será explorada, motivo pelo qual serão apresentadas no capítulo a seguir algumas características que afetam o desempenho de redes sem fio.

CAPÍTULO 3

ASPECTOS DE DESEMPENHO EM REDES SEM FIO

3.1 INTRODUÇÃO

Com a evolução das redes sem fio, sua utilização tornou-se cada vez mais comum. Até mesmo em residências, atualmente as redes sem fio são utilizadas em substituição às redes tradicionais em aplicações idênticas às utilizadas em redes com fio. A tecnologia de rede sem fio mais utilizada atualmente segue o padrão IEEE 802.11.

Porém, estas aplicações e seus protocolos foram desenvolvidos quando a tecnologia de rede comum era a rede com fio. Por esse motivo, os protocolos que suportavam as aplicações de rede foram desenvolvidos visando esta tecnologia padrão de rede na época. Em consequência, alguns protocolos como o TCP, têm desempenho degradado quando utilizado em uma rede sem fio.

Alguns estudos demonstram que as redes sem fio, em especial as sem infra-estrutura, apresentam desempenho bastante inferior ao desempenho obtido em redes com fio (LI, BLAKE, DE COUTO et al, 2001; HOLLAND e VAIDYA, 1999).

Neste capítulo, são apresentados alguns fatores que deterioram o desempenho em redes sem fio e, particularmente, em relação ao protocolo TCP. O protocolo de rede sem fio escolhido para a análise deste capítulo foi o IEEE 802.11.

3.2 FATORES QUE AFETAM O DESEMPENHO DE REDES SEM FIO

3.2.1 Controle de acesso ao meio

Um dos principais problemas enfrentados pelo protocolo 802.11 é o controle do acesso ao meio de transmissão. Como a faixa de frequência destinada ao uso em uma rede sem fio é compartilhada por todos os nós que participam da rede, é essencial que exista um algoritmo conhecido por todos os nós que regulamente o acesso ao meio de transmissão. O padrão 802.11 (IEEE, 1999) descreve dois diferentes tipos de controle de acesso ao meio de

transmissão, conhecidos por DCF (*Distributed Coordination Function*) e PCF (*Point Coordination Function*).

O padrão 802.11 determina que o suporte ao DCF é obrigatório enquanto o suporte ao PCF é opcional. Isso na prática fez com que o controle para acesso ao meio em redes sem fio padrão 802.11 seja sempre o DCF, já que é extremamente incomum encontrar no mercado equipamentos que suportem o PCF (GEIER, 2002). Neste documento será explorado somente o formato DCF – o leitor pode obter mais informações sobre o PCF no descritivo do padrão 802.11 (IEEE, 1999).

O principal objetivo do DCF é evitar que duas ou mais estações transmitam informações ao mesmo tempo, o que resultaria em colisão, impossibilitando a correta recepção da informação. A técnica usada no DCF é conhecida por CSMA/CA (*Carrier Sense Multiple Access / Collision Avoidance*). Como o próprio nome deixa claro, o que se procura é evitar colisões, ao invés de detectar colisões como acontece em uma rede com fio do tipo *Ethernet* (802.3).

Uma estação utilizando DCF, antes de iniciar a transmissão de dados, “escuta” o meio para detectar se ele está em uso ou não, o que é chamado de detecção de portadora. A detecção de portadora consiste em medir o nível de energia do sinal presente no canal, e caso a intensidade deste sinal esteja acima de um determinado valor, isto significa que o sinal é proveniente da transmissão de uma outra estação presente na rede. Se o meio não estiver em uso, a estação poderá iniciar o processo de transmissão de dados após aguardar por um tempo chamado de IFS (*Inter Frame Space*) mais, caso ocorra colisão, um intervalo de retenção aleatório (*Backoff*). A função do intervalo aleatório é evitar que duas ou mais estações com dados a serem transmitidos iniciem a transmissão simultaneamente, o que causaria colisão.

Um sinal transmitido por uma estação pode ser interpretado corretamente pelo receptor se não houver colisão e se o sinal chegar ao receptor com potência suficientemente maior que o ruído do canal e outras interferências. O raio de alcance dentro do qual um sinal transmitido pode ser corretamente recebido é chamado de *alcance de transmissão*.

Existe ainda o chamado *alcance de detecção de portadora*, que é a área na qual o sinal transmitido tem potência suficiente para que uma estação possa detectar a existência de portadora, ou seja, consegue detectar que existe outra estação transmitindo, ainda que a potência do sinal não seja suficiente para a correta recepção do dado transmitido.

Naturalmente, o alcance de detecção de portadora é maior que o alcance de transmissão. Ainda maior é o *alcance de interferência*, que é a área dentro da qual um sinal pode interferir em outra transmissão, causando colisão, mesmo que não seja forte o suficiente

para que uma estação detecte o sinal e o considere como portadora (XU, GERLA e BAE, 2002).

Quando uma estação A pretende transmitir dados para uma estação B, e existe ainda uma estação C (que também deseja transmitir dados para B), sempre que a estação C estiver fora do alcance de detecção da portadora de A, pode ocorrer o problema do terminal escondido (*hidden terminal*): A e C podem transmitir para B simultaneamente, causando colisão, pois C não consegue detectar a transmissão de A e vice-versa.

Para minimizar o problema do terminal escondido, o DCF pode utilizar quadros de controle RTS (*Request to send*) e CTS (*Clear to send*) antes de enviar o quadro com o dado propriamente dito. Além desses dois quadros existe ainda o quadro de controle ACK (*acknowledgment*) que confirma o correto recebimento da informação transmitida. A estação que deseja transmitir um quadro de dados deve enviar um quadro RTS e aguardar a recepção do quadro CTS enviado pela estação receptora antes de transmitir o quadro com dados. Após receber corretamente o dado, o receptor responde com um quadro ACK para que o transmissor saiba que o dado foi recebido corretamente.

Os quadros de controle carregam informações sobre o tempo necessário para a transmissão, reservando assim o canal pelo tempo necessário. Todas as estações que estiverem dentro do alcance de transmissão dos quadros RTS e CTS devem reter suas transmissões pelo tempo informado nos quadros de controle.

No exemplo dado anteriormente para o problema do terminal escondido, caso seja utilizado controle de fluxo via RTS/CTS, o problema é solucionado, pois a estação C detecta o quadro CTS enviado por B antes do início da transmissão de A para B, e assim pode aguardar o término da comunicação antes de iniciar sua transmissão.

Pela capacidade do controle de fluxo usando RTS/CTS de minimizar o problema do terminal escondido, ele é também chamado de detecção de portadora virtual (*virtual carrier sensing*). Vale ressaltar que o controle RTS/CTS somente é utilizado para quadros de dados de tamanho razoável, pois em quadros muito pequenos não se justifica o consumo de banda adicionado pelos quadros de controle. Os quadros ACK são sempre utilizados, independente do uso ou não de RTS/CTS.

O tempo de espera após a detecção de canal disponível, IFS, pode ser de dois diferentes tipos, o SIFS (*Short IFS*) ou DIFS (*Data IFS*). O DIFS é usado sempre antes do início do envio de dados, situação que pode ser antes de um quadro RTS (quando se está usando RTS/CTS), antes de um quadro de dados (quando não se está usando RTS/CTS), ou ainda após a reserva do meio usando RTS/CTS. Nas outras situações utiliza-se o tempo SIFS.

Como o SIFS é menor que o DIFS, os quadros seguintes ao início da transmissão têm prioridade sobre o quadro que dá início ao envio de dados, assim assegura-se que nenhuma outra estação irá iniciar o processo de transmissão até que o final da transmissão em curso.

Se o único tempo de espera fosse o IFS, logo após a meio ficar livre várias estações começariam a transmitir ao mesmo tempo. Por isso, o *backoff* aleatório é importante, aumentando a probabilidade de que mesmo após o meio ficar livre pelo tempo DIFS somente uma estação começará a transmitir.

O tempo de *backoff* é um valor aleatório, distribuído uniformemente entre 0 e $CW-1$. CW é a janela de contenção (*contention window*), que inicia com um valor mínimo e tem seu valor dobrado a cada falha de transmissão. Assim, quando uma estação pretende transmitir dados, primeiramente utiliza a detecção de portadora para identificar quando o canal está livre, em seguida aguarda pelo tempo DIFS, e depois aguarda pelo tempo de *backoff*, contando regressivamente até que ao chegar a zero, caso o canal ainda esteja livre, a estação inicia sua transmissão. A seqüência de eventos que ocorre antes de uma transmissão está apresentada na Figura 4 (QIAO e SHIN, 2003).

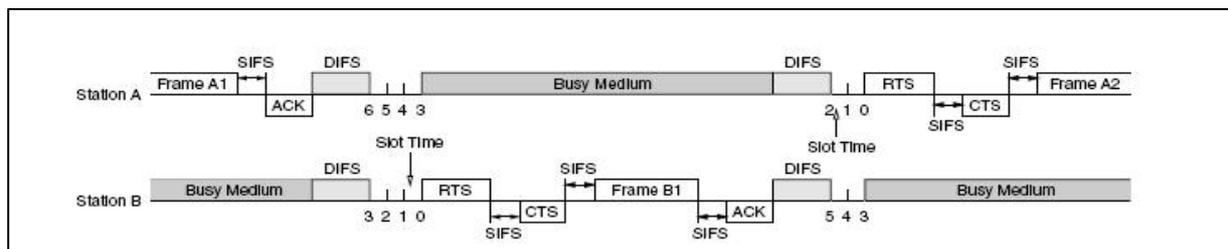


Figura 4: Seqüência de eventos para transmissão usando DCF

Quando durante o tempo de *backoff* o canal volta a ficar ocupado, o contador regressivo do *backoff* é parado, e no próximo intervalo de *backoff* a contagem recomeça deste ponto, aumentando a chance da transmissão poder acontecer antes de outra estação.

Enquanto em uma rede estruturada normalmente cada estação está dentro do alcance de transmissão (ou de detecção de portadora) das demais, em uma rede *ad hoc* é muito comum que uma estação esteja fora do alcance de detecção de portadora de outra. Nesta situação, o DCF ainda conseguiria evitar colisão com o uso da detecção virtual de portadora. No entanto, segundo Xu, Gerla e Bae (2002), o alcance de interferência costuma ser 1,78 vezes maior que o alcance de transmissão, além de ser maior que o alcance de detecção de portadora.

