



DISSERTAÇÃO DE MESTRADO

**O Protocolo SCTP e suas variações: uma avaliação
da funcionalidade de *multi-homing* com base em
redes WLAN**

Helder Cleber Almeida Pereira

Brasília, setembro de 2010

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

**O Protocolo SCTP e suas variações: uma avaliação
da funcionalidade de *multi-homing* com base em
redes WLAN**

Helder Cleber Almeida Pereira

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Mestre em Engenharia Elétrica*

Banca Examinadora

Prof. Dr. Paulo R. de L. Gondim, ENE/UnB _____
Orientador

Prof. Dr. Georges Amvame Nze, Redes/UnB _____
Examinador interno

Prof. Dr. Sidney Cerqueira Bispo dos Santos _____
Examinador externo

FICHA CATALOGRÁFICA

PEREIRA, HELDER CLEBER ALMEIDA

Protocolo SCTP e suas variações: uma avaliação da funcionalidade de multi-homing com base em redes WLAN
2010 xv, 130p., 210x297 mm

(ENE/FT/UnB, Mestre, Engenharia Elétrica, 2010)

Dissertação de Mestrado - Universidade de Brasília

Faculdade de Tecnologia - Departamento de Engenharia Elétrica

1. Protocolo SCTP

2. Multihoming

3. Handover

4. proativo adaptativo

REFERÊNCIA BIBLIOGRÁFICA

PEREIRA, H. C. A. de (2010) Protocolo SCTP e suas variações: uma avaliação da funcionalidade de multi-homing com base em redes WLAN. Dissertação de Mestrado em Engenharia Elétrica, Publicação 430/2010, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 130p.

CESSÃO DE DIREITOS

AUTOR: Helder Cleber Almeida Pereira

TÍTULO: Protocolo SCTP e suas variações: uma avaliação da funcionalidade de multi-homing com base em redes WLAN.

GRAU: Mestre ANO: 2010

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor se reserva a outros direitos de publicação e nenhuma parte desta dissertação de Mestrado pode ser reproduzida sem a autorização por escrito do autor.

Helder Cleber Almeida Pereira

Agradecimentos

Agradeço, primeiramente a Deus, por ter me permitido chegar até aqui, agradeço a Nossa Senhora que me acompanha em minha caminhada, me orienta, me dá forças e me educa para saber ouvir aquele que é Pai, Filho e Espírito Santo.

Agradeço de maneira muito especial à minha querida esposa, que amo muito, que abriu mão das suas prioridades para fazer suas as minhas prioridades. A você, Valéria, o meu amor de servidão, de doação, de companheirismo e de fidelidade.

A você, meu filho João Vitor (em memória), que nos trouxe felicidade desde o dia em descobrimos que estava no ventre da sua mãe. É uma pena que não esteja conosco para ver a alegria do seu pai, mas você sabe como contribuiu para que eu aqui chegasse. Beijos, meu filho querido e amado, estamos com saudades.

Agradeço e dedico essa dissertação aos meus queridos pais, irmãos, sobrinhos e tios. Em especial a você minha mãe, que vive muito para a sua família.

Não poderia deixar de agradecer à minha tia e madrinha Cici, que, mesmo lhe causando desconforto, me acolheu em sua casa do primeiro dia do mestrado até o término. Obrigado, minha tia querida, essa minha conquista também é sua.

Meu tio Caiô e tia Marcia e seus filhos, obrigado pelas vezes que me acolheram, me proporcionaram lanches variados, além de almoços deliciosos, fora a companhia agradável durante esses dois anos de convivência.

Agradeço ao IFTO-Palmas, especificamente à minha coordenação de informática por ter aprovado o meu afastamento para cursar o mestrado.

Obrigado ao professor Paulo Gondim, que me deu a oportunidade de ingressar no mestrado e por contribuir de forma significativa na minha formação acadêmica.

Não poderia de deixar de agradecer aos meus amigos de estudo, de convivência, Cláudio, Manoel, Vinícius, Vanice e Vaguete. Quero agradecer de forma especial a você Claudio de Castro Monteiro, pois me incentivou a ingressar no mestrado.

E, por fim, agradeço às pessoas que me ajudaram no decorrer desta maratona, Wandenberg, Adriana, Wesley e Diego, obrigado meus amigos, sempre que precisava de alguma coisa vocês prontamente me ajudavam.

RESUMO

O presente trabalho, inicialmente, descreve o protocolo SCTP ("Stream Control Transmission Protocol") e algumas de suas variações, e em seguida apresenta um estudo para obter um maior ganho de desempenho no emprego desse protocolo quando em *multi-homing*, beneficiando de maneira direta a sua utilização por aplicações, através da proposta de agregação de novos mecanismos àquele protocolo, tornando-o mais proativo no acompanhamento do estado dos diferentes recursos de transmissão disponibilizados pelo protocolo e na adaptação ao uso do caminho que melhor atenda à necessidade das aplicações, seja em cenário estático ou dinâmico em relação à movimentação das partes envolvidas na associação SCTP.

A solução proposta, aqui chamada de PA-SCTP (Proativo Adaptativo SCTP), busca fazer uso do *multi-homing* de forma mais adaptativa e proativa às necessidades das aplicações. Para demonstrar o ganho de desempenho com o uso do PA-SCTP, gráficos relativos a atraso, atraso médio, variação da janela de transmissão (cwnd) e vazão foram utilizados de forma comparativa com os gráficos gerados pelo SCTP-Padrão (SCTP com valores de parâmetros configuráveis sugeridos pela RFC 4960), SCTP-Parametrizado (SCTP com valores de parâmetros configuráveis mais sensíveis à condição do meio de transmissão) e RSCTP (solução proposta por Kim et al. [1] que busca otimizar o protocolo SCTP nas decisões de troca do caminho de transmissão dos pacotes de dados).

O PA-SCTP funciona como um módulo auxiliar em uma comunicação fim-a-fim SCTP, que computa informações referentes a paradas no recebimento de pacotes, picos de atraso, atraso médio e variação da janela de transmissão (CWND). Por meio desse acompanhamento da comunicação por parte do cliente, o PA-SCTP ditará o melhor momento de fazer a troca do caminho de transmissão, ou seja, menos danoso para a aplicação.

Abstract

This work, initially, describes the Stream Control Transmission Protocol (SCTP) and some variations. On the continuation, it presents a study to obtain a higher gain in performance with multi-homing in the utilization of the mentioned protocol, benefiting directly to the applications, considering the aggregation of new mechanisms to the SCTP protocol, making it more proactive in following, in the most active modes, the status of the different transmission resources available by the protocol and adaptive to the use of the path that best meets the application needs, whether in dynamic or static scenario in relation for the movement of the parties involved in the SCTP association.

The solution proposed, which is here called PA-SCTP (Proactive Adaptive SCTP), aims to make use of multi-homing, considering the needs of applications with an adaptive and proactive behaviour. To demonstrate the gain of performance with the use of the PA-SCTP, graphics related to the delay, the average delay, the variation of the transmission window (CWND) and flow rate were used in a comparative way with the graphics generated by Default-SCTP (SCTP with values configurable parameters suggested by RFC 4960), SCTP-Parametrized (SCTP with values of configurable parameters most sensitive to the condition of the way transmission) and R-SCTP (solution proposed by Kim et al. [1] which aims to optimize the protocol SCTP in decisions to change the transmission path of data packets.

The PA-SCTP operates as an auxiliary module in SCTP-based end-to-end communication, which computes the information referring to the stop of receiving packets, peak delay, average delay and variation of the transmission window (CWND) . Through this monitoring of the communication by the client, the PA-SCTP will dictate the best time to make the switch the transmission path, that is, less damage to the application.

SUMÁRIO

ABSTRACT	III
1 INTRODUÇÃO	1
1.1 MOTIVAÇÃO E PROBLEMA	2
1.2 OBJETIVOS.....	3
1.2.1 GERAL.....	3
1.2.2 ESPECÍFICOS.....	3
1.3 ABORDAGEM	3
1.4 PROPOSTA DE TRABALHO.....	4
1.5 ORGANIZAÇÃO.....	5
2 SCTP - STREAM CONTROL TRANSMISSION PROTOCOL.....	6
2.1 VISÃO GERAL DO PROTOCOLO SCTP.....	6
2.2 CRONOLOGIA DO PROTOCOLO SCTP	8
2.3 BREVE COMPARATIVO: SCTP E TCP	9
2.4 ASSOCIAÇÃO	10
2.5 MULTI-HOMING NO SCTP.....	11
2.6 MULTISTREAMING NO SCTP	12
2.7 PACOTE SCTP.....	12
2.8 FRAGMENTAÇÃO NO SCTP.....	14
2.9 CARGA ÚTIL DO SCTP.....	14
2.10 DETALHAMENTO DO MULTI-HOMING NO SCTP	15
2.11 DETALHAMENTO DO MULTISTREAMING NO SCTP	16
2.12 ENVIO E RECEBIMENTO DE CHUNKS <i>SACKs</i>	17
2.13 GERENCIAMENTO DE RETRANSMISSÕES	19
2.14 CONTROLE DE CONGESTIONAMENTO DO SCTP.....	20
2.14.1 SLOW-START.....	20
2.14.2 CONGESTION AVOIDANCE	21
2.14.3 COMPORTAMENTO DO SCTP EM AMBIENTES DE PERDA DE PACOTE	22
2.15 DETECÇÃO DE CAMINHOS FALHOS.....	22
2.16 VALORES DE PARÂMETROS CONFIGURÁVEIS DO SCTP	23
2.17 COMPARAÇÃO ENTRE PROTOCOLOS DA CAMADA DE TRANSPORTE	24

3	MSCTP - MOBILE STREAM CONTROL TRANSMISSION PROTOCOL	25
3.1	VISÃO GERAL DO MSCTP	25
3.2	TRABALHANDO COM IPS DINÂMICOS EM UMA ASSOCIAÇÃO	26
3.3	NOVAS ESTRUTURAS TLVS DO MSCTP	27
3.4	SCTP E MSCTP MULTI-HOMINGS	28
3.5	GERENCIAMENTO DE MOBILIDADE COM O SCTP-MSCTP	29
4	HANDOVER NO SCTP	32
4.1	POLÍTICA DE TRATAMENTO DE HANDOVER NO SCTP	32
4.2	PARÂMETROS SCTP QUE INFLUENCIAM NA POLÍTICA HANDOVER	36
5	TRABALHOS RELACIONADOS	37
5.1	ORGANIZAÇÃO	37
5.1.1	ARTIGO 1 - HANDOVER BASEADO NA LARGURA DE BANDA	38
5.1.2	ARTIGO 2 - HANDOVER BASEADO NA LARGURA DE BANDA	40
5.1.3	ARTIGO 3 - HANDOVER BASEADO NA LARGURA DE BANDA	42
5.1.4	ARTIGO 4 - HANDOVER BASEADO EM MEDIDAS DE ATRASO	43
5.1.5	ARTIGO 5 - HANDOVER BASEADO EM MEDIDAS DE ATRASO	44
5.1.6	ARTIGO 6 - HANDOVER BASEADO EM PERDA DE CONECTIVIDADE	45
5.1.7	ARTIGO 7 - HANDOVER BASEADO EM PERDA DE CONECTIVIDADE	45
5.1.8	ARTIGO 8 - HANDOVER BASEADO EM PERDA DE CONECTIVIDADE	47
5.1.9	ARTIGO 9 - HANDOVER BASEADO EM MEDIDAS DE SRTT	47
5.1.10	ARTIGO 10 - HANDOVER BASEADO EM MEDIDAS DE RTT	48
6	SCTP PROATIVO ADAPTATIVO (PA-SCTP)	51
6.1	TRATAMENTO PROATIVO ADAPTATIVO DO HANDOVER COM O PA-SCTP	52
6.2	CONTEXTUALIZAÇÃO DA PROBLEMÁTICA DA POLÍTICA DE HAN- DOVER DO SCTP	53
6.3	SOLUÇÃO PROPOSTA PA-SCTP	55
6.3.1	CUSTO DA ALTERAÇÃO DO DESTINO PRIMÁRIO	56
6.3.2	PA-SCTP - LADO TRANSMISSOR DOS PACOTES DE DADOS SCTP	57
6.3.3	PA-SCTP - LADO RECEPTOR DOS PACOTES DE DADOS SCTP	59
6.4	APRESENTAÇÃO DO TESTBED	65
6.5	DESCRIÇÃO DOS EXPERIMENTOS	68
6.6	EXPERIMENTO 1 - CONDIÇÃO 1	71
6.6.1	COM RELAÇÃO À VARIAÇÃO DA JANELA DE TRANSMISSÃO CWND (GRÁFICOS DA FIGURA 6.7)	71
6.6.2	CONCLUSÃO DO EXPERIMENTO	75
6.7	EXPERIMENTO 2 - CONDIÇÃO 2	75
6.7.1	COM RELAÇÃO AO ATRASO E AO ATRASO MÉDIO (GRÁFICOS DAS FIG- URAS 6.8 E 6.9)	76

6.7.2	COM RELAÇÃO À VARIAÇÃO DA JANELA DE TRANSMISSÃO CWND (GRÁFICOS DA FIGURA 6.10)	80
6.7.3	COM RELAÇÃO À VAZÃO (GRÁFICOS DA FIGURA 6.11)	82
6.7.4	CONCLUSÃO DO EXPERIMENTO	84
6.8	EXPERIMENTO 3 - CONDIÇÃO 3	85
6.8.1	COM RELAÇÃO AO ATRASO E AO ATRASO MÉDIO (GRÁFICOS DAS FIG- URAS 6.12 E 6.13)	86
6.8.2	COM RELAÇÃO À VARIAÇÃO DA JANELA DE TRANSMISSÃO CWND (GRÁFICOS DA FIGURA 6.14)	89
6.8.3	COM RELAÇÃO À VAZÃO (GRÁFICOS DA FIGURA 6.15)	92
6.8.4	CONCLUSÃO DO EXPERIMENTO	94
6.9	EXPERIMENTO 4 - CONDIÇÃO 4	94
6.9.1	COM RELAÇÃO AO ATRASO E AO ATRASO MÉDIO (GRÁFICOS DAS FIG- URAS 6.16 E 6.17)	95
6.9.2	COM RELAÇÃO À VARIAÇÃO DA JANELA DE TRANSMISSÃO CWND (GRÁFICOS DA FIGURA 6.18)	99
6.9.3	COM RELAÇÃO À VAZÃO (GRÁFICOS DA FIGURA 6.19)	101
6.9.4	CONCLUSÃO DO EXPERIMENTO	103
6.10	EXPERIMENTO 5 - CONDIÇÃO 5	103
6.10.1	COM RELAÇÃO AO ATRASO E AO ATRASO MÉDIO (GRÁFICOS DAS FIG- URAS 6.20 E 6.21)	104
6.10.2	COM RELAÇÃO À VARIAÇÃO DA JANELA DE TRANSMISSÃO CWND (GRÁFICOS DA FIGURA 6.22)	108
6.10.3	COM RELAÇÃO À VAZÃO (GRÁFICOS DA FIGURA 6.23)	110
6.10.4	CONCLUSÃO DO EXPERIMENTO	112
7	CONCLUSÃO	113
	REFERÊNCIAS BIBLIOGRÁFICAS	115
	REFERÊNCIAS BIBLIOGRAFICAS	115

LISTA DE FIGURAS

2.1	Localização do SCTP na pilha TCP/IP, referenciado em Costa [2] e Stewart [3]	7
2.2	RFCs do SCTP e suas atualizações	8
2.3	RFCs que estendem as possibilidades de uso do SCTP	9
2.4	Tratamento dado à mensagem do usuário, conforme Stalvig [4]	10
2.5	Conexão TCP e a associação SCTP, conforme Stalvig [4]	10
2.6	Encerramento da conexão TCP e associação SCTP, conforme Stalvig [4]	11
2.7	Modelo conceitual de associação, conforme Costa [2].....	13
2.8	Pacote SCTP.....	13
2.9	Pacote SCTP que tem um <i>chunk</i> genérico como carga útil.....	15
3.1	Representação genérica de um pacote SCTP com um <i>Chunk ASCONF/ASCONF-ACK</i>	27
3.2	Gerenciamento de mobilidade usando o mSCTP.....	31
4.1	Interligação entre duas estações SCTP	33
4.2	Tratamento do SCTP para realizar o <i>handover</i> quando PMR=5, RTO.Min=1s, RTO.Max=60s e RTO.Atual=1s	35
6.1	Detalhamento da transmissão de pacotes SCTP e PA-SCTP	58
6.2	Detalhamento da recepção de pacotes SCTP e PA-SCTP	59
6.3	Fluxo de execução no módulo Calc_Atraso_Medio	60
6.4	Fluxo de execução no módulo Contabilizar_Tempo_Espera	61
6.5	Fluxo de execução no módulo Disparar_Sondagem.....	63
6.6	<i>Testbed</i> utilizado para realizar os experimentos.....	66
6.7	Variação da CWND no transmissor referente a arquivos de 15MB e 41MB.....	74
6.8	Atrasos durante o recebimento de arquivos de 15 e 41MB.....	78
6.9	Atrasos médios durante o recebimento de arquivos de 15 e 41MB	79
6.10	Variação da CWND no transmissor referente a arquivos de 15MB e 41MB.....	81
6.11	Vazão na ótica do receptor referente a arquivos de 15MB e 41MB	83
6.12	Atrasos durante o recebimento de arquivos de 15 e 41MB.....	87
6.13	Atrasos médios durante o recebimento de arquivos de 15 e 41MB	88
6.14	Variação da CWND no transmissor referente a arquivos de 15MB e 41MB.....	91
6.15	Vazão na ótica do receptor referente a arquivos de 15MB e 41MB	93
6.16	Atrasos durante o recebimento de arquivos de 15 e 41MB.....	97

6.17	Atrasos médios durante o recebimento de arquivos de 15 e 41MB	98
6.18	Variação da CWND no transmissor referente a arquivos de 15MB e 41MB.....	100
6.19	Vazão na ótica do receptor referente a arquivos de 15MB e 41MB	102
6.20	Atrasos durante o recebimento de arquivos de 15 e 41MB.....	106
6.21	Atrasos médios durante o recebimento de arquivos de 15 e 41MB	107
6.22	Variação da CWND no transmissor referente a arquivos de 15MB e 41MB.....	109
6.23	Vazão na ótica do receptor referente a arquivos de 15MB e 41MB	111

LISTA DE TABELAS

2.1	Tipos de <i>chunks</i> do protocolo SCTP.....	15
2.2	Comparativo entre protocolos de transporte. [Legenda NA - Não Aplicável] ...	24
3.1	Novas estruturas TLVs, conforme RFC 5061.....	28
3.2	Novas estruturas TLVs para tratamento de erros, conforme RFC 5061	28
5.1	Lista de parâmetros e coeficientes.....	39
6.1	Variáveis SCTP que influenciam na latência de <i>handover</i>	54
6.2	Tempo gasto em segundos para efetuar o <i>handover</i> , segundo Fallon et al. [5] ..	54
6.3	Resultado de testes do impacto dos <i>handovers</i> na vazão da rede	57
6.4	Lista de parâmetros limiares usados pelo PA-SCTP	64
6.5	Condições impostas ao tráfego de pacotes de dados pela rede	68
6.6	Protocolos utilizados nos experimentos.....	69
6.7	Valores para os parâmetros configuráveis do protocolo SCTP	69

Capítulo 1

Introdução

O protocolo SCTP (*Stream Control Transmission Protocol*), que foi inicialmente apresentado para a suíte de protocolos TCP/IP como um protocolo de transporte de uso geral, tem características especiais que o diferenciam dos seus equivalentes (TCP e UDP), tais como o *multi-homing*, que permite o uso de mais de uma saída para os pacotes de dados em função de uma única aplicação, e o *multistreaming*, que permite mais de um fluxo de dados utilizando uma única conexão lógica. O *multi-homing* é usado com duas finalidades: redundância a falhas por prover múltiplos caminhos de transmissão e como mecanismo usado para gerenciamento de mobilidade por promover a participação do terminal dotado de protocolo SCTP em diferentes redes IPs.

Com o uso do *multi-homing* no SCTP, é apresentada para a aplicação, de forma implícita, a disponibilidade de vários caminhos disponíveis dentro da associação estabelecida. A escolha de qual caminho inicialmente utilizar fica a cargo do protocolo SCTP implementado (FUNASAKA et al. [6]).

Na RFC 4960, que rege o SCTP, não existe um mecanismo de avaliação inicial de qual o melhor caminho para transmissão de pacotes, entre os disponíveis, a ser utilizado inicialmente para a transmissão dos pacotes de dados. Existem políticas para detectar a falha do caminho primário e, a partir desse ponto, começar a fazer uso de um caminho secundário, se houver algum disponível.

A metodologia padrão do protocolo SCTP para chegar à conclusão de que um caminho é falho e reiniciar a transmissão dos pacotes de dados pelo caminho secundário conduz a grandes atrasos no recebimento de pacotes de dados durante esse período. Como consequência, esse procedimento pode aumentar significativamente o tempo de transmissão de um arquivo de dados, nem sempre a patamares aceitáveis para o usuário (GRACE et al. [7]).

A tomada de decisão sobre o momento da troca do caminho para o envio de pacotes de dados SCTP é baseada em parâmetros que não englobam as necessidades da aplicação em relação ao serviço que está sendo prestado ao usuário (FUNASAKA et al. [6]). Para que, de alguma forma, os ajustes nos parâmetros configuráveis do SCTP atendam às necessidades

das aplicações, seria preciso fazer algum tipo de correlação baseada na eventual alteração desses valores, como, por exemplo comparação com, o limite máximo do atraso entre pacotes suportados pela aplicação.

Esta pesquisa, portanto, apresenta questões relacionadas ao protocolo SCTP, a partir da disponibilidade do *multi-homing*, que permite a troca do caminho para transmissão de pacotes de dados. Os impactos causados para aplicações que estejam utilizando o protocolo SCTP como protocolo de transporte, na maioria das vezes, não são contornados devido à forma de gerenciamento do *multi-homing*, que toma decisões de troca do caminho de transmissão de pacotes de dados não levando em consideração ao desempenho da aplicação. No que se refere à solução, a pesquisa buscou gerenciar as múltiplas saídas propiciadas pelo protocolo SCTP, para a transmissão de pacotes de dados, de forma que amenizem possíveis problemas que possam vir a se apresentar em aplicações que façam uso desse protocolo, como, por exemplo, atraso excessivo no recebimento de pacotes de dados decorrido de congestionamento que possam estar ocorrendo em algum ponto da rede durante uma conexão fim-a-fim.

1.1 Motivação e problema

O uso mais eficiente e satisfatório para o transporte de pacotes de dados é uma busca constante em praticamente todas as camadas de protocolos envolvidos em uma comunicação. No caso do protocolo SCTP, ele se apresenta como um verdadeiro canivete suíço, capaz de atender a necessidades simples, como transporte de pacotes de dados, e complexas, como a de aplicações com algum tipo de exigência relacionada ao tempo de recebimento de pacotes de dados. Para atender a diferentes necessidades inerentes a vários tipos de aplicação, o SCTP suporta a forma de entrega de pacotes ordenada/não ordenada e também a parcialmente ordenada com o PR-SCTP (*Partial Reliable SCTP*). O SCTP também consegue promover o gerenciamento de mobilidade com o mSCTP (*Mobile SCTP*).

Em todos os cenários, seja usando o SCTP, o PR-SCTP ou o mSCTP, um dos seus grandes atrativos é o *multi-homing*, ou seja, a disponibilização de múltiplos caminhos para a entrega de pacotes de dados. A gerência desses múltiplos caminhos é realizada da mesma forma, tanto no SCTP como nas suas extensões, no caso o PR-SCTP e mSCTP. A troca do caminho para a transmissão de pacotes de dados está associada à perda de conectividade. Diagnosticar a perda de conectividade em um caminho é um procedimento lento que traz insatisfação com o SCTP, seja quando utilizado apenas no transporte de pacotes de dados ou em situações mais complexas, como atender a aplicações com algum tipo de exigência na entrega de pacotes ou no gerenciamento de mobilidade com o mSCTP.

1.2 Objetivos

1.2.1 Geral

Esta dissertação propõe o gerenciamento de *multi-homing*, provido pelo protocolo SCTP, de forma que priorize as necessidades das aplicações que porventura venham a fazer uso desse protocolo em uma comunicação fim-a-fim. A pesquisa foi direcionada para criar um mecanismo capaz de tornar o protocolo SCTP adaptativo à aplicação, que proponha a troca do caminho de transmissão de pacotes de dados quando se depara com um ambiente que cause a perda de desempenho. Na versão mais recente do protocolo em apreço, de recomendação de troca do caminho de transmissão de dados é baseada na presença ou não de conectividade sendo que isso infere prejuízos significativos para aplicações inclusive inviabilizando, em alguns contextos, o seu uso como protocolo de transporte.

1.2.2 Específicos

Para alcançar o objetivo geral desta pesquisa, foram definidos alguns objetivos específicos:

- avaliar a habilidade do SCTP para atender à aplicação, em cenários *multi-homed*;
- avaliar as soluções já propostas para sanar/resolver o problema da lentidão do SCTP em tomar a decisão de troca do caminho de transmissão de pacotes de dados;
- propor método que determine o melhor momento para a troca do caminho de transmissão de pacotes de dados e que esteja familiarizado com as necessidades das aplicações;
- definir e implantar um *testbed* em que os experimentos da pesquisa possam ser realizados;
- desenvolver uma solução que faça uso do *multi-homing* do SCTP de forma que priorize o atendimento aos requisitos das aplicações.

1.3 Abordagem

Em um primeiro momento, foi feita uma revisão da literatura. Isso foi necessário para conhecer melhor o protocolo SCTP, problemas apontados, propostas de soluções, formas de empregabilidade e suporte ao protocolo.

De forma mais direcionada, foi feito um estudo do uso do *multi-homing* no SCTP e possíveis problemas apontados no uso do atual mecanismo de troca do caminho para transmissão de pacotes de dados.

Posteriormente, foi sondada a plataforma de sistema operacional que proveria maior suporte ao novo protocolo de transporte para a implementação de uma possível solução relacionada ao uso do *multi-homing* SCTP.

E, por fim, buscou-se uma solução que priorize as necessidades das aplicações no uso do *multi-homing* SCTP por determinar quando realizar o procedimento de troca do caminho de transmissão de pacotes de dados não causando/amenizando possíveis prejuízos à aplicação. Paralelamente a essa etapa, foi idealizado um *testbed* para demonstrar o ganho de desempenho obtida com a solução.

1.4 Proposta de trabalho

Uma importante característica do novo protocolo da camada de transporte SCTP é, sem dúvida, o *multi-homing*. Essa característica peculiar permite ao SCTP agregar múltiplos endereços IPs a uma única porta de comunicação dentro de uma conexão, de forma transparente para a aplicação. Isso é um grande diferencial, se comparado aos protocolos clássicos da camada de transporte UDP e TCP.

A característica de *multi-homing* do SCTP foi inicialmente projetada para fins de prover confiabilidade à camada de rede. Pouco tempo depois, a sua funcionalidade foi estendida e passou a atender o gerenciamento de mobilidade via mSCTP. Hoje existem propostas para usar a característica de *multi-homing* do SCTP para fazer balanceamento de carga. Mas isso ainda não está ratificado na forma de RFC. Outras propostas existentes fazem uso do *multi-homing* para atender à necessidade da aplicação, como, por exemplo, a exigência de tempo máximo na chegada de pacotes de dados transmitidos.

O esquema padrão utilizado pelo SCTP para desencadear o melhor momento para a tomada de decisão de troca do caminho primário não o torna muito atrativo em relação ao TCP/UDP como protocolo de transporte, a ponto de reescrever programas para utilizar esse novo protocolo. O atraso gerado no procedimento de detecção de falha no atual caminho primário e a troca por um caminho secundário podem ser inviáveis para alguns tipos de aplicações.

Como o protocolo SCTP é *multi-homing*, o termo *handover*, neste trabalho, foi utilizado para definir o ato do protocolo SCTP de alterar o endereço de destino para o envio de pacotes de dados.

O termo *multi-homing*, neste trabalho, foi utilizado para definir uma estação SCTP que possua duas interfaces de comunicação, que propicia acesso a redes IPs diferentes.

Todo o trabalho de pesquisa culminou no PA-SCTP (Proativo Adaptativo-SCTP) que busca amenizar o problema relacionado ao melhor momento de realizar *handover* (troca do destino primário), baseando-se na computação de paradas no recebimento de pacotes, picos de atraso, atraso médio e variação da janela de transmissão (CWND). Com esses dados é es-

perado que o PA-SCTP seja capaz de prever possíveis oscilações na rede e preventivamente recomendar a troca do caminho primário, resguardando, dessa forma, a aplicação.

1.5 Organização

Esta dissertação se encontra organizada da seguinte maneira:

- Capítulo 2 - este capítulo apresenta um estudo sobre o protocolo SCTP e suas principais características. Essa leitura é importante pelo fato do objeto de estudo dessa dissertação ser baseada nas características do SCTP;
- Capítulo 3 - este capítulo apresenta um estudo sobre a extensão do protocolo SCTP chamada mSCTP. A importância desse capítulo é justificada pelo fato da solução proposta nessa dissertação fazer uso do mSCTP como meio para alcançar as metas propostas;
- Capítulo 4 - a importância deste capítulo é justificada pelo detalhamento de uma das principais características do SCTP, que é o *multi-homing*. Essa característica foi bastante explorada na solução proposta nessa dissertação;
- Capítulo 5 - este capítulo faz um levantamento comparativo entre várias publicações, no que refere ao uso do protocolo SCTP, especificamente explorando o *multi-homing*. A importância desse capítulo é pela apresentação das várias formas de empregabilidade do *multi-homing* e observações sobre essa característica na ótica de outros pesquisadores;
- Capítulo 6 - este capítulo apresenta uma solução, denominada PA-SCTP (Proativo Adaptativo SCTP), que faz uso do *multi-homing* no SCTP de forma adaptativa a aplicação;
- Capítulo 7 - este capítulo expõe a conclusão e indica trabalhos futuros.

Capítulo 2

SCTP - Stream Control Transmission Protocol

2.1 Visão geral do protocolo SCTP

O SCTP foi originalmente projetado para ser o protocolo de transporte da rede de telefonia pública comutada (PSTN – *Public Switched Telephone Network*), sinalizando mensagens sobre redes IP, conforme definido em Stewart [3]. Contudo mostrou-se uma tendência em desempenhar um papel de protocolo de transporte de uso geral devido à sua capacidade para atender a amplas necessidades das aplicações.

Como boa parte das aplicações requer confiabilidade para transportar informações, embora nem sempre seja possível devido a variados fatores, o protocolo SCTP tenta preencher lacunas da falta de confiabilidade no uso da estrutura de rede preexistente, no caso, redes IPs.