Assim, em uma rede *ad hoc* com múltiplos saltos, o problema do terminal escondido volta a ser relevante, e neste caso o DCF e a detecção virtual de portadora nada podem fazer para evitar o problema. As colisões voltam a acontecer, pois uma estação que está alguns saltos à frente na seqüência de transmissão entre origem e destino final dos dados não será capaz de identificar um quadro CTS enviado por uma estação anterior, e sua transmissão causará colisão. Esta situação é típica em uma rede sem fio de múltiplos saltos onde as estações estão organizadas em linha, cada uma próxima do limite do alcance de transmissão uma da outra.

O 802.11 tenta retransmitir os dados caso não receba um quadro ACK, e descarta os dados após quatro tentativas sem sucesso, ou seja, sem que um ACK seja recebido. Além disso, o 802.11 também descarta dados após o envio de sete quadros RTS sem receber o quadro CTS. Além da diminuição da eficiência da rede devido à necessidade de retransmissões, o aumento no tempo de *backoff* causado pelas falhas contribui ainda mais para a degradação do desempenho da rede.

3.2.2 *Overheads* impostos pelo MAC/DCF e pelo protocolo TCP

Em uma rede sem fio, a maior taxa de transmissão possível é aquela realizada somente entre dois nós, sendo que a origem/destino dos dados corresponde, respectivamente, ao transmissor/receptor da rede sem fio. Ainda neste cenário, é necessário que nenhum outro nó além daqueles envolvidos na transmissão esteja compartilhando do mesmo meio de transmissão, deixando assim o canal livre de concorrência.

Nessa situação, onde não existem nós intermediários responsáveis pelo roteamento dos pacotes entre a origem e o destino, nem concorrência pelo uso do canal, a única limitação para que a taxa de transmissão não seja exatamente a velocidade máxima dos dispositivos de rede é o *overhead* imposto pelos vários protocolos envolvidos na transmissão, em diferentes níveis da rede. O *overhead* pode ser devido a cabeçalhos necessários em cada pacote de dados ou então devido a pacotes de controle dos protocolos.

No nível de enlace da rede, em uma rede padrão 802.11, o protocolo de acesso ao meio (MAC ou *media access control*) faz uso de quadros de controle do tipo CTS e RTS (*clear to send* e *request to send*, respectivamente) para alocar o canal pelo tempo necessário para que a transmissão seja realizada. Além disso, após cada transmissão de dados o receptor envia um quadro ACK para confirmar que os dados foram transmitidos corretamente.

Sabendo que os quadros do tipo RTS tem tamanho de 40 bytes, que os quadros CTS e ACK tem tamanho de 39 bytes, e que cada pacote de dados pode transportar até 1500 bytes usando um cabeçalho de 47 bytes, verifica-se que a taxa de transmissão efetiva no padrão 802.11 é de aproximadamente 92% da velocidade dos rádios dos dispositivos (LI, BLAKE, DE COUTO *et al*, 2001). Este valor é ainda menor se forem considerados os tempos entre os vários tipos de quadros, previstos no MAC do padrão 802.11, chegando a apenas 85% de eficiência considerando um rádio com taxa de transmissão bruta de 2 Mbps (LI, BLAKE, DE COUTO *et al*, 2001). Além disso, a existência de outros nós na rede com rádios operando a uma taxa de transmissão inferior causa um impacto na taxa de transmissão de todos os nós.

Os níveis três e quatro de uma rede (IP e TCP, por exemplo) também possuem cabeçalhos, contendo campos de controle em cada pacote de dados, que contribuem ainda mais para a diminuição da velocidade efetiva da rede.

3.2.3 Comunicação *multi-hop*

Em uma rede *ad hoc*, cada nó é também um potencial roteador para os pacotes de outras estações. Conseqüentemente, a capacidade para transmissão de dados disponível para cada nó é afetada pelos fluxos de dados de outras estações da rede.

Simulações e experimentos realizados em redes *ad hoc* mostram que a capacidade de uma rede deste tipo é muito inferior à capacidade máxima dos rádios utilizados na rede. Além disso, esta capacidade é inversamente proporcional ao número de nós na rede. Embora seja razoável imaginar que, com o aumento do número de nós e seu distanciamento, a capacidade total da rede deveria aumentar, devido à reutilização espacial do espectro de frequências, isso somente ocorre se o interesse de tráfego entre as estações for localizado, isto é, se não houver transmissão de dados entre estações distantes uma da outra dentro da rede, que demandam um grande número de saltos para que a informação trafegue da origem até seu destino.

Considerando uma rede *ad hoc* como a apresentada na Figura 5 onde os nós estão posicionados em linha, com cada nó a uma distância próxima do alcance de transmissão do outro, e considerando ainda uma transmissão de dados partindo do primeiro nó da linha com destino ao último nó, o maior desempenho possível para esta transmissão seria de 1/3 da velocidade de transmissão disponível em cada nó. (LI, BLAKE, DE COUTO *et al*, 2001). Isto porque os nós precisam cadenciar suas transmissões de forma a evitar colisões. Por exemplo, se o nó 1 está transmitindo, então o nó 2 não pode transmitir ao mesmo tempo, pois não pode

fazer isso enquanto está recebendo dados. O nó 3 também não pode transmitir, pois sua transmissão causaria colisão na recepção do nó 2.

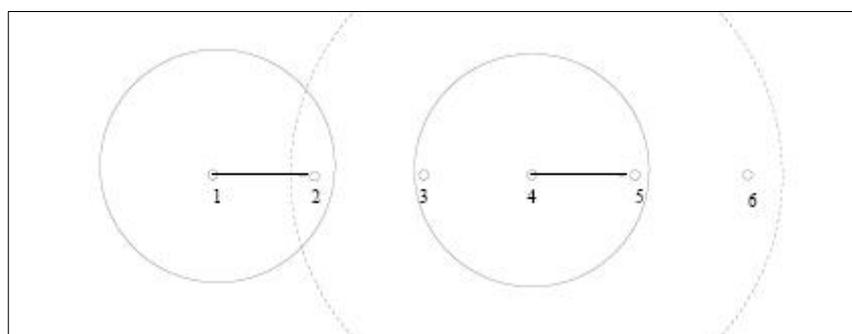


Figura 5: Exemplo de rede sem fio *ad hoc* com topologia em linha

Ainda neste exemplo, sob condições ideais, o nó 4 poderia transmitir ao mesmo tempo que o nó 1. Daí a conclusão de que a capacidade máxima teórica é de $1/3$, pois somente um a cada 3 nós pode transmitir dados simultaneamente.

Porém, se neste exemplo for considerado o alcance de interferência, nem mesmo o nó 4 pode transmitir ao mesmo tempo que o nó 1, uma vez que mesmo com o nó 2 fora de seu alcance de transmissão, o alcance de interferência é maior e a transmissão do nó 4 acaba causando interferência (e colisão) em 2. Assim, a taxa de transmissão acaba sendo de somente $1/4$, pois somente um a cada 4 nós pode transmitir simultaneamente em uma topologia em linha.

Embora seja esperada uma taxa de transmissão de $1/4$ da taxa nominal do equipamento em simulações utilizando topologia em linha, quando o número de saltos entre a origem de dados e o destino aumenta, a capacidade obtida é de cerca de $1/7$. Isso mostra que o controle de acesso ao meio usado no 802.11 não é sempre capaz de atingir a capacidade máxima do canal.

Simulações conduzidas por Li, Blake, de Couto, *et al*, (2001) mostraram que se uma estação enviar dados para a rede de forma cadenciada, controlando a oferta de pacotes de modo a não ultrapassar a taxa de $1/4$ da capacidade nominal da rede, então o 802.11 será capaz de trafegar os dados à mesma taxa que eles são ofertados, atingindo assim a capacidade máxima teórica da rede. No entanto, basta que esta mesma estação passe a ofertar pacotes para a rede a uma maior taxa para que a rede, ao invés de manter o desempenho anterior, passe a demonstrar um desempenho muito inferior. O resultado desta simulação mostra que,

embora o controle de acesso ao meio do 802.11 não consiga obter sempre o melhor desempenho possível, existem situações onde o melhor desempenho pode ser obtido, mesmo sem alterações no controle de acesso. A falha do protocolo está na incapacidade de descobrir e aplicar a taxa de transmissão ideal.

A disponibilidade de uma estação para enviar pacotes depende da concorrência exercida pelas demais estações, uma vez que o meio de transmissão é compartilhado. Em uma topologia em linha, as estações situadas mais ao início e ao fim da rota sofrem uma menor concorrência pelo canal, ao passo que as estações mais centrais têm um nível de concorrência maior. Nesse cenário, as estações do início da rota conseguem então enviar pacotes a uma taxa maior do que as estações mais ao centro podem dar vazão. O resultado disso é um enfileiramento de pacotes nas estações centrais da rota, já que elas têm uma capacidade de transmissão menor devido à concorrência, e o conseqüente descarte de pacotes quando as estações não conseguem uma janela de transmissão em tempo hábil. O congestionamento no centro da rede causa ainda um maior desperdício na ocupação do canal, pois sempre que uma estação não consegue transmitir corretamente um pacote ela aumenta o tempo de *backoff*, muitas vezes desperdiçando valioso tempo no qual o canal poderia estar sendo utilizado.

O descarte de pacotes no interior da rede causa a retransmissão de dados, e mais uma vez os nós iniciais conseguem enviar os dados retransmitidos rapidamente, deixando novamente os nós centrais em desvantagem e contribuindo para o congestionamento do núcleo da rede.

É possível então resumir o comportamento do controle de acesso ao meio como sendo incapaz de alocar a banda de transmissão disponível de forma homogênea entre os nós da rede, alocando sempre mais banda para os nós iniciais. É importante lembrar que, ao contrário das redes sem fio, uma rede tradicional (usando circuitos com fios) sempre é capaz de operar próximo à sua capacidade máxima, mesmo que a oferta de pacotes seja superior à capacidade do circuito.

Outro fator que afeta o desempenho de redes sem fio *multi-hop* é a sincronização dos relógios de cada nó da rede. Quando existe um ponto de acesso coordenando os nós, esse equipamento pode ser a referência comum para os relógios dos nós, que ficam então sincronizados. Porém, se não existe um ponto de acesso, os nós não têm uma referência única para sincronização e seus relógios podem ter pequenas diferenças, o que afeta negativamente a eficiência dos mecanismos de controle de acesso ao meio.

Se o conjunto de protocolos de nível dois do modelo OSI em de uma rede sem fio é incapaz de controlar corretamente o fluxo de pacotes para obter o melhor desempenho, existe

ainda a possibilidade de controlar este fluxo em uma camada mais alta da rede, transferindo para as camadas superiores da rede esta responsabilidade.

Protocolos como o TCP procuram controlar a oferta de pacotes para a rede, na intenção de obter o maior desempenho possível e ao mesmo tempo evitando o congestionamento da rede. O TCP consegue realizar essa tarefa de modo muito eficiente em redes com fio, porém as características que foram levadas em consideração quando o TCP foi projetado eram justamente aqueles de uma rede típica da época, ou seja, uma rede com fio.

Características particulares de redes sem fio como as apresentadas neste capítulo tornam o TCP incapaz de controlar corretamente a oferta de pacotes para a rede. O item a seguir detalha os motivos pelos quais o TCP não possui a mesma eficiência de redes com fio quando usado em uma rede sem fio.

3.3 LIMITAÇÕES DO TCP EM REDES SEM FIO

A larga adoção do TCP como protocolo de transporte em diversas aplicações, tais como navegação *web* (HTTP), transferências de arquivos (FTP), correio eletrônico (SMTP e POP3) e a necessidade de uso destas aplicações em redes de qualquer tipo, tornou o protocolo TCP indispensável e necessariamente utilizado mesmo em redes sem infra-estrutura (HOLLAND e VAIDYA, 1999).

Em redes tradicionais (com infra-estrutura), as mudanças de topologia da rede e falhas em circuitos de transmissão são pouco comuns. Nesse cenário, o TCP já provou ser um protocolo muito robusto, capaz de se adaptar a situações de congestionamento, até mesmo por ter sido desenvolvido para este fim. No entanto, em redes sem fio *ad hoc*, as constantes mudanças na topologia e falhas de transmissão não são corretamente interpretadas pelo TCP padrão.

O TCP não é capaz de diferenciar uma falha na transmissão em um pacote de um congestionamento na rede. Por esse motivo, o desempenho do TCP em redes sem fio *ad hoc* é seriamente comprometido, pois as constantes e temporárias falhas de transmissão, típicas em redes móveis, disparam indevidamente os processos de controle de congestionamento do TCP.

O processo de controle de congestionamento do TCP é baseado em janelas de transmissão e temporizadores de retransmissão. Quando o TCP não recebe a confirmação do recebimento dos dados pelo destino, isto é interpretado como um congestionamento na rede e

a ação do TCP é reduzir o tamanho da janela de transmissão e aumentar o tempo entre as tentativas de reenvio. Isto acontece porque, se existe congestionamento, o melhor a fazer é reduzir a taxa de transmissão até que a situação seja resolvida.

Nas redes sem fio, é comum que os canais de transmissão sofram constantes interrupções, porém tipicamente curtas. Durante estes momentos de interrupção, pacotes de dados ou de confirmação de recebimento podem ser perdidos. Nesta situação o melhor a fazer é simplesmente reenviar o pacote perdido, pois é provável que o canal de comunicação já esteja normalizado quando deste reenvio. Entretanto, este não é o comportamento adotado, uma vez que o TCP não é capaz de detectar a diferença entre congestionamento e falha no canal de comunicação.

Holland e Vaidya (1999) apresentam uma análise utilizando simulação, para a taxa esperada de transferência, que é considerada como sendo a taxa máxima que pode ser obtida em uma transmissão TCP, em função do número de saltos entre origem e destino da transmissão. A Tabela 8 mostra a taxa esperada de um a dez saltos, considerando uma rede com taxa de transmissão bruta de 2Mbps.

Tabela 8: Taxa de transferência esperada para o TCP em redes sem fio

Número de saltos (<i>hops</i>)	Velocidade máxima (Kbps)
1	1352,9
2	671,1
3	446,1
4	269,3
5	181,0
6	164,8
7	150,7
8	144,1
9	139,7
10	139,2

É importante lembrar que os valores da Tabela 8 foram obtidos por simulação e consideram também os *overheads* dos protocolos IP e TCP.

3.4 CONCLUSÃO

Neste capítulo, vários fatores que motivam o baixo desempenho de uma rede sem fio, quando comparado a uma rede com fio, foram apresentados:

- O meio de transmissão é compartilhado entre os vários nós participantes de rede, o que por si só causa uma divisão da banda disponível na rede entre os nós;
- A técnica utilizada no padrão 802.11 para evitar colisões e minimizar o efeito *hidden terminal*, chamada de *virtual carrier sensing*, com uso de pacotes CTS e RTS, implica em um maior volume de pacotes de controle durante a transmissão de dados, e por conseqüência um maior overhead na transmissão;
- Em redes com múltiplos saltos, como o alcance de interferência costuma ser maior que o alcance de transmissão, a técnica *virtual carrier sensing* não é suficiente para evitar colisões;
- O controle de acesso ao meio utilizado no protocolo 802.11 não é capaz de extrair da rede sem fio o maior desempenho possível;
- Problemas de sincronização dos relógios dos nós em redes *multi-hop*.

Além destes fatores, que se aplicam ao protocolo 802.11, o protocolo TCP também sofre degradação no desempenho. Isto é devido à incorreta identificação de falhas pontuais de transmissão (na rede sem fio) como sendo congestionamento.

Embora um panorama geral tenha sido apresentado, vários dos problemas mencionados têm impacto direto sobre o comportamento do tráfego em redes *ad hoc*. Este impacto, por sua vez, afetará o comportamento de protocolos de controle de tráfego como o TCP.

Assim, o capítulo seguinte propõe analisar, à luz do modelo proposto para redes com fio do capítulo 2, os efeitos observados numa rede sem fio *ad hoc*.

CAPÍTULO 4

ANÁLISE DO MODELO TCP MAX-PLUS PARA UMA REDE SEM FIO *AD HOC*

4.1 INTRODUÇÃO

O modelo TCP max-plus proposto por Baccelli e Hong (2000), apresentado no capítulo 2, faz uso de vários roteadores entre a origem e o destino da transmissão TCP. O uso de vários roteadores procura representar o que normalmente acontece em uma transmissão TCP quando origem e destino estão em redes distintas, interconectadas via outras redes, demandando vários passos de roteamento durante o trânsito dos pacotes entre as redes. Esta é a situação típica que ocorre, por exemplo, em uma conexão TCP entre dois computadores na Internet.

Particularmente, topologia semelhante é usada em uma rede sem fio *ad hoc*. Neste tipo de rede, como foi detalhado na subseção 3.2.3, pode ser necessário o uso de vários saltos (*multi-hops*) durante a comunicação entre dois nós participantes da rede.

Como uma conexão TCP entre dois nós de uma rede *ad hoc* apresenta uma topologia semelhante àquela utilizada no modelo TCP max-plus, este capítulo analisa o comportamento do modelo max-plus para uma rede sem fio *ad hoc*, procurando identificar comportamentos possivelmente não modelados, que afetam determinadas redes sem fio.

4.2 METODOLOGIA

Para verificar a aplicabilidade do modelo TCP max-plus em redes *ad hoc*, um exemplo de rede *ad hoc* com múltiplos saltos foi escolhido, e a este exemplo de rede foi aplicado o modelo max-plus para obtenção dos valores de tempos de saída dos pacotes e também uma medida de desempenho da rede.

O mesmo exemplo de rede foi construído em um simulador de redes de computadores (*ns-2*), para que os resultados da simulação pudessem ser posteriormente comparados com os resultados obtidos com o uso do modelo max-plus.

Em resumo, a metodologia utilizada para testar a aplicabilidade do modelo max-plus em redes *ad hoc* foi a comparação de medidas de desempenho obtidas com o modelo max-

plus com medidas de desempenho obtidas por simulação. Esta comparação mostrou-se adequada para redes com fio, conforme apresentado na seção 2.8.

O modelo max-plus para a rede *ad hoc* foi construído com auxílio das ferramentas para *Matlab* desenvolvidas e apresentadas nas seções 2.6 e 2.7.

Por sua vez, a simulação foi realizada usando o simulador *ns-2*, que é um dos simuladores mais utilizados para este tipo de aplicação, especialmente em ambiente acadêmico, por ser um *software* livre e com uma comunidade de desenvolvimento bastante ativa.

Na comparação, duas diferentes medidas foram consideradas. A primeira delas é o tempo de chegada da confirmação de entrega de pacotes (ACK) na origem da conexão TCP. Alguns pacotes, dentre os vários pacotes da transmissão TCP, foram escolhidos para coleta desta medida.

A segunda medida utilizada foi a taxa máxima de transmissão da rede, obtida usando a técnica apresentada na subseção 2.7 e que foi aplicada em uma comparação semelhante na subseção 2.8.

Para montar a topologia de rede no simulador *ns-2*, foi necessário fazer uso de um protocolo de roteamento para redes *ad hoc*. Caso contrário, não seria possível a comunicação TCP entre origem e destino por falta de conectividade da rede.

Ainda sobre a simulação *ns-2*, é importante citar que em todas as simulações executadas os nós foram considerados estáticos. Embora o *ns-2* permita simulações com mobilidade dos nós, não era objetivo deste estudo analisar a mobilidade dos nós da rede e seus efeitos.

4.3 CENÁRIO DE ESTUDO I

A rede escolhida para o primeiro cenário de estudo foi uma rede sem fio *ad hoc* composta por um total de cinco nós, sendo um deles a origem e outro o destino da comunicação TCP, e os três nós remanescentes atuando como roteadores intermediários para os pacotes do fluxo TCP.

Para modelar a rede do primeiro cenário utilizando o modelo TCP max-plus, é necessário conhecer o número de roteadores, definir a latência na transmissão entre cada roteador, definir o tempo de serviço nos roteadores, e determinar uma estratégia para a

evolução do tamanho da janela do TCP. Neste primeiro cenário de estudo, o modelo max-plus utilizado não sofre alterações em relação ao modelo apresentado no capítulo 2.

Para construir a mesma rede no simulador *ns-2*, vários parâmetros são necessários. Para que a simulação possa ser realizada, é necessário definir:

- o protocolo de enlace utilizado na rede sem fio;
- a velocidade da transmissão sem fio;
- o posicionamento dos nós (e conseqüentemente a distância entre eles);
- o tamanho dos pacotes de dados do TCP;
- o tamanho dos pacotes de confirmação de entrega do TCP.