Entre as características do protocolo SCTP, destaca-se a execução de todas as requisições de serviços dentro de um contexto de associação, semelhante à ideia de conexão no TCP, só que entre duas estações SCTP. Por natureza, o SCTP é orientado à conexão, assim não suporta o envio de fluxo de dados *multicast* ou *broadcast*, conforme comentado por Xie et al. [8]. Durante o período de configuração de uma associação, cada estação SCTP provê informações sobre a lista de possíveis endereços IPs pelos quais podem ser referenciados na rede, sendo todos associados a uma única porta de comunicação na camada de transporte. Conforme apresentado na figura 2.1, o SCTP está localizado na camada de transporte em relação ao padrão TCP/IP.

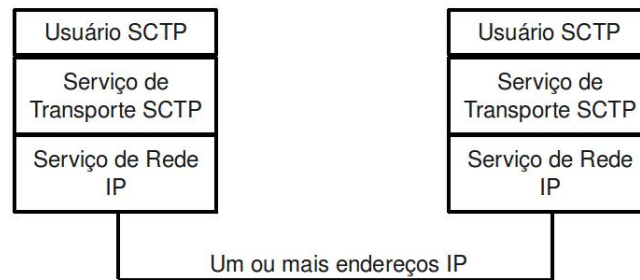


Figura 2.1: Localização do SCTP na pilha TCP/IP, referenciado em Costa [2] e Stewart [3]

Conforme Costa [2] e Stewart [3], para proporcionar confiabilidade sobre a estrutura existente, o SCTP oferece os seguintes serviços:

- confirmação livre de erros;
- tentativa de não transferência de dados duplicados;
- fragmentação de dados conforme a descoberta do MTU da rede (mais detalhes no tópico 2.8);
- entrega de dados de forma ordenada conforme o TCP e de não ordenada como no UDP;
- controle diferenciado de fluxos de dados por meio do uso do *multistreaming* (mais detalhes na seção 2.6);
- tolerância a falhas no nível de rede pelo suporte ao *multi-homing* (mais detalhes na seção 2.5);
- apropriado controle de congestionamento (mais detalhes na seção 2.14);
- proteção contra ataques do tipo mascaramento e inundação.

As características mais importantes desse protocolo são, sem dúvida, o *multi-homing* e o *multistreaming*. A técnica de *multi-homing* possibilita o uso de mais de um endereço IP pela mesma associação para enviar e receber pacotes de dados SCTP, constituindo, dessa maneira, uma importante ferramenta de tolerância a falhas na transferência de dados. Já o *multistreaming* se caracteriza pela criação de mais de um canal virtual de transporte de dados sobre um único destino estabelecido na associação, possibilitando fluxos diferentes em cada caminho virtual, além da aplicação de técnicas distintas de entrega, como, por exemplo, a entrega de pacotes de forma ordenada/não ordenada/parcialmente ordenada.

2.2 Cronologia do protocolo SCTP

Inicialmente, o protocolo foi projetado para transportar sinalização de tráfego de telecomunicação de voz sobre redes IPs. Logo o SCTP mostrou a sua tendência de ser um protocolo de transporte de uso geral para aplicações orientadas a mensagens. A definição do protocolo foi feita pelo grupo de trabalho SIGTRAN (*Signaling Transport*) do IETF (*Internet Engineering Task Force*), que emitiu o primeiro documento de padronização do SCTP em outubro de 2000.

A cronologia sobre o protocolo SCTP é representada na figura 2.2, a qual apresenta RFCs que foram publicadas e que são descritas na sequência.

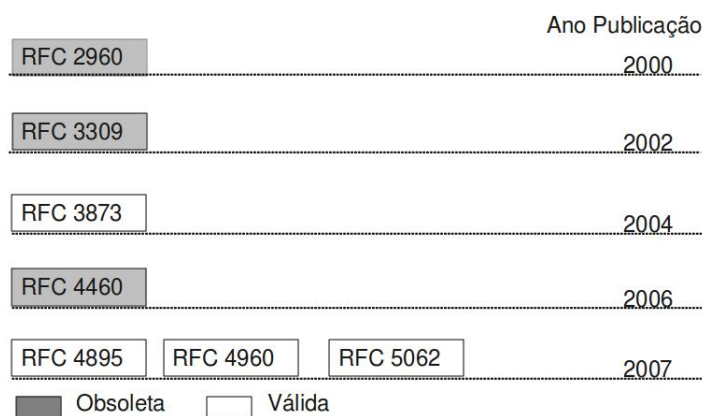


Figura 2.2: RFCs do SCTP e suas atualizações

- RFC 2960 - primeira RFC contendo informações de definição e funcionamento do protocolo SCTP. Tornou-se obsoleta após o lançamento da RFC 4960.
- RFC 3309 - trata de uma correção na forma de cálculo do *checksum* do pacote SCTP descrito na RFC 2960. Tornou-se obsoleta após o lançamento da RFC 4960.
- RFC 3873 - trata do processo do uso do SCTP e de MIB (*Management Information Base*).
- RFC 4460 - trata de correções na RFC 2960.
- RFC 4895 - trata do processo de autenticação de *chunks* no SCTP.
- RFC 4960 - nova RFC referente ao SCTP, tornando obsoletas as RFCs 2960, 3309 e 4460.
- RFC 5062 - trata de segurança contra ataques ao SCTP.

Na figura 2.3, são apresentadas RFCs que propõem um uso estendido do SCTP, para além do proposto pela RFC 2960 e, posteriormente, pela RFC 4960.

	Ano Publicação
RFC 3758	2004
RFC 5061	2007

Figura 2.3: RFCs que estendem as possibilidades de uso do SCTP

- RFC 3758 - estende a capacidade de transporte do SCTP com o uso de confiabilidade parcial na entrega de pacotes (PR-SCTP).
- RFC 5061 - estende o SCTP para tratar gerenciamento de mobilidade (mSCTP).

2.3 Breve comparativo: SCTP e TCP

O TCP tem sido usado como um protocolo para transferência de dados de forma confiável em redes IP, porém certo número de aplicações tem encontrado limitações no TCP, não atendendo, de maneira adequada, aos requisitos impostos por essas aplicações, tais como aquelas sensíveis a atraso no recebimento de pacotes. Por conta disso, elas fazem uso do UDP e incorporam, na camada de aplicação, os requisitos adicionais pretendidos e não atendidos pelo UDP.

Entre algumas limitações encontradas no TCP (UFRS [9]):

- o TCP provê confiabilidade, mas, em contrapartida, exige estrita ordenação na entrega dos dados. Algumas aplicações necessitam de confiabilidade na entrega dos dados, mas não exigem a sequência correta na entrega deles; enquanto outros se contentam com uma ordem de entrega parcial dos dados. Em ambos os casos, o TCP causará um atraso desnecessário na entrega dos dados. O SCTP consegue trabalhar com entrega ordenada, não ordenada e parcialmente ordenada;
- a natureza orientada à conexão do TCP é frequentemente um inconveniente para certas atividades. Aplicações devem adicionar seus próprios marcadores de registro para delimitar suas mensagens e devem fazer uso explícito das facilidades de *push* para assegurar que uma mensagem completa possa ser transferida em um tempo razoável (figura 2.4). O SCTP contorna isso com o uso de *chunks*;
- o escopo do TCP complica a tarefa de prover *hosts multi-homing*, pois não consegue gerenciar mais de um destino IP por conexão. O SCTP consegue trabalhar com múltiplos endereços de transporte nativamente e de forma transparente para aplicações;
- o TCP é relativamente vulnerável a ataques de negação de serviço. O SCTP contorna essa falha por lançar desafios a cada solicitação de conexão.

Um quadro comparativo envolvendo o TCP, UDP e SCTP é apresentado no final deste capítulo, na tabela 2.2.

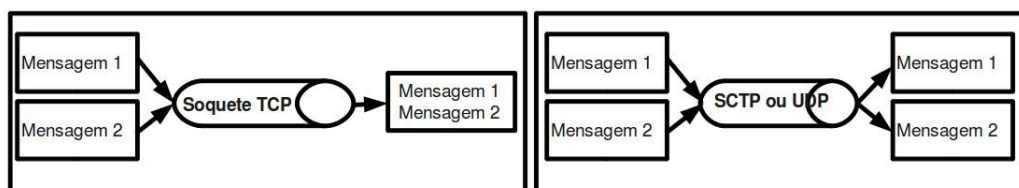


Figura 2.4: Tratamento dado à mensagem do usuário, conforme Stalvig [4]

2.4 Associação

Associação é uma terminologia utilizada para configuração de um meio de comunicação fim-a-fim entre duas estações SCTP, que, no caso do TCP, é chamada de conexão (figura 2.5). A associação tem maior amplitude que a conexão devido a fatores, como, por exemplo, gerenciamento de múltiplos destinos.

Uma associação é inicializada por uma solicitação de usuário SCTP. Um mecanismo de *cookie* (uma espécie de desafio para o solicitante) é usado para prover proteção contra ataques de sincronização e alocação de recursos. O processo de configuração da associação utiliza uma forma de comunicação de quatro vias (no TCP, utiliza-se um mecanismo de três vias) para posterior envio de mensagens. As duas são demonstradas na figura 2.5.

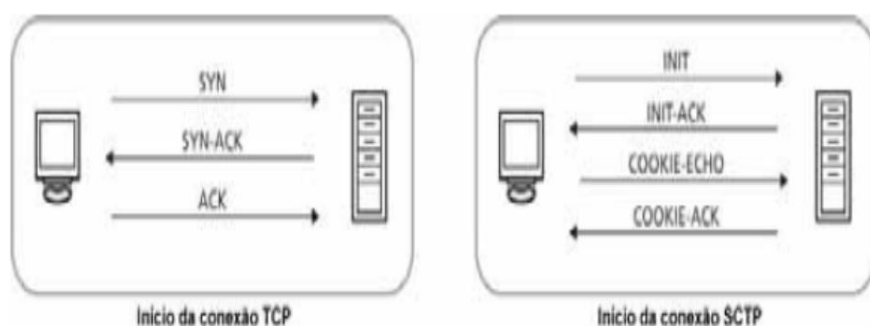


Figura 2.5: Conexão TCP e a associação SCTP, conforme Stalvig [4]

O SCTP permite dois tipos de encerramento de uma associação. A forma mais suave é por meio do envio de mensagem de controle denominada *SHUTDOWN*, como apresentado na figura 2.6, que permite o encerramento suave da associação sem perda de dados. Existe também o modo áspero, por meio do envio de uma mensagem de controle denominado *ABORT*, que equivaleria, de maneira grosseira, a uma interrupção repentina do meio

de comunicação. Essas duas técnicas de encerramento de uma associação podem ser induzidas pelo usuário. Todavia a técnica mais áspera, no caso, o *ABORT*, pode ser empregada sem o consentimento do usuário, devido a fatores que a implementação do protocolo SCTP considere inaceitável para se manter uma associação ativa, tais como o recebimento de um *chunk DATA* sem dados.

O SCTP não suporta o estado *half-open* (como o TCP), situação em que um lado continua a enviar dados, enquanto o outro já fechou a conexão. Em uma associação SCTP, quando um dos lados executa *SHUTDOWN* ou *ABORT*, cada estação interrompe o envio e o recebimento de pacotes de dados da sua camada superior.

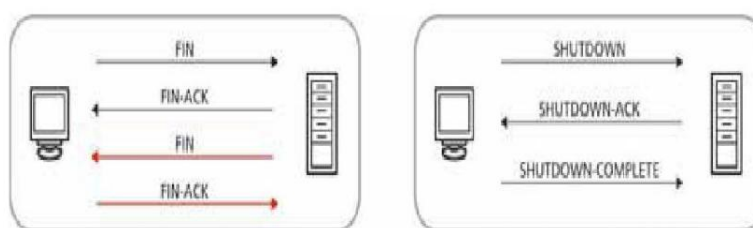


Figura 2.6: Encerramento da conexão TCP e associação SCTP, conforme Stalvig [4]

2.5 Multi-homing no SCTP

Uma estação SCTP é considerada *multi-homed* se existir mais de um endereço de transporte usado como destino por outra estação SCTP, conforme citações encontradas em Budzisz et al. [10], Caro et al. [11], Kelly et al. [12] e Siddiqui et al. [13].

O protocolo SCTP consegue fazer com que vários endereços IP estejam associados a uma porta, sendo isso totalmente transparente para a aplicação (CARO et. al. [14]).

A principal característica desse protocolo com essa técnica, segundo Stewart [3] e Caro et al. [15], é proporcionar tolerância a falhas de comunicação na camada de rede. Outra variação do uso dessa técnica de *multi-homing*, proposta em artigos como o de Caro et al. [16] e Iyengar et al. [17], é fazer uso dessa característica para fins de balanceamento de carga ou transmissão redundante.

Segundo Kelly et al. [12], o *multi-homing* é proposto no SCTP para dispor redundância de caminhos, oferecendo, dessa forma, certo grau de confiabilidade na rede. Ele também poderia ser usado para prover certo nível de QoS (*Quality of Service*) para aplicações que tenham necessidade de trabalhar com comunicação de tempo real, semelhante a VoIP (*Voice over IP*) ou aplicações de *streaming* de vídeo. O *multi-homing* também pode ser usado para facilitar o *handover*.

O SCTP propõe fazer uso dos destinos pertencentes à associação apenas de maneira sequencial. A saída classificada como primária é utilizada para transferência de dados, e o

destino secundário fica ocioso. As exceções ocorrem em retransmissões de pacotes perdidos, que se utilizam do caminho secundário, pois, caso contrário, esse caminho seria utilizado apenas para transmissão de dados de controle (sondagem).

Quando o caminho primário falha, é acionado o mecanismo de tratamento de *handover*, e é eleito um novo endereço IP como o destino para os pacotes de dados SCTP. Vários artigos tratam sobre o *handover* no SCTP, entre os quais podemos citar Budzisz et al. [10], Caro et al. [14], Caro et al. [11] e Fitzpatrick et al. [18].

Artigos que enfocam basicamente o uso de *multi-homing* para resolver problemas de interrupção de serviços podem ser encontrados em Budzisz et al. [10], Fallon et al.[5], Fitzpatrick et al. [18], Funasaka et al. [6], Grace et al. [7] e Siddiqui et al. [13].

2.6 MULTISTREAMING no SCTP

Na configuração de uma conexão TCP, é criado apenas um canal de comunicação, que transmite dados somente no modo de entrega ordenada, o que pode ocasionar um tipo de bloqueio na entrega dos dados denominado *head-of-line* (bloqueia a entrega de pacotes fora de ordem para a camada de aplicação), inviabilizando o uso do protocolo TCP por aplicações com algum tipo de exigência de atraso mínimo intrapacotes.

Na configuração de associação SCTP (mesmo sentido de conexão para TCP), é possível criar vários canais de comunicação (*multistream*) para uma única associação. Isso é pré-acordado na inicialização da associação, precisamente nas duas primeiras mensagens trocadas entre as partes no início da configuração (*INIT* e *INIT ACK*).

Uma diferença marcante entre o SCTP e o TCP é apresentada na sua associação/conexão lógica fim-a-fim. Segundo Costa [2], o SCTP apresenta um número arbitrário de fluxos *simplex* (unidirecionais) acertados no início da associação (precisamente nas trocas das mensagens *INIT* e *INIT ACK*), enquanto que no TCP é formado apenas por um fluxo *full-duplex*.

Cada fluxo SCTP (figura 2.7) é um canal de comunicação unidirecional, ao qual podem ser aplicadas técnicas de entrega de dados ordenada/não ordenada. Esses fluxos de comunicação não precisam, necessariamente, estar em quantidades iguais em cada direção, conforme afirmado por Costa [2].

2.7 Pacote SCTP

O formato da estrutura de cabeçalho de um pacote SCTP (figura 2.8) é bem simples, contém apenas quatro campos. Essa simplicidade aparente se dá pelo tipo de técnica utilizada pelo protocolo, já que promove de dois a três tipos de encapsulamento de estruturas de cabeçalho na mesma camada, no caso, a de transporte.

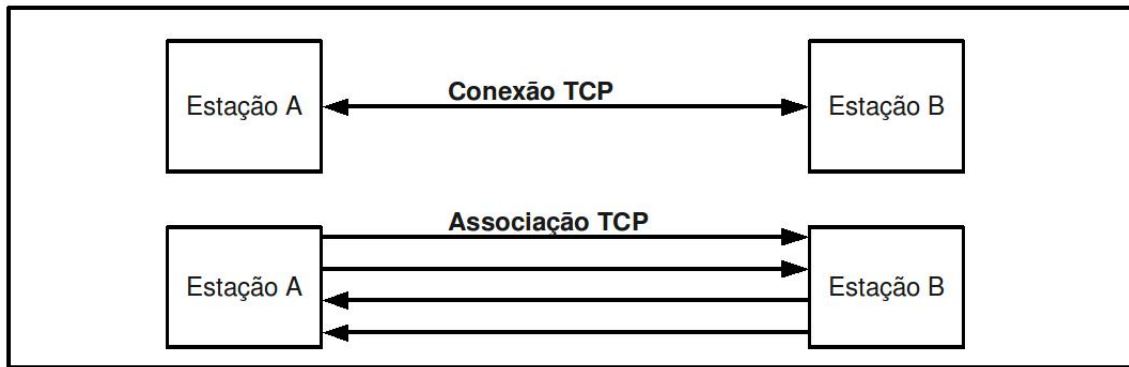


Figura 2.7: Modelo conceitual de associação, conforme Costa [2]

A proposta do protocolo trabalha com a ideia de *chunks* (*payload* do SCTP), que nada mais é que um segmento da mensagem do usuário/controlador com estrutura de cabeçalho própria, mas dependente do cabeçalho SCTP para trafegar pela rede. No *payload* do *chunk*, também existe uma estrutura de cabeçalho própria, chamada genericamente de TLV (abreviação *Type-Length-Value*), que também é dependente da estrutura de cabeçalho do *chunk*, que, por sua vez, é dependente da estrutura de cabeçalho do pacote SCTP.

Basicamente, o cabeçalho SCTP contém informações para identificação da associação na estação receptora e indicação para qual porta a mensagem deve ser entregue.

A carga útil do protocolo SCTP pode conter um ou mais *chunks* (dados do usuário ou informações de controle), do mesmo tipo ou de tipos diferentes, limitados pelo MTU da rede. Existem certos tipos de *chunks* que devem vir obrigatoriamente sozinhos como carga útil de um pacote SCTP, da mesma forma que, na multiplexação de *chunks* no *payload* de pacotes SCTP, certas “hierarquias” de posição física devem ser respeitadas dentro da carga útil do pacote SCTP.

Quando encapsulados vários *chunks* no *payload* do mesmo pacote SCTP, os de controle devem estar localizados fisicamente antes dos de dados.

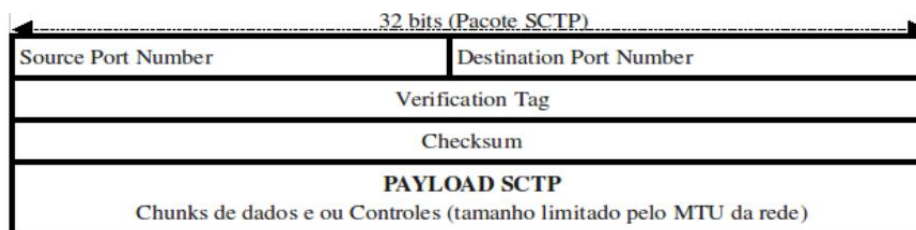


Figura 2.8: Pacote SCTP

2.8 Fragmentação no SCTP

Como em qualquer situação, seja qual for o protocolo que se encontra na camada de transporte (TCP/UDP/SCTP), o tamanho da mensagem entregue pela camada de aplicação pode exceder a carga útil do protocolo de transporte. É preciso fragmentar essa mensagem em pedaços para que possa ser enviada pela estrutura de rede. O SCTP e o TCP realizam essa fragmentação de acordo com o MTU da rede e, no caso do UDP, a limitação é desencadeada pela capacidade de carga útil do protocolo. As partes da mensagem fragmentadas pelo SCTP recebem o nome de *chunk DATA*.

Tanto o TCP como o SCTP trabalham com um processo de entrega fim-a-fim, ou seja, eles precisam de confirmação de recebimento dos pacotes enviados. O TCP faz esse controle em relação à quantidade de *bytes* enviados, enquanto que o SCTP usa a ideia de TSN (parâmetro numérico de identificação) presente no cabeçalho dos *chunks DATA*. Atribuindo valores inteiros de identificação (chamados TSNs) a esses *chunks*, é possível pelas informações de confirmação verificar quais pedaços da mensagem foram entregues.

Com a introdução da ideia de *chunks* adicionada ao *multistream*, é possível fazer o processo de multiplexação de dados/controle na camada de transporte, podendo, dessa maneira, tentar disponibilizar, de forma mais ágil, os dados para o usuário.

2.9 Carga útil do SCTP

Como já mencionado neste documento, *chunk* é a carga útil de um pacote SCTP. Mais de um *chunk* pode ser inserido dentro de um pacote SCTP. Isso é limitado pelo MTU da rede.

Os *chunks* podem ser classificados como de dados ou de controle. Essa informação é especificada no seu cabeçalho. Os *chunks* de dados podem vir encapsulados junto com os de controle dentro de um pacote SCTP. Exceções ficam por conta dos *chunks INIT*, *INIT ACK* e *SHUTDOWN COMPLETE*. Ainda temos os *chunks SHUTDOWN*, *SHUTDOWN ACK* e *ABORT*, que podem ser encapsulados dentro de um pacote SCTP junto com outros pacotes de controle, mas não com dados, devido a regras mencionadas na RFC 4960, na qual é dito que todo *chunk* de dado deve vir após *chunks* de controle quando encapsulados dentro do mesmo pacote SCTP.

Na figura 2.9, existe a representação de um pacote SCTP com um *chunk* genérico. Cada *chunk* tem seus campos específicos de cabeçalho, mas também existem campos genéricos que estão presente em todos os *chunks*, listados como *Type*, *Flags* e *Length*.

Conforme figura 2.9, existe uma referência à estrutura TLV (abreviação de *Type-Length-Value*), que pode vir dentro de um *chunk* de controle (dependendo do seu tipo) e que não está presente em *chunks DATA*.

As estruturas TLVs são utilizadas como invólucro, por exemplo, para mensagens de erro

que trafegam pela rede e praticamente para todos os subtipos de mensagens enviadas por *chunks* (*INIT*, *INIT-ACK*, *COOKIE-ECHO* e *COOKIE-ACK*) na abertura de associação para, por exemplo, informar por quais endereços IPs responde cada estação SCTP.

Tipos de *chunks* existentes para o protocolo SCTP, de acordo com a RFC 4960, estão listados na tabela 2.1. Eles são referenciados quando necessário neste documento.

Tabela 2.1: Tipos de *chunks* do protocolo SCTP

Nome do <i>chunk</i>	Função do <i>chunk</i>
<i>INIT</i>	Usado para iniciar uma associação
<i>INIT ACK</i>	Segunda mensagem de configuração de uma associação
<i>COOKIE ECHO</i>	Terceira mensagem de configuração de uma associação
<i>COOKIE ACK</i>	Quarta mensagem de configuração de uma associação
<i>DATA</i>	Dados da aplicação
<i>SACK</i>	Confirmação seletiva de recebimento de <i>chunks</i> de dados
<i>ERROR</i>	Mensagens de erro com origens diversas
<i>ABORT</i>	Fechamento imediato de uma associação
<i>SHUTDOWN</i>	Primeira fase para o fechamento gracioso de uma associação
<i>SHUTDOWN ACK</i>	Segunda fase para o fechamento gracioso de uma associação
<i>SHUTDOWN COMPLETE</i>	Terceira fase para o fechamento gracioso de uma associação
<i>HEARTBEAT</i>	Sondagem de caminho secundário ou ocioso
<i>HEARTBEAT ACK</i>	Confirmação de sondagem de caminho secundário ou ocioso

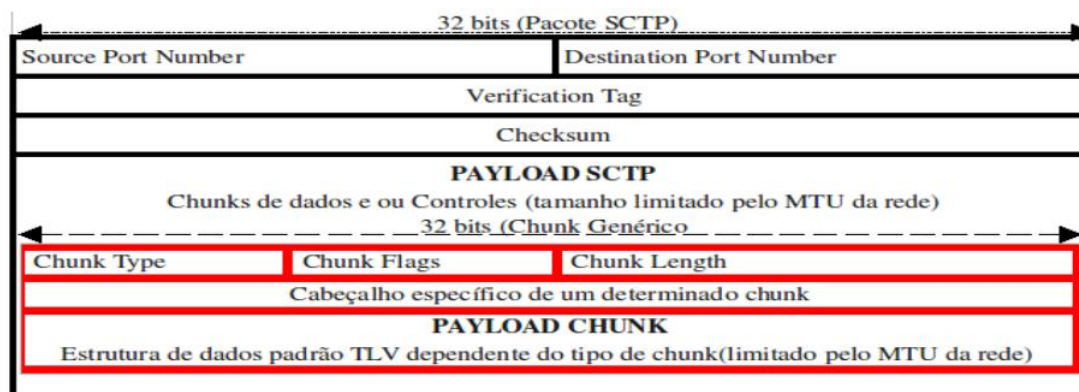


Figura 2.9: Pacote SCTP que tem um *chunk* genérico como carga útil

2.10 Detalhamento do multi-homing no SCTP

Uma estação SCTP é considerada *multi-homed* se tiver mais de um endereço de transporte que possa ser utilizado para alcançar outra estação SCTP em uma associação já estabelecida. Porém a estação SCTP deve selecionar apenas um endereço de destino, como sendo o endereço primário para a transmissão de dados.

A transmissão de *chunks* de dados ocorre em apenas um dos endereços de destino disponível, no caso, o intitulado como primário. O tipo de dado transmitido nos outros

endereços de destino, exceto o primário quando ativo, são os chamados *chunk HEARTBEAT* e *HEARTBEAT ACK*. Conforme Kelly et al. [12], esses *chunks* monitoraram a alcançabilidade dos destinos e calculam o RTT para possível uso na transmissão de *chunks* de dado/controlado quando não for possível o seu envio pelo endereço intitulado como primário.

Por padrão, uma estação SCTP sempre transmite dados pelo endereço primário, a menos que de outra forma seja compelido a usar o caminho secundário.

Uma estação *multi-homed* deve tentar transmitir *chunks* de dados/controlado e receber confirmações sempre pelo mesmo caminho. De acordo com Caro et al. [19], se ocorrer um *timeout* (envio de dado e não recebimento de confirmação) para um *chunk* de dado, a retransmissão deverá ser feita por meio de um caminho secundário, se houver, e novas transmissões de dados devem continuar a ser enviadas pelo caminho primário, até quando alcançar o limite máximo de tentativas de envio por um caminho e não receber confirmação. A partir desse ponto, o caminho primário é taxado como inalcançável e a estação SCTP passa a eleger um dos caminhos secundários como primário, seguindo, dessa maneira, o fluxo normal de transmissão dos dados.

No caso do caminho primário, ao se tornar inativo, passarão a trafegar apenas *chunks HEARTBEAT*. A partir do momento que os *chunks HEARTBEAT* passarem a ser confirmados por *chunks HEARTBEAT ACK*, esse caminho automaticamente poderá tornar-se primário novamente. Isso depende de como foi implementado o protocolo SCTP.

Segundo Budzisz et al. [10], a RFC 4960, que documenta o SCTP, deixa brechas na interpretação de como implementar o protocolo. Como exemplo, os autores citam que no caso de falha na entrega de um pacote, o mesmo deve ser retransmitido por um dos caminhos secundários e um novo pacote deve ser tentado a transmissão pelo caminho primário. Ainda segundo Budzisz et al. [10], existem implementações do protocolo que realizam isso de maneira sequencial, prejudicando o desempenho, porque deveriam ser simultâneas, uma vez que as saídas tendem a ser diferentes.

Outro caso citado por Budzisz et al. [10] é que, em algumas implementações do protocolo SCTP, novas transmissões de dados só ocorrem quando todos os pacotes marcados para retransmissão forem reenviados, levando os autores a concluírem que só existe uma fila de transmissão, enquanto o correto seria uma fila para pacotes novos e outra para retransmissão. Isso causa impacto no desempenho do protocolo, pois deveriam existir mais de uma saída para a transmissão de dados, que poderiam ser usadas simultaneamente.

2.11 Detalhamento do multistreaming no SCTP

Na configuração de uma associação entre duas estações SCTP, há a possibilidade de se criar mais de um canal virtual de comunicação com características próprias de entrega de dados (ordenada/não ordenada). A quantidade de canais é determinada no cabeçalho dos

chunks INIT e *INIT ACK* por meio dos parâmetros *Number of Outbound Streams* e *Number of Inbound Streams*.

No cabeçalho de cada *chunk* de dado, existe uma variável denominada *Stream Identifier*, que identifica o canal virtual por onde tráfegará a informação, e um campo chamado *Stream Sequence Number*, que identifica cada mensagem oriunda da camada de aplicação/rede. É por meio do campo SSN (*Stream Sequence Number*) que se faz o controle quanto à forma de entrega de pacotes de dados SCTP, podendo ser ordenada/não ordenada.

A mensagem identificada pelo *Stream Sequence Number* pode caber em apenas um *chunk* de dado, ou ser segmentada dentro de vários *chunks* de dados. Cada *chunk* de dado é identificado pela variável TSN. É por meio dessa variável que se realiza o controle de confirmação e retransmissão dos dados na rede, e é por meio do *Stream Sequence Number* que se controla a entrega ordenada/não ordenada dos dados.

Conforme Costa [2] e Seo et al. [20], o *multistream* ajuda, por exemplo, nos casos de acesso a páginas *web*, no qual retornam três tipos de dados: texto, som e imagem. Utilizando técnicas de *multistream*, é possível enviar cada tipo de dado por meio de diferentes canais empregando separadamente distintas técnicas de controle de fluxo, contornando, dessa maneira, o problema do *Head-of-Line* encontrado no TCP e permitindo a apresentação da página *web* no terminal do usuário de forma mais instantânea, mesmo não contendo todas as partes, como, por exemplo, a apresentação do texto antes da imagem. No caso do uso do TCP, teria de abrir três conexões, uma para cada tipo de dado. Isso acarretaria um atraso por ter de abrir e fechar três conexões. No servidor, existe um limite para abertura de conexões TCP simultâneas.

Segundo estudos feitos por Shi et al. [21], o protocolo SCTP se mostrou mais hábil para prover suporte a transações *e-commerce* que o TCP e UDP, devido ao uso do *multistreaming*, que proporciona ao SCTP o contorno ao problema do *Head-of-Line*.

2.12 Envio e recebimento de chunks SACKs

O SACK é um *chunk* de controle utilizado na confirmação de recebimento de dados (por meio do campo TSN em cada *chunk DATA*) que contém informações, tais como maior TSN *chunk DATA* recebido e que exista a confirmação de todos os TSNs numericamente inferiores, falhas na sequência dos *chunk DATA* recebido e TSNs de *chunks DATA* duplicados. Por meio dessas informações de retorno, o protocolo SCTP pode aplicar as técnicas de controle de congestionamento que se encontram ao seu alcance.