Comparando os parâmetros da simulação com o modelo max-plus, percebe-se que o protocolo de enlace não é relevante no modelo max-plus, bem como o posicionamento dos nós. A velocidade da transmissão sem fio participa na formação do tempo agregado de serviço para o modelo max-plus, bem como o tamanho dos pacotes do TCP.

Para respeitar as premissas do modelo max-plus inicialmente proposto, foi considerado ainda que os *ACKs* do protocolo TCP trafegam por um circuito de comunicação independente, diferente do percurso seguido pelos pacotes de dados.

4.3.1 Modelo TCP max-plus

Para construir o modelo max-plus da rede proposta, foi necessário primeiramente determinar o número de roteadores, o tempo de serviço em cada roteador e a latência na comunicação entre roteadores. Estes são os valores que determinam as características da rede nas equações do modelo max-plus.

O número de roteadores a ser utilizado no modelo max-plus fica evidente após a descrição da topologia da rede.

Como a rede utilizada no cenário é uma rede sem fio, o valor escolhido para a latência entre os roteadores deve representar o tempo de transmissão entre os nós na tecnologia de transmissão sem fio utilizada.

Para possibilitar uma comparação justa entre o modelo max-plus e a simulação *ns-2*, o valor de latência entre os roteadores determinado para o modelo max-plus foi o mesmo valor utilizado pelo *ns-2*. Observou-se que nas transmissões de pacotes de dados do TCP entre dois

nós no simulador *ns-2*, o tempo de transmissão foi de aproximadamente 10ms. Este foi o valor então utilizado como latência entre os roteadores no modelo max-plus.

Da mesma forma, como a velocidade da rede sem fio utilizada na simulação em *ns-2* foi de 2Mbps, e o tamanho dos pacotes de dados foi de 1250 *bytes*, o valor utilizado como tempo agregado de serviço nos roteadores do modelo max-plus foi definido como 5ms. Esse cálculo é realizado da seguinte forma: como cada pacote de 1250 *bytes* possui 10.000 *bits*, em uma taxa de 2Mbps cada pacote necessita $10.000/2M = 5ms$ para ser processado.

Na Tabela 9, é apresentado um resumo dos parâmetros da rede utilizados no modelo max-plus:

Tabela 9: Parâmetros utilizados no modelo max-plus (cenário I).

Parâmetro	Valor
Índice do roteador destino (K)	4
Latência entre roteadores ($d_{0,1}, d_{1,2}, d_{2,3}, d_{3,K}$)	10ms
Tempo agregado de serviço nos roteadores ($\sigma_i(n)$)	5ms
Tempo de retorno do ACK ($d_{K,0}$)	1ms

O valor para o tempo de retorno do ACK foi determinado como sendo 1ms tanto na simulação como no modelo max-plus. Além dos valores para caracterizar a rede, o modelo max-plus requer ainda que a evolução do tamanho da janela TCP seja previamente determinado. Assim sendo, neste estudo foi considerado que a janela TCP inicia com o valor 1, e evolui seguindo o método TCP *Tahoe* sem *slow-start*, mesmo método usado na comparação da seção 2.8.

Como o objetivo final é comparar o modelo max-plus com uma simulação em *ns-2*, ao invés de considerar que a janela do TCP continua crescendo como se não houvesse nenhum congestionamento, foi determinado que a janela do TCP permaneceria com um valor constante assim que fosse detectada a existência de congestionamento utilizando a técnica *Rate Based Loss Detection* apresentada por Baccelli e Hong (2000), e utilizada na seção 2.7. Caso a janela do TCP não fosse limitada, ela continuaria aumentando como se a rede não apresentasse congestionamento em momento algum, o que não é uma consideração realista. Assim, o valor vigente da janela de transmissão do TCP no momento da detecção do congestionamento passa a ser o valor utilizado até o final da transmissão.

Para se determinar este valor de janela vigente no momento de detecção de congestionamento, utilizou-se o seguinte artifício. O modelo foi construído inicialmente sem o limitador para o tamanho da janela. Em seguida foi utilizada a técnica *Rate Based Loss Detection* para determinar o instante de congestionamento e o valor da janela vigente nesse instante, e então o modelo foi alterado usando o limitador encontrado para o tamanho da janela.

Os valores de tempo de saída dos pacotes nos roteadores e também o valor de velocidade máxima na rede foram obtidos utilizando o programa *Matlab*, com auxílio das ferramentas apresentadas na seção 2.6 e 2.7.

A Figura 6 apresenta o programa utilizado para modelar a rede.

```

clear;
K=4; % Número de roteadores
N=300; % Número máximo de pacotes

% Definindo o tempo de serviço
ts = ones(N,K);
ts(:,1)=5;
ts(:,2)=5;
ts(:,3)=5;
ts(:,4)=5;

% Definindo o tempo de propagação entre os roteadores
for n = 1:N;
    d(n,:) = [10,10,10,10];
end;

% Definindo o tempo de retorno do ack
d_ack = ones(N,1);
d_ack(:,1) = 1;

% Definindo evolução do tamanho da janela
% OBS: o índice do vetor já representa Vn-1
v = tcp_v(1,N);

% Calculando passo a passo os tempos de saída:
y = zeros(N,K+1);
for n = 1:N;
    if n == 1
        y(1,1) = 0;
    else
        delta = n - v(1,n);
        if delta < 1 delta = 1; end;

        y(n,1) = mp_multi( y(delta,K+1),d_ack(delta,1) );

    end;
    for i = 2:K+1;
        if n == 1
            saida_pct_ant = 0;
        else
            saida_pct_ant = y(n-1,i);
        end;

        y(n,i) = mp_multi( mp_sum( mp_multi( y(n,i-1),d(n,i-1) ),saida_pct_ant ), ts(n,i-1));

    end;
end;

% Teste de congestionamento
G1 = tcp_G1(y,v,ts,d_ack);

for n=1:N
    if G1(n,1) == 0
        pct_loss = n;
        break;
    end;
end;

ppsmax = pct_loss / y(pct_loss,K+1) * 1000;
kbps = ppsmax * 1250 / 1024;

```

Figura 6: Código *Matlab* para o cenário I

4.3.2 Resultados obtidos com o modelo max-plus

O modelo TCP max-plus implementado no Matlab detectou congestionamento durante a transmissão do pacote de número 92. Neste momento, a janela de transmissão do TCP era de 13 pacotes, e este valor de janela foi utilizado como limitador para o tamanho máximo da janela.

A velocidade determinada utilizando a técnica *Rate Based Loss Detection* foi de 107,35 pacotes por segundo, velocidade atingida durante a transmissão do pacote de número 92, após aproximadamente 850ms do início da transmissão TCP.

O tempo de chegada da confirmação de entrega de alguns pacotes, segundo o modelo max-plus, pode ser observado na Tabela 10.

Tabela 10: Tempo de chegada de ACKs para o modelo max-plus (cenário I).

Pacote	Tempo de chegada do ACK (ms)	Taxa de transmissão (pct/s)
5	188	26,60
10	259	38,61
50	630	79,37
100	898	111,35
250	1648	151,69
500	2898	172,53
1000	5398	185,25

Ainda na Tabela 10, o valor para a taxa de transmissão é uma medida da velocidade média de transmissão da rede desde o início da transmissão até o instante da confirmação de entrega do respectivo pacote apresentado na tabela. O valor é obtido dividindo o número de pacotes enviados até o momento pelo tempo de recepção do *ACK* para o pacote. É possível notar que a taxa de transmissão até a entrega dos primeiros pacotes é inferior, pois no início da transmissão a janela do TCP ainda é pequena.

4.3.3 Simulação da rede usando *ns-2*

Como no modelo TCP max-plus é considerado que as confirmações de entrega dos pacotes de dados do TCP seguem não pelo caminho reverso dos pacotes de dados, mas via um caminho alternativo, foi necessário reproduzir essa premissa na simulação.

Para isso, na topologia da simulação, foi utilizado um circuito ligando o nó de destino do fluxo TCP ao nó de origem. Esse circuito foi configurado de forma a permitir somente comunicação em um sentido (o sentido que os pacotes ACK do TCP devem seguir), evitando assim que os pacotes de dados pudessem utilizar esse circuito para chegar ao destino, além de forçar os pacotes de dados a trafegar pela rede sem fio, nó a nó, até chegar ao destino.

A Figura 7 mostra uma captura de tela do programa NAM (*Network Animator*), programa que faz parte do simulador *ns-2*. Nesta figura, pode-se observar a topologia da rede simulada, incluindo o circuito ligando a origem (nó 0) ao destino (nó 4) da comunicação TCP.

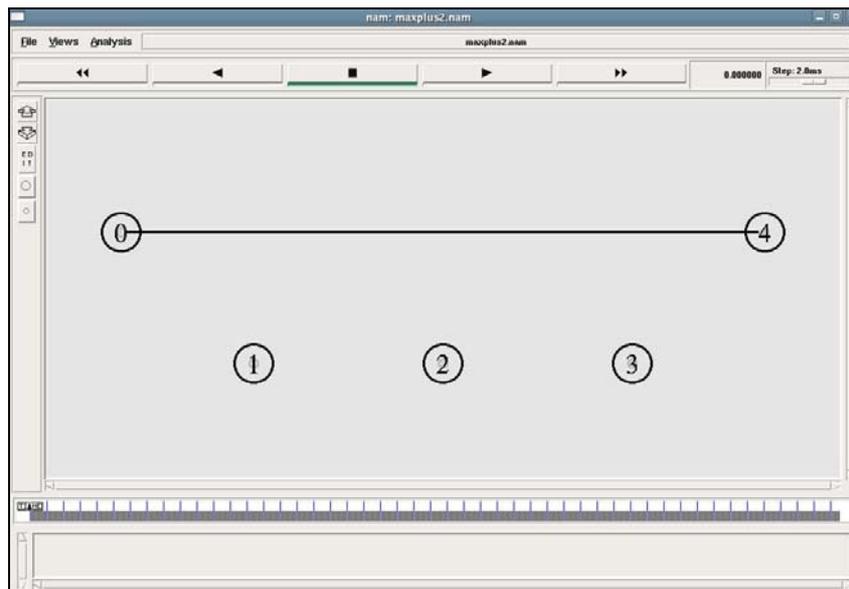


Figura 7: Topologia da rede para a simulação (cenário I).