A cada *chunk SACK* que chega ao emissor de dados é executada uma série de processamento em relação aos valores encontrados nesse *chunk*, tais como o cálculo da capacidade da estação receptora em receber pacotes de dados.

O emissor de dados tem uma variável chamada RWND (*Receiver Window*), que detém

um valor que venha a ser o mais aproximado possível em *bytes*, representativo da capacidade de recepção de dados por parte do receptor no momento. Para chegar a esse valor, o emissor utiliza as informações oriundas do *chunk SACK* recebido, no caso, as variáveis *Cumulative TSN ACK* (último TSN recebido em sequência), *Advertised Receiver Window Credit* (A_RWND - capacidade de recepção do cliente, quando gerado o SACK) e *Gap ACK Blocks* (buracos na sequência de TSNs recebidos).

Outra variável também criada e mantida pelo emissor de dados é a *Cumulative TSN ACK Point*, sempre atualizada pelo *Cumulative TSN ACK* recebido a cada *SACK*. Essa é uma das variáveis que um emissor de dados pode observar para descobrir *chunks SACKs* recebidos fora de ordem, uma vez que não existe um identificador explícito para determinar a ordem sequencial de *SACKs* enviados. Para identificar *SACKs* fora de ordem, utiliza-se geralmente heurística (RFC 4960).

A qualquer momento que um *SACK* chega, uma estação SCTP executa as seguintes tarefas, segundo Stewart [3]:

- se o *Cumulative TSN ACK* for menor do que o *Cumulative TSN ACK Point*, o *chunk SACK* é descartado, indicativo de que foi recebido fora de ordem;
- ajusta o valor da variável RWND igual ao valor do A_RWND (recebido) menos o número de *bytes* (de *chunks DATA* enviados) ainda pendentes de confirmação;
- se reportada pela variável *Gap ACK Block* a falta de algum *chunk* de dado, um contador é incrementado para esse específico TSN. Ao atingir seu limiar, quantidade de vezes que recebe a solicitação de envio de um *chunk DATA* específico, um tratamento especial será desencadeado para resolver o problema, no caso, *Fast Retransmitted*.

O primeiro *chunk* de dados recebidos em uma associação recém estabelecida deve ser imediatamente confirmado. *chunks* de dados posteriormente recebidos não necessariamente precisam ser confirmados a cada pacote SCTP, mas um *SACK* poderá ser imediatamente gerado e transmitido quando se detectar um buraco na sequência de TSNs recebidos, ou quando houver o recebimento de *chunks* de dados duplicados.

Conforme Siddiqui et al. [13], quando a estação SCTP é *multi-homed*, o *chunk SACK* deve ser enviado pelo mesmo caminho que recebeu o *chunk DATA* a ser confirmado.

A cada *chunk* de dado descartado, devido à falta de espaço para armazenamento, um novo *SACK* deve ser gerado imediatamente, informando que o tamanho da janela de recepção é zero.

Em uma situação perfeita, um *SACK* deve ser gerado no mínimo a cada segundo de pacotes SCTP recebidos. Em casos de recebimento de *chunks* de dados com intervalos na sequência TSN, um *SACK* poderá ser gerado a cada 200 ms, conforme citado por Jungmaier et al. [22] e Noonan et al. [23]. Segundo Noonan et al. [23], Wang et al. [24] e Stewart [3], uma estação não deve gerar mais do que um *SACK* para cada pacote SCTP que receba.

2.13 Gerenciamento de retransmissões

Uma estação SCTP emissora usa *timer* para assegurar que *chunks DATA* enviados e não confirmados no espaço de tempo determinado sejam novamente retransmitidos. A duração desse tempo é referida como RTO (*Retransmission Timeout*).

Quando uma estação SCTP é *multi-homed*, existe um RTO separado para cada endereço de destino. O RTO é recalculado a cada SACK recebido ou a cada ocorrência de um *timeout*.

O ajuste do RTO é feito a cada RTT (*Round-Trip Time*), tempo decorrido entre a transmissão e a confirmação de um *chunk* de dado. No caso da não confirmação do recebimento de um *chunk* em um período RTO, esse valor será duplicado e o *chunk* retransmitido. Existem três formas de ajustes do parâmetro RTO, empregados conforme a situação que originou a chamada da rotina para alterar esse valor. Essas três formas e quando cada uma deve ser empregada estão documentadas na RFC 4960.

Em estações SCTP *multi-homed*, a retransmissão do *chunk* sempre ocorre por um caminho diferente daquele utilizado para o envio do *chunk* de dado no qual ocorreu o *timeout*. A transmissão de pacotes de dados novos sempre se dará pelo caminho primário até alcançar o número máximo de tentativas de transmissão permitida sem o recebimento de confirmação.

A não confirmação de um *chunk* de dado influencia diretamente no tipo de controle de congestionamento aplicado a um determinado caminho e no tempo de espera para retransmissão.

Segundo Noonan et al. [23] e Brennan et al. [25], na ausência de perdas de *chunks* de dados na rede, o receptor executa um atraso para transmissão de um SACK. Porém, se detectada uma quebra de sequência no recebimento de TSNs, desencadeará uma transmissão de um SACK imediatamente a todo o momento que um pacote chega com dados, até que a sequência seja normalizada.

A todo o momento que uma estação SCTP receber um SACK que indica que algum TSN foi perdido, é preciso esperar por mais duas indicações de perdas (via subseqüentes SACKs) do mesmo TSN antes de marcá-lo para *Fast Retransmit*, que é o processo de retransmitir pacotes pendentes imediatamente, ignorando a capacidade de recepção por parte do destinatário.

Resumindo, conforme Noonan et al. [23], pacotes perdidos no SCTP podem ser detectados de duas formas. Se um simples pacote foi perdido e subseqüentes não foram, a informação contida dentro de um SACK permite ao emissor deduzir a perda e retransmitir o pacote. Se nenhum SACK for recebido após o período de um RTO, o contador de erro do caminho é incrementado, o pacote é marcado como perdido e retransmitido por um dos caminhos alternativos, e o RTO é dobrado.

2.14 Controle de congestionamento do SCTP

É uma das funções básicas do protocolo SCTP. Para algumas aplicações, ele trabalha de maneira adequada para garantir que, mesmo em momentos críticos, de muito tráfego na rede, possa continuar o fluxo de dados da aplicação, nem que seja de maneira mais lenta.

O controle de congestionamento deverá ser capaz de operar sobre condições adversas, seja devido ao surgimento de falhas parciais de conectividade na rede ou até na presença de inesperado tráfego concorrente. Na ausência de congestionamento na rede, o algoritmo não deverá impactar no desempenho do protocolo SCTP (RFC 4960).

O controle de congestionamento do protocolo SCTP é sempre aplicado para a associação inteira e não para *streams* ou caminhos individuais em ambientes *multi-homing*. Existem trabalhos, como de Seo et al. [20], que propõem controle de congestionamento separados, para aumentar a eficiência do protocolo.

No SCTP, qualquer *chunk* de dado que for confirmado por um *SACK*, incluindo aqueles que chegaram fora de ordem, não são considerados totalmente entregues até que o *Cumulative TSN ACK Point* (mantido pelo emissor de dados) iguale ou ultrapasse o TSN do *chunk* de dado confirmado (BRENNAN et al. [25]).

Basicamente, o protocolo SCTP utiliza duas técnicas de controle de congestionamento, *Slow-Start* e *Congestion Avoidance*. A primeira é utilizada para ambientes congestionados ou desconhecidos (com relação ao RTT); e a segunda, para caminhos conhecidos, estáveis e sem congestionamento.

Para cada endereço de destino em uma associação SCTP, seja *multi-homing* ou *single-homing*, sempre o mecanismo *Slow-Start* é o primeiro a ser utilizado como controle de congestionamento para transmissão de dados.

Uma variável chamada de *SSTHRESH* (*Slow-Start Threshold*) é mantida por cada estação SCTP e é utilizada como limiar para determinar qual mecanismo de controle de congestionamento está sendo usado no momento. Quando o *CWND* (*Congestion Window*) for maior que *SSTHRESH*, o gerenciamento de congestionamento utilizado é o *congestion avoidance*, caso contrário é o *slow start*.

2.14.1 Slow-start

Esse algoritmo para controle de congestionamento é empregado nas seguintes situações (RFC 4960):

- quando a transmissão de dados se dá por um caminho onde as condições de tráfego são ainda indefinidas;
- quando o caminho pretendido tem uma ociosidade relativamente grande, e o SCTP teve de fazer sondagens para determinar a disponibilidade;

- quando houve perdas de pacotes no caminho;
- após a ocorrência de um *timeout* em um caminho de rede.

As medidas, expostas a seguir, devem ser adotadas ao entrar em vigor o *slow-start*, segundo Stewart [3].

- O valor inicial da variável CWND, antes da transmissão de dados ou após um longo período ocioso (espaço de tempo entre geração de *chunks HEARTBEATS*), deve ser ajustado para $\text{MIN}(4 * \text{MTU}, \text{MAX}(2 * \text{MTU}, 4380\text{B}))$.
- Quando uma estação SCTP não faz transmissão de dados sobre um determinado endereço de transporte, o CWND do endereço de transporte deve ser ajustado para $\text{MAX}(\text{CWND}/2, 4 * \text{MTU})$ a cada RTO.
- O CWND pode aumentar o seu tamanho quando há a junção dos fatores: CWND menor ou igual a Ssthresh; CWND usada em sua totalidade; uma entrada SACK aumenta o *Cumulative TSN ACK Point*; e o emissor de dados não está no modo *Fast Recovery*. O CWND deve ser aumentado em $\text{MIN}((\text{chunks DATA pendentes confirmados}), (\text{MTU do destino}))$.
- O valor inicial da variável CWND, após um *timeout*, não deve ser maior do que $1 * \text{MTU}$.

2.14.2 Congestion avoidance

Esse controle de congestionamento é adotado a partir do momento que a CWND se tornar maior que o Ssthresh, passando, nesse momento, de acordo com a RFC 4960, a vigorar as seguintes regras:

- a *partial_bytes_acked* é inicializada com zero. Essa variável é especificamente para ser utilizada nesse tipo de controle de congestionamento;
- a cada recebimento de um *chunk SACK*, aumentará o *Cumulative TSN ACK Point*; aumentará o *partial_bytes_acked* para o número total de todos os *bytes* de *chunks* confirmados dentro de um SACK, incluindo *chunks* de dados confirmados por novos *Cumulative TSN ACK* e por *Gap Ack Blocks*;
- quando *partial_bytes_acked* é igual ou maior do que o CWND e, antes da chegada do SACK, tendo o emissor pendências de confirmação igual ou maior ao tamanho da CWND (antes da chegada do SACK, pacotes em trânsito com tamanho total maior ou igual ao CWND), o CWND deverá ser incrementado em um MTU e reiniciar *partial_bytes_acked* para $(\text{partial_bytes_acked} - \text{CWND})$;

- do mesmo modo que no *slow start*, quando o emissor não faz transmissão de dados sob um determinado endereço de transporte, o CWND do endereço de transporte deve ser ajustado para $\text{MAX}(\text{CWND}/2, 4*\text{MTU})$ por RTO;
- quando todos os dados transmitidos pelo emissor tiverem sido confirmados pelo receptor, o *partial_bytes_acked* é inicializado com zero.

2.14.3 Comportamento do SCTP em ambientes de perda de pacote

Quando detectados pacotes perdidos por meio de um *chunk SACK*, uma estação SCTP deve seguir as seguintes regras, segundo Stewart [3]:

- $\text{SSTHRESH} = \text{MAX}(\text{CWND}/2, 4*\text{MTU})$
- $\text{CWND} = \text{SSTHRESH}$
- $\text{partial_bytes_acked} = 0$

Basicamente, um pacote perdido causa a redução da CWND pela metade. Na prática, isso acarreta o uso do método de controle de congestionamento *slow start* no limite de alteração para o *congestion avoidance*.

Quando ocorre o *timeout* de um dado transmitido, o SCTP deve executar as seguintes regras, segundo Stewart [3]:

- $\text{SSTHRESH} = \text{MAX}(\text{CWND}/2, 4*\text{MTU})$
- $\text{CWND} = 1*\text{MTU}$

As regras acima listadas asseguram que apenas um pacote de dados SCTP esteja em trânsito pelo caminho primário. Essa condição permanece até que seja recebida a confirmação de sucesso da entrega do pacote. A ocorrência do *timeout* conduz ao uso imediato do controle de fluxo *slow start*, devido à redução repentina no tamanho da CWND, forçando, assim, uma maior permanência nesse tipo de controle de congestionamento até que o caminho primário esteja apto novamente a trabalhar no modo *congestion avoidance*.

2.15 Detecção de caminhos falhos

A detecção de um caminho falho é muito difícil, principalmente em ambientes intermitentes, que fazem o SCTP subutilizar a sua capacidade de transmissão de dados quando há a possibilidade de acesso *multi-homing*.

Uma estação SCTP mantém um contador para o número total de tentativas de retransmissão por caminho e outro por associação. Quando a soma de todas as tentativas de transmissão se esvaem, a pilha SCTP anuncia a incapacidade de contato com a estação SCTP de destino, cabendo a implementação do protocolo SCTP decidir se fecha a associação ou entra em estado de dormência, conforme comentado por Caro et al. [11].

O contador de cada caminho deve ser incrementado a cada RTO. Quando o contador exceder o limite, o caminho deve ser marcado como inativo. No caso da estação SCTP ser *single-homing*, a associação poderá ser imediatamente fechada. Caso seja *multi-homed*, um outro caminho primário deve ser eleito entre os secundários existentes. No caso de um caminho primário ou secundário ser taxado de inativo pela pilha SCTP, ele continuará a ser sondado por *chunks HEARTBEAT*, sem haver incremento do contador (mensagens enviadas e não confirmadas) para que, quando confirmadas, tornem a ser taxados como ativos.

O contador, para cada caminho, deve ser reiniciado a cada *chunk SACK* ou *HEARTBEAT ACK* recebido. Quando configurar uma estação SCTP, o usuário deve evitar ter o valor de contador de tentativas e envio de *chunks* para associação menor do que a soma dos contadores para todos os endereços de destino. De outra forma, todos os endereços de destino podem tornar-se inativos mesmo quando uma estação SCTP considera o receptor alcançável.

Caminhos ociosos, aqueles considerados secundários ou mesmo taxados como inalcançáveis, sempre devem ser sondados por *chunks HEARTBEAT*. O emissor do *HEARTBEAT* tem a informação do momento de envio do *chunk*. O receptor de um *HEARTBEAT* deve imediatamente responder com um *HEARTBEAT ACK*. O receptor do *HEARTBEAT ACK* executa a medida do RTT usando o *timer* transportado dentro do *payload*. É aconselhável que apenas um *chunk HEARTBEAT* seja enviado a cada RTO.

2.16 Valores de parâmetros configuráveis do SCTP

Os seguintes parâmetros são recomendados pela RFC 4960 como padrões para uma estação SCTP:

- *RTO.Initial* = 3 segundos - valor inicial adotado para o RTO quando uma estação SCTP ainda não tem medidas RTT;
- *RTO.Min* = 1 segundo - valor mínimo que pode ser atribuído a um RTO;
- *RTO.max* = 60 segundos - valor máximo que pode ser atribuído a um RTO;
- *Max.Burst* = 4 - quantidade máxima de pacotes transmitidos e ainda não confirmados;
- *RTO.Alpha* = 1/8 - valor padrão usado para cálculo do RTO;
- *RTO.Beta* = 1/4 - valor padrão usado para cálculo do RTO;

- *Valid.Cookie.Life* = 60 segundos - tempo de validade um *cookie* SCTP;
- *Association.Max.Retrans* = 10 tentativas - quantidade máxima de tentativas para retransmissão em uma associação;
- *Path.Max.Retrans* = 5 tentativas - quantidade máxima de tentativas para retransmissão por caminho;
- *Max.Init.Retransmits* = 8 tentativas - quantidade máxima de tentativas de transmissão do *chunk* INIT;
- *HB.Max.Burst* = 1 - quantidade máxima de *chunk HEARTBEAT* transmitido sem confirmação por RTO.

2.17 Comparação entre protocolos da camada de transporte

A tabela 2.2 expõe algumas diferenças/semelhanças entre os protocolos mais comuns de camada de transporte e o SCTP. Essa tabela foi gerada com base nas informações obtidas em Kelly et al. [12], Fu et al. [26] e Lei et al. [27].

Tabela 2.2: Comparativo entre protocolos de transporte. [Legenda NA - Não Aplicável]

Função, Serviço, Comportamento	SCTP	TCP	UDP
Orientado a mensagem	Sim	Não	Sim
Orientado a <i>bytes</i>	Não	Sim	Não
<i>Acks</i> seletivos	Sim	Opcional	Não
Bloqueio <i>head-of-line</i>	Não	Sim	NA
Capaz de usar ECN (<i>Explicit Congestion Notification</i>)	Sim	Sim	Não
Controle de fluxo	Sim	Sim	Não
Controle de congestionamento	Sim	Sim	Não
Descoberta do MTU do caminho	Sim	Sim	Não
<i>Full-Duplex</i>	Sim	Sim	Sim
<i>Multi-homing</i>	Sim	Não	Não
<i>Multistreaming</i>	Sim	Não	Não
Orientado a conexão	Sim	Sim	Não
Permite conexões <i>half-closed</i>	Não	Sim	NA
Preserva limites da mensagem	Sim	Não	Sim
Protege contra ataques de inundação SYN	Sim	Não	NA
Pseudocabeçalho para <i>checksum</i>	Usa Vant.	Sim	Sim
Transferência confiável de dados	Sim	Sim	Não
Transferência não ordenada de dados	Sim	Não	Sim
Transferência ordenada dos dados	Sim	Sim	Não
Transferência parcial confiável de dados	Opcional	Não	Não
Verificação de acessibilidade	Sim	Sim	Não
<i>Multi-Homing</i> dinâmico	Opcional	Não	Não
Autenticação	Opcional	Opcional	NA

Capítulo 3

mSCTP - Mobile Stream Control Transmission Protocol

3.1 Visão geral do mSCTP

Devido à popularização das tecnologias *wireless* nos últimos anos, hoje podemos encontrar facilmente múltiplas redes adjacentes, com padrões de comunicações iguais e/ou diferentes.

A crescente necessidade de prover acessibilidade em todos os lugares e a qualquer momento ao usuário poderá prontamente ser atendida pelo uso de mecanismos de controle de gerenciamento de mobilidade proporcionados pelo uso do *multi-homing* do SCTP, conforme Fallon et al. [5], Siddiqui et al. [13], Wang et al. [28] e Randhawa [29]. O mSCTP pode proporcionar um *handover* categorizado como horizontal/vertical que utiliza técnicas de *soft-handover/hard-handover* e busca manter a conexão com serviços preestabelecidos ativos em cenários de mudança de rede devido à movimentação das unidades móveis.

Basicamente existem várias soluções para tratamento de mobilidade sem perda de conexão dos serviços preestabelecidos. As propostas mais populares se encontram nas camadas de rede, transporte e aplicação em relação ao padrão TCP/IP.

O protocolo SCTP foi concebido para trabalhar em estruturas de rede cabeada, ou seja, pensado inicialmente para ambientes com dispositivos fixos. Uma das principais características do SCTP, no caso *multi-homing*, de acordo com Xie et al. [8], é tolerância a falhas, pelo uso de mais de um endereço IP como destino para transferência de dados. Essa é a característica do protocolo SCTP que é trabalhada para permitir mobilidade.

A partir da característica de *multi-homing* do protocolo SCTP (RFC 2960 de 2000), surgiram publicações sobre o uso do SCTP para gerenciar mobilidade. Como a RFC do protocolo SCTP não suporta mobilidade, começaram a surgir *drafts* sobre o assunto a partir de 2002 e uma variedade de artigos propondo essa possibilidade, conforme Ezzouhairri et al. [30],

Kelly et al. [12] e Stewart et al. [31].

Para o protocolo SCTP trabalhar com a inserção dinâmica de endereços IPs em uma associação já constituída, foi publicada uma extensão, no caso, a RFC 5061 (mSCTP). A RFC concretiza a técnica de inserir/deletar dinamicamente um endereço IP em uma associação, além de ter um mecanismo para determinar qual será o endereço primário a ser utilizado na transferência de pacotes SCTP.

Essa adaptação ao protocolo além de suporte a IPs dinâmicos, permite também interfaces dinâmicas, conforme Tüexen et al. [27].

Essa nova RFC (5061) atribui mais dois novos tipos de *chunks* e doze novos tipos de estruturas TLVs para auxiliá-los. Cinco estruturas TLVs são referentes a causas de erro que podem ser desencadeados devido ao processo de inserir/deletar, ou ajustar um novo endereço IP como primário em uma associação SCTP.

3.2 Trabalhando com IPs dinâmicos em uma associação

Como já mencionado anteriormente, o SCTP foi projetado como um protocolo de transporte de uso geral. A sua característica de *multi-homing* foi inicialmente pensada para ser usada como um mecanismo de tolerância a falhas, mas, no decorrer dos anos, foram apresentados vários artigos propondo o uso dessa técnica para balanceamento de carga, como em Caro et al. [16] e Iyengar et al. [17], e mobilidade, em Ezzouhairri et al. [30], Kelly et al. [12], Stewart et al. [31] e Stewart et al.[32]. A última se tornou mais próspera em relação ao uso desvirtuado do propósito original do *multi-homing* no SCTP.

Na proposta original do SCTP, a funcionalidade de *multi-homing* é permitir redundância ao caminho primário. Na configuração de uma associação, os endereços IPs pelo qual cada uma das partes estão respondendo, são apresentados especificamente durante o envio dos *chunks INIT* e *INIT ACK* encapsulados dentro de estruturas TLVs. Uma vez apresentadas, não poderiam ser mais alteradas, caso contrário, sumariamente uma das partes fecharia a associação por meio do envio de um *chunk ABORT*.

Para proporcionar o uso dinâmico de endereços IPs em uma associação já estabelecida, foi proposta uma extensão ao SCTP denominada mSCTP (RFC 5061), que propõe o uso de novos tipos de *chunks*, denominados *ASCONF* (*Address Configuration*) e *ASCONF-ACK*, representados na figura 3.1. Por intermédio desses novos *chunks*, é possível adicionar/deletar IPs de uma associação sem causar o seu fechamento (RANDHAWA [29]). Esses novos *chunks* devem ser enviados de modo autenticado de acordo com mecanismo definido em Tüexen et al. [33].

Por meio do *chunk ASCONF*, é possível inicializar as seguintes operações:

- adicionar endereços IPs a uma associação;

- deletar endereços IPs de uma associação;
- definir o endereço primário de uma associação para transmissão de pacotes SCTP.

Como já foi comentado nas seções 2.5 e 2.10, o SCTP não permite fazer uso de mais de um endereço de destino IP ao mesmo tempo para transmissões de dados do usuário, exceto em casos de retransmissão. Os dados são sempre transmitidos pelo caminho taxado como primário.

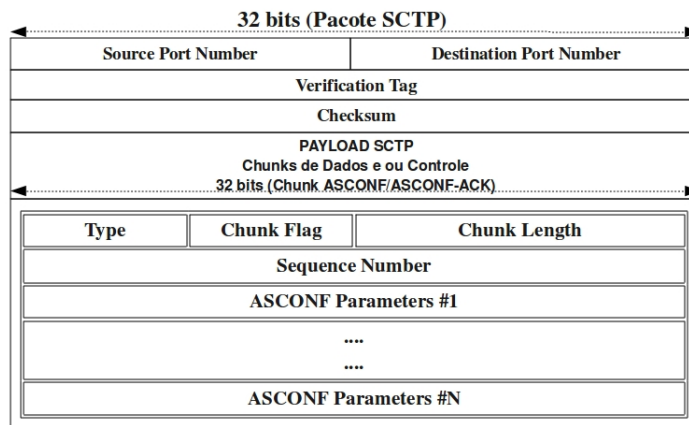


Figura 3.1: Representação genérica de um pacote SCTP com um *Chunk ASCONF/ASCONF-ACK*

O *chunk ASCONF-ACK* (figura 3.1) é gerado como resposta ao recebimento do *chunk ASCONF*. O recebimento do *chunk ASCONF-ACK*, necessariamente, não significa o sucesso da operação iniciada por parte do gerador do *chunk ASCONF*. O sucesso da operação pode ser reportada referindo-se a todas as estruturas TLVs presentes no *chunk ASCONF* ou de forma individualizada. As estruturas TLVs que podem estar presentes no *chunk ASCONF* se referem a adicionar/deletar endereços IPs da associação e ajustar o endereço primário.

O *chunk ASCONF ACK* pode ser reportado sem nenhuma estrutura TLV, interpretada, dessa maneira, como tido sucesso em todas as solicitações enviadas por meio do *chunk ASCONF*.

3.3 Novas estruturas TLVs do mSCTP

Doze novas estruturas TLVs foram adicionadas àquelas já documentadas na RFC 4960 para proporcionar o uso de IPs de forma dinâmica (inserir/deletar) em uma associação já estabelecida.

O uso de estruturas TLVs, como as do padrão SCTP, é dependente dos tipos de *chunks*. O desrespeito em relação à restrição de uso poderá conduzir ao encerramento automático da associação. As estruturas TLVs se apresentam dentro de *chunks ASCONF* como *ASCONF Parameters*, conforme apresentado na figura 3.1.

Nas tabelas 3.1 e 3.2 estão listadas as novas estruturas TLVs e os *chunks* aos quais estão restritos o seu uso.

Tabela 3.1: Novas estruturas TLVs, conforme RFC 5061.

Estrutura TLV	INIT	INIT ACK	ASCONF	ASCONF ACK	Funcionalidade
<i>Set Primary Address</i>	X	X	X	-	Delega um novo endereço primário para a associação.
<i>Adaptation Layer Indication</i>	X	X	-	-	Repassado para a camada de aplicação.
<i>Supported Extensions</i>	X	X	-	-	Informa extensões de <i>chunks</i> suportadas na associação.
<i>Add IP Address</i>	-	-	X	-	Adiciona um endereço IP à associação.
<i>Delete IP Address</i>	-	-	X	-	Deleta um endereço IP da associação.
<i>Error Cause Indication</i>	-	-	-	X	Uma operação de erro reportada.
<i>Success Indication</i>	-	-	-	X	Indicação de sucesso da operação.

Tabela 3.2: Novas estruturas TLVs para tratamento de erros, conforme RFC 5061

Estrutura TLV - Causas de erro	Reportar erros
<i>Request to Delete Last Remaining IP Address</i>	Transações com IPs
<i>Operation Refused Due to Resource Shortage</i>	Transações com IPs
<i>Request to Delete Source IP Address</i>	Transações com IPs
<i>Association Aborted Due to Illegal ASCONF-ACK</i>	Segurança
<i>Request Refused - No Authorization</i>	Segurança

3.4 SCTP e mSCTP multi-homings

Para demonstrar a funcionalidade do *multi-homing* SCTP e também as facilidades proporcionadas pelo mSCTP por permitir inserir/deletar/alterar destino primário, alguns experimentos foram feitos utilizando como plataforma o sistema operacional FreeBSD, no qual já está presente de forma nativa, conforme Tüxen et al. [27].

O SCTP trabalha com a premissa de que as rotas lhe serão fornecidas e que a associação simplesmente fará uso delas. Essas rotas podem ser inseridas manualmente ou por algum mecanismo que o faça dinamicamente.

Como já comentado anteriormente, inserir/apagar IPs de uma associação só é possível pelo uso da extensão do SCTP no caso o mSCTP.

A qualquer momento que um IP é inserido ou alterado em uma estação onde há uma associação vigente, é solicitado automaticamente, pelo protocolo SCTP, a sua inserção na outra extremidade da associação por meio do envio do *chunk ASCONF*, que naturalmente responderá com um *chunk ASCONF ACK* informando o seu sucesso ou falha.

O SCTP também aceita naturalmente que ocorra uma nova inserção de interface de comunicação, como, por exemplo, um *usb-wireless* ou *usb-ethernet*. Consequentemente,

um novo IP será atribuído para que se possa fazer uso do novo caminho, cabendo, então, ao protocolo mSCTP fazer a inserção desse novo IP em uma associação já vigente se for o caso.

O mecanismo, para excluir um IP de uma associação vigente, não é automaticamente propagado de uma estação SCTP para outra. É preciso ter uma ação direta do usuário para que isso ocorra. Esse processo é feito pelo envio de um *chunk ASCONF* que contém como *payload* uma estrutura TLV *Delete IP Address* para que isso ocorra. O receptor responderá com um *chunk ASCONF ACK* informando o sucesso ou o fracasso da operação.

3.5 Gerenciamento de mobilidade com o SCTP-mSCTP

No cenário apresentado na figura 3.2, temos um MN (*Mobile Node*), um CN (Correspondente *Node*) e dois *Access Point* (AP1 e AP2). O MN possui duas interfaces. A *Net1* (interface de rede 1) se encontra dentro da sub-rede 192.168.0.0/24, tendo como IP 192.168.0.10. A segunda interface (*Net2*) não tem IP atribuído a ela (ND - não definido). O CN tem apenas uma interface, a qual se encontra no momento na rede 10.10.10.0/24, tendo como IP 10.10.10.50.

Em um primeiro momento, conforme representado na figura 3.2, não existe uma associação SCTP estabelecida entre MN e CN. Após o estabelecimento da associação e a troca de alguns pacotes de dados, o MN começa a se movimentar em direção a outro *Access Point* (no caso AP2).

De acordo com a figura 3.2, temos a seguinte cronologia na troca de mensagens:

1. envio por parte do MN do *chunk INIT* com a lista de endereços IPs que o representa, iniciando, dessa forma, a configuração da associação;
2. envio por parte do CN do *chunk INIT ACK* com a sua lista de endereços IPs, é a segunda mensagem de um total de quatro para configurar a associação;
3. envio por parte do MN do *chunk COOKIE ECHO*, sendo a terceira mensagem de um total de quatro para configurar a associação;
4. envio por parte do CN do *chunk COOKIE ACK*, sendo a quarta e última mensagem de controle para terminar de configurar uma associação;
5. envio de *chunk DATA* do CN para o MN;
6. envio de *chunk DATA* do CN para o MN e movimentação em direção ao AP2;
7. envio de *chunk SACK* do MN para o CN, e, em paralelo, ocorre associação da interface *Net2* do MN para com o AP2;
8. envio de *chunk DATA* do CN para o MN, e, em paralelo, ocorre a associação da interface *Net2* do MN para com o AP2;

9. solicitação do MN para o CN para inserção do IP 11.11.11.30 na associação SCTP;
10. confirmação por parte do CN para o MN da inserção do IP 11.11.11.30 na associação SCTP;
11. envio de *chunk SACK* do MN para o CN;
12. envio de um pacote *HEARTBEAT* para sondar a conectividade do novo endereço IP;
13. se tudo correr bem, o emissor do *chunk HEARTBEAT* receberá como resposta um *chunk Heartbeat-Ack* trazendo a confirmação de conectividade e informações que ajudaram a traçar o RTT e o RTO do novo caminho;
14. solicitação por parte do MN em direção ao CN para alteração do IP primário para envio de dados de 192.168.0.10 para 11.11.11.30;
15. confirmação por parte do CN para o MN da alteração do IP primário na associação SCTP;
16. solicitação por parte do MN em direção ao CN para excluir o IP 192.168.0.10 da associação SCTP;
17. confirmação por parte do CN para o MN da exclusão do IP da associação SCTP;
18. envio de *chunk DATA* do CN para o MN;
19. confirmação de recebimento do *chunk DATA* pelo envio de um *chunk SACK* do MN para o CN.

O custo do processo de associação da interface *wireless* com um *access point* não faz parte do escopo do mSCTP. Dependendo de como foi implementado o protocolo SCTP, o processo de inserir/deletar IP em uma associação pode não ser automatizado e, por isso, passará a depender de implementações extras da aplicação, já que a RFC do SCTP não aborda esse nível de detalhamento.

As diretrizes utilizadas pelo mSCTP para alteração do endereço IP de destino dos pacotes de dados, devido à perda de conectividade, continuam utilizando as diretrizes definidas pela RFC 4960 do SCTP. E o fator usado pelo SCTP para determinar se um caminho é falho e solicitar a alteração do endereço de destino é baseado no não recebimento da confirmação de pacotes de dados SCTP enviados. O custo desse tempo poderá não atender à necessidade da aplicação.

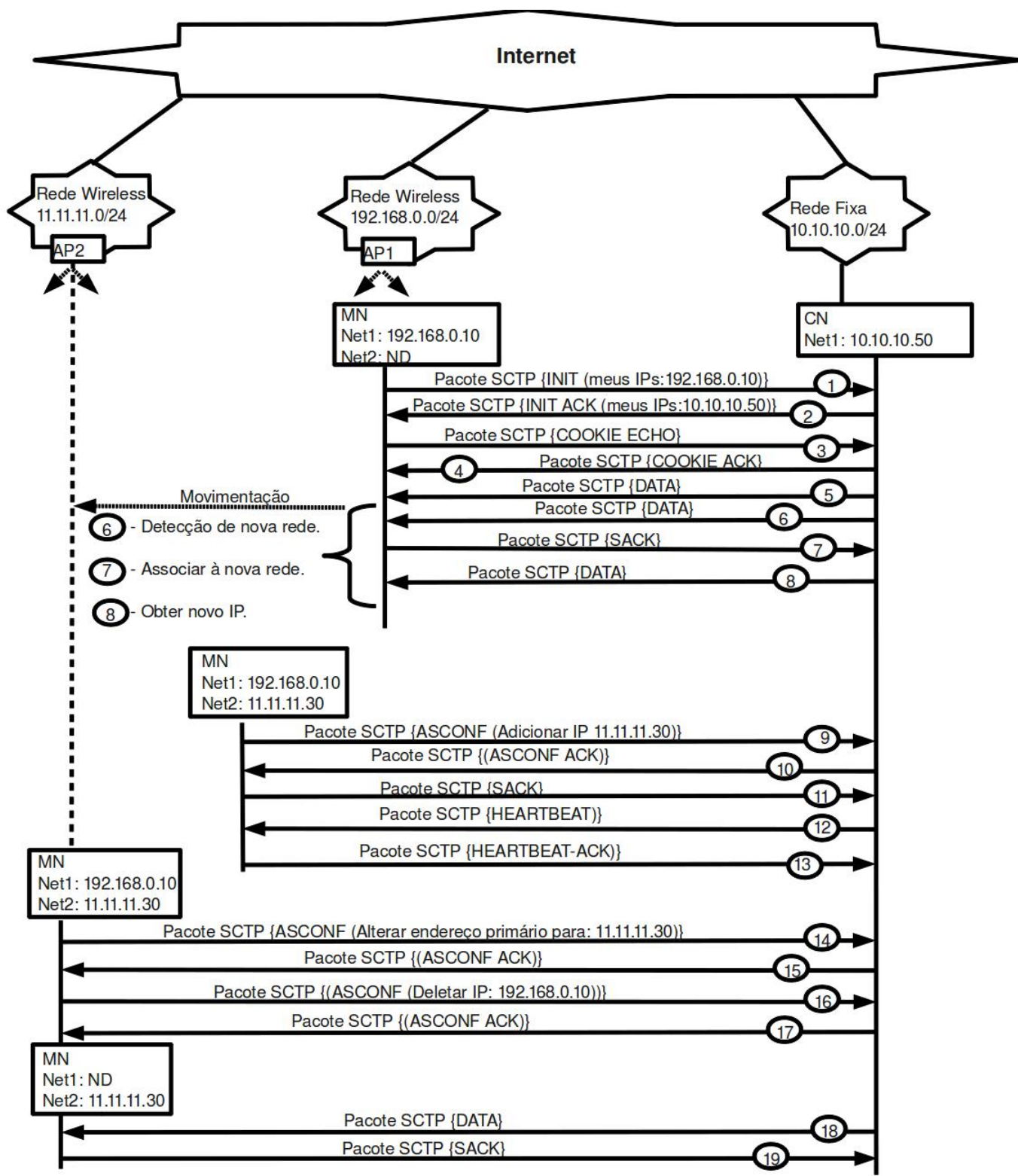


Figura 3.2: Gerenciamento de mobilidade usando o mSCTP

Capítulo 4

Handover no SCTP

4.1 Política de tratamento de handover no SCTP

Basicamente em sua composição simplória, a estação SCTP faz uso de duas ou mais interfaces de comunicação de rede (*wireless* e/ou cabeada) pertencentes a diferentes redes IPs e, a partir de um determinado contexto, é trocada a saída para envio de pacotes devido ao caminho primário não estar mais atendendo às expectativas de transmissão de dados impostos pelo protocolo SCTP.

O critério do protocolo SCTP, para realizar a troca do caminho-destino dos pacotes, é condicionado ao número de tentativas de transmissões consecutivas de pacotes de dados SCTP e o não recebimento de confirmação. Após a ocorrência do primeiro *timeout*, envio de um pacote de dados e o não recebimento de confirmação, por padrão, deverão ocorrer cinco novas tentativas de transmissões consecutivas pelo caminho primário sem recebimento de confirmação, antes que o caminho seja taxado como inalcançável, elegendo-se, a partir dessa condição, um novo caminho como destino primário para os pacotes de dados.

Aparentemente, cinco tentativas de transmissão por um caminho e o não recebimento de confirmação parecem um número razoável para se fazer o *handover*, no entanto existe o intervalo entre essas tentativas. Na melhor das hipóteses, essas cinco tentativas são realizadas em um intervalo de tempo de 63 segundos e, na pior das hipóteses, elas ocorreriam em um intervalo de tempo de 360 segundos, considerando os valores de parâmetros sugeridos pela RFC 4960 do SCTP, como, por exemplo, RTO.Min e RTO.Max.

O uso do caminho secundário para transmissão de pacotes de dados, no período de detecção do primeiro *timeout* até a ocorrência do *handover*, é mínimo. Os *chunks SACKs* reportam o maior TSN (*chunk DATA*) já recebido e o maior TSN ao qual todos os seus antecessores já foram confirmados, e a diferença entre os dois após 30 segundos de transmissão pode alcançar dezenas de pacotes. Essa diferença reportada a cada *SACK* poderá ser zerada no intervalo entre o período da primeira detecção de *timeout* até a ocorrência do *handover*, pois esses pacotes poderão ser retransmitidos pelo caminho secundário durante esse intervalo

(essa opção é dependente de implementação do protocolo SCTP).

Em um contexto de transferência de dados entre aplicações sem requerimento de tempo real, como no caso de *downloads* da internet, na melhor das hipóteses de *handover* no SCTP, o tempo gasto é tolerável, mas não seria uma escolha razoável para esse tipo de serviço, já no pior caso, esse intervalo seria inaceitável, pois a chegada de pacotes aconteceria com atrasos cada vez maiores até que realmente ocorresse o *handover*.

Os valores de 63 a 360 segundos podem sofrer algumas variações devido ao que realmente ocorre em um contexto de transição de endereços de destino SCTP. Essa variação no tempo de *handover* é detalhado mais abaixo de acordo com a figura 4.1, na qual existem duas estações SCTP, cada uma com duas interfaces de rede. A interface1 do *host1* envia e recebe pacotes para/da interface1 do *host2*. Mesmo caso ocorre com a interface2 do *host1*, que envia e recebe dados para/da interface2 do *host2*.

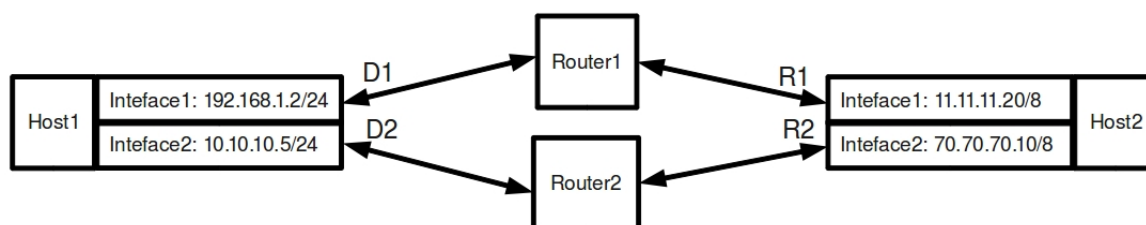


Figura 4.1: Interligação entre duas estações SCTP

Presumindo que já exista uma associação estabelecida entre *host1* e *host2*, sendo as interfaces1 taxadas como primárias para a transmissão de dados, analisemos o contexto, descrito logo abaixo, para se obter o tempo total de *handover*. As regras do SCTP determinam que, a cada *timeout* ocorrido em um caminho, o valor do RTO seja dobrado.

O protocolo SCTP utiliza seus parâmetros configuráveis com valores sugeridos pela RFC 4960, como, por exemplo, o limiar de retransmissões consecutivas a ser alcançado para que ocorra o *handover*, sendo aqui considerado o valor cinco. O valor do RTO no momento da ocorrência do primeiro *timeout* estava em um segundo. Nesse contexto, cronologicamente teríamos os seguintes acontecimentos (obs. também apresentados na forma gráfica como demonstrado na figura 4.2.):

1. um pacote é enviado pelo caminho primário;
2. após um segundo de espera e o não recebimento de confirmação, um novo pacote é transmitido pelo caminho primário, e o pacote perdido é retransmitido pelo caminho secundário. Agora o RTO do caminho primário passou para dois segundos, e o contador de perdas para o caminho é incrementado de zero para um;
3. após dois segundos de espera e o não recebimento de confirmação do pacote enviado pelo caminho primário e o recebimento de confirmação do pacote retransmitido pelo

caminho secundário, um novo pacote é transmitido pelo caminho primário, enquanto que o pacote perdido é retransmitido pelo caminho secundário. O RTO do caminho primário passa a quatro segundos, e o contador de perdas para o caminho passa para dois;

4. após quatro segundos de espera, ocorreu o mesmo caso que o item anterior, e o RTO do caminho primário passou a oito segundos, e o contador de perdas passou para três;
5. após oito segundos de espera, ocorreu o mesmo caso que o item anterior, e o RTO do caminho primário passou a dezesseis segundos e o contador de perdas passou para quatro;
6. após dezesseis segundos de espera, ocorreu o mesmo caso que o item anterior, e o RTO do caminho primário passou a trinta e dois segundos e o contador de perdas passou para cinco;
7. após trinta e dois segundos de espera e o não recebimento de confirmação do pacote enviado pelo caminho primário, o protocolo SCTP atinge o valor máximo de tentativas de transmissão consecutiva pelo caminho primário sem obter uma confirmação e automaticamente começa a utilizar o caminho secundário como primário para transmissão de pacotes, e o caminho primário (original) é taxado como inalcançável até segunda ordem;
8. se o caminho primário ficar muito intermitente, ou seja, ora confirma pacotes, ora não confirma, o sistema poderá ficar por um tempo razoável em estado de quase *handover*, conseqüentemente, haverá um prolongamento de tempo para efetuar o *handover*. Isso é muito prejudicial para aplicações, principalmente, aquelas mais sensíveis ao atraso, uma vez que a cada confirmação o contador de perdas de pacote do caminho é inicializado com zero.

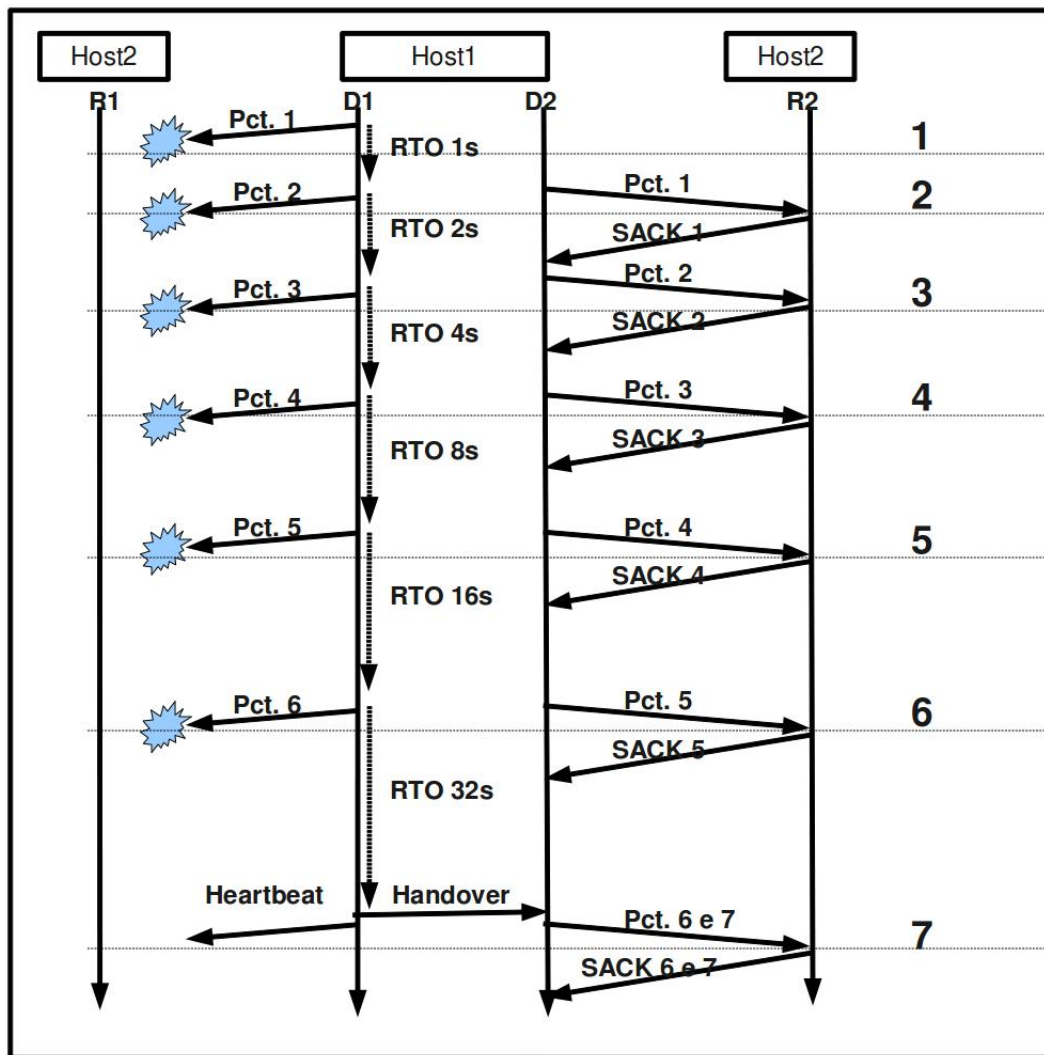


Figura 4.2: Tratamento do SCTP para realizar o *handover* quando $PMR=5$, $RTO.Min=1s$, $RTO.Max=60s$ e $RTO.Atual=1s$

A RFC 4960 do SCTP fala que, a partir do momento que o atual caminho primário não apresenta condições para transmissão de pacote de dados, o protocolo SCTP elegerá um dos seus caminhos secundários para continuar a transmissão de pacotes de dados dentro da associação vigente. As implementações do SCTP para os sistemas operacionais *Linux* e *FreeBSD* fazem um processo de eleição temporária de um caminho secundário como primário até que o antigo caminho primário apresente novamente conectividade ou até que a associação termine. Se o antigo caminho primário apresentar conectividade novamente, não levando em consideração a largura de banda disponibilizada, o protocolo SCTP passa a utilizá-lo automaticamente como destino primário, independentemente da atual situação do vigente caminho primário.

Na RFC do SCTP, não se fala explicitamente de eleição temporária do caminho primário, no caso de falha do original, mas esse comportamento é encontrado nas implementações, como visto em Kelly et al. [12], Caro et al. [11], Fracchia et al. [34], Grace et al. [7] e Caro

et al. [15].

Segundo Caro et al. [14], a mudança permanente do caminho primário pode melhorar o desempenho uma vez que, se houver o retorno para o antigo caminho primário, o controle de congestionamento vigente será o *slow start* com a CWND de um MTU. Também Caro et al. [14] afirmam que não existe especificação por parte da RFC do SCTP de no caso de falha do caminho primário qual critério deve ser utilizado para a escolha do caminho secundário que exercerá a função de primário, para continuar a transmissão dos pacotes de dados, se houver mais de um disponível.

4.2 Parâmetros SCTP que influenciam na política handover

O desempenho satisfatória do protocolo SCTP, principalmente em redes *wireless*, tende a depender fortemente de ajustes nos parâmetros do protocolo SCTP (GRACE et al. [7]) que influenciam de maneira direta/indireta no momento do *handover* e, conseqüentemente, no tempo total de transmissão de dados. Entre esses parâmetros, podemos destacar:

- *Path.Max.Retrans* (PMR) - que representa a quantidade máxima de tentativas de transmissão de dados por um caminho sem que haja confirmação de recebimento. Como demonstrado por Caro et al. [14] e Grace et al. [7], isso influencia diretamente no tempo decorrido na transferência de dados. O valor padrão (RFC 4960) é de cinco tentativas;
- *RTO.Min* - que representa qual o menor tempo de espera pela confirmação de recebimento de um dado. O valor padrão (RFC 4960) é de um segundo;
- *RTO.Max* - que representa qual o maior tempo de espera pela confirmação de recebimento de um dado. O valor padrão (RFC 4960) é de 60 segundos;
- *HB.interval* - que representa o tempo decorrido entre o envio de cada *chunk* de sondagem (*HEARTBEAT*) para verificar a alcançabilidade de um caminho e, conseqüentemente, calcular o RTT. O valor padrão (RFC 4960) é de 30 segundos. Para caminhos primários intermitentes, o valor dessa variável influencia de maneira direta o desempenho da transmissão de um arquivo pela rede.

Capítulo 5

Trabalhos relacionados

5.1 Organização

Os trabalhos relacionados apresentam novas soluções para gerenciar o *multi-homing* do SCTP buscando diminuir a latência no procedimento de tomada de decisão de *handover*.

Os artigos são apresentados conforme a similaridade das propostas em tomar decisões de *handover*.

Os artigos são apresentados da seguinte forma:

- ***handover* baseado na largura de banda** - as tomadas de decisões do momento da troca do caminho primário são baseadas principalmente na largura de banda disponibilizada pelo caminho primário e secundário. Nesse contexto, enquadram-se os artigos de 1 a 3;
- ***handover* baseado no atraso** - as tomadas de decisões do momento da troca do caminho primário são baseadas principalmente em medidas de atraso. Nesse contexto, enquadram-se os artigos 4 a 5;
- ***handover* baseado em perda de conectividade** - as tomadas de decisões do momento da troca do caminho primário são baseadas principalmente na perda de conectividade. Nesse contexto, enquadram-se os artigos de 6 a 8;
- ***Handover* baseado em RTT/SRTT** - as tomadas de decisões do momento da troca do caminho primário são baseadas no valor nominal do RTT/SRTT. Nesse contexto, enquadra-se o artigo 9 a 10.

5.1.1 Artigo 1 - Handover baseado na largura de banda

5.1.1.1 Descrição da proposta

Nesse artigo apresentado por Kashihara et al. [35], é proposto ao SCTP *multi-homing* um novo mecanismo para seleção de rede que busca atender, dessa forma, a aplicações de tempo-real em ambientes com redes sobrepostas.

Na proposta padrão do SCTP, a tomada de decisão do *handover* só ocorre quando for detectada a falha no atual caminho primário. O tempo tomado para diagnosticar que o caminho primário é falho geralmente leva dezenas de segundos. Nesse artigo, os autores alteram essa percepção de quando deve trocar o caminho primário, baseando-se no tamanho da largura de banda disponível em relação à necessidade imposta pela aplicação e de maneira secundária utiliza o valor do SRTT para tomada de decisão, quando, por exemplo, nenhum dos caminhos atende à necessidade da aplicação com relação à largura de banda.

Para conseguir implementar a solução, os autores fizeram modificação no funcionamento do protocolo SCTP, desabilitaram seu controle de congestionamento, confirmação de pacotes recebidos e passaram a enviar pacotes de sondagem *HEARTBEATS* para todos os caminhos, inclusive o primário. Nessa nova proposta, os pacotes de sondagem são de 60B.

Os pacotes de sondagem passaram a ser transmitidos a cada segundo a partir do momento do estabelecimento da associação. Com o resultado da sondagem feita a cada segundo, foi calculado o RTT e o BBW (largura de banda do caminho). Como o RTT flutua muito durante o processo de sondagem, foi usado o SRTT como um fator mediador dessa oscilação.

A fórmula de cálculo do SRTT foi feita da seguinte maneira: $SRTT = \alpha SRTT + (1 - \alpha) * RTT$ ($0 \leq \alpha \leq 1$). Obs.: para o experimento demonstrado no artigo, o valor de α é 0.9.

De acordo com os autores, se houver perda de pacotes *HEARTBEATS*, o cálculo do SRTT será computado tomando como base o valor do RTO, ao invés de utilizar o valor do RTT.

A fórmula utilizada para calcular a largura de banda (BBW) do caminho também foi feita a partir dos pacotes de sondagem *HEARTBEAT*. Na prática, são transmitidos dois pacotes de sondagem obtendo o RTT de cada pacote. O tamanho do pacote transmitido dividido pela diferença dos RTT dos dois pacotes consecutivos resulta na largura de banda do caminho. $BBW = (HB \text{ packet size}) / (\text{diff-Time})$.

Os coeficientes e os limiares adotados na solução proposta estão listados conforme a tabela 5.1.

Tabela 5.1: Lista de parâmetros e coeficientes

Parâmetro	Função
HB.Interval	Intervalo entre a geração de pacotes de sondagem. O valor adotado foi 1segundo.
RTT	Intervalo de tempo computado da diferença, entre o envio e um pacote de sondagem e o recebimento de confirmação.
SRTT	Métrica utilizada para auxiliar o momento da tomada de decisão de <i>handover</i> . Esse valor, na verdade, é um valor moderado das métricas RTT.
BBW	Representa a largura de banda disponível em um determinado caminho no momento.
Ba	Largura de banda requerida pela aplicação.
Br	Largura de banda requerida pela aplicação adiciona um fator de segurança.
γ	Coeficiente de segurança usado na computação do Br. Esse é menor ou igual a 1.
β	Fator de segurança usado quando comparado à largura do caminho primário com o do caminho secundário. Esse valor é maior ou igual a 1.
ϵ	Fator de segurança usado quando comparado o SRTT do caminho primário com o do caminho alternativo.
d	Período de avaliação da situação. O valor usado para o experimento foi 5 segundos.

Os autores se baseiam em quatro regras para tomada de decisão do momento de *handover*.

- **Caso 1** – somente a BBWp (largura de banda primária) atende à necessidade de banda imposta pela aplicação (Br).
- **Caso 2** – somente a BBWa (largura de banda secundária) atende à necessidade de banda imposta pela aplicação (Br). Essa situação deve ser observada durante um período de **d** segundos.
- **Caso 3** – tanto BBWp como BBWa atendem a Br. Nesse caso, o destino primário só é alterado quando atendidas as condições listadas logo abaixo por um período de **d** segundos.

Condição1: ($\beta BBWp < BBWa$)

Condição2: ($SRTTp > \epsilon BBWa$)

- **Caso 4** – tanto BBWp como BBWa não atendem a Br. Nesse caso, o destino primário só é alterado quando atendidas as condições listadas logo abaixo por um período de **d** segundos.

Condição1: ($\beta BBWp < BBWa$)

Condição2: ($SRTTp > \epsilon BBWa$)

5.1.1.2 Comentário sobre a proposta

O primeiro ponto a ser considerado é que essa proposta altera profundamente o protocolo SCTP a ponto de ele deixar de ser um protocolo de propósito geral para a finalidade específica de transporte de aplicações de tempo-real sem confiabilidade na entrega.

Um levantamento a ser considerado também é um constante fluxo de mensagens de controle em todos os caminhos, podendo desencadear sérios problemas em ambientes *half-duplex*, como *wireless 802.11*, ou mesmo pensando de forma macro com vários nós usando esse mesmo protocolo.

Como a proposta utiliza pacotes de sondagem para calcular o RTT, SRTT e o BBW, eles podem obter dados imprecisos da real condição do meio devido a ser mais fácil propagar pequenos pacotes pela rede *wireless* em relação a grandes pacotes.

A tomada de decisão de alteração do destino primário é baseada na comparação entre a largura de banda requerida pela aplicação (B_r) e a largura de banda disponibilizada pelo caminho (BBW). O artigo não deixa claro o que faz disparar essa comparação.

Quando a BBWp não atende a B_r , deverá ser feita uma comparação durante um intervalo de d segundos para, só após isso, tomar a decisão de *handover*. Se por ventura houvesse a interrupção do meio primário, a permuta do caminho só seria efetuado após cinco segundos. Esse intervalo, provavelmente, não seria bem digerido pela aplicação.

5.1.2 Artigo 2 - Handover baseado na largura de banda

5.1.2.1 Descrição da proposta

Nesse artigo apresentado por Funasaka et al. [6], é proposta uma alteração na solução submetida por Kashihara et al. [35] com finalidade de torná-lo mais eficiente. Nessa nova proposta, é inserido também o atraso máximo para chegada de pacotes como uma condição também a ser observada na hora da tomada de decisão de *handover*.

Conforme levantamento apontado no artigo, a solução proposta por Kashihara et al. [35] apresenta as seguintes desvantagens:

- quando ambos os caminhos apresentam uma largura de banda que atenda à aplicação, a troca do caminho primário é desnecessária se o atraso no caminho secundário for maior que o atraso detectado no meio primário;
- condições importantes a serem consideradas no momento da troca do caminho primário são omitidas, como, por exemplo, o atraso máximo permitido pela aplicação para o recebimento de pacotes de dados;
- quando dois caminhos apresentam diferentes características de atraso em seu meio, a aplicação tende a experimentar grandes atrasos após a troca do caminho primário.

No artigo, é inserido um novo condicionador a ser considerado no momento da alteração do atual caminho primário, tomando como base a comparação entre os atrasos no recebimento de pacotes e o limiar de atraso imposto pela aplicação para o recebimento de pacotes de dados.

As regras para tomada de decisão sobre a troca do caminho primário foram um pouco alteradas para atender à nova proposta:

- **Caso 1** – a regra permaneceu a mesma documentada por Kashihara et al. [35];
- **Caso 2** – a alteração do caminho primário só irá ocorrer se, após o período de avaliação ϕ , no caso 30 milissegundos, o atraso na chegada de pacotes no caminho secundário se mantiver menor do que no atual caminho primário;
- **Caso 3** – a alteração do caminho primário se dará pelas seguintes situações: condição 1 e condição 2 serem verdadeiras ou condição 3 ser verdadeira. (T_p e T_a representam o atraso detectado, respectivamente, no caminho primário e secundário e D_r é o atraso máximo requerido pela aplicação)
Condição 1: ($\beta BBW_p < BBW_a$)
Condição 2: ($SRTT_p > \epsilon BBW_a$)
Condição 3: ($T_a < D_r$ e $T_p > D_r$) durante um período de avaliação ϕ
- **Caso 4** – a alteração do caminho primário só será efetuada se as seguintes condições forem satisfeitas:
Condição 1: ($\beta BBW_p < BBW_a$)
Condição 2: ($T_p > T_a$) durante um período de avaliação ϕ

5.1.2.2 Comentário sobre a proposta

Como essa proposta é baseada na proposta apresentada por Kashihara et al. [35], praticamente todas as observações citadas anteriormente são válidas.

O fluxo de pacotes de sondagem, nessa nova proposta, aumenta muito se comparado com a proposta original apresentada por Kashihara et al. [35], pois a sondagem, que era feita a cada um segundo, passou a ser feita a cada 30 milissegundos. Esse fluxo de pacotes de controle paralelo ao de dados pode influenciar significativamente na percepção de largura de banda de outras aplicações que possam usar segmentos de rede que façam parte dessa associação SCTP.

Quando os caminhos se encontram no Caso 3 e Caso 4, poderá haver uma exagerada troca do caminho primário, quando esse algoritmo for submetido a ambientes intermitentes com relação ao recebimento de pacotes de dados.

5.1.3 Artigo 3 - Handover baseado na largura de banda

5.1.3.1 Descrição da proposta

O artigo apresentado por Fracchia et. al [36] submete uma nova proposta de protocolo chamado WiSE (baseado na alteração do SCTP), que, em ambientes *wireless*, busca discernir a perda de pacotes de dados ocasionada por congestionamento daquelas causadas por interferências no meio. A partir dessa informação, o WiSE aplica um controle de fluxo mais apropriado de acordo com o originador da falha, buscando, com isso, otimizar recursos e garantir a continuidade do serviço sem perdas para a aplicação.

Com o WiSE, os autores pretendem discernir perdas relacionadas a congestionamento na rede daquelas decorridas do meio de comunicação *wireless*. Os autores utilizam esse protocolo para, eficientemente, usar a disponibilidade da tecnologia *multi-homing* para sempre utilizar o melhor caminho disponível para a aplicação.

Para selecionar o melhor caminho, os autores se baseiam no cálculo da largura de banda disponível em cada caminho. Utilizam o WiSE para aplicar uma eficiente redução do CWND no caminho primário de acordo com o motivo que ocasionou a perda do pacote de dados.

Na proposta dos autores, a tomada de decisão para alterar o caminho primário acontece a cada *timeout*. Após isso, é feita a comparação da largura de banda disponível do caminho primário com o do secundário e realizar a troca de caminho somente se a largura de banda do caminho secundário for maior do que a do caminho primário. Para que não ocorra o efeito “ping-pong”, a alternância do caminho primário só poderá ser realizada no mínimo a cada 60 segundos.