Essa topologia forçou o caminho seguido pelos pacotes de dados e de confirmação de entrega do TCP de acordo com as premissas do modelo max-plus.

Nesse primeiro cenário de estudo optou-se por construir uma simulação usando valores típicos para a maioria dos parâmetros, conforme mostra a Tabela 11. Foi considerado como taxa de transferência entre os nós o valor de 2 Mbps. Este valor foi escolhido por ser o valor padrão para simulação de redes padrão 802.11 no simulador *ns-2*.

Tabela 11: Parâmetros utilizados na simulação (cenário I)

Parâmetro	Valor
Protocolo de enlace para rede sem fio	802.11
Velocidade de transmissão da rede sem fio	2 Mbps
Protocolo de roteamento	DSDV ⁵
Distância entre nós	250 m
Tamanho do pacote de dados TCP	1210
Tamanho do pacote ACK TCP	40
Tamanho das filas de transmissão	50 pacotes

O programa apresentado na Figura 8 cria a topologia para simulação no *ns-2*, utilizando os parâmetros apresentados na Tabela 11.

⁵ *Destination-Sequenced Distance Vector*

```

set ns [new Simulator]

# Cor dos pacotes (NAM)
$ns color 1 Red

# Setando parâmetros da rede sem fio
set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround
set val(ant) Antenna/OmniAntenna
set val(ll) LL
set val(ifq) Queue/DropTail/PriQueue
set val(ifqlen) 50
set val(netif) Phy/WirelessPhy
set val(mac) Mac/802_11
set val(rp) DSDV
set val(nn) 5
set val(x) 950
set val(y) 300

# Arquivo para o NAM
set nf [open maxplus2.nam w]
$ns namtrace-all-wireless $nf $val(x) $val(y)

# Arquivo de saída (trace-file)
$ns use-newtrace
set tf [open maxplus2.trace w]
$ns trace-all $tf

# Função executada ao fim da simulação

proc finish {} {
    global ns nf tf
    $ns flush-trace
    close $nf
    close $tf
    exec nam maxplus2.nam &
    exit 0
}

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

create-god $val(nn)

set canall [new $val(chan)]

# Definindo padrões para os nós sem fio
$ns node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -topoInstance $topo \
    -wiredRouting ON \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace OFF \
    -channel $canall

# Cria nós
for {set i 0} {$i < $val(nn)} {incr i} {
    set no($i) [$ns node ]
        $no($i) random-motion 0
    }

$ns simplex-link $no(4) $no(0) 1Mb lms DropTail
$ns queue-limit $no(4) $no(0) 2

# Localização dos nós
$no(0) set X_ 50.0
$no(0) set Y_ 225.0
$no(0) set Z_ 0.0

$no(1) set X_ 225.0
$no(1) set Y_ 50.0
$no(1) set Z_ 0.0

$no(2) set X_ 475.0
$no(2) set Y_ 50.0
$no(2) set Z_ 0.0

$no(3) set X_ 725.0
$no(3) set Y_ 50.0
$no(3) set Z_ 0.0

$no(4) set X_ 900.0
$no(4) set Y_ 225.0
$no(4) set Z_ 0.0

for {set i 0} {$i < $val(nn)} {incr i} {
    $ns initial_node_pos $no($i) 50
        $no($i) color gray
    }

# Envio de dados em TCP
set tcp [new Agent/TCP]
$tcp set class_ 1
$tcp set packetSize_ 1210
$ns attach-agent $no(0) $tcp
set sink [new Agent/TCPSink]
$sink set packetSize_ 40
$ns attach-agent $no(4) $sink
$ns connect $tcp $sink
$tcp set fid_ 1

# Envia FTP sobre o TCP
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

# Inicio e fim da simulação
$ns at 100.0 "$ftp start"
$ns at 200.0 "$ftp stop"
$ns at 200.1 "finish"

$ns run

```

Figura 8: Código gerador do cenário I no simulador *ns-2*

O tempo total de simulação foi 200s, sendo que a primeira metade do tempo foi destinada à estabilização do protocolo de roteamento. Desta forma, a transmissão TCP é iniciada somente aos 100s de simulação, quando as rotas já estão estabilizadas, evitando que a comunicação entre os nós para troca de informações do protocolo de roteamento afetasse o desempenho da comunicação TCP.

O resultado da simulação é armazenado em um arquivo de *log* que contém detalhes sobre todos os eventos ocorridos durante a simulação. A partir deste arquivo, um filtro foi implementado para extrair as informações sobre o tempo de recepção dos *ACKs*, uma vez que o objetivo é analisar o comportamento do protocolo TCP.

Com as informações extraídas do arquivo de *log* da simulação, foi possível determinar o desempenho da rede simulada e obter as informações necessárias para comparar o resultado com os obtidos utilizando o modelo max-plus, conforme apresentado na seção seguinte.

4.3.4 Resultados obtidos na simulação *ns-2*

Para evitar que os pacotes transmitidos pelo próprio protocolo de roteamento afetassem o resultado da simulação, os primeiros segundos da simulação foram descartados da análise, de forma que somente foram considerados na comparação os pacotes enviados após a estabilização do protocolo de roteamento. Na prática isso significa dizer que foi considerado como instante de tempo zero, a situação da rede aos 100 segundos de simulação.

A simulação executada no *ns-2* obteve taxas de transferência muito diferentes das taxas determinadas pelo modelo max-plus. No momento da transmissão do pacote de número 1000, a taxa de transmissão média era de 19,58 pacotes por segundo.

Os tempos de chegada da confirmação de entrega obtidos por simulação podem ser observados na Tabela 12.

Tabela 12: Tempo de chegada de ACKs com a simulação (cenário I).

Pacote	Tempo de chegada do ACK (ms)	Taxa de transmissão (pct/s)
5	249	20,08
10	486	20,58
50	2394	20,89
100	4957	20,17
250	12714	19,66
500	25186	19,85
1000	51049	19,58

Note que a Tabela 12 apresenta os resultados de simulação para os mesmos pacotes da Tabela 10, obtida com o modelo max-plus. A comparação destes resultados é apresentada a seguir.

4.3.5 Análise do cenário I

Comparando a taxa de transmissão obtida a partir do modelo max-plus com a taxa obtida por simulação, nota-se uma grande disparidade de valores, conforme pode ser observado na Tabela 13.

Enquanto no modelo max-plus o tempo agregado de serviço nos roteadores foi calculado para representar uma taxa de transmissão de 2Mbps, na simulação a taxa efetiva de transmissão entre dois nós é muito menor que 2Mbps. Isso pode ser explicado pelo compartilhamento da taxa de transmissão entre os vários nós que concorrem pelo meio de transmissão.

Tabela 13: Comparação entre as taxas de transmissão (cenário I)

Pacote	Taxa de transmissão modelo	Taxa de transmissão simulação
	max-plus (pct/s)	(pct/s)
5	26,60	20,08
10	38,61	20,58
50	79,37	20,89
100	111,35	20,17
250	151,69	19,66
500	172,53	19,85
1000	185,25	19,58

O modelo max-plus utilizado no cenário I não considera a concorrência pelo meio de transmissão, e portanto não é capaz de reproduzir o congestionamento imposto pela rede sem fio. Assim, o modelo max-plus na forma como foi apresentado neste cenário modela uma rede sem fio ideal, na qual não existe concorrência pelo uso do canal.

Outra característica não considerada no modelo max-plus usado no cenário I é o impacto sobre a janela de transmissão do TCP causada pelos pacotes descartados pela rede sem fio. Conforme foi apresentado na sub-seção 3.2.1, o protocolo 802.11 descarta o pacote após várias tentativas de reserva do meio de transmissão sem sucesso, ou após várias transmissões sem uma posterior confirmação de entrega.

Esses pacotes descartados são interpretados pelo TCP como congestionamento, causando diminuição no tamanho da janela. Já no modelo max-plus utilizado a janela permanece estável após atingir o maior valor possível que não causa congestionamento.

No cenário II, apresentado na seção a seguir, os parâmetros utilizados no modelo max-plus são alterados na intenção de capturar um comportamento mais próximo ao de uma rede sem fio real.

4.4 CENÁRIO DE ESTUDO II

O objetivo do cenário II é novamente comparar os resultados obtidos pelo modelo max-plus com a simulação. Desta vez, modificando o modelo max-plus para que ele represente melhor as características de uma rede sem fio.

O cenário II utiliza a mesma rede proposta no cenário I. O modelo max-plus modificado será então comparado com a simulação ns-2 já realizada no cenário I, sem alterações.

4.4.1 Modelo TCP max-plus modificado

Como a taxa de transmissão em uma rede sem fio com múltiplos saltos é inferior à taxa nominal dos equipamentos utilizados, devido ao congestionamento imposto pelo compartilhamento do canal e pela interferência entre os nós, esse comportamento deve ser inserido no modelo max-plus usado neste cenário.

O parâmetro do modelo max-plus que modela a taxa de transmissão é o tempo agregado de serviço nos roteadores. No primeiro cenário, esse parâmetro recebeu o valor de 5ms, conforme foi apresentado na seção 4.3.1. No cenário atual, o valor de tempo agregado de serviço foi aumentado para reproduzir a taxa de transmissão inferior imposta pela rede sem fio.

O antigo valor para o tempo agregado de serviço (5ms) foi multiplicado por diferentes valores, e a cada novo valor o modelo foi calculado. Um resultado mais compatível com os valores da simulação foi obtido quando o tempo de serviço nos roteadores foi multiplicado por 3, resultando em um tempo agregado de serviço de 15ms em cada roteador do modelo.

Este valor escolhido como tempo agregado de serviço foi obtido por experimentação. Diferentes valores foram usados, e para cada valor utilizado foram calculados os tempos de saída usando o modelo max-plus. Assim, empiricamente, o valor foi ajustado até encontrar o valor mais compatível.

Além da modificação no tempo agregado de serviço, a evolução da janela do TCP utilizada no modelo também foi alterada. Para reproduzir o efeito dos *timeouts* do TCP, ao invés de estabilizar o tamanho da janela quando esta atingia o valor 13, conforme realizado no cenário I, o valor da janela TCP foi limitado em apenas 2 pacotes no cenário II.