A estimação da largura de banda é realizada no caminho secundário pelo envio de seis pacotes de sondagem (dois pequenos, dois grandes e dois pequenos). Os pacotes pequenos são de 40B, e os pacotes grandes são de 1500B. A largura de banda (B) é computada a partir do tamanho dos pacotes grandes (P) divididos pela variação (Δt) dos RTT dos pacotes grandes.

$$B = P/\Delta t$$

Para computar a largura de banda no caminho primário, foi utilizada uma fórmula que leva em consideração a quantia de dados confirmados (Dk) divididos pela variação do tempo de confirmação.

$$B = Dk/\Delta t$$

5.1.3.2 Comentário sobre a proposta

A solução apresentada pelos autores necessita fazer uso de um SCTP modificado, no caso o WiSE, que é apresentado apenas em simuladores, não existe uma implementação real para consolidar os dados.

O mecanismo de tomada de decisão da troca do destino primário é desencadeado após a ocorrência de um *timeout* e seria possível existir casos em que o RTO do caminho primário seja grande, podendo, desencadear um lento processo de alteração do caminho primário, uma vez que o mecanismo de comparação de tomada de decisão de alternância do caminho primário só se realiza após a ocorrência de um *timeout*.

Devido ao fluxo mais intenso de pacotes de dados pelo caminho primário, a percepção da largura de banda do atual caminho primário sempre será mais exato do que a computada no caminho secundário, em que o fluxo de pacotes de sondagem se dá de forma bem mais lenta.

5.1.4 Artigo 4 - Handover baseado em medidas de atraso

5.1.4.1 Descrição da proposta

A proposta apresentada por Nooman et al. [37] busca solucionar/amenizar os problemas encontrados por aplicações, com algum tipo exigências de atraso, no uso do SCTP quando se tentam utilizar as facilidades de *multi-homing*.

O momento da tomada de decisão para alterar o caminho primário, quando utilizada a política padrão do SCTP, gera atrasos que nem sempre são aceitáveis para as aplicações. Esse artigo apresenta uma alternativa que se baseia no atraso fim-a-fim, buscando, dessa forma, contornar a latência apresentada atualmente quando se usa a política padrão do SCTP, que se baseia na falha de comunicação do seu meio, para tomar a decisão de *handover*.

A computação do atraso no caminho primário é obtida dos pacotes de dados transmitidos e confirmados, enquanto que no caminho secundário é por meio dos pacotes de sondagem *HEARTBEAT* enviados, em intervalos de 1 segundo, e posteriormente confirmados.

5.1.4.2 Comentário sobre a proposta

Como a computação do atraso do caminho primário e secundário é efetuada de forma diferente, ou seja, o primário usa pacotes de dados, e o secundário, pacotes de controle *HEARTBEAT*, isso é propenso a trazer informações imprecisas sobre a condição do meio, uma vez que, no caminho primário, a percepção da atual condição do meio é mais realista devido ao constante tráfego de pacotes de dados, enquanto que, no caminho secundário, essa leitura é obtida apenas a cada segundo de transmissão.

Outro fator que chama a atenção é que, como os pacotes utilizados para obter o atraso são de tamanhos diferentes, existe a possibilidade de se obter também leituras diferentes sobre a condição atual do meio devido ao fato de que é mais fácil a propagação de pacotes pequenos do que grandes em ambientes *wireless*.

No artigo, não é definido de forma clara qual o mecanismo que desencadeará o momento

da comparação desses atrasos e por quanto tempo se dará essa observação. A condição para se fazer a alternância do caminho primário é baseada na simples comparação, não leva em conta os possíveis picos de oscilação do meio devido a interferências ou do mecanismo de controle de congestionamento do protocolo SCTP.

5.1.5 Artigo 5 - Handover baseado em medidas de atraso

5.1.5.1 Descrição da proposta

Segundo Hynes et al. [38], o algoritmo usado atualmente para determinar a execução do *handover* é falho para atender às aplicações devido ao fato de não levar em consideração que, durante o uso do caminho primário, pode existir um secundário em melhores condições para atuar como primário. O caminho secundário poderia ser utilizado para sempre propiciar um ótimo desempenho para a aplicação. Com base nessa conclusão, os autores propõem um outro mecanismo de controle para tomada de decisão de quando fazer o *handover*.

Para resolver o problema, foi proposto um novo algoritmo que gerenciará a computação de medidas de atraso a fim de determinar o caminho que mais otimizará a aplicação. Esse algoritmo propõe o uso de técnicas de filtros para remover as altas variações de atraso não comum ao meio.

O diferencial desse algoritmo para outras propostas, segundo os autores, é que a tomada de decisão de *handover* não é baseada no atraso instantâneo, mas utiliza uma série de procedimentos estatísticos e probabilísticos sobre a gama de dados de atraso computados no decorrer da associação.

5.1.5.2 Comentário sobre a proposta

O algoritmo aparentemente se apresenta de forma complexa, talvez hoje seja inviável sua implementação devido à gama de informação que deva ser processada em um período curto de tempo para uma possível tomada de decisão de *handover*.

No artigo, fica claro que a troca do caminho primário será executada a partir do momento que o caminho primário se apresentar em piores condições que o secundário quando comparado os seus atrasos de propagação. Não está claro o que iniciará essa comparação. Se isso ocorrer a cada pacote, as recentes medidas sempre seriam agregadas à gama de informações coletadas no decorrer da associação e deveriam ser processadas novamente para chegar a um resultado final. Isso praticamente tornaria inviável o mecanismo.

Como já comentado em outros artigos, o fluxo de dados gerados no caminho primário é imensamente maior do que o do secundário. Haveria maior exatidão das informações computadas para o caminho primário do que para o secundário, uma vez que para o secundário o atraso é computado por meio de envios de pacotes de sondagem *heartbeat* de tamanhos bem

menores do que os pacotes de dados.

5.1.6 Artigo 6 - Handover baseado em perda de conectividade

5.1.6.1 Descrição da proposta

O artigo apresentado por Grace et al. [7] não é uma proposta, mas uma visão globalizada da empregabilidade do SCTP *multi-homing* para melhorar o desempenho de aplicações em ambientes *wireless*.

Segundo os autores, existe uma tendência muito forte de usar a estrutura de redes IPs para prestar serviços de telefonia. Essa estrutura seria mais fácil de administrar e produziria um custo operacional menor para as empresas. A dificuldade encontrada é que os atuais protocolos de transporte (TCP/UDP) não conseguem suprir as necessidades impostas pelo tráfego de voz. O SCTP, por contornar algumas deficiências dos atuais protocolos de transporte e suportar o *multi-homing*, apresenta-se como saída atraente para resolver essa questão deficitária dos atuais protocolos de transporte em relação às necessidades das aplicações de telefonia.

Como o SCTP apresenta vários parâmetros configuráveis que permitem tornar o protocolo mais ágil em reagir a possíveis problemas de perda de conectividade, pelo uso do *multi-homing*, esse artigo busca apresentar quais seriam as melhores configurações para o SCTP atender a essa demanda em um cenário móvel localizado sobre redes sobrepostas.

Os parâmetros configuráveis do SCTP que são alterados nesse experimento, a fim de proporcionar uma maior fluxo de dados na associação, são os listados na tabela 6.1.

5.1.6.2 Comentário sobre a proposta

No artigo, percebe-se claramente que os parâmetros existem para se adequar à necessidade de cada aplicação, mas as variações de cenários e dos padrões de comunicação fazem de cada caso um cenário novo, ou seja, os parâmetros configurados para uma rede X provavelmente não levarão ao mesmo resultado de vazão em uma rede Y. Conclui-se, dessa forma, que, para cada novo cenário, necessita-se de uma nova pesquisa para obtenção dos melhores valores para os parâmetros do protocolo SCTP.

5.1.7 Artigo 7 - Handover baseado em perda de conectividade

5.1.7.1 Descrição da proposta

O artigo apresentado por Caro et al. [14] faz um estudo do comportamento do protocolo SCTP quando alterado o valor dos seus parâmetros configuráveis (tabela 6.1), com a finalidade de o tornar mais sensível à condição do meio. Os autores estudam o impacto do uso de

valores de parâmetros agressivos que podem provocar uma errônea ideia de que o caminho primário é falho e, devido a isso, efetuar a troca do atual caminho primário.

Como as implementações do protocolo SCTP trabalham com a ideia de que o caminho primário usado inicialmente na associação é sempre o melhor. No caso de falha desse caminho, há o uso temporário de um caminho secundário como primário até que o antigo caminho primário apresente conectividade novamente. O retorno para o caminho primário primogênito é feito automaticamente pelo protocolo SCTP ao primeiro sinal de conectividade. Os autores propõem uma solução para que o retorno nem sempre ocorra. Para isso, baseiam-se em estimativas do RTT para a tomada de decisão.

Segundo os autores, o custo por alterar o caminho primário é ter de utilizar o controle de congestionamento no modo de gerenciamento *slow start*, que propicia um uso mais conservador na utilização do meio para transmitir pacotes de dados.

Em seus estudos, os autores argumentam que a atual política de tratamento de alternância do caminho primário é prejudicial para muitos tipos de aplicações e questionam que isso pode ser bastante melhorado se for usada a estimativa exata do RTT para tomada de decisão da alternância do atual caminho primário.

5.1.7.2 Comentário sobre a proposta

O artigo é claramente um estudo de caso do impacto causado no SCTP ao fazer ajustes agressivos nos parâmetros configuráveis do protocolo, fazendo com que ele tenha uma atitude mais enérgica em relação a possíveis falhas no caminho primário. No estudo apresentado, chegou-se à conclusão de que uma falsa percepção de falha no caminho primário não degrada o desempenho e, em muitos casos, chega a melhorar a vazão na rede, pela troca do atual caminho primário por um secundário.

A problemática trazida por esse modo agressivo é que a alternância do caminho primário poderá proporcionar, para uma aplicação, um caminho pior do que o anterior. Essa troca do atual caminho primário pode ter sido desencadeada devido a uma falsa detecção de caminho falho, tanto é que os próprios autores consideram, em seu artigo, que se obtêm ganhos de desempenho em se ajustar os parâmetros configuráveis do SCTP para valores agressivos, não levando em consideração as condições (largura de banda, atraso, perda de pacotes etc.) de todos os caminhos envolvidos na associação SCTP.

Os autores concluem que deveria ser implementada uma política de mudança do caminho primário, quando apresentasse condições impróprias para transmissão de pacotes de dados, de forma definitiva, desde que o transmissor pudesse usar informações de RTT para essa tomada de decisão. O problema é que o RTT é muito suscetível aos picos de oscilações do meio e tende a apresentar variações bruscas que, dependendo de como for usado, pode inviabilizar o seu uso para tomada de decisão de alternância do caminho primário.

Como nos artigos anteriores, esse também busca fazer um processo mais rápido de

detecção e mudança do atual caminho primário. Já a forma conservadora do SCTP, em tomar essa decisão, deteriora a sua capacidade de fazer uso proveitoso do *multi-homing* para aumentar a vazão dos dados.

5.1.8 Artigo 8 - Handover baseado em perda de conectividade

5.1.8.1 Descrição da proposta

O artigo apresentado por Fallon et al. [5] mostra o comportamento inadequado do SCTP no que tange à sua política de *handover* quando utilizado em ambientes *wireless*. O mecanismo utilizado pelo SCTP, para tomar a decisão do momento da troca do caminho primário, é baseada na falha do atual caminho primário. A falha é determinada quando certo número de tentativas de envio de pacotes não é confirmado. O problema que ocorre é que o intervalo de envio de pacotes é regulado pelo RTO que, por sua vez, é regulado pelo RTT. Segundo os autores, o valor do RTT, nesse ambiente, distorce muito o cálculo do RTO, devido à variação, fazendo com que, em um processo de falha do caminho primário nesse ambiente, ocorra uma demora bem maior do que em outros cenários.

5.1.8.2 Comentário sobre a proposta

De acordo com Fallon et al. [5], é inapropriada a política padrão adotada hoje no SCTP para a tomada de decisão do momento do *handovers* em ambientes intermitentes (que apresentam muita variação no meio durante o período de transmissão). Os autores consideram que a forma de cálculo do RTT/RTO para o caminho é ineficiente para ambientes *wireless*, nos quais interferências ocasionais, comuns nesses ambientes, podem produzir interpretações erradas sobre a condição do meio. O artigo também relata que o tempo total gasto para a realização do *handover* em meios de transmissão intermitentes pode prolongar e muito a subutilização da rede devido à demora do protocolo SCTP em fazer o *handover*. Em trabalhos futuros, os autores consideram que o acompanhamento de crescimento do RTT pode ser utilizado como indicador de possível falha no atual caminho primário.

5.1.9 Artigo 9 - Handover baseado em medidas de SRTT

5.1.9.1 Descrição da proposta

Nesse artigo apresentado por Kelly et al. [12], é proposto que a alternância do caminho primário seja realizada tomando como base medidas de atraso fim-a-fim, uma vez que a tomada de decisão centrada na falha do caminho primário não é adequada à necessidade de muitas aplicações.

Como na proposta atual (protocolo SCTP), o tempo gasto para tomada de decisão de

alterar o caminho primário não atende à necessidade de aplicações de tempo-real, é proposto, nesse artigo, que essa decisão seja condicionada à variação do SRTT. A escolha do SRTT ao invés do RTT, segundo os autores, foi devido ao fato de que, com o uso do SRTT, é possível contornar problemas relacionados às oscilações, principalmente em meio *wireless*, que podem causar variações bruscas do RTT.

5.1.9.2 Comentário sobre a proposta

Os autores não esclarecem no artigo com que intensidade foi realizada essa sondagem para obter o SRTT. Como já comentado em propostas anteriores, se o fluxo de informações de controle for muito intenso, pode trazer prejuízos para outras aplicações que possam usar segmentos de rede que façam parte dessa associação SCTP vigente, diminuindo, assim, a sua noção de largura de banda disponível. No entanto, se a sondagem para obter o SRTT for realizada com intervalos grandes, poderá proporcionar uma noção não real da atual condição do meio secundário, quando comparada com a do meio primário.

Outro fato é que o SRTT do caminho primário é baseado no envio de pacotes de dados e recebimento de confirmação, enquanto que o SRTT do caminho secundário é baseado em pacotes de sondagem. Como os pacotes de sondagem geralmente são bem menores que os pacotes de dados e, em ambientes *wireless*, é mais fácil a propagação de pacotes pequenos em relação a grandes, é possível que o caminho secundário apresente uma falsa impressão de que o seu meio está apto a atender à necessidade da aplicação que atualmente está fazendo uso do caminho primário.

No artigo, os autores fazem uma comparação franca entre os SRTT do caminho primário e do secundário, ou seja, não existe nenhum fator de segurança para prevenir possíveis oscilações curtas do meio primário, evitando, dessa forma, a desnecessária alteração do caminho primário na associação SCTP.

Outro ponto a ser considerado é qual a condição que desencadeará a comparação entre os SRTTs do caminho primário e do secundário e se existe algum período mínimo de permanência do SRTT do caminho primário maior do que o secundário para que somente após isso se justifique a alternância do caminho primário.

5.1.10 Artigo 10 - Handover baseado em medidas de RTT

5.1.10.1 Descrição da proposta

Nos artigos apresentado por Kim et al. [1] e Koh et al. [39], foi proposto um mecanismo de seleção do melhor caminho baseado em leituras do RTT obtido de todos os caminhos disponibilizados por uma associação SCTP *multi-homing*.

Foi definida uma regra geral para a tomada de decisão do momento da troca do atual

caminho primário baseada na seguinte condição: $\text{if } (RTT_p > \alpha * RTT_a)$ (obs.: RTT_p refere-se ao caminho primário e RTT_a , ao caminho secundário).

Segundo os autores, α é o fator de segurança utilizado para evitar o efeito “ping-pong” de troca do caminho primário que força o uso do mecanismo de controle de congestionamento *slow start* degradando, dessa forma, o desempenho.

Conforme afirmações dos autores, o valor de α influencia diretamente no desempenho, ou seja, quando o RTT_p for significativamente maior que o RTT_a e se valor de α for grande, isso causará queda do desempenho. A queda de desempenho também ocorrerá se o RTT_p do caminho primário for muito próximo ao RTT_a . Se o valor α for pequeno, poderá causar o efeito “ping-pong”, e a cada novo caminho primário o SCTP usará o mecanismo de controle de congestionamento *slow start*, que diminuirá o desempenho na rede.

Para contornar esses problemas, os autores definiram duas formas de atuação adaptativa à condição do meio: modo conservador e modo agressivo:

Modo agressivo: $\text{if } |RTT_p - RTT_a| \leq \beta$ (obs. $\beta = 1\text{seg}$)

α_1 = valor agressivo (ou seja, um valor menor que α_2)

Condição para o *handover*: $\text{if } (RTT_p > \alpha_1 * RTT_a)$

Modo conservador: $\text{if } |RTT_p - RTT_a| > \beta$ (obs. $\beta = 1\text{seg}$)

α_2 = valor conservador (ou seja, um valor maior que α_1)

Condição para o *handover*: $\text{if } (RTT_p > \alpha_2 * RTT_a)$

Conforme os resultados apresentados no artigo, os melhores valores para α_1 e α_2 são, respectivamente, 3 e 9.

5.1.10.2 Comentário sobre a proposta

No artigo, não está claro como será feito o cálculo do RTT do caminho primário, se via pacotes de sondagem ou via o próprio pacote de dados. O artigo também não deixa claro com que constância serão disparados os pacotes de sondagem.

Se a sondagem do caminho primário for feita via pacotes de dados, o RTT será muito mais preciso em relação à condição do meio do que o RTT do caminho secundário, além de ser mais sensível a qualquer oscilação no meio. Outro fator também a ser ponderado é que o tamanho dos pacotes de sondagem será diferente, uma vez que os pacotes de dados têm valores provavelmente de 1500B, e os de sondagem *HEARTBEAT* têm valores de aproximadamente 40B. É mais fácil trafegarem pacotes de dados pequenos do que a grandes em ambientes *wireless*.

A redução do intervalo entre o envio de pacotes de sondagem tem como consequência um fluxo constante de informações de controle, que é prejudicial para ambientes *half-duplex* e também problemático quando se pensa no uso desse protocolo em uma escala macro. É notório que essas informações de controle (sondagem) podem não prejudicar o caminho primário da associação SCTP, mas podem trazer prejuízo para qualquer outra aplicação que

possa estar usando algum segmento que pertença ao conjunto dessa associação que dispara de tempo em tempos pacotes de sondagem.

O uso do valor do RTT como condicionador de tomada de decisão do momento de alterar o atual caminho primário pode não trazer bons resultados em ambientes intermitentes, uma vez que o RTT se apresenta com variações bruscas no decorrer de uma associação, isso, por exemplo, é mencionado em Kashihara et al. [35].

Capítulo 6

SCTP Proativo Adaptativo (PA-SCTP)

O PA-SCTP é uma extensão proposta ao SCTP, que faz uso do *multi-homing* em prol das necessidades da aplicação em relação à condição do meio primário para transmissão/recepção de pacotes de dados.

O protocolo SCTP faz a alternância do destino primário baseando-se exclusivamente em percepções das condições do meio captadas pelo emissor de pacotes de dados em uma associação estabelecida. O receptor de dados é passivo (só recebe pacotes de dados e envia confirmação, ou seja, *chunks SACKs*), não participa no processo de tomada de decisão da troca do destino primário. Na verdade, ele só participa quando for omissor, ou seja, não enviar confirmação (*SACKs*).

A alternância do destino primário pelo protocolo SCTP ocorre quando este declara o atual caminho primário sem conectividade. Um caminho primário é visto pelo SCTP como sem condições para transmissão de pacotes de dados, quando esse caminho passa a não confirmar mais os pacotes de dados enviados. O tempo decorrido para que ocorra a alteração do destino primário geralmente pode não ser apto para a necessidade da aplicação. O tempo gasto na transição é referenciado na tabela 6.2.

A extensão mSCTP proporciona maior liberdade de se trabalhar com o protocolo SCTP por permitir inserir, deletar e solicitar a alteração do destino primário de uma associação. Isso não é concebido na proposta do SCTP (RFC 4960).

O mSCTP faz uso das mesmas técnicas do protocolo SCTP para chegar à conclusão de perda de conectividade do atual caminho primário, ou seja, baseada nas percepções do transmissor de pacotes de dados com relação ao recebimento de confirmações *SACKs*.

A proposta do PA-SCTP é fazer o uso das percepções de picos de atraso, atraso médio e variação do CWND a cada recebimento de pacotes de dados e, por meio disso, fazer a solicitação de alteração de um caminho primário instável, desde que o caminho secundário se apresente em melhor situação em relação ao atual caminho primário diante da necessidade da aplicação. O PA-SCTP não altera o mecanismo atual de troca do caminho primário, apenas cria um outro mecanismo, baseado nas percepções do receptor de dados, mais sensível às

condições do meio e ao mesmo tempo consciente das condições reais do caminho secundário.

6.1 Tratamento proativo adaptativo do handover com o PA-SCTP

Estações SCTP tendem a ser auxiliadas por mais de uma interface de comunicação que podem já estar conectadas a diferentes redes IPs. O termo *handover*, nesse caso, cabe apenas, para a alternância da saída IP, para a transmissão de dados.

O grande gargalo visível no SCTP utilizando a sua característica de *multi-homing* é prever qual o melhor momento de realizar o *handover* para amenizar possíveis problemas que possam vir a incorrer para a aplicação que está fazendo uso desse protocolo.

Os seis *timeouts* consecutivos para tomada de decisão de *handover* (configuração padrão do SCTP) se mostram ineficazes para a realização dessa tarefa. A redução desse *timeout* para valores muito baixos também não se mostra muito atraente quando o caminho secundário tem uma capacidade de banda bem menor ou que esteja com uma largura de banda limitada devido a um tráfego intenso e que a falha no caminho primário é uma coisa passageira.

A redução do período de *timeout* (período de silêncio sem a confirmação do recebimento de pacotes de dados) é delimitada pela própria RFC do SCTP, que aconselha no mínimo um segundo como sendo um valor que atenderia bem às variadas formas de propagação da informação por diferentes tipos de enlace. O valor de *timeout* abaixo de um segundo tem uma funcionalidade aceitável com bons resultados, mas é dependente da situação do meio de transmissão.

A preocupação do protocolo SCTP, após a perfeita conclusão da associação, é mantê-la ativa até que naturalmente haja uma solicitação para o seu encerramento. Não existe qualquer tipo de preocupação com as necessidades de aplicações que porventura estejam utilizando esse protocolo como meio de transporte, como, por exemplo, exigência máxima de atraso na chegada dos pacotes.

A solução adotada e mostrada em resultados mais adiante é o uso de uma política de monitoramento no lado cliente, baseada na percepção de picos de atraso, atraso médio e variação da CWND na recepção de pacotes de dados SCTP, a fim de verificar a operacionalidade do atual caminho primário em relação à necessidade da aplicação. Para tentar sanar/atenuar a falta de condição do atual caminho primário à necessidade da aplicação, é possível fazer sondagens para verificar a condição do caminho secundário, sempre que a atual condição do caminho primário não estiver mais atendendo à necessidade da aplicação. Nesse cenário, obtendo-se uma melhor condição no caminho secundário, é solicitada a alternância do destino primário do ponto de vista do servidor.

6.2 Contextualização da problemática da política de handover do SCTP

Toda política de tomada de decisão do momento de *handover* no SCTP é baseada nos *chunks SACKs* recebidos pelo servidor. Em um ambiente puramente cliente-servidor, um lado só transmite pacotes de dados SCTP e o outro só recebe, a tomada de decisão de *handover* caberia exclusivamente ao transmissor de pacotes SCTP. O lado cliente ficará alheio a esse processo.

Para entender melhor o processo, tomando como base o *testbed* da figura 6.6 (apresentada na seção 6.4), o cliente iniciará uma negociação para configuração de uma associação com o servidor para posterior recebimento de pacotes de dados SCTP. A solicitação de início da associação é feita pelo cliente 1. É gerado o *chunk INIT* que será enviado pela rota 192.168.100.3 – 192.168.50.2. Ao receber o *chunk INIT*, o servidor responderá com um *INIT-ACK* enviado pela rota 192.168.50.2 – 192.168.100.3. A partir da conclusão da associação, para o cliente, seu destino primário para todo pacote de dados SCTP dentro dessa associação será 192.168.50.2, enquanto que para o servidor será 192.168.100.3.

Apenas o transmissor de pacote de dados SCTP é capaz de identificar a interrupção/congestionamento do meio e tomar a decisão do momento de *handover*. A interpretação dessa falha de comunicação pelo servidor ocorre pelo não recebimento do *chunk SACK*, que deve ser gerado pelo cliente para confirmar o recebimento de pacotes de dados SCTP. A confirmação é esperada por um período de tempo determinado pelo RTO, que, por sua vez, é gerado a partir de medidas obtidas pelo cálculo do RTT, que é gerado a partir de informações oriundas de *chunks SACKs* recebidos.

O não recebimento de *chunks SACKs*, por parte do servidor em um período de tempo determinado pelo RTO, gera um *timeout*. Na prática, expirou o tempo de receber a confirmação do pacote de dado SCTP, por isso esse pacote é retransmitido pelo servidor. Após a ocorrência do primeiro *timeout*, cinco consecutivos *timeouts* serão necessários para que ocorra o *handover*. Esse é o limiar a ser alcançado, conforme determinado pela política padrão do SCTP, para só então ocorrer o *handover*.

Ao ocorrer o *handover*, no cenário descrito acima, o transmissor de pacotes de dados SCTP passará a utilizar o IP secundário do cliente, como seu IP de destino para transmissão de dados.

O RTT, que é base para o valor do RTO, é recalculado a cada SACK recebido. A ocorrência de um *timeout* altera a forma de cálculo para o RTO. Nesse caso, $RTO = RTO * 2$.

Se a interrupção do meio de comunicação ocorresse quando o RTO fosse igual a 1 segundo, teríamos teoricamente um tempo total entre a detecção do primeiro *timeout* e os cinco consecutivos necessários, um valor acumulado de 63 segundos ($1 + 1*2 + 2*2 + 4*2 + 8*2 + 16*2 = 63$) de aproveitamento ínfimo da rede, devido a essa política conservadora para

tomada de decisão de *handover*.

A parametrização do SCTP é permitida para que se possa adequar o protocolo às condições variadas de rede. Entende-se por SCTP parametrizado quando há alteração dos valores padrões dos parâmetros listados na tabela 6.1. Nem sempre essa solução é eficaz e geralmente não se adéqua aos caminhos que não se comportam de forma padronizada, ou seja, apresentam oscilações aleatórias e com períodos não constantes.

Parâmetros do SCTP que podem ser alterados para tentar promover um melhor uso do ambiente *multi-homing* estão listados na tabela 6.1.

Tabela 6.1: Variáveis SCTP que influenciam na latência de *handover*

Parâmetro	Função
<i>Path.Max.Retrans(PMR)</i>	Determina a quantidade máxima de <i>timeouts</i> no caminho primário para tomar a decisão de <i>handover</i> .
<i>RTO.Max</i>	Maior RTO que pode ser determinado para um caminho.
<i>RTO.Min</i>	Menor RTO determinado para um caminho.
<i>HB.Interval</i>	Intervalo entre a geração de <i>chunks HEARTBEAT</i> para sondagem de conectividade de caminho.

Testes feitos por Fallon et al. [5], apresentados na tabela 6.2, mostram os tempos de *handover* baseados em simulações feitas em NS2 (*Network Simulator*) alterando os parâmetros referenciados na tabela 6.1.

Tabela 6.2: Tempo gasto em segundos para efetuar o *handover*, segundo Fallon et al. [5]

PMR	RTO.MAX(S)				
	60	30	10	5	1
5	186,61s	126s	58,24s	34,37s	9,14s
4	130,91s	95,4s	47,65s	28,77s	7,53s
3	70,31s	64,8s	37,05s	23,17s	6,53s
2	34,1s	34,81s	26,44s	17,57s	5,53s
1	15,84s	15,84s	15,84s	11,97s	4,52s
0	6,36s	6,36s	6,36s	6,36s	2,92s

A alteração dos parâmetros do SCTP com a finalidade de diminuir a latência de *handover*, segundo Caro et al. [14], geralmente traz melhorias na vazão, desde que se considere que o caminho secundário tenha uma vazão igual ou superior a do primário.

Como, no momento do *handover*, o mecanismo de eleição de um novo destino primário, entre os IPs que identificam o cliente na associação, não é tratado diretamente na RFC do SCTP, o novo destino primário pode não atender adequadamente à aplicação.

Atente-se também ao fato de que, para cada mudança de caminho de transmissão, o mecanismo de controle de congestionamento inicialmente utilizado é o *slow start*, e o tamanho da janela CWND do caminho se inicia em 4380 *bytes*. Mas o grande problema

é permanecer com a utilização de um caminho de transmissão em detrimento a outro em melhor condição para realizar a tarefa.

6.3 Solução proposta PA-SCTP

O *PA-SCTP* (Proativo Adaptativo - SCTP) é uma solução que busca atender à necessidade da aplicação em relação ao atraso médio máximo permitido para o recebimento de pacotes e dados SCTP. Para isso, faz o monitoramento de picos de atraso, atraso médio e variação da CWND a cada recebimento de pacotes e solicita do servidor a alternância do destino primário sempre que este não estiver atendendo às necessidades da aplicação e que exista outro destino em melhores condições.

A política adotada hoje pelo SCTP para tomada de decisão do momento de realizar o *handover* é toda fundamentada no transmissor. A solução aqui proposta incube essa tarefa também ao receptor sem, no entanto, desabilitar ou alterar a forma como é feita hoje, ou seja, fundamentada no transmissor.

A grande diferença entre as duas técnicas é que a fundamentada no transmissor é mais reativa do que proativa quanto à detecção de caminho falho e tomada de decisão do momento da realização do *handover*, sendo que, em nenhum momento, é levada em conta a necessidade da aplicação. A fundamentada no receptor é mais proativa e adaptativa; proativa por ser mais sensível às oscilações do meio e adaptativa, pois tenta usar o melhor caminho para atender às necessidades da aplicação, baseando-se em picos de atraso, atraso médio e variação da CWND detectados a cada recepção de pacotes. Como as duas técnicas passarão a monitorar a condição do meio de forma paralela, a tendência é que sempre o PA-SCTP seja executado primeiro devido à sua característica mais agressiva de acompanhamento das condições do meio primário.

Os parâmetros configuráveis do SCTP proporcionam uma tentativa de adaptabilidade à realidade do meio de transmissão. Como já comentado aqui e em vários artigos referenciados, essa solução por si só não garante uma boa saída para todos os cenários, além do fato de contar com o desgaste da procura por tentativa e erro dos melhores valores.

Por fim, a solução aqui proposta buscará se adaptar dinamicamente à necessidade da aplicação em relação às condições do meio de transmissão. Para isso, busca diminuir a latência de alternância do caminho primário em relação à sua política padrão e, ao mesmo tempo, evita o processo desgastante de tentar descobrir qual o melhor ajuste de parâmetros SCTP para rapidamente detectar uma tendência à falha do caminho de transmissão de pacotes de dados e, conseqüentemente, alterar o destino primário do ponto de vista do servidor.

Para demonstrar o ganho de desempenho a ser alcançado por adoção do PA-SCTP, toda a solução foi implementada em linguagem C utilizando a plataforma *FreeBSD* e API documentada por Xie et al. [8], a qual está baseada na RFC 4960 (SCTP-2007) e na RFC5061

(mSCTP-2007).

A escolha do sistema operacional *FreeBSD* se deu pelo fato de que, a partir da sua versão 7.1, já vir nativamente instalado o SCTP e o mSCTP, baseado nas RFCs mais recentes e em conformidade com a API documentada por Xie et al. [8] de 2009. A escolha pela linguagem C se deu também pelo fato da API estar escrita em C.

Mesmo a solução tendo sido implementada na camada de aplicação, os resultados se mostraram eficazes em relação ao ganho de desempenho, como é demonstrado mais adiante por meio de gráficos.

6.3.1 Custo da alteração do destino primário

De acordo com pesquisas relatadas em Caro et al. [14] e Paul et al. [11], surpreendentemente o processo de alternância do destino primário de transmissão de pacotes de dados SCTP não causam impacto significativo no tempo total de transmissão de um arquivo pela rede. Complementando essa informação, essa transição não causa custo desde que seja feita a solicitação pela aplicação antes da ocorrência de um *timeout*, pois, após isso, o protocolo SCTP (implementado no *FreeBSD*) cobra um custo geralmente entre 1 e 2 segundos para alterar o destino primário.

Entendendo o mecanismo de funcionamento do SCTP, teoricamente, conclui-se que haverá um prejuízo se ocorrer mudança do caminho de transmissão devido o novo destino, do protocolo SCTP, ter de inicialmente utilizar o mecanismo de controle de congestionado *slow start*. Na prática, há uma queda na capacidade de transmissão, pressupondo que no caminho anterior o mecanismo de controle de congestionamento utilizado era o *congestion avoidance*.

Ao fazer testes utilizando o *testbed* da figura 6.6, mostrou-se que a alternância do destino primário pela aplicação não causa um custo significativo conforme resultados apresentados na tabela 6.3, desde que as condições dos caminhos disponíveis na associação sejam iguais. Nesse contexto, houve a transmissão de um arquivo do servidor para o cliente em que foram transmitidos aproximadamente 30.000 pacotes de 1.400B. Os caminhos (primário e secundário) se apresentam com as mesmas condições e sem nenhum tipo de tráfego paralelo (obs.: os enlaces primário e secundário são de 5Mbps).

Coforme a análise dos resultados, o controle de congestionamento em seu ápice promove um CWND médio entre 55.000 a 75.000B de capacidade de transmissão. Ao executar o *handover*, o novo caminho primário utiliza inicialmente o modo *slow start* com a CWND inicial de 4380B. Como as condições dos dois caminhos estão perfeitas, a CWND do novo caminho primário alcança o valor reportado anteriormente pelo seu antigo caminho primário em um período muito curto. Provavelmente, isso justifique o pouco impacto negativo provocado pela alternância de destino primário.

Um outro fator que justifica isso é a capacidade de transmissão do caminho primário no

Tabela 6.3: Resultado de testes do impacto dos *handovers* na vazão da rede

Interrupção	Handovers	Tempo de Transmissão				
		EXP1	EXP2	EXP3	EXP4	EXP5
A cada 35000 pcts	0	144s	145s	144s	144s	145s
A cada 10000 pcts	3	144s	144s	145s	144s	144s
A cada 8000 pcts	3	144s	144s	145s	145s	144s
A cada 7000 pcts	4	145s	144s	144s	145s	147s
A cada 6000 pcts	5	145s	145s	145s	145s	145s
A cada 5000 pcts	6	145s	145s	144s	144s	145s
A cada 4000 pcts	7	145s	145s	145s	145s	145s
A cada 3000 pcts	10	145s	146s	144s	145s	145s
A cada 2000 pcts	15	145s	146s	146s	145s	145s
A cada 1000 pcts	30	145s	144s	145s	145s	145s
A cada 500 pcts	61	146s	147s	146s	146s	147s
A cada 100 pcts	306	146s	146s	146s	147s	146s

seu ápice, ou seja, no momento antes da transição de destino. A capacidade de transmissão de pacotes sem recebimento de confirmação pode estar em um valor inferior ao permitido pelo novo caminho primário, que inicialmente é 4 MTU.

Em Caro et al. [14], também são apresentados experimentos que comprovam a não percepção de perda na transmissão de pacotes de dados no SCTP quando há mudanças constantes do destino primário de um servidor.

O efeito negativo da alternância do destino primário em espaços de tempos curtos fica por conta de uso de um caminho com características piores do que o que está atualmente em uso. A utilização do SCTP com os seus parâmetros alterados, a fim de promover uma menor latência de *handover*, aumenta a possibilidade de se migrar para um caminho com piores condições que o anterior, impondo, dessa maneira, um custo que nem sempre a aplicação pode suportar.

Ao comparar as tabelas 6.2 e 6.3, aparentemente elas apresentam valores diferentes para o mesmo custo, ou seja, o tempo gasto na alternância do destino primário. A diferença é que, na primeira tabela, o custo é medido em um cenário no qual houve a ocorrência de *timeouts*. Já na segunda tabela, praticamente não há custo, pois a solicitação da alternância do destino primário foi feita pela aplicação em um cenário onde o atual caminho primário se apresentava em condições de largura de banda disponível e aceitável para qualquer tipo de aplicação.

6.3.2 PA-SCTP - lado transmissor dos pacotes de dados SCTP

A figura 6.1 apresenta o transmissor de pacotes de dados SCTP. Os retângulos mais escuros representam os módulos que foram inseridos pela extensão PA-SCTP. Ao retirar os módulos mais escuros da figura, tem-se o SCTP puro como proposto pela sua RFC.

Os módulos documentados aqui serão apenas aqueles inseridos pelo PA-SCTP, os outros

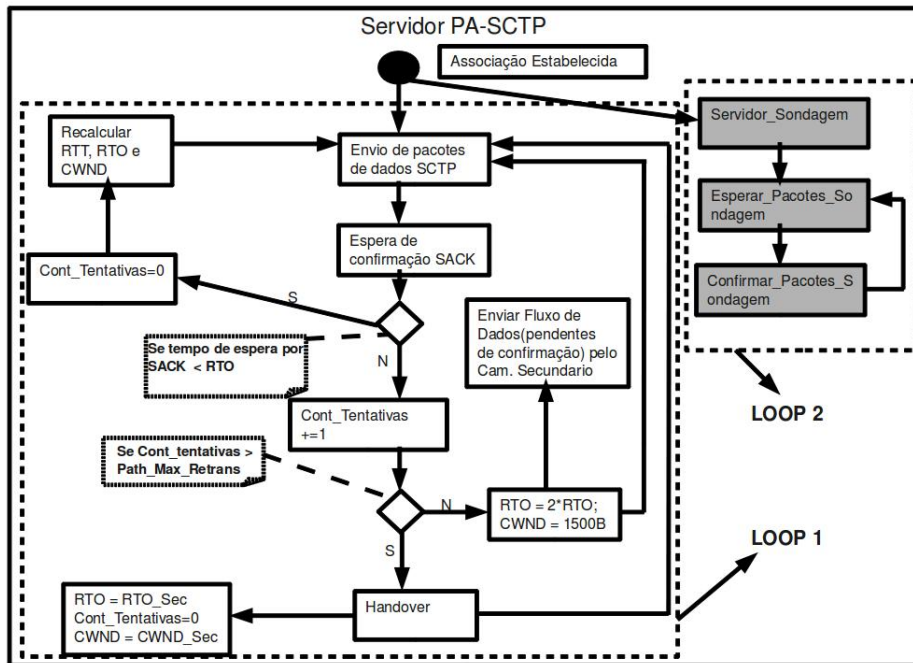


Figura 6.1: Detalhamento da transmissão de pacotes SCTP e PA-SCTP

foram detalhados de forma direta ou indireta dentro deste documento.

O módulo **Servidor_Sondagem** é acionado logo após a conclusão da configuração da associação e fica à espera de pacotes de sondagem originados por um cliente PA-SCTP. É aberto um *socket* UDP à espera da chegada de pacotes de sondagem.

O *socket* é UDP, pelo fato do SCTP não poder proporcionar um atraso mais exato devido à sua política de retransmissão de pacotes e controle de fluxo.

O módulo **Esperar_Pacotes_Sondagem** fica a cargo de tratar os pacotes UDP provenientes do receptor de dados SCTP. Uma vez recebido esse pacote, é ativado o tratamento de confirmação.

O módulo **Confirmar_Pacotes_Sondagem** informa ao solicitante o recebimento dos pacotes de sondagem. Os pacotes de sondagem que chegam são de apenas 40B, e os de confirmação enviados pelo servidor são de 1.400B. Essa diferença entre os tamanhos dos pacotes é similar ao processo de envio de pacotes de dados SCTP e o recebimento de confirmação SACK. Com isso, obtém-se um atraso mais similar a um fluxo normal de dados do caminho primário de uma associação SCTP.

Como o protocolo UDP não tem, na sua essência, nenhum mecanismo de retransmissão de pacote, a cada solicitação de sondagem que chega, são gerados três pacotes UDP de confirmação, tendo cada um o tamanho de 1.400B. O não recebimento de no mínimo dois pacotes UDP de resposta é sinalização para o cliente de que o caminho não está apto para transmissão de dados.

O tamanho de 1.400 *bytes* para o pacote de confirmação é usado devido ao tamanho do

MTU da camada de enlace de redes mais comuns, no caso a *ethernet*, é de aproximadamente 1500 *bytes*. A transmissão do pacote de confirmação de tamanho grande se traduz em uma leitura mais fiel das condições do meio por onde foi transmitido o pacote. Segundo Liu et al. [40], existe uma maior probabilidade de sucesso na transmissão de pequenos pacotes em relação ao de tamanho grande em ambientes *wireless*. Então, segundo essa afirmação, é mais fácil obter sucesso na transmissão de um pacote de 40 *bytes* do que um de 1400 *bytes*.

6.3.3 PA-SCTP - lado receptor dos pacotes de dados SCTP

A figura 6.2 apresenta o receptor de pacotes de dados SCTP. Os retângulos mais escuros representam os módulos propostos pela extensão PA-SCTP. Ao retirar os módulos mais escuros da figura, tem-se o SCTP puro como proposto pela sua RFC.

Os módulos documentados aqui são apenas os propostos pela extensão PA-SCTP. Os outros foram detalhados de forma direta ou indireta dentro deste documento.

O *Loop1* e *Loop2* são executados paralelamente e ficam em constante execução até que a associação seja finalizada.

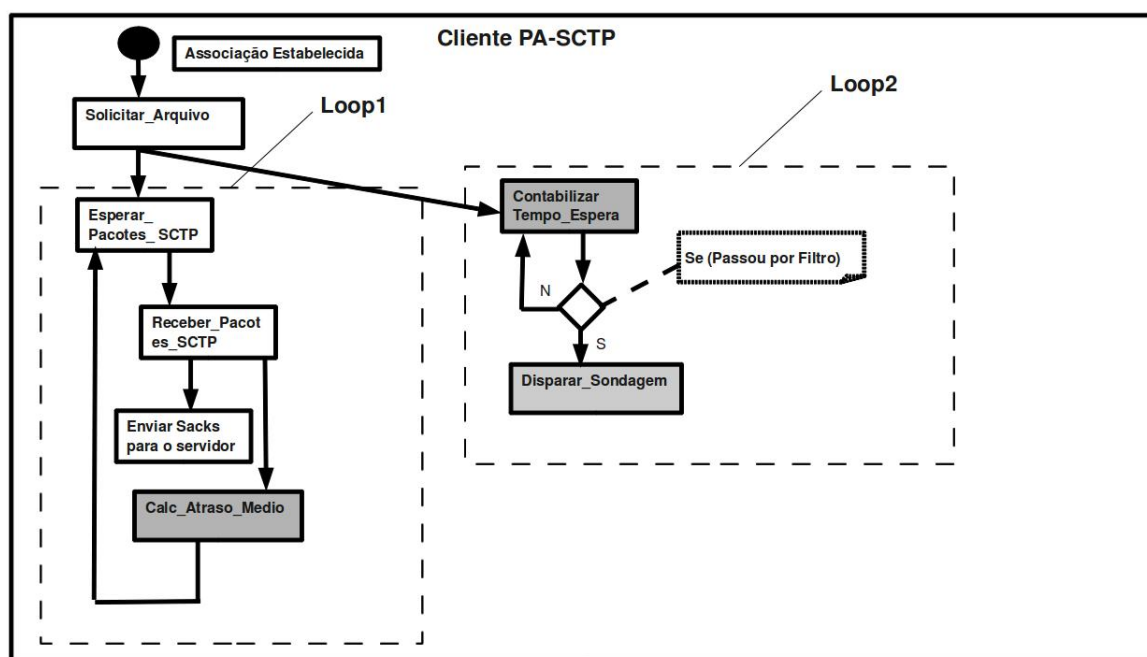


Figura 6.2: Detalhamento da recepção de pacotes SCTP e PA-SCTP

6.3.3.1 Módulo Calc_Atraso_Medio

Esse módulo é responsável por contabilizar o atraso médio na chegada de pacotes de dados no caminho primário da associação estabelecida. Seu funcionamento é demonstrado na figura 6.3.

A cada pacote de dado recebido é calculado o atraso médio. Como, por padrão, o protocolo SCTP permite apenas o recebimento de pacotes de dados de forma bloqueante, esse submódulo não será usado para, por exemplo, acionar o procedimento de disparo de pacotes de sondagem a fim de conhecer a situação do caminho secundário, devido a atrasos muito grandes contabilizados a cada pacote recebido. Existe um mecanismo próprio para isso.

Por meio desse módulo, é monitorado o caminho primário buscando detectar um comportamento intermitente em relação ao recebimento de pacotes de dados SCTP, ou seja, com espaços de tempo no recebimento de pacotes de dados não uniforme. Esse comportamento poderá gerar uma sinalização de que o atual caminho primário pode não estar suportando manter a exigência da aplicação com relação ao atraso médio máximo permitido no recebimento de pacotes de dados.

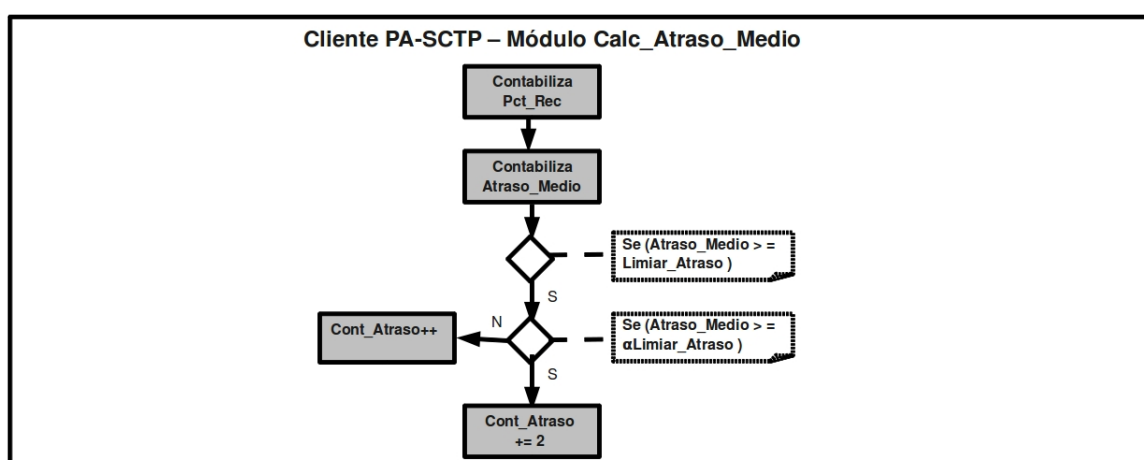


Figura 6.3: Fluxo de execução no módulo Calc_Atraso_Medio

A função **Contabiliza Pct_Recebido** é um sinalizador para o sistema do recebimento de um pacote de dados pelo caminho primário.

A função **Contabiliza Atraso_Medio** calcula o atraso médio no recebimento de pacotes de dados atualmente.

O parâmetro **Atraso_Medio** guarda o atraso médio no recebimento de pacotes de dados SCTP em milissegundos.

O parâmetro α é um coeficiente usado para determinar o quão grande foi um determinado espaço de tempo entre o recebimento de pacotes de dados. Isso determinará a intensidade do incremento ao sinalizador (**Cont_Atraso**) que informa para o sistema uma proeminente possibilidade de falha do atual caminho primário em atender à necessidade da aplicação.

O parâmetro **Cont_Atraso** é um sinalizador para o sistema sobre a atual condição do caminho primário em atender à necessidade da aplicação. Quanto maior esse valor, pior é a condição do atual caminho primário.

6.3.3.2 Módulo Contabilizar_Tempo_Espera

Esse módulo computa o tempo decorrido entre o recebimento de pacotes de dados. É por meio desse módulo que é disparada a sondagem de pacotes de dados pelos caminhos secundários. Seu funcionamento é demonstrado na figura 6.4.

Existem duas possibilidades desse módulo acionar o mecanismo **Disparar_Sondagem** (módulo para disparar pacotes de sondagem pelo caminho secundário). O disparo de pacotes de sondagem pode ser acionado devido ao atraso muito grande entre recebimento de pacotes de dados, sendo isso computado nesse módulo, e também devido ao comportamento intermitente do caminho primário detectado pelo módulo **Calc_Atraso_Medio**.

A cada acionamento do módulo **Disparar_Sondagem**, o módulo **Contabilizar_Tempo_Espera** é interrompido temporariamente até que o processo de sondagem e possível realização de troca do destino primário sejam realizados.

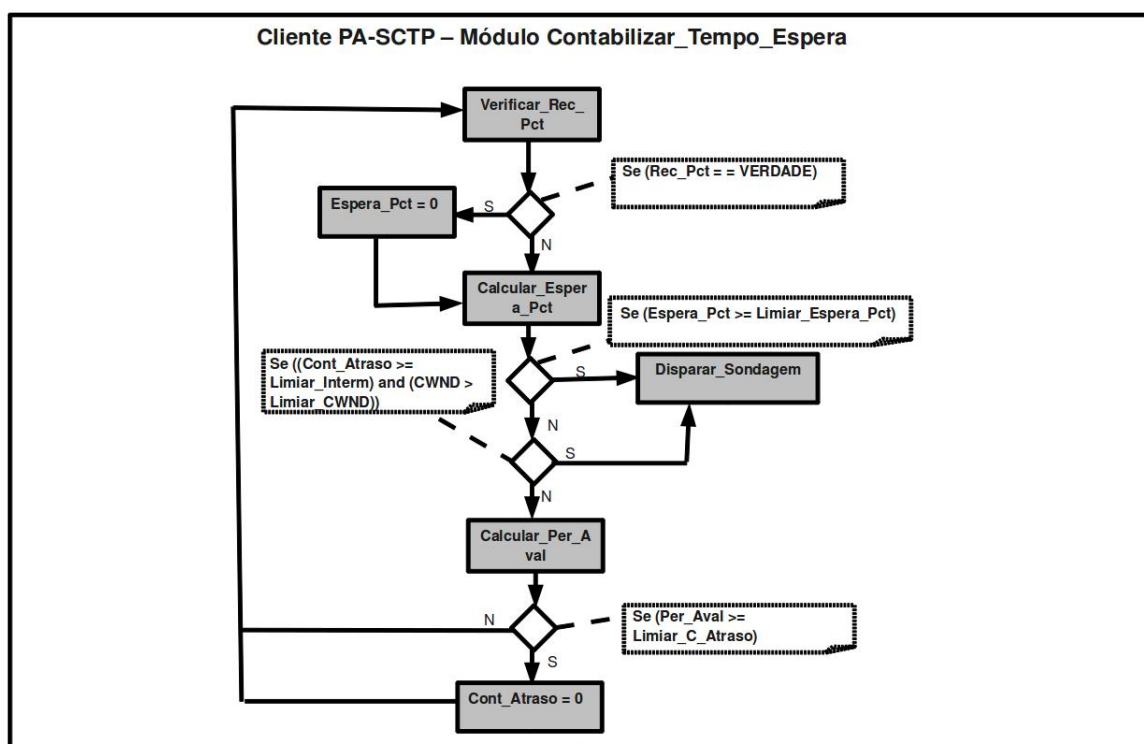


Figura 6.4: Fluxo de execução no módulo Contabilizar_Tempo_Espera

A função **Verificar_Rec_Pct** busca sinalização do recebimento de pacotes de dados a cada *loop* desse módulo de acompanhamento do tempo de espera para recebimento de pacotes de dados. Uma vez que haja a sinalização do recebimento de um pacote de dados, o parâmetro **Espera_Pct** é reinicializado com zero.

O parâmetro **Espera_Pct** tem o tempo decorrido desde o recebimento do último pacote.

A função **Calcular_Espera_Pct** contabiliza o período, em milissegundos, desde o recebimento do último pacote de dado pelo caminho primário.

O disparo do mecanismo para sondar o caminho secundário pode ocorrer por meio de dois possíveis fatores: período muito longo sem receber pacotes de dados ou devido ao atual caminho primário apresentar um comportamento intermitente com relação ao recebimento de pacotes de dados.

Um período sem receber pacotes de dados maior ou igual ao parâmetro **Limiar_Espera_Pct** ativa o mecanismo de sondagem do caminho secundário.

Se o índice de caminho intermitente, o parâmetro **Cont_Atraso**, computado durante a execução do módulo **Calc_Atraso_Medio**, for maior ou igual ao parâmetro **Limiar_Interm** e a janela de transmissão (CWND) do transmissor for menor do que o **Limiar_cwnd**, o mecanismo de sondagem do caminho secundário também poderá ser disparado.

A função **Calcular_Per_Aval** computará há quanto tempo está executando o módulo **Contabilizar_Tempo_Espera**. O valor gerado por essa função será comparado com o parâmetro **Limiar_C_Atraso**, que limita o tempo máximo permitido para validade do valor do parâmetro **Cont_Atraso**.

6.3.3.3 Módulo Disparar_Sondagem

Esse módulo é responsável por fazer o processo de sondagem sobre o caminho secundário existente dentro da associação. A sondagem sempre se inicia do receptor de pacotes de dados de uma associação SCTP estabelecida pelo envio de um pacote de 40B e a espera do pacote resposta de 1400B. Seu funcionamento é demonstrado na figura 6.5.

Como o processo de sondagem é efetuado via um *socket* UDP, que é um protocolo não confiável para a entrega de pacotes, sempre são enviados, no mínimo, três pacotes de sondagem, e o caminho é considerado apto quando se recebe a confirmação de pelo menos dois dos pacotes de sondagem enviados.

A opção por enviar os pacotes de sondagem via uma outra conexão que não seja a SCTP, no caso o UDP, é devido ao fato de que, para os pacotes de sondagem, não é conveniente trabalhar com mecanismos de controle de fluxo ou retransmissão de pacotes de dados. Essas características são implícitas quando se usa o protocolo de transporte SCTP, podendo produzir resultados imprecisos quanto à real qualidade do meio de transmissão.

A sondagem padrão SCTP tem como finalidade testar a conectividade e, ao mesmo tempo, obter um RTT do caminho para o cálculo do RTO. O RTT pode, em muitos casos, não representar a realidade, pois, em uma rede *wireless*, é mais fácil se obter sucesso na transmissão de um pacote de 60B do que um de 1500B.

De acordo com Kelly et al. [12], a sondagem no caminho secundário gera um tráfego insignificante em relação ao que ocorre no caminho primário, e o RTT obtido com a sondagem do caminho secundário é meramente um valor guia quando utilizado na transmissão de pacotes de dados SCTP.

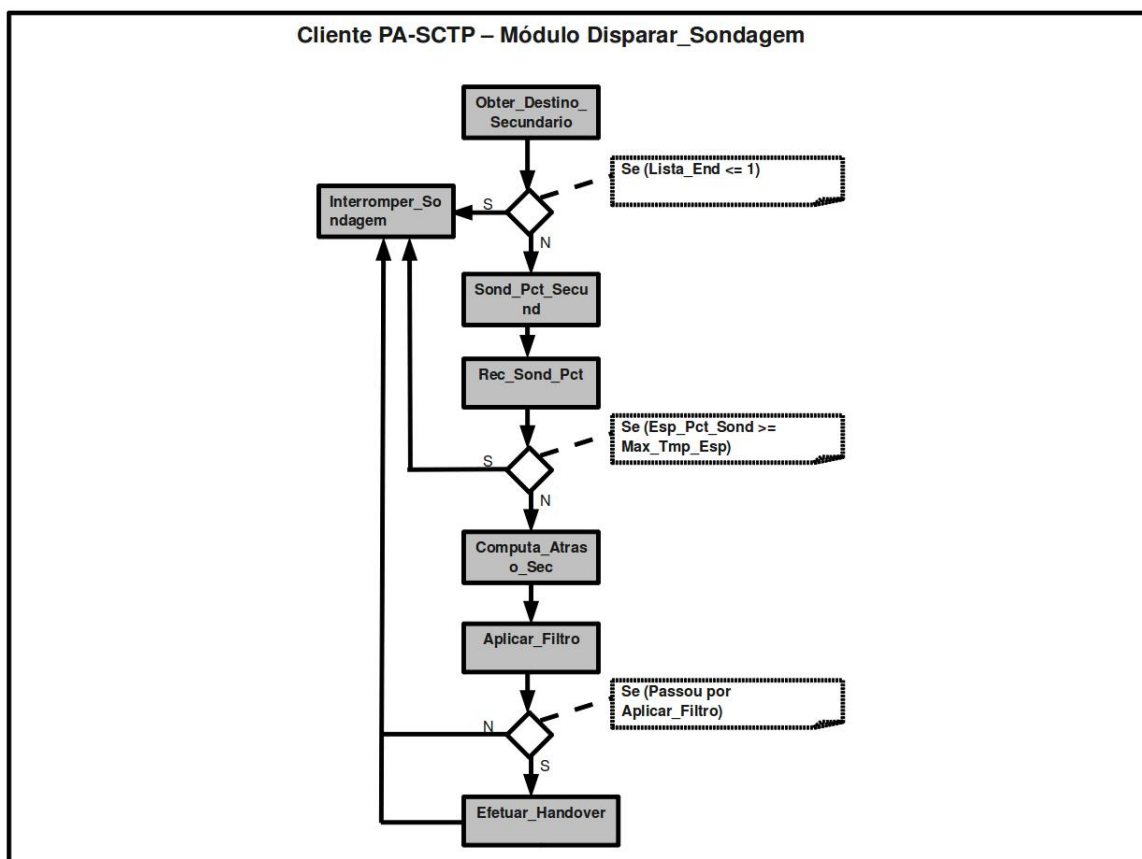


Figura 6.5: Fluxo de execução no módulo Disparar_Sondagem

A função **Obter_Destino_Secundario** é utilizada para se obterem todos os endereços de destino IPs, do ponto de vista do receptor de dados, que fazem parte da associação SCTP no atual momento. A API do SCTP fornece uma função chamada *sctp_getpaddrs*, que disponibiliza todos os endereços IPs que fazem parte da associação ativa no momento.

Naturalmente se não for fornecido nenhum endereço além do atual endereço primário, é porque a atual associação está no modo *single-homed*, o que impediria o ganho de desempenho obtido pelo uso do PA-SCTP.

A função **Sond_Pct_Secund** é encarregada de enviar pacotes de sondagem para os endereços secundários existentes na associação.

A função **Rec_Sond_Pct** é encarregada de monitorar o recebimento das respostas dos pacotes de sondagem enviados. O período máximo para ficar esperando a resposta dos pacotes de sondagem é definido pelo parâmetro **Max_Tmp_Esp**. Após esse período, o caminho é considerado impróprio para uso como destino dos pacotes de dados da aplicação.

A função **Aplicar_Filtro** executa um tipo de filtro que depende de qual mecanismo acionou o disparo de pacotes de sondagem, podendo ser por tempo de espera de pacotes de dados ou por intermitência do caminho primário. O filtro se aplicam da seguinte forma:

- quando o disparo de pacotes de sondagem foi acionado por uma espera longa no rece-

bimento de pacotes de dados: se houver o recebimento de sinalização da chegada de novos pacotes de dados e o valor do parâmetro **Atraso_Medio** for menor ou igual ao atraso médio computador no caminho secundário multiplicado por um coeficiente de segurança - a interrupção do mecanismo de sondagem deve ocorrer imediatamente;

- quando o disparo de pacotes de sondagem for acionado por um comportamento intermitente: valor do parâmetro **Atraso_Medio** for menor ou igual ao atraso médio computador no caminho secundário multiplicado por um coeficiente de segurança - a interrupção do mecanismo de sondagem deve ocorrer imediatamente.

A função **Efetuar Handover** utiliza a funcionalidade disponibilizada ao SCTP pela extensão documentada na RFC 5061 para solicitar a alteração do atual destino primário do ponto de vista do servidor de dados da associação SCTP.

6.3.3.4 Parâmetros de limiar no algoritmo PA-SCTP

Os parâmetros limiares adotados na proposta PA-SCTP são todos utilizados pela parte receptora de uma associação SCTP para determinar quando se devem acionar certos mecanismos para resguardar a boa condição do caminho de recebimento de pacotes de dados do ponto de vista da necessidade da aplicação.

Tabela 6.4: Lista de parâmetros limiares usados pelo PA-SCTP

Parâmetro Limiar	Valor	Função
Threshold_Delay	50 ms	Atrasos no recebimento de pacotes de dados que excedem esse limiar sinalizam para o sistema um caminho intermitente.
Threshold_Interm	2	Delimita quantas vezes o Limiar_Atraso deve ser alcançado, para que o atual caminho primário seja considerado intermitente, e o mecanismo de sondagem seja acionado.
Threshold_C_Atraso	2s	Determina o tempo máximo de validade do parâmetro Cont_Atraso, que contabiliza quantas vezes se obteve um atraso no recebimento de pacotes de dados igual ou superior ao Limiar_Atraso.
Threshold_Espera_Pct	1s	Determina o tempo máximo permitido de atraso no recebimento de pacotes e aciona o mecanismo de sondagem caso esse valor seja alcançado.
Threshold_CWND	40.000 <i>bytes</i>	Determina qual o tamanho máximo da janela de transmissão para que ocorra o <i>handover</i> .
Per_Probe	3s	Determina o intervalo mínimo de ocorrência entre sondagem de pacotes. Esse parâmetro é válido quando o disparo de pacotes de sondagem é gerado devido a um comportamento intermitente do meio.
Max_Time_Wait	2s	Determina o tempo máximo de espera por respostas dos pacotes de sondagem enviados.

6.4 Apresentação do testbed

No *testbed* (figura 6.6), existem 10 segmentos de enlace numerados de 1 a 10. Os enlaces 1 e 5 são *wireless* padrão 802.11b/g, enquanto que os outros são *fast ethernet*.

Em todos os experimentos a serem realizados usando o *testbed*, o **cliente1** sempre solicitará o arquivo para o **servidor1**.

O **cliente1** é um nó móvel que tem duas interfaces *wireless* padrão 802.11 b/g, e todos os outros elementos da rede utilizam o padrão 802.3 *fast ethernet* como forma de interligação entre os segmentos da rede. A interligação entre as redes *wireless* e *ethernet* se faz via os *access points*.