Embora limitar o tamanho da janela em dois pacotes possa parecer uma redução muito drástica, na verdade esse valor é adequado para uma rede sem fio com 4 saltos como a rede proposta nesse capítulo, pois não mais que dois nós podem transmitir ao mesmo tempo nessa topologia. Por exemplo, se o primeiro nó (origem da transmissão TCP) estiver transmitindo, o nó 2 não pode transmitir (por estar recebendo dados), e o nó 3 também não pode transmitir pois sua transmissão causaria colisão na recepção do nó 2. Sendo assim, neste exemplo, somente o nó 4 poderia transmitir ao mesmo tempo que o nó 1, e assim por diante.

A latência entre os roteadores e o tempo de retorno dos pacotes *ACK* do cenário I foram mantidos no cenário II, pois a alteração no tempo agregado de serviço mostrou-se suficiente para capturar o comportamento da rede sem fio. A seguir são apresentados os resultados obtidos.

4.4.2 Resultados obtidos com o modelo max-plus modificado

Como no modelo max-plus modificado o tamanho da janela foi limitado em apenas dois pacotes, não foi possível utilizar a técnica *Rate Based Loss Detection* para detectar o momento de congestionamento da rede, pois o limitado tamanho da janela evitou que a taxa de envio de pacotes para a rede fosse superior à taxa de serviço dos roteadores.

O tempo de chegada de alguns pacotes *ACK*, calculados usando o modelo max-plus modificado, pode ser observado na Tabela 14.

Tabela 14: Tempo de chegada de ACKs com o modelo max-plus modificado (cenário II).

Pacote	Tempo de chegada do ACK (ms)	Taxa de transmissão (pct/s)
5	317	15,77
10	605	16,53
50	2625	19,05
100	5150	19,42
250	12725	19,65
500	25350	19,72
1000	50600	19,76

É interessante notar que a taxa de transmissão do modelo max-plus, neste cenário, obteve um valor bastante estável após a transmissão dos primeiros pacotes. Após o pacote de número 50, a taxa de transmissão manteve-se próxima a 19 pacotes por segundo. Isso mostra que as alterações realizadas nos parâmetros do modelo max-plus são capazes de reproduzir uma situação de congestionamento na rede, limitando a taxa de transmissão.

4.4.3 Análise do cenário II

Enquanto no cenário anterior a diferença de desempenho entre o modelo max-plus e a simulação ns-2 chegou a apresentar diferenças consideráveis, com as mudanças aplicadas nos parâmetros do modelo max-plus neste cenário foram obtidos valores de desempenho bastante compatíveis entre o modelo max-plus modificado e a simulação, conforme pode ser observado na Tabela 15.

Tabela 15: Comparação entre as taxas de transmissão (cenário II).

Pacote	Taxa de transmissão max-plus modificado (pct/s)	Taxa de transmissão simulação (pct/s)
5	15,77	20,08
10	16,53	20,58
50	19,05	20,89
100	19,42	20,17
250	19,65	19,66
500	19,72	19,85
1000	19,76	19,58

Com a evolução da transmissão TCP, os valores para a taxa de transmissão apresentam diferenças da ordem de apenas 1%, o que sugere que o modelo TCP max-plus pode ser aplicado para modelar uma transmissão em TCP sobre uma rede sem fio, desde que seus parâmetros sejam alterados para representar corretamente a taxa de transmissão efetiva de uma rede sem fio e também o impacto sobre a evolução do tamanho da janela do TCP.

Nota-se ainda que, após os instantes iniciais de simulação, a taxa de transmissão sofre uma pequena diminuição, evidenciada nas três últimas linhas da tabela acima. Esse efeito é devido ao congestionamento dos nós centrais da rede, que sofrem maior interferência assim que a janela do TCP aumenta. Como no modelo max-plus a janela do TCP foi limitada em 2 pacotes, os nós centrais não foram sobrecarregados pelo TCP e o mesmo efeito não ocorreu.

4.5 CENÁRIO DE ESTUDO III

Considerando que a grande diferença entre as medidas obtidas no cenário I é fruto principalmente da concorrência pelo uso do canal sem fio (que não estava sendo representada no modelo max-plus) e também das interferências que causam retransmissões e perdas de pacotes, é razoável supor que uma rede sem fio desprovida dos efeitos de concorrência e interferência teria desempenho equivalente ao de uma rede com fio, e, portanto, comparável ao desempenho obtido pelo modelo max-plus original usado no cenário I.

Para testar essa hipótese, a simulação em ns-2 no cenário III é modificada para reduzir a concorrência pelo uso do canal sem fio, assim como procurar eliminar as interferências entre os nós. Em seguida, os resultados obtidos são comparados com os resultados calculados com o modelo max-plus original usado no cenário I.

Assim, o que o cenário III apresenta é o oposto do cenário II: enquanto no cenário II o modelo max-plus é modificado para se adequar a uma rede sem fio, no cenário III a simulação é alterada, mudando inclusive as características da rede proposta, para que a rede tenha um comportamento mais próximo ao de uma rede com fio.

4.5.1 Simulação da rede modificada

Para eliminar a concorrência pelo uso do canal na simulação, os nós foram separados em células com somente dois nós cada, sendo que cada célula está separada da outra por uma distância de 1000 metros para que também não exista interferência entre elas. Assim, os dois efeitos que, por hipótese, seriam os causadores do menor desempenho em uma rede sem fio desaparecem.

A topologia obtida no simulador após as alterações mantém o número de saltos sem fio entre a origem e o destino da transmissão TCP, porém para montar essa topologia foi necessário adicionar nós auxiliares na topologia, conforme mostra a Figura 9.

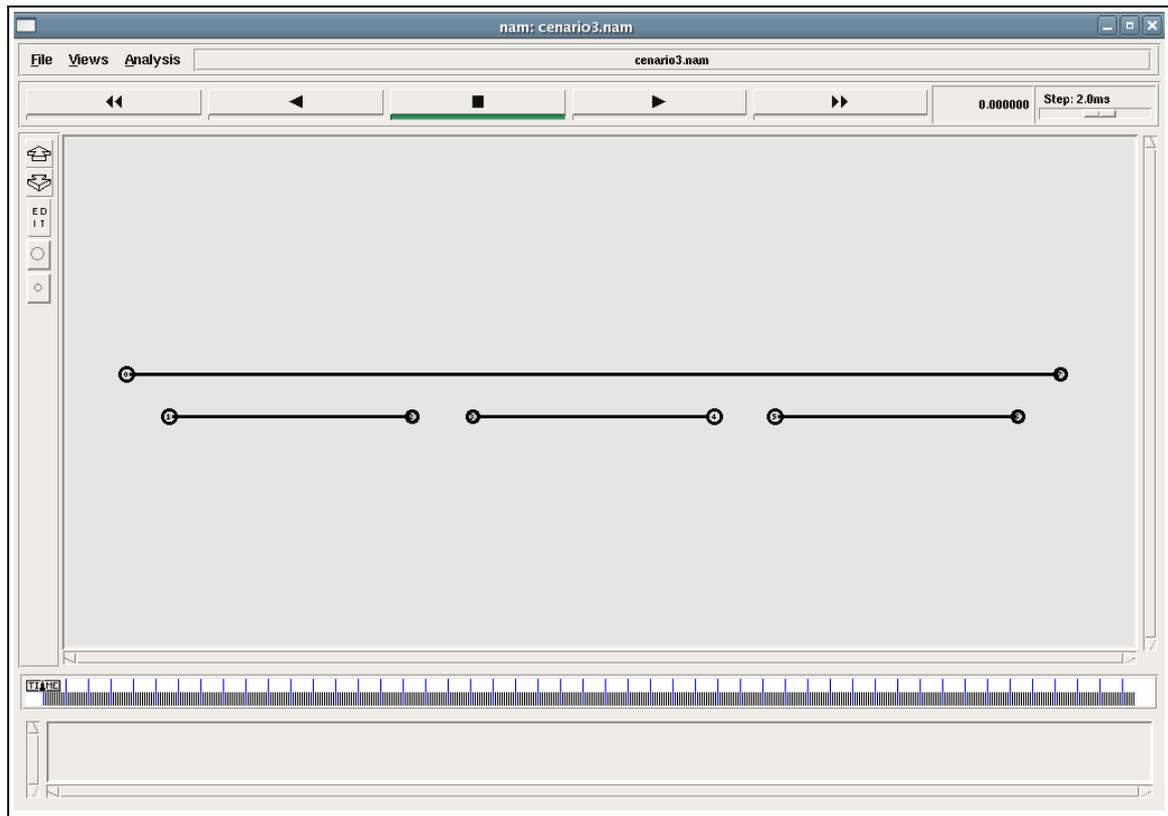


Figura 9: Topologia da rede para a simulação (cenário III).

As células sem fio foram ligadas entre si por um circuito com fio de alta velocidade e baixa latência, para que a transmissão dos pacotes TCP por esse circuito não afetasse o resultado final. Estes circuitos estão representados na Figura 9 pelos três circuitos entre os nós centrais. Desta forma, a intenção é fazer com que o pacote seja transportado entre as células em um tempo pequeno o suficiente para não afetar o resultado da simulação, tal como se os nós de origem e de destino do circuito entre as células representassem de fato o mesmo nó.