Como o nó móvel apresenta duas interfaces *wireless* associadas sobre a área de influências dos dois *access points* (AP1 e AP2), ele passará a ser uma estação *multi-homed* do ponto de vista do protocolo SCTP. A partir desse contexto, a associação estabelecida entre o **cliente1** e **servidor1** tem dois possíveis caminhos distintos que podem ser utilizados para a transmissão/recepção de pacotes de dados. Do ponto de vista do **servidor1**, temos: 4 – 3 – 2 – 1 como um possível destino, e 8 – 7 – 6 – 5, como um segundo possível caminho.

As estações fixas denominadas **cliente2** e **servidor2** são utilizadas quando forem necessárias para geração de tráfego concorrente, sempre utilizando o caminho 9 – 2 – 3 – 10.

A presença de duas tecnologias de protocolos da camada de enlace (*wireless* e *ethernet*) no *testbed* é importante, pois elas apresentam comportamentos diferentes em relação ao atraso e, naturalmente, podem apresentar maior sensibilidade aos parâmetros ajustáveis do protocolo SCTP. Dessa maneira, ficará comprovado que a extensão proposta ao protocolo não está sujeita a funcionar apenas em um determinado tipo de tecnologia de enlace.

Nos experimentos serão sempre transferidos arquivos de 15 e 45MB de tamanho. O tamanho do arquivo não é importante poderia se utilizar arquivos de qualquer tamanho.

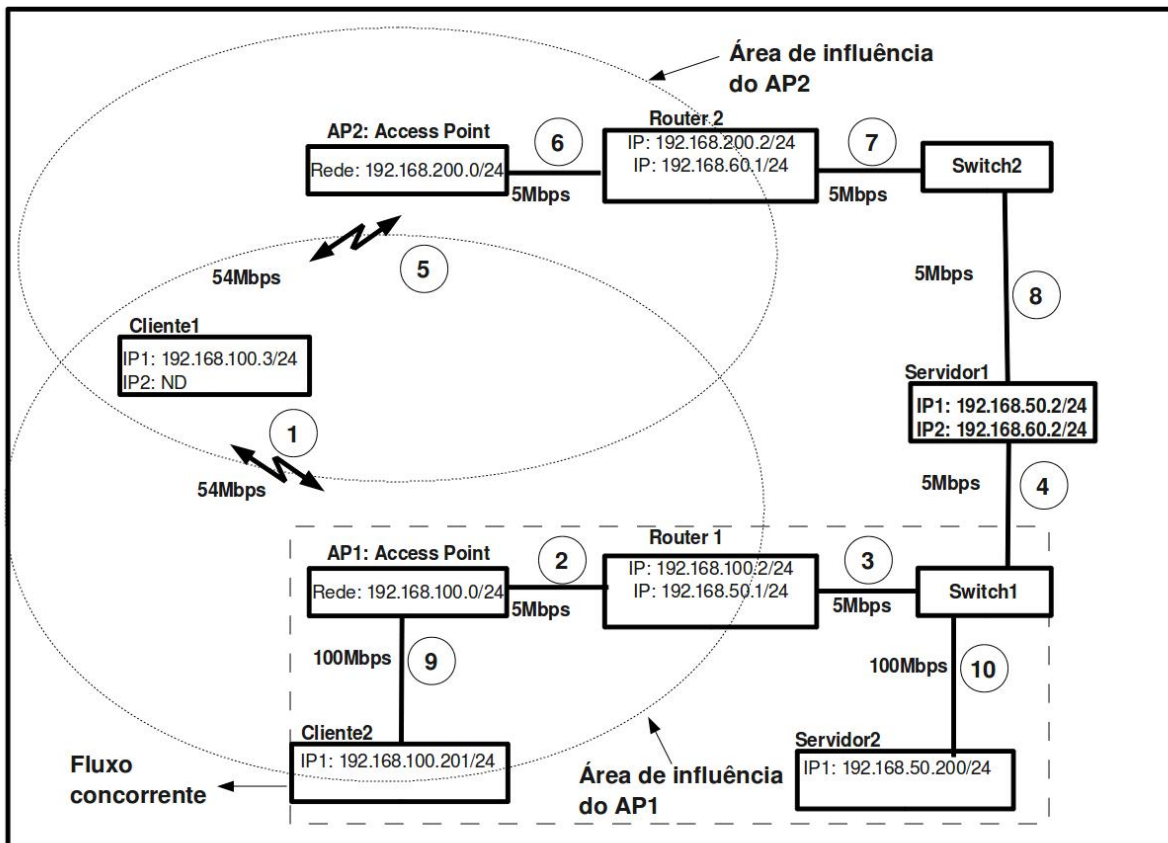


Figura 6.6: Testbed utilizado para realizar os experimentos

Configuração da máquina cliente1

- Processador: *Intel Pentium D 2.2 GHz*
- Memória: *1 GB - 667MHz - DDR2*
- Sistema operacional: *FreeBsD 7.2*
- Interfaces *wireless*: 2 interfaces *wireless-usb* da *Dlink* padrão 802.11A/B/G

Configuração da máquina servidor1

- Processador: *Pentium 4 de 1.8 GHz*
- Memória: *512 MB - 400 MHz - DDR*
- Sistema operacional: *FreeBsD 7.2*
- Interface *ethernet*: *Via Rhine II 100 Mbps*
- Interface *ethernet*: *Via Rhine II 100 Mbps*

Configuração do Router1

- Processador: *Pentium 4 de 1.8 GHz*
- Memória: *512 MB - 400 MHz - DDR*
- Sistema operacional: *FreeBsD 7.2*
- Interface *ethernet*: *SIS 100Mbps (onboard)*
- Interface *ethernet*: *Realtek 100Mbps*

Configuração do Router2

- Processador: *Pentium 4 de 1.8 GHz*
- Memória: *512 MB - 400 MHz - DDR*
- Sistema operacional: *FreeBsD 7.2*
- Interface *ethernet*: *SIS 100Mbps (onboard)*
- Interface *ethernet*: *Realtek 100Mbps*

Configuração do Cliente2

- Processador: *Pentium 4 de 1.8 GHz*
- Memória: *512 MB - 400 MHz - DDR*
- Sistema operacional: *FreeBsD 7.2*
- Interface *ethernet*: *SIS 100Mbps (onboard)*

Configuração do Servidor2

- Processador: *Pentium 4 de 1.8 GHz*
- Memória: *512 MB - 400 MHz - DDR*
- Sistema operacional: *FreeBsD 7.2*
- Interface *ethernet*: *SIS 100Mbps (onboard)*

Infraestrutura de rede

- Configuração dos *Access Point*: dois *Access Point* modelo *TP-LINK TD-W8920G*
- *Access Point* 1 (AP1): modelo *TP-LINK TD-W8920G* usando canal 1
- *Access Point* 1 (AP2): modelo *TP-LINK TD-W8920G* usando canal 8

- *Switch1*: *tenda* de 5 portas (modelo S105)
- *Switch2*: *encore* de 8 portas

É utilizado o sistema operacional *FreeBSD* em todas as estações devido à sua implementação nativa do SCTP baseada na RFC4960 e na RFC5061 (extensão do SCTP para mobilidade - mSCTP), conforme comentado por Lei et al. [27] em seu artigo.

6.5 Descrição dos experimentos

Os experimentos aqui apresentados buscam demonstrar o comportamento do protocolo SCTP-Padiao, SCTP-Parametrizado, R-SCTP (proposto por Kim et al. [1]) e o PA-SCTP quando submetidos a situações de interrupção do meio, presença de tráfego concorrente no caminho primário e movimentação dos nós da associação.

Todos os experimentos documentados posteriormente foram submetidos ao *testbed* (descrito anteriormente) para transferir arquivos com tamanhos de 15 e 41MB do servidor para o cliente. Em cada experimento, o arquivo foi transmitido por cinco vezes sempre no sentido do servidor para o cliente.

Cada experimento (descritos mais adiante) executado no *testbed* gerou dados que produziram vários gráficos. Os gráficos gerados se referem ao atraso, ao atraso médio, à variação da CWND e à vazão.

A aplicação cliente e servidora, utilizada em todos os experimentos, foi escrita em linguagem C e compilada para a plataforma FreeBSD utilizando a API documentada por Xie et al. [8] de 2009, baseada na RFC4960 de 2007.

Os experimentos realizados, para demonstrar o comportamento do SCTP-Padiao, SCTP-Parametrizado, R-SCTP e PA-SCTP em cenários diversos, estão representados nas tabelas 6.5, 6.6 e 6.7.

Tabela 6.5: Condições impostas ao tráfego de pacotes de dados pela rede

Protocolo	Condição	Interrupção do meio	Tráfego concorrente	Movimentação
SCTP-Padiao, SCTP-Param, R-SCTP e PA-SCTP	Condição 1	Não	Não	Não
SCTP-Padiao, SCTP-Param, R-SCTP e PA-SCTP	Condição 2	Sim	Não	Não
SCTP-Padiao, SCTP-Param, R-SCTP e PA-SCTP	Condição 3	Não	Sim	Não
SCTP-Padiao, SCTP-Param, R-SCTP e PA-SCTP	Condição 4	Não	Não	Sim
SCTP-Padiao, SCTP-Param, R-SCTP e PA-SCTP	Condição 5	Não	Sim	Sim

Tabela 6.6: Protocolos utilizados nos experimentos

Protocolo	Comentário
SCTP-Padrão	Usa o SCTP com valores de parâmetros padrões (conforme tabela 6.7).
SCTP-Parametrizado	Usa o SCTP com valores de parâmetros alterados (conforme tabela 6.7). Essa nomenclatura foi aqui criada para diferenciar o SCTP com os valores de parâmetros sugeridos pela RFC, do SCTP o qual teve os seus parâmetros configuráveis alterados pelo programador.
R-SCTP	Solução baseada no RTT (Proposta por Kim et. al. [1]). Usa valores de parâmetros conforme tabela 6.7.
PA-SCTP	Mecanismo proposto para efetuar o <i>handover</i> no SCTP.

Tabela 6.7: Valores para os parâmetros configuráveis do protocolo SCTP

Protocolo	PMR	RTO.Min	RTO.Max	HB.Interval
SCTP-Padrao	5 Tent.	1 seg	60 seg	30 seg
SCTP-Parametrizado	1 Tent.	100 miliss.	101 miliss.	1 seg
R-SCTP	5 Tent.	1 seg	60 seg	1 seg
PA-SCTP	5 Tent.	1 seg	60 seg	30 seg

Descrição das colunas das tabela 6.5 e 6.7

- **Interrupção do meio** - informa que, em algum momento do período de transmissão, ocorreu a interrupção do meio primário.
- **Tráfego concorrente** - informa que, em algum momento durante o período de transmissão, houve algum tipo de tráfego de dados concorrentes.
- **Movimentação** - informa que o cliente se movimentou de um *access point* em direção a outro, forçando, dessa forma, a deterioração de um caminho em relação ao outro.
- **PMR** - quantidade máxima de retransmissão permitida a um caminho primário antes que o protocolo SCTP efetue um *handover*.
- **RTO.Min** - informa qual o menor valor de RTO permitido pelo protocolo SCTP.
- **RTO.Max** - informa qual o maior valor de RTO permitido pelo protocolo SCTP.
- **HB.Interval** - informa o intervalo para geração de chunks *HEARTBEAT* utilizado pelo protocolo SCTP.

Para alguns cenários, os valores dos parâmetros configuráveis do protocolo SCTP foram alterados (tabela 6.7) para tornar o protocolo SCTP mais ágil para detectar oscilações que puderam ocorrer no meio de transmissão. A seguir, a justificativa do porquê desses valores especificamente.

- PMR - o valor padrão recomendado pela RFC 4960 é cinco. Para o SCTP-Parametrizado, esse parâmetro foi reduzido numa tentativa de deixar o protocolo mais sensível a condições do meio. Uma redução significativa do PMR pode provocar uma alternância constante do caminho primário, tornando-o muito sensível a qualquer tipo de interferência mínima do meio (CARO et. al. [11]). No artigo de Grace et al. [7], é comprovado que a redução do PMR não é garantia de ganho de desempenho em uma associação SCTP, uma vez que os caminhos podem ser assimétricos.
- RTO.Min – o valor mínimo para esse parâmetro recomendado pela RFC 4960 é de um segundo. O ajuste dessa variável é em milissegundos e foi alterado para 100 milissegundos no caso do SCTP-Parametrizado para deixá-lo bem sensível às condições do meio, mesmo tendo ciência de que valores abaixo de um segundo são dependentes do nível de qualidade do meio de transmissão, ou seja, valores abaixo de um segundo poderão trazer melhorias quando se usa apenas o padrão *ethernet*, mas, em ambientes *wireless*, ele pode gerar prejuízos na transmissão de pacotes de dados, devido a fatores como aqueles documentados por Ye et al. [41]. Valores abaixo de um segundo são suscetíveis a desnecessários *timeouts* (STEWART [3]) e consequente a retransmissões de pacotes de dados de forma desnecessária, conforme comentado por Jungmaier et al. [22].
- RTO.Max – o valor desse parâmetro recomendado pela RFC 4960 é de 60 segundos. Ao ser computado um novo valor para o RTO, o seu tamanho é limitado ao máximo pelo RTO.Max. Para o SCTP-Parametrizado, o RTO.Max foi alterado para 110 milissegundos buscando deixar o protocolo mais ágil em reagir a oscilações do meio.
- HB.Interval – o valor desse parâmetro recomendado pela RFC 4960 é de 30 segundos. Esse parâmetro teve o seu valor reduzido para um segundo quando usado o SCTP-Parametrizado e o R-SCTP, ou seja, o teste de conectividade do caminho secundário se deu a cada segundo de inatividade. Esse ajuste possibilitará ao protocolo conhecer de forma mais instantânea a situação do caminho secundário, além de poder proporcionar um melhor valor de RTT. Valores de HB.Interval muito baixo geram um tráfego constante de informações de controle que podem prejudicar outras aplicações (necessariamente não precisam ser SCTP) que possam estar, em algum momento, utilizando enlaces (que façam parte de um dos caminhos de destinos possíveis de uma associação SCTP), limitando, dessa forma, desnecessariamente, a largura de banda disponível. Nesse caso, o SCTP utilizará dois caminhos distintos. Um transmitirá pacote de dados e outro, apenas pacotes de controle em excesso.

Os parâmetros configuráveis (PMR, RTO.Min, RTO.Max e HB.Interval) do protocolo SCTP (Parametrizado) foram reduzidos especificamente para esses valores, comentados anteriormente, para que pudessem alcançar um desempenho similar ao R-SCTP e ao PA-SCTP.

Em todos os experimentos, o destino primário do servidor sempre foi de 192.168.50.2 para 192.168.100.3, e o destino secundário foi de 192.168.60.2 para 192.168.200.3 (endereço

IP sempre obtido quando o nó móvel se encontram na área de influência do AP2). Essa regra só foi desrespeitada quando explicitamente comentada no texto.

Com relação a picos de atrasos que possam aparecer nos gráficos, segundo Ye et al. [41] e Shi et al. [21], quando se utiliza um meio de transmissão *wireless*, é factível a presença oscilação na capacidade de transmissão. Noonan et al. [37] afirmam que o atraso sobre o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) é altamente variável e particularmente sensível ao ambiente de atuação.

Em relação aos gráficos, cada experimento foi realizado por cinco vezes consecutivas e o experimento utilizado para gerar os gráfico sempre foi aquele que obteve o tempo de transmissão mais próximo da média computada na realização do mesmo experimento por cinco vezes.

6.6 Experimento 1 - condição 1

Nesse experimento, é demonstrado o comportamento do SCTP-Padrão, SCTP-Parametrizado, R-SCTP e o PA-SCTP em uma situação ideal em que a condição do caminho primário poderá atender perfeitamente a qualquer necessidade da aplicação durante todo o tempo de vigência da associação. Nesse experimento, o meio de transmissão se encontra na **Condição 1** (tabela 6.5).

Os gráficos da figura 6.7 apresenta, a variação da CWND. Todos os gráficos foram gerados do procedimento de transferência de um arquivo de 15MB e posteriormente de 41MB usando o SCTP-Padrao, SCTP-Parametrizado, R-SCTP e PA-SCTP.

Para cada gráfico gerado, foi realizado o mesmo experimento cinco vezes. O fluxo de pacotes de dados SCTP sempre fluiu do servidor para o cliente, conforme os papéis sugeridos pelo *testbed* documentado na figura 6.6.

Durante o experimento, foi considerado que **Cliente1** permaneceu parado sobre a área de influência dos dois AP, sendo que recebeu o IP 192.168.100.3 do AP1 e 192.168.200.3 do AP2. O **Cliente1** inicia a associação com o **Servidor1** via 192.168.100.3 e permaneceu imóvel até o término da associação.

Na ótica do servidor, o destino primário inicial foi 192.168.100.3, e o secundário 192.168.200.3.

6.6.1 Com relação à variação da janela de transmissão CWND (gráficos da figura 6.7)

Os gráficos comentados nesse experimento estão representados por meio da figura 6.7.

Ao analisarmos a variação da CWND, é perceptível que o SCTP-Padrao e o PA-SCTP

mantiveram uma variação previsível e constante. Nesses gráficos, é perfeitamente visível que não houve alteração do caminho primário nem perdas de pacotes. Se houvesse a troca do caminho primário a CWND seria reduzida a 4.380B. No caso de perda de pacotes essa redução seria para $1 * MTU$.

Ao analisarmos os gráficos referentes ao SCTP-Parametrizado (15 MB), é perceptível que ocorreram *timeouts* várias vezes, forçando, dessa forma, a redução da janela de transmissão (CWND) para 1 MTU. Isso ocorreu devido ao valor reduzido do RTO.Min e RTO.Max, o que resultou na interpretação errônea por parte do protocolo SCTP sobre a perda de pacotes de dados, uma vez que o tempo de envio de um pacote de dados e o recebimento de uma confirmação SACK não foram suficientes. Isso também aconteceu com muita frequência quando analisamos o SCTP-Parametrizado (41MB). É visível também que houve pelo menos duas alterações do caminho primário quando analisamos o gráfico resultante da transferência de 15 MB e várias vezes no de 41MB.

Ainda analisando os gráficos referentes ao SCTP-Parametrizado (15MB), os momentos de *timeouts* podem ser percebidos nos períodos entre 0-2s, 10-12s, 30-32s, 32-34s, 36-38s, 40-42s e 44-46s. Os momentos de troca do caminho primário podem ser percebidos nos tempos 2-4s e 38-40s.

Com relação aos gráficos referentes ao SCTP-Parametrizado (41MB), os momentos de *timeouts* podem ser percebidos nos períodos entre 0-5s, 25-30s, 30-35s, 40-45s, 50-55s, 60-65s, 80-85s, 85-90s, 95-100s, 100-105s, 115-120s, 120-125s, 125-130s, 130-135s, 135-140s e 140-145s. Os momentos de troca do caminho primário podem ser percebidos nos tempos 20-25s, 25-30s, 45-50s, 50-55s, 55-60s, 80-85s, 85-90s, 95-100 e 125-130s.

No caso dos gráficos do R-SCTP, como não houve alteração do parâmetro do RTO em relação ao valor sugerido pela RFC do SCTP, não teve a ocorrência de *timeouts* e, consequentemente, a redução da CWND para o valor de 1MTU. É perceptível que houve várias alterações do caminho primário visto que, em vários momentos, a janela de transmissão foi reduzida para 4.380B.

Especificamente no caso do desempenho com o R-SCTP (15MB), apresentou-se com frequência a troca do caminho primário de forma desnecessária. Essa troca ficou clara nos períodos entre 0-2s e entre 24-30 segundos, em que houve aproximadamente quatro trocas do caminho primário nesse curto período de tempo.

Em relação ao R-SCTP (41MB), as trocas do caminho primário se apresentaram entre 5-10s, 35-40s, 50-55s, 85-90s e 105-110s. Como os dois caminhos da associação se encontravam em perfeitas condições, fica claro que as trocas do caminho primário foram feitas de forma desnecessária.

É importante observar que, mesmo em condições perfeitas, tanto o SCTP-Parametrizado como o R-SCTP consideraram o seu atual caminho primário não aptos segundo a sua política de *handover*. Com isso, não se obteve nenhum ganho para o SCTP-Parametrizado e um ganho insignificante para o R-SCTP, mas poderia ter havido uma perda significativa para

aplicações concorrente que estivessem utilizando segmentos de enlaces que compunham o caminho da associação Sctp-Parametrizado/R-Sctp. Uma perda significativa seria imposta à aplicação se a capacidade de transmissão do meio primário e secundário fossem diferentes.

A alternância do caminho primário, quando bem sintonizada, pode trazer melhorias em relação ao tempo total de transmissão do arquivo, para aplicações de tempo real e até mesmo para um cenário de troca de rede em um ambiente de múltiplas redes *wireless* sobrepostas ou parcialmente sobrepostas. O problema é que, quando mal administrado, esse recurso poderá trazer uma disputa danosa em relação a outros protocolos de transporte (UDP e TCP) que não façam uso desse recurso. Na prática, uma aplicação Sctp pode fazer uso constante de dois caminhos para transmissão. O uso de apenas um, de forma constante, traria o mesmo resultado. Um bom exemplo disso é quando comparado, nesse experimento, o R-Sctp e o Sctp-Parametrizado com o PA-Sctp.

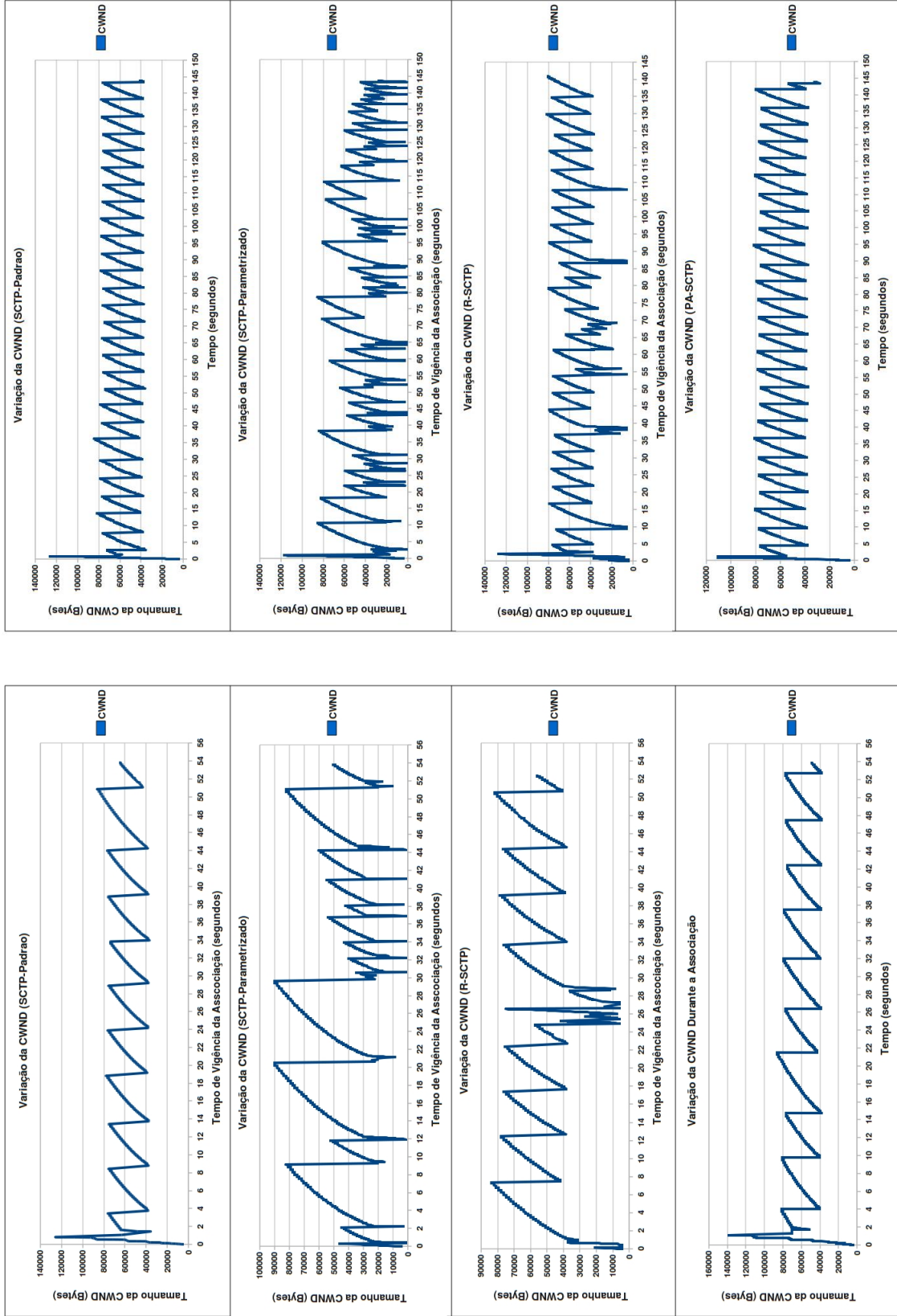


Figura 6.7: Variação da CWND no transmissor referente a arquivos de 15MB e 41MB

6.6.2 Conclusão do experimento

O PA-SCTP e o SCTP-Padiao se mostraram mais sensatos por não fazerem alterações do caminho primário já que este não apresentava qualquer tipo de interferência.

O SCTP-Parametrizado se comportou de forma desleal na disputa por recursos de rede com possíveis aplicações concorrentes que pudessem fazer uso de algum enlace que compõe a associação do SCTP-Parametrizado, uma vez que praticamente durante todo o tempo fez uso dos dois caminhos disponíveis para transmitir pacotes de dados. O caminho primário foi usado para a transmissão normal dos pacotes de dados, e o secundário para retransmissão dos pacotes de dados que sofreram *timeout* no caminho primário. Junta-se a isso a realização da troca do caminho primário de forma desnecessária.

Apesar do R-SCTP ter conseguido o melhor tempo de transmissão, houve a troca do caminho primário várias vezes desnecessariamente. No cenário da transferência do arquivo de 15MB, é perceptível a queda de desempenho entre 25 e 30 segundos devido à constante troca do caminho primário, ou seja, o seu algoritmo fez interpretações equivocadas sobre a condição do meio, levando a execução *handover* precipitadamente. Esse comportamento, na verdade, poderia levar a uma degradação da desempenho que poderia ter sido sentida pela aplicação, uma vez que a troca do caminho primário levou a redução da CWND para 4.380B.

Considerando as observações citadas, conclui-se que o PA-SCTP e o SCTP-Padiao se mostraram mais aptos nesse ambiente.

6.7 Experimento 2 - condição 2

Nesse experimento, é demonstrado o comportamento do SCTP-Padiao, SCTP-Parametrizado, R-SCTP e o PA-SCTP em uma situação em que o caminho primário foi interrompido fisicamente de forma abrupta. Após o estabelecimento da associação, o destino primário foi interrompido definitivamente, forçando, dessa maneira, a tomada de decisão de realizar a troca do destino primário para envio de pacotes de dados, ou seja, o *handover*.

O momento da interrupção do caminho primário de transmissão, adotado nesse experimento, foi de 30 segundos após o início da associação. A interrupção pode ser feita a qualquer momento, portanto 30 segundos não é nenhum valor especial.

A interrupção do meio de transmissão, adotada para esse experimento, é a forma manual, ou seja, desconectando o cabo de rede em qualquer ponto que interrompa o fluxo pelo caminho primário. O ponto de interrupção não exerce qualquer tipo de influência no procedimento de tomada de decisão de *handover*.

Os gráficos das figuras 6.8, 6.9, 6.10 e 6.11 apresentam, respectivamente, o atraso, o atraso médio, a CWND e a vazão gerados no recebimento de um arquivo de 15MB e 41MB usando o SCTP-Padiao, SCTP-Parametrizado, R-SCTP e PA-SCTP.

A cada arquivo transferido foram gerados gráficos referentes a atraso, atraso médio, CWND e vazão. Alguns desses, no caso, o atraso, o atraso médio e a vazão foram formados a partir da computação dos pacotes de dados recebidos, e a CWND, a partir da computação dos enviados.

Para cada gráfico gerado, foi realizado o mesmo experimento cinco vezes. O fluxo de pacotes de dados SCTP sempre fluiu do servidor para o cliente, conforme os papéis sugeridos pelo *testbed* documentado na figura 6.6.

Durante o experimento, foi considerado que o **Cliente1** esteve parado sobre a área de influência dos dois AP, sendo que recebeu o IP 192.168.100.3 do AP1 e 192.168.200.3 do AP2. O **Cliente1** iniciou a associação com o **Servidor1** via 192.168.100.3 e permaneceu imóvel até o término da associação.

Na ótica do servidor, o destino primário inicial foi 192.168.100.3, e o secundário 192.168.200.3.

6.7.1 Com relação ao atraso e ao atraso médio (gráficos das figuras 6.8 e 6.9)

Quando analisado o tempo total de transmissão, o SCTP-Parametrizado (15 e 45MB) foi o que conseguiu concluir o processo de transmissão do arquivo em menor tempo. Ele conseguiu isso devido aos seus parâmetros reduzidos (RTO.Min e RTO.Max) provocarem uma maior sensibilidade quanto a situação do meio, fazendo um chaveamento de troca do caminho primário de forma mais rápida que as demais soluções aqui analisadas.

Ao analisarmos o atraso e o atraso médio, os melhores resultados foram conseguidos com o uso do SCTP-Parametrizado, que, devido à constante interpretação de perda dos pacotes enviados, forçou um frequente reenvio de pacotes. Com isso, houve a geração de pacotes de controle SACK a todo momento, devido ao frequente recebimento de pacotes duplicados. Isso ocorreu devido ao equívoco do SCTP-Parametrizado em taxar o pacote como perdido baseando-se no valor do RTO reduzido, não adequado, para o meio de transmissão em uso. Com um maior fluxo de pacotes SACKs recebidos, o transmissor de pacotes de dados pôde construir uma visão mais imediatista da situação do meio de transmissão e, assim, conseguiu transmitir pacotes a todo momento, entretanto em uma menor frequência devido aos ajustes de o RTO.Min e do RTO.Max produzirem uma constante sensação de perda de pacotes por parte do transmissor inibindo, dessa forma, o crescimento da sua janela de transmissão.

Os grandes atrasos gerados pelo SCTP-Padrao já eram esperados, conforme documentado na seção 4.1. O que se tem a acrescentar nesse gráfico (figura 6.8) é que o protocolo SCTP usado para executar esse experimento nem sempre se comportou de forma tão clara na geração desse gráfico (figura 6.8), devido aos tamanhos dos picos de atrasos gerados. Algumas vezes os testes mostraram apenas duas grandes paradas de 32 segundos cada uma, e outras vezes o procedimento de *handover* foi executado em aproximadamente 45 segundos

após a primeira ocorrência de *timeout*.