Essa topologia mista, usando transmissão com e sem fio, requer o uso de um esquema hierárquico de endereçamento no simulador *ns-2*, onde cada célula é um diferente domínio de endereçamento, com um dos nós fazendo o papel de ponto de acesso entre as células. A Figura 10 apresenta o gerador usado na simulação.

```

set ns [new Simulator]
$ns node-config -addressType hierarchical
AddrParams set domain_num_ 4
lappend cluster_num 1 1 1 1
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 2 2 2 2
AddrParams set nodes_num_ $eilastlevel

# Cor dos pacotes (NAM)
$ns color 1 Red

# Setando parâmetros da rede sem fio
set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround
set val(ant) Antenna/OmniAntenna
set val(ll) LL
set val(ifq) Queue/DropTail/PriQueue
set val(ifqlen) 50
set val(netif) Phy/WirelessPhy
set val(mac) Mac/802_11
set val(rp) DSDV
set val(nn) 8
set val(x) 4000
set val(y) 300

set nf [open cenario3.nam w]
$ns namtrace-all-wireless $nf $val(x) $val(y)
set tf [open cenario3.trace w]
$ns trace-all $tf

proc finish {} {
    global ns nf tf
    $ns flush-trace
    close $nf
    close $tf
    #Execute NAM on the trace file
    exec nam cenario3.nam &
    exit 0
}

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
set canall [new $val(chan)]

# Definindo padrões para os nós
$ns node-config \
    -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -topoInstance $topo \
    -wiredRouting ON \
    -agentTrace OFF \
    -routerTrace OFF \
    -macTrace ON \
    -movementTrace OFF \
    -channel $canall

# Cria nós
set no(0) [$ns node 0.0.1 ]
$no(0) random-motion 0

set bs(0) [$ns node 0.0.0 ]
$bs(0) random-motion 0
$no(0) base-station \
[AddrParams addr2id [$bs(0) node-addr]]

set no(1) [$ns node 1.0.1 ]
$no(1) random-motion 0

set bs(1) [$ns node 1.0.0 ]
$bs(1) random-motion 0
$no(1) base-station \
[AddrParams addr2id [$bs(1) node-addr]]

set no(2) [$ns node 2.0.1 ]
$no(2) random-motion 0

set bs(2) [$ns node 2.0.0 ]
$bs(2) random-motion 0
$no(2) base-station \
[AddrParams addr2id [$bs(2) node-addr]]

set no(3) [$ns node 3.0.1 ]
$no(3) random-motion 0

set bs(3) [$ns node 3.0.0 ]
$bs(3) random-motion 0
$no(3) base-station \
[AddrParams addr2id [$bs(3) node-addr]]

# Criação dos links
$ns simplex-link $bs(0) $no(1) 10Mb 0.001ms DropTail
$ns queue-limit $bs(0) $no(1) 2
$ns simplex-link $bs(1) $no(2) 10Mb 0.001ms DropTail
$ns queue-limit $bs(1) $no(2) 2
$ns simplex-link $bs(2) $no(3) 10Mb 0.001ms DropTail
$ns queue-limit $bs(2) $no(3) 2
$ns simplex-link $bs(3) $no(0) 1Mb 1ms DropTail
$ns queue-limit $bs(3) $no(0) 2

# Localização dos nós
$no(0) set X_ 50.0
$no(0) set Y_ 225.0
$no(0) set Z_ 0.0
$bs(0) set X_ 225.0
$bs(0) set Y_ 50.0
$bs(0) set Z_ 0.0
$no(1) set X_ 1225.0
$no(1) set Y_ 50.0
$no(1) set Z_ 0.0
$bs(1) set X_ 1475.0
$bs(1) set Y_ 50.0
$bs(1) set Z_ 0.0
$no(2) set X_ 2475.0
$no(2) set Y_ 50.0
$no(2) set Z_ 0.0
$bs(2) set X_ 2725.0
$bs(2) set Y_ 50.0
$bs(2) set Z_ 0.0
$no(3) set X_ 3725.0
$no(3) set Y_ 50.0
$no(3) set Z_ 0.0
$bs(3) set X_ 3900.0
$bs(3) set Y_ 225.0
$bs(3) set Z_ 0.0

for {set i 0} {$i < 4} {incr i} {
    $ns initial_node_pos $no($i) 50
    $no($i) color gray
}
for {set i 0} {$i < 4} {incr i} {
    $ns initial_node_pos $bs($i) 50
    $bs($i) color gray
}

# Envio de dados em TCP
set tcp [new Agent/TCP]
$tcp set class_ 1
$tcp set packetSize_ 1210
$ns attach-agent $no(0) $tcp
set sink [new Agent/TCPSink]
$sink set packetSize_ 40
$ns attach-agent $bs(3) $sink
$ns connect $tcp $sink
$tcp set fid_ 1

# Envia FTP sobre o TCP
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

# Inicio e fim da simulação
$ns at 100.0 "$ftp start"
$ns at 200.0 "$ftp stop"
$ns at 200.1 "finish"

$ns run

```

Figura 10: Código gerador do cenário III no simulador *ns-2*.

4.5.2 Resultados obtidos na simulação *ns-2* modificada

A topologia modificada para a simulação obteve um resultado de taxa de transmissão bastante superior à simulação executada no cenário I, porém ainda inferior ao desempenho determinado pelo modelo max-plus do cenário I.

O desempenho da simulação modificada pode ser observado na Tabela 16.

Tabela 16: Tempo de chegada de ACKs com a simulação modificada (cenário III).

Pacote	Tempo de chegada do ACK (ms)	Taxa de transmissão (pct/s)
5	123	40,65
10	182	54,95
50	635	78,74
100	1197	83,54
250	2886	86,63
500	5699	87,73
1000	11330	88,26

4.5.3 Análise do cenário III

Mesmo com as alterações incluídas na simulação com o objetivo de minimizar as interferências e concorrência pelo uso do canal, os valores de taxa de transmissão obtidos não se aproximaram dos valores calculados pelo modelo max-plus do cenário I.

Isso demonstra que, mesmo sem a interferência e com somente dois nós em cada célula para evitar congestionamento na rede sem fio, existem outros fatores envolvidos em uma transmissão sem fio que deterioram o desempenho da rede.

O comparativo com os valores obtidos para a taxa de transmissão pode ser encontrado na Tabela 17.

Um dos motivos pelo qual o desempenho da simulação permaneceu inferior poderia ser o controle do acesso ao meio do protocolo 802.11, que mesmo com somente dois nós utilizando o mesmo canal manteve o uso dos pacotes CTS e RTS, anteriormente descritos na subseção 3.2.2, que são utilizados antes da transmissão dos dados. Como no modelo max-plus, o tempo adicional dispensado por essa troca de pacotes de controle não é considerado,

isto poderia justificar a diferença encontrada. Para verificar essa hipótese, uma simulação foi mais uma vez executada, desta vez adicionando a configuração “Mac/802.11 set RTSThreshold_ 3000”, que tem por objetivo desativar o uso de pacotes de controle RTS e CTS.

Verificou-se então que o uso de pacotes RTS e CTS não causa, neste cenário, uma grande diferença na taxa de transmissão, motivo pelo qual seu uso foi mantido no cenário.

Tabela 17: Comparação entre as taxas de transmissão (cenário III)

Pacote	Taxa de transmissão max-plus (pct/s)	Taxa de transmissão simulação modificada (pct/s)
5	26,60	40,65
10	38,61	54,95
50	79,37	78,74
100	111,35	83,54
250	151,69	86,63
500	172,53	87,73
1000	185,25	88,26

4.6 CONCLUSÃO

O estudo apresentado na seção 2.8, onde se compara o desempenho de uma rede TCP determinado via simulação e via modelo max-plus, mostra que o último obtém valores comparáveis aos obtidos por simulação, e reforça sua aplicabilidade para análise do protocolo TCP quando utilizado sobre uma rede com fio.

Neste capítulo, uma comparação semelhante foi feita em redes sem fio. Em um primeiro momento (cenário I), quando o modelo max-plus foi parametrizado usando valores de tempo de serviço e valores para a evolução da janela do TCP sem considerar as características de desempenho típicas das redes sem fio, os resultados obtidos sugeriam que o modelo não era capaz de representar corretamente o comportamento do TCP em uma rede sem fio.

Porém, quando o modelo TCP max-plus foi alterado, usando desta vez valores adequados para os parâmetros de tempo agregado de serviço nos roteadores e para a evolução

da janela do TCP, o resultado obtido novamente obteve valores semelhantes aos obtidos por simulação.

É interessante ressaltar que, logo após a execução do cenário I, quando se identificou a grande diferença nas medidas de desempenho, sendo necessário alterar os parâmetros do modelo max-plus para formar o cenário II, o parâmetro que modela o tempo de transmissão entre os roteadores foi inicialmente alterado. Entretanto, quando esta alteração foi realizada, notou-se que a alteração não reproduzia corretamente os efeitos da rede sem fio.

Após uma reflexão mais demorada, ficou evidente o motivo pelo qual a primeira alteração não representou corretamente o desempenho de uma rede sem fio. Conforme mostrado no capítulo 3, o compartilhamento do meio de transmissão é que causa a retenção dos pacotes nas filas de saída dos roteadores. Quando um pacote consegue ser enviado após a reserva do meio de transmissão pela camada MAC da rede sem fio, a transmissão de um roteador a outro é rápida. Por esse motivo, o parâmetro que melhor modela a retenção dos pacotes nos roteadores, aguardando pelo momento adequado para o envio, é o tempo agregado de serviço.

Assim, quando os valores para o tempo agregado de serviço foram alterados, o modelo max-plus passou a ter um comportamento mais coerente com o comportamento da rede obtido por simulação. Então, bastou ajustar a evolução da janela do TCP para que o modelo max-plus fosse capaz de obter resultados semelhantes aos obtidos por simulação.

O cenário III, por sua vez, mostrou que a diferença de desempenho de uma rede sem fio não é devido somente ao congestionamento e/ou interferências existentes na rede, mas sugere que existam ainda outros fatores, como o *overhead* inserido pelos pacotes de controle de acesso ao meio.

CAPÍTULO 5

DISCUSSÃO E CONCLUSÕES

5.1 CONCLUSÕES

Esta dissertação apresentou uma análise detalhada do modelo TCP max-plus proposto na literatura quando aplicado a uma rede sem fio. Uma rede sem fio *ad hoc* foi escolhida por apresentar comunicação *multi-hop*, que gera efeitos bem diferentes dos observados em redes com fio. Para permitir uma melhor compreensão das possíveis restrições do modelo quando aplicado a uma rede sem fio, foram explorados no capítulo 3 os desafios enfrentados por este tipo de rede. Com base nesta fundamentação teórica, o capítulo 4 apresentou uma metodologia para analisar o modelo max-plus aplicado a uma rede sem fio.

Além disso, embora a formulação matricial apresentada no capítulo 2 tenha sido utilizada para demonstrar a linearidade do modelo max-plus, esta não é a mais adequada para uma implementação computacional, devido às dimensões das matrizes envolvidas e a característica esparsa dessas matrizes. Mesmo assim, algumas ferramentas foram desenvolvidas para serem utilizadas em pequenos modelos.

Os resultados obtidos mostraram que o modelo TCP max-plus pode ser aplicado a uma rede sem fio, desde que alguns parâmetros do modelo sejam alterados para modelar as restrições da rede. Porém, mais importante do que isso, é a identificação de quais parâmetros devem ser alterados para reproduzir o comportamento do protocolo TCP sobre uma rede sem fio. Notadamente, o tempo de serviço nos roteadores pode ser aumentado de forma a refletir a diminuição da taxa de transmissão na rede causada pela concorrência no acesso ao meio de transmissão, e a evolução da janela do TCP pode ser determinada observando a topologia e o número de nós da rede.

De maneira geral, o modelo max-plus modela apenas comportamentos de sincronização e atraso, conforme visto no capítulo 2. Entretanto, abre a possibilidade de modelar comportamentos normalmente descritos por incertezas através de parâmetros cuja evolução pode ser adequadamente ajustada, conforme mostrado no capítulo 4. Geralmente, nos modelos estocásticos de redes, essas incertezas (falta de informação sobre o comportamento da rede) são modeladas através de distribuições de probabilidade adequadas para as variáveis aleatórias envolvidas. Embora os parâmetros assim obtidos possam ser

variantes no tempo, uma função de variação bem definida pode ser proposta, obtendo-se, neste caso, um modelo max-plus linear variante no tempo.

5.2 TRABALHOS FUTUROS

O modelo max-plus explorado neste estudo tem como objetivo representar as características do protocolo TCP, especialmente em relação ao controle de fluxo dos dados, realizado pela variação da janela de transmissão do TCP. Por ter seu foco no protocolo TCP, o modelo utiliza uma representação bastante genérica para a rede pela qual o protocolo trafega. Essa representação genérica não traz prejuízo para a modelagem das características do TCP, porém oculta detalhes interessantes sobre o funcionamento da rede.

As características da álgebra max-plus, adequadas para a modelagem de situações de sincronização e de atrasos, sugerem a possibilidade de sua aplicação para modelar detalhes sobre o funcionamento de uma rede sem fio, em especial do que diz respeito ao controle de acesso ao meio. Desta forma, uma sugestão de trabalho futuro seria utilizar a álgebra max-plus para propor um modelo do controle de acesso ao meio de uma rede sem fio. Este modelo poderia inclusive ser incorporado ao modelo TCP max-plus, substituindo a representação genérica da rede pela nova representação detalhada de uma rede sem fio.

Enquanto no modelo TCP max-plus a evolução dos pacotes pela rede é representada basicamente pelos tempos de entrada e de saída dos pacotes nos roteadores, na representação detalhada da rede sem fio poderiam ser exploradas outras características como o tempo de espera até que o meio de transmissão se torne disponível (que aparentemente é um problema de sincronização), e tempos dispendidos com pacotes de controle do tipo CTS/RTS (que poderia ser modelado como um atraso).

REFERÊNCIAS BIBLIOGRÁFICAS

- ANSI/IEEE Std 802.11. **Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)**, IEEE Computer Society, 1999. Disponível em: <http://standards.ieee.org/getieee802>
- BACCELLI, F., COHEN, G., OLSDER, G., QUADRAT, J. **Synchronization and Linearity: An Algebra for Discrete Event Systems**. Chichester, England: John Wiley & Sons, Inc., 1992.
- BACCELLI, F., HONG, D. TCP is Max-Plus Linear and what it tells us on its throughput. **Proceedings of ACM SIGCOMM'00**, p. 219-230, Stockholm, Sweden, 2000.
- COHEN, G., GAUBERT, S, QUADRAT, J. Max-plus algebra and system theory: where we are and where to go now. **Annual Reviews in Control**, v. 23, n. 1, p. 207-219, 1999.
- COMER, D. **Internetworking with TCP/IP - Vol. I**, 2ª. ed. New Jersey: Prentice-Hall, 1991.
- FALL, K., VARADHAN, K. **The ns Manual**. 2007. Disponível em: <http://www.isi.edu/nsnam/ns/ns-documentation.html>. Acessado em 10/01/2007.
- GEIER, J. 802.11 Medium Access Methods. 2002. Disponível em: <http://www.wi-fiplanet.com/tutorials/article.php/1548381>. Acessado em 12/01/2007.
- HOLLAND, G., VAIDYA, N. Analysis of TCP Performance over Mobile Ad Hoc Networks. **Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking**, p. 219-230, ACM Press, New York, 1999.
- LAKSHMAN, T., MADHOW, U. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. **IEEE/ACM Transactions on Networking**, v. 5 n. 3, p. 336-350, 1997.
- LE BOUDEC, J. Y.; THIRAN, P. Network Calculus: A Theory of Deterministic Queuing Systems for the Internet. **Lecture Notes in Computer Science**, n. 2050, Springer-Verlag New York, LLC, 2001.

- LI, J., BLAKE, C., DE COUTO, D. S. J., LEE, H. I., MORRIS, R. Capacity of Ad Hoc wireless networks. **Proceedings of the 7th annual international conference on Mobile computing and networking**, p. 61-69, ACM Press, New York, 2001.
- MATHIS, M., SEMKE, J., MAHDAVI, J., OTT, T. The macroscopic behavior of the TCP Congestion Avoidance algorithm. **Computer Communications Review**, v.27, n.3, 1997.
- TANENBAUM, A. **Computer Networks**, 3^a. ed. New Jersey: Prentice-Hall, 1996.
- PADHYE, J., FIROIU, V., TOWSLEY, D. A stochastic model of TCP Reno congestion avoidance and control. **CMPSCI Technical Report**, n. 99-02, University of Massachusetts, 1999.
- QIAO, D., SHIN, K. G. UMAV: A Simple Enhancement to the IEEE 802.11 DCF. **Proceedings of the 36th Annual Hawaii International Conference on System Sciences**, v.9, p.306, IEEE Computer Society, 2003.
- STAŃCZYK, J. **Max-Plus Algebra Toolbox for Matlab[®] and GNU Octave**, v. 1.5. Otto-von-Guericke Universität Magdeburg, 2005.
- XU, K., GERLA, M., BAE, S. How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks? **Global Telecommunications Conference. GLOBECOM '02**, v.1, p. 72–76, IEEE Computer Society, 2002.

ANEXO 1**CÓDIGOS-FONTE DAS FUNÇÕES *MATLAB***

Função tcp_Z:

```
function result = tcp_Z(y,n,v)

w=max(v(1,1:n+1));
Z = [];

for f=n:-1:(n-w+1);
    if f < 1 f = 1; end;
    Z = [ Z , y(f,2:size(y,2)) ];
end;

result = Z.';
```

Função tcp_D:

```
function D = tcp_D(n,K,v)

w=max(v(1,1:n));
D = mp_zeros(K*w,K*w);

for i=1:K*(w-1)
    D(K+i,i)= mp_one;
end;
```

Função tcp_M:

```

function M = tcp_M(n,K,ts,d)

M = mp_zeros(K);
for i=1:K
    for j=1:K
        if i>=j
            x = 0;
            for k=j:i
                x = x + ts(n,k);
            end;
            for k=j:i-1
                x = x + d(n,k+1);
            end;
            M(i,j) = x;
        end;
    end;
end;

```

Função tcp_Ml:

```

function Ml = tcp_Ml(n,K,ts,d,d_ack)

Ml = mp_zeros(K);
for i=1:K
    x = 0;
    for k=1:i
        x = x + d(n,k) + ts(n,k);
    end;
    x = x + d_ack(n);
    Ml(i,K) = x;
end;

```

Função tcp_A:

```
function A = tcp_A(a,n,v,K,ts,d,d_ack)

w=max(v(1,1:n));

A = mp_zeros(K*w,K*w);

if a == 1
    A(1:K,1:K) = mp_sum (tcp_M(n,K,ts,d) , tcp_Ml(n,K,ts,d,d_ack) );
else
    A(1:K,1:K) = tcp_M(n,K,ts,d);
    A( 1:K , (a-1)*K+1 :(a-1)*K+K ) = tcp_Ml(n,K,ts,d,d_ack);
end;

A = mp_sum(A, tcp_D(n,K,v));
```

Função tcp_G1:

```
function G1 = tcp_G1(y,v,ts,d_ack)

ts_max = max(ts,[],2);
K = size(y,2);

S(:,1) = y(:,K)-y(:,1)+d_ack(:,1);

G1 = (v'./S) - (1./ts_max);
```

Função tcp_v:

```
function v = tcp_v(tipo,N,limite)

v = ones(1,N);
if tipo == 1 % TAHOE sem slow-start
    jan = 1;
    cont = 1;
    for n=1:N
        v(1,n) = jan;
        if cont > jan
            cont = 0;
            jan = jan +1;
        end;
        cont = cont+1;
    end;
end;
if tipo == 2
    jan = 1;
    cont = 1;
    for n=1:N
        v(1,n) = jan;
        if cont == 2
            cont = 0;
            jan = jan +1;
        end;
        cont = cont+1;
    end;
end;
if tipo == 3 % TAHOE sem slow-start limitado
    jan = 1;
    cont = 1;
    for n=1:N
        if jan == limite
            v(1,n) = limite;
        else
            v(1,n) = jan;
            if cont > jan
                cont = 0;
                jan = jan +1;
            end;
            cont = cont+1;
        end;
    end;
end;
end;
```


RESUMO:

Este trabalho propõe a aplicação de um modelo para o protocolo de transporte TCP, utilizando a álgebra max-plus, em redes sem fio. A abordagem determinística e até mesmo formal do modelo TCP max-plus é pouco tradicional, sendo que na literatura é mais comum encontrar métodos estocásticos para modelagem de redes. Neste estudo, são apresentados um breve resumo do protocolo TCP e da álgebra max-plus, seguido de uma descrição do modelo TCP max-plus. Um resumo dos fatores que afetam o desempenho de redes sem fio e do protocolo TCP sobre este tipo de rede é também apresentado. Uma metodologia para verificar a aplicabilidade do modelo em redes sem fio é proposta, comparando taxas de transmissão obtidas a partir do modelo max-plus e por simulação. O modelo TCP max-plus utilizado mostrou-se capaz de reproduzir o desempenho de redes sem fio, quando seus parâmetros são corretamente definidos de forma a inserir no modelo os efeitos da rede sem fio sobre a evolução da janela do TCP e na retenção dos pacotes durante a disputa pelo uso do meio de transmissão. A alteração nos parâmetros do modelo TCP max-plus não altera sua formulação original, mantendo a linearidade do modelo. Ainda neste estudo, um conjunto de ferramentas computacionais que implementam o modelo TCP max-plus foi construído.

PALAVRAS-CHAVE

Redes sem Fio, Álgebra Max-plus, Desempenho de Redes.

ÁREA/SUB-ÁREA DE CONHECIMENTO

1.03.04.04 - 5	Teleinformática
1.03.03.00 - 6	Modelos Analíticos e de Simulação
1.03.04.00 - 2	Sistemas de Computação

2007

Nº: 441

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)