No caso do R-SCTP, não foi possível executar esse experimento devido ao fato de ele, na maioria das vezes, entrar no modo de *loop* eterno. Quando há a interrupção do meio, o RTT do caminho interrompido não pode ser atualizado e sempre ficará com o mesmo valor, e a transmissão dos pacotes, na maioria das vezes, fica presa no caminho que não apresenta conectividade. O R-SCTP foi feito para trabalhar apenas em ambientes *multi-homing*.

O PA-SCTP apresentou uma interrupção na transmissão de pacotes de dados entre um e dois segundos. O PA-SCTP possui dois mecanismos que o auxiliam na tomada de decisão de troca do caminho primário, sempre usando pacotes de sondagem. Um dos mecanismos dispara pacotes de sondagem baseado-se no atraso médio gerado a cada recebimento de pacotes de dados. A sondagem do caminho secundário é executada quando o atraso médio computado no caminho primário ultrapassa a 50 milissegundos por no mínimo duas vezes em um período de dois segundos. Mas, para esse mecanismo funcionar, precisa haver a conectividade do meio. Quando não há conectividade, outro mecanismo é executado, o qual dispara automaticamente pacotes de sondagem quando não ocorre recebimento de pacotes de dados por um período igual ou superior a um segundo. Foi esse mecanismo, nesse experimento, que disparou sondagem que ocasionou a posterior troca do caminho primário no PA-SCTP.

Mesmo o pico de atraso e do atraso médio gerado pelo SCTP-Parametrizado ter sido bem menor do que o gerado pelo PA-SCTP, o tempo de transmissão foi praticamente o mesmo. Isso aconteceu devido ao SCTP-Parametrizado ter a constante sensação de *timeout* forçando constantes retransmissões e a redução da sua janela de transmissão de forma desnecessária, já que o meio estava sendo usado apenas por ele e, com isso, se perderam milissegundos que acumulados praticamente inviabilizaram o ganho de tempo ocorrido na detecção e na troca do caminho primário, quando comparado ao PA-SCTP.

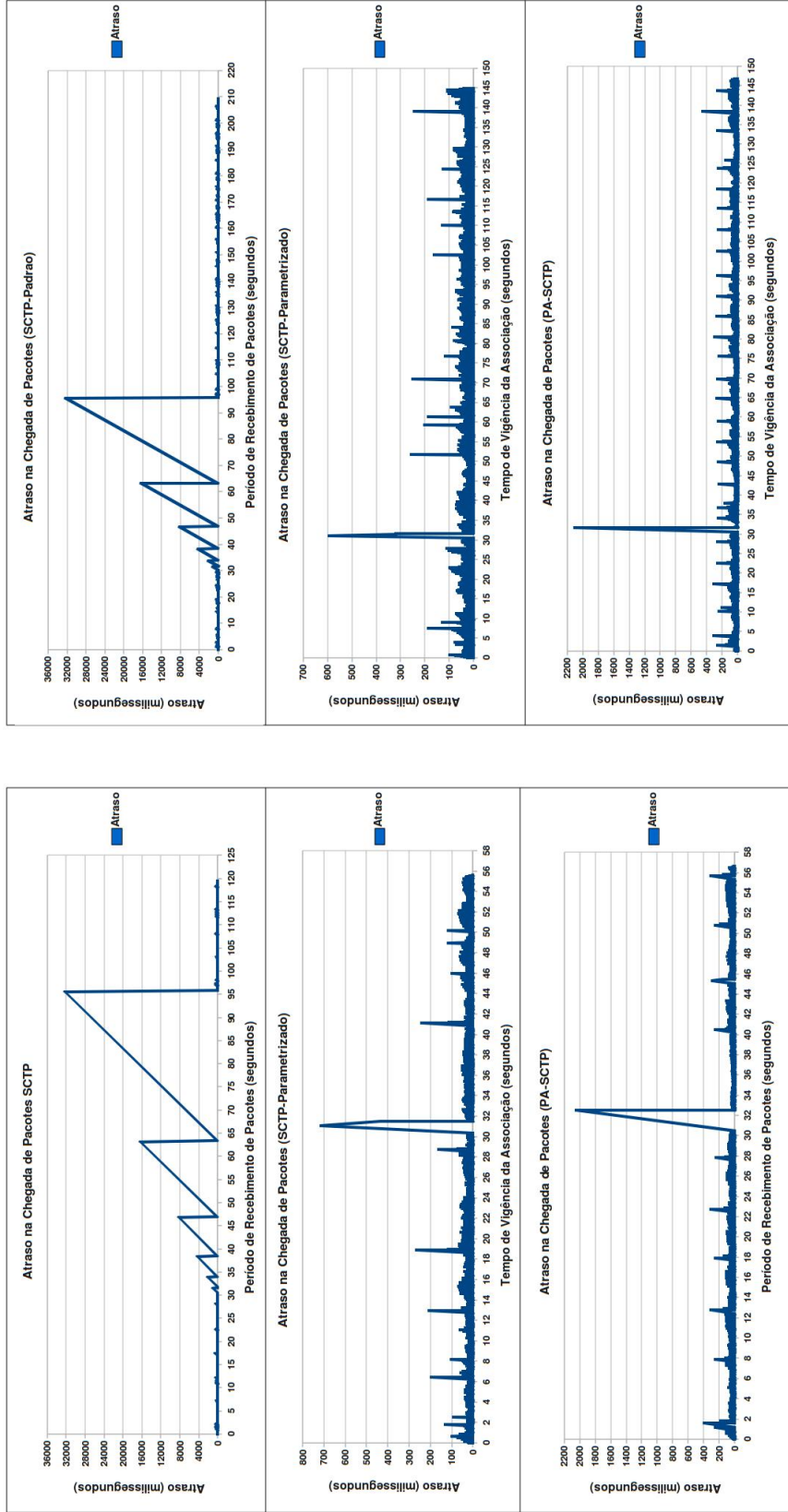


Figura 6.8: Atrasos durante o recebimento de arquivos de 15 e 41MB

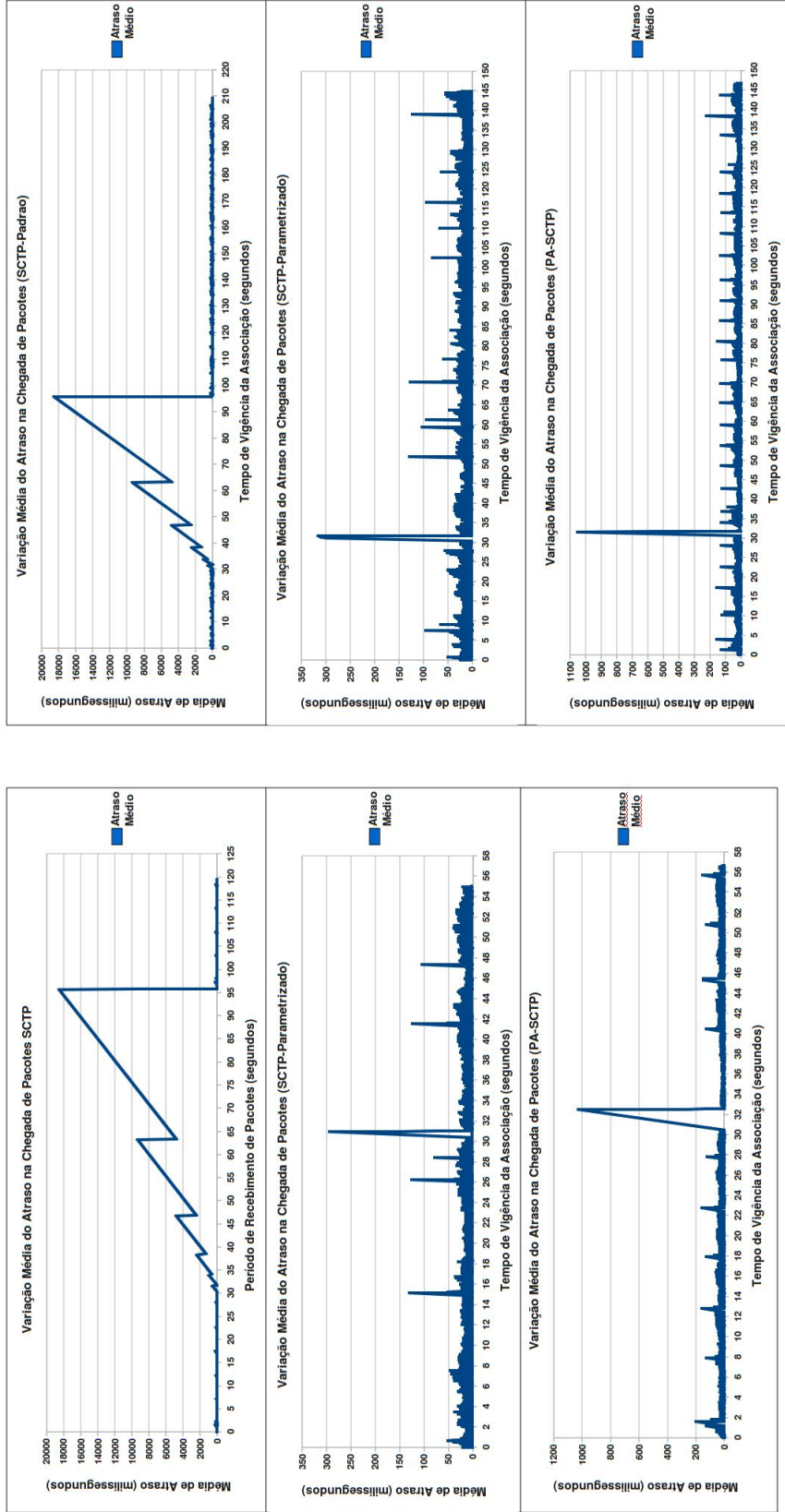


Figura 6.9: Atrasos médios durante o recebimento de arquivos de 15 e 41MB

6.7.2 Com relação à variação da janela de transmissão CWND (gráficos da figura 6.10)

Os gráficos comentados nesse experimento estão representados por meio da figura 6.10.

Ao analisarmos a variação da CWND, é perceptível que o SCTP-Padrao e o PA-SCTP mantiveram uma variação previsível. O único momento de troca do destino dos pacotes ocorreu quando o caminho primário foi interrompido.

Quando analisamos os gráficos referentes ao SCTP-Padrao (15 e 41 MB - figura 6.10), fica claro que houve a interrupção do fluxo de transmissão e sua janela de transmissão foi a zero. Esse gráfico mostra claramente a latência do SCTP em tomar a decisão de realizar o *handover*.

Com relação aos gráficos referentes ao uso do SCTP-Parametrizado, também fica claro a redução brusca e muito significativa da sua janela de transmissão devido a fatores relacionados aos ajustes feitos no RTO.Min e no RTO.Max. Mesmo estando a largura de banda toda disponível para a aplicação, o protocolo teve nítidas sensações de perda de pacotes e é claro que isso acarretou atrasos relacionados a reenvio e redução da sua janela de transmissão.

O PA-SCTP se mostrou objetivo, ou seja, sondou o caminho secundário e naturalmente alterou o caminho primário recuperando logo, em seguida, praticamente a mesma capacidade de transmissão que detinha quando usava o antigo caminho primário.

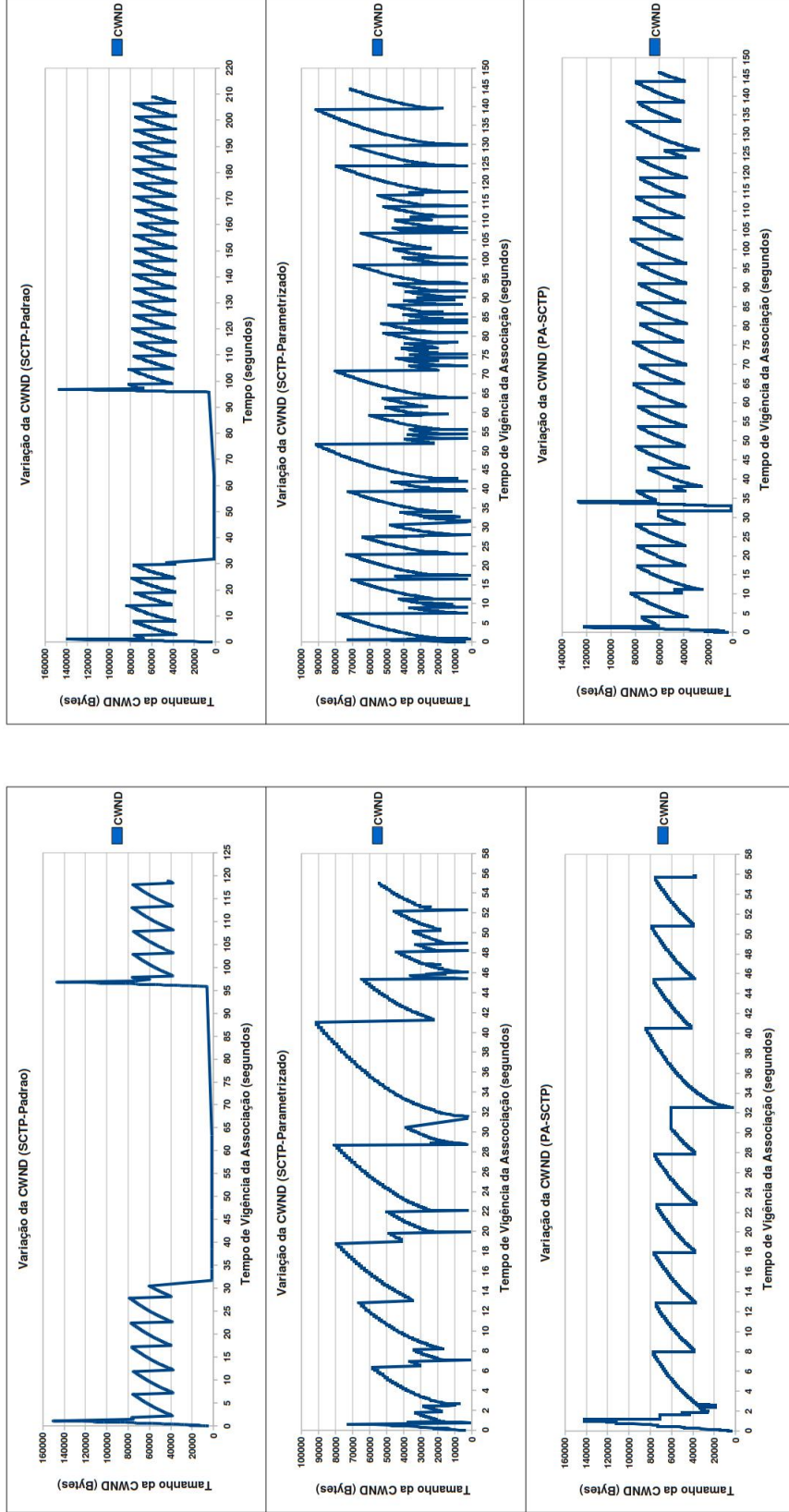


Figura 6.10: Variação da CWND no transmissor referente a arquivos de 15MB e 41MB

6.7.3 Com relação à vazão (gráficos da figura 6.11)

Quando analisado o gráfico gerado pelo uso do SCTP-Padrao, é perceptível a interrupção no recebimento de pacotes de dados desde o período da ocorrência do primeiro *timeout* até a efetivação do *handover*. A quantidade de pacotes transmitidos durante esse período foi menor do que uma dezena, pois, após a ocorrência do *timeout*, a capacidade de transmissão do protocolo se reduziu a 1MTU, que, nesse caso, é de 1.500B.

A percepção visual de interrupção no recebimento de pacotes de dados com o SCTP-Parametrizado (41MB) foi irrelevante. O tempo perdido, em detectar a falha no atual caminho primário e a sua posterior troca é insignificante no contexto do tempo gasto para a transmissão do arquivo. No entanto o tempo perdido é perceptível quando analisado o gráfico relacionado a atraso e atraso médio.

Com o uso do PA-SCTP, é visível uma pequena parada no recebimento dos pacotes de dados, pois o algoritmo responsável por executar o *handover* espera um segundo sem recebimento de pacotes de dados, antes de autorizá-lo.

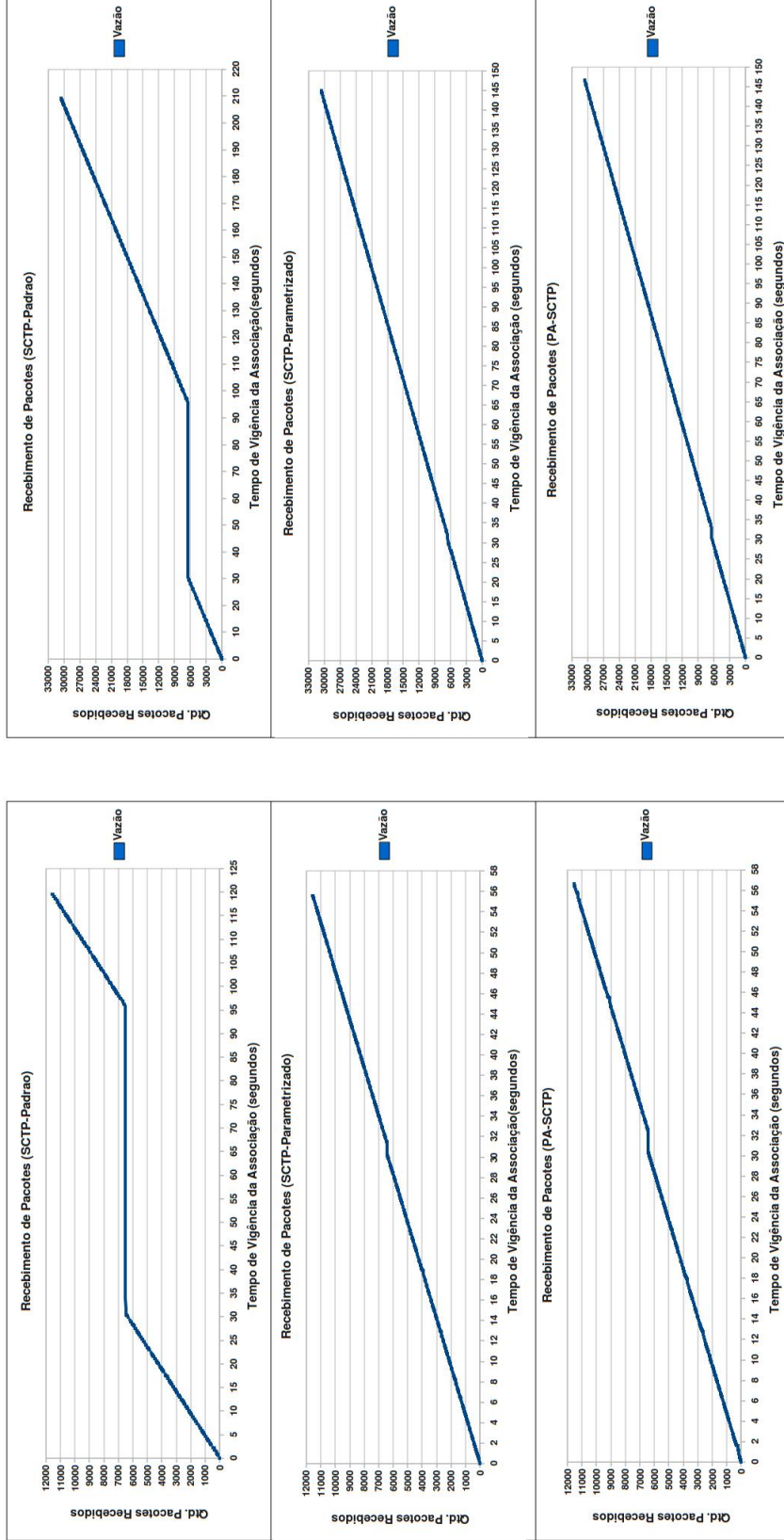


Figura 6.11: Vazão na ótica do receptor referente a arquivos de 15MB e 41MB

6.7.4 Conclusão do experimento

Nesse experimento, ficou clara a ineficácia do SCTP-Padrao, mesmo para aplicações que não exigem algum tipo de atraso máximo na chegada de pacotes. A latência gerada pelo seu mecanismo de troca do caminho primário não o fez atraente em relação a outros protocolos de transporte, como podemos perceber nos gráficos do experimento 2.

O SCTP-Parametrizado, por sua vez, observando os resultados de forma macro, aparentemente, resolveu o problema da latência. No caso, bastou alterar os parâmetros configuráveis do protocolo SCTP, RTO.Min, RTO.Max, PMR e HB.Interval, para produzir um efeito praticamente idêntico a um ambiente que não apresente perda da conectividade. Isso fica claro quando comparado os gráficos desse experimento com o do experimento 1.

O problema de ajustes nos parâmetros configuráveis do protocolo SCTP, no nosso caso aqui o SCTP-Parametrizado, é que, para obter o resultado desejado, os ajustes nos parâmetros configuráveis são dependentes do meio de transmissão, ou seja, os valores dos parâmetros configuráveis estão associados a uma determinada condição do meio. Modificando esse meio, como, por exemplo, largura de banda disponibilizada no momento, poderá conduzir a resultados indesejados.

Outro problema associado a esses ajustes dos parâmetros configuráveis do protocolo SCTP é que, ao observar mais atentamente os gráficos referentes a CWND do SCTP-Parametrizado, em vários momentos, a janela de transmissão ficou em 4.380B e 1.500B. O primeiro valor é referente a troca do caminho primário e o segundo valor foi referente a ocorrência de *timeout*. Após a perda de conectividade, observando o gráfico referente a CWND, houve vários momentos de redução da janela de transmissão para 1MTU. Esses *timeouts* forçam o uso, praticamente durante todo o tempo de vigência da associação, do *slow start* como controlador de congestionamento para o SCTP-Parametrizado. Isso reduziu claramente a capacidade do protocolo de manter um fluxo alto no envio de pacotes de dados, ou seja, o meio foi mau aproveitado.

No caso do R-SCTP, houve uma falha de execução do protocolo, uma vez que não ocorreu alterações do RTT quando existiu a perda da conectividade, levando o protocolo a um comportamento de *loop*, ou seja, preso ao caminho que não apresenta conectividade.

O PA-SCTP se mostrou mais uma vez uniforme e equilibrado uma vez fez a tomada de decisão de troca do caminho primário foi feito em função da demanda da aplicação. Seu procedimento não causa dano a possíveis aplicações concorrentes e mantém um fluxo mais uniforme na transmissão dos pacotes de dados, quando, por exemplo, comparado ao SCTP-Parametrizado, que se observado de forma macro precipitadamente poderia concluir-se que obteve o melhor desempenho.

6.8 Experimento 3 - condição 3

Nesse experimento, é demonstrado o comportamento do SCTP-Padrão, SCTP-Parametrizado, R-SCTP e o PA-SCTP em um meio de transmissão que existe a presença do tráfego concorrente. Nesse experimento, o meio de transmissão se encontra na **Condição 3** (tabela 6.5).

Para esse experimento, foi gerado um fluxo, similar ao VBR, concorrente àquele gerado por uma associação SCTP estabelecida. O fluxo concorrente é gerado pelo uso do programa **iperf**. O tráfego concorrente fluiu entre o **Cliente2** e **Servidor2**, conforme sugerido pelo *testbed* (figura 6.6).

O fluxo concorrente só usou os segmentos de rede numerados como 9-2-3-10, conforme demonstrado na figura 6.6.

A execução do *iperf* do lado cliente e servidor:

```
iperf -s -m -u – lado servidor
```

```
iper -c 192.168.50.200 -m -t3600 -P5 -u - lado cliente
```

O tráfego concorrente só foi iniciado 30 segundos após o estabelecimento da associação. O momento de geração do tráfego concorrente não é importante, pode ser gerado a qualquer momento. Nesse experimento, foi determinado que ocorresse após 30 segundos de atividade da associação SCTP para que pudesse ser melhor visualizado nos gráficos.

Os gráficos das figuras 6.12, 6.13, 6.14 e 6.15 apresentam, respectivamente, o atraso, o atraso médio, a CWND e a vazão gerada no recebimento de um arquivo de 15MB e 41MB usando o SCTP-Padrao, SCTP-Parametrizado, R-SCTP e PA-SCTP.

A cada arquivo transferido foram gerados gráficos referentes a atraso, atraso médio, CWND e vazão. Alguns desses, no caso, o atraso, o atraso médio e a vazão foram formados a partir da computação dos pacotes de dados recebidos, e a CWND, a partir da computação dos enviados.

Para cada gráfico gerado, foi realizado o mesmo experimento cinco vezes. O fluxo de pacotes de dados SCTP sempre fluiu do servidor para o cliente, conforme os papéis sugeridos pelo *testbed* documentado na figura 6.6.

Durante o experimento, foi considerado que o **Cliente1** estava parado sobre a área de influência dos dois AP, sendo que recebeu o IP 192.168.100.3 do AP1 e 192.168.200.3 do AP2. O **Cliente1** é que iniciou a associação com o **Servidor1** via 192.168.100.3. O Cliente1 ficou parado até o término da associação.

Na ótica do servidor, o destino primário inicial foi 192.168.100.3, e o secundário 192.168.200.3.

6.8.1 Com relação ao atraso e ao atraso médio (gráficos das figuras 6.12 e 6.13)

Quando analisado o tempo total de transmissão, o PA-SCTP (15 e 45MB) foi o que conseguiu concluir o processo de transmissão do arquivo em menor tempo. Ele conseguiu isso devido à sua melhor percepção das condições do meio de transmissão.

Ao observar o desempenho do SCTP-Padrazo, é visível que houve um aumento significativo do atraso e do atraso médio após o surgimento do tráfego concorrente. Em nenhum momento, o SCTP-Padrazo realizou o *handover*, o que acarretou na extensão do tempo de transmissão de 54 segundos para 130 segundos, no caso de transferência do arquivo de 15MB, e de 144s para 510s, no caso de transferência do arquivo de 41MB. Com esse experimento, é plausível concluir que o SCTP-Padrazo é bem conservador em relação à execução da troca do caminho primário.

No caso do SCTP-Parametrizado, houve naturalmente uma troca do caminho primário de forma rápida. Mesmo usando um caminho onde a largura de banda estava toda a seu dispor, o protocolo ainda insistia em fazer uso do caminho congestionado como primário ao primeiro sinal de insatisfação com o caminho usado atualmente para transmissão de pacotes de dados. Com isso, naturalmente, houve um crescimento do atraso e do atraso médio, conforme demonstrado nos gráficos (figura 6.12).

Com o R-SCTP, aconteceu um comportamento parecido ao do SCTP-Parametrizado. O R-SCTP alterou o caminho primário rapidamente ao primeiro sinal de congestionamento, mas, quando o novo caminho primário utilizado não atendia às suas expectativas momentâneas, o protocolo insistia em utilizar o caminho congestionado como primário. Isso provocou um aumento do atraso e do atraso médio, mostrado através dos gráficos (figura 6.12).

Já o PA-SCTP, nos dois casos, efetuou a troca do endereço de destino dos pacotes transmitidos ao interpretar que o atual caminho primário não mais atendia às necessidades da aplicação e, em nenhum momento, voltou a utilizar o caminho congestionado como primário. Isso mostrou mais uma vez a eficácia do PA-SCTP, que detectou a deterioração do caminho primário e tomou preventivamente a decisão de *handover*.

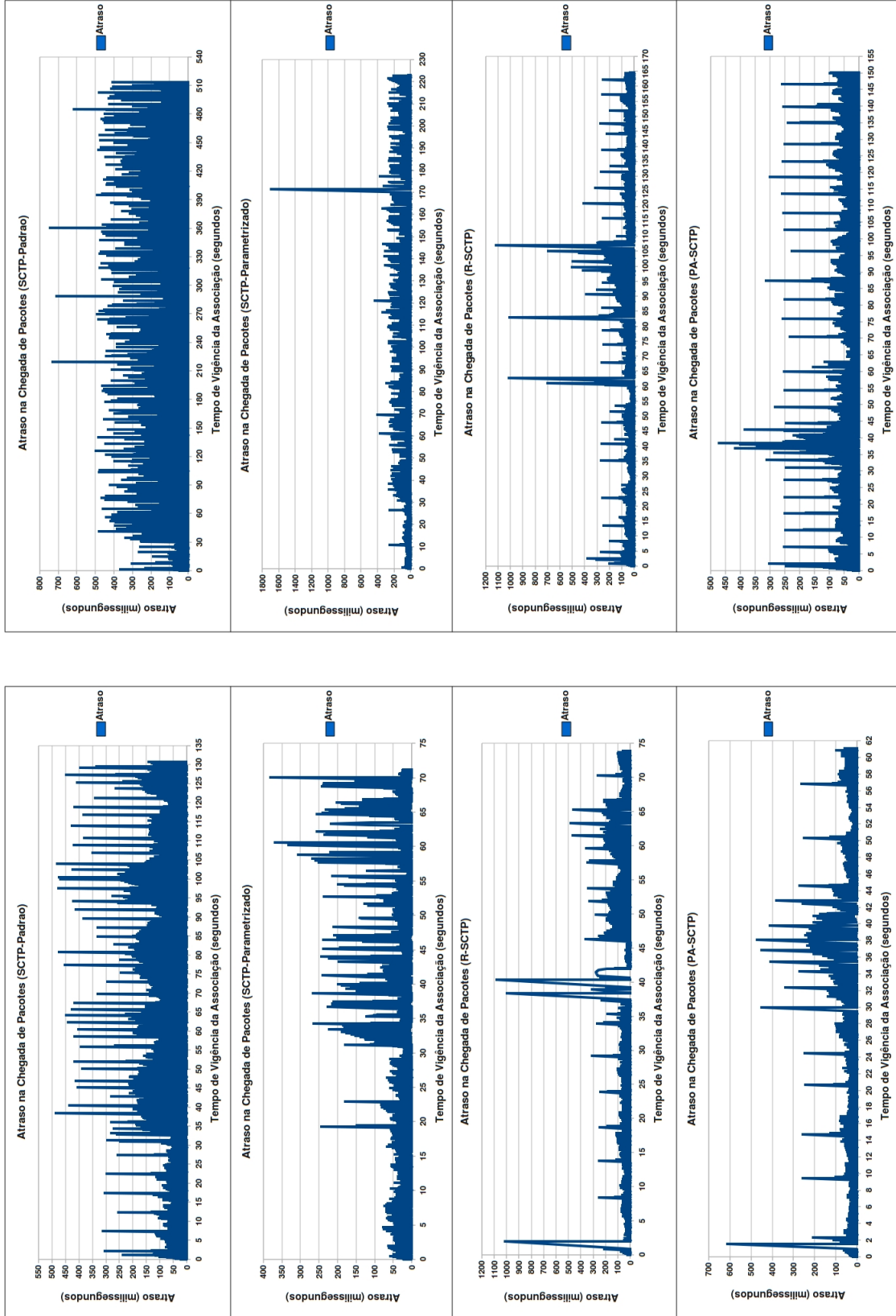


Figura 6.12: Atrasos durante o recebimento de arquivos de 15 e 41MB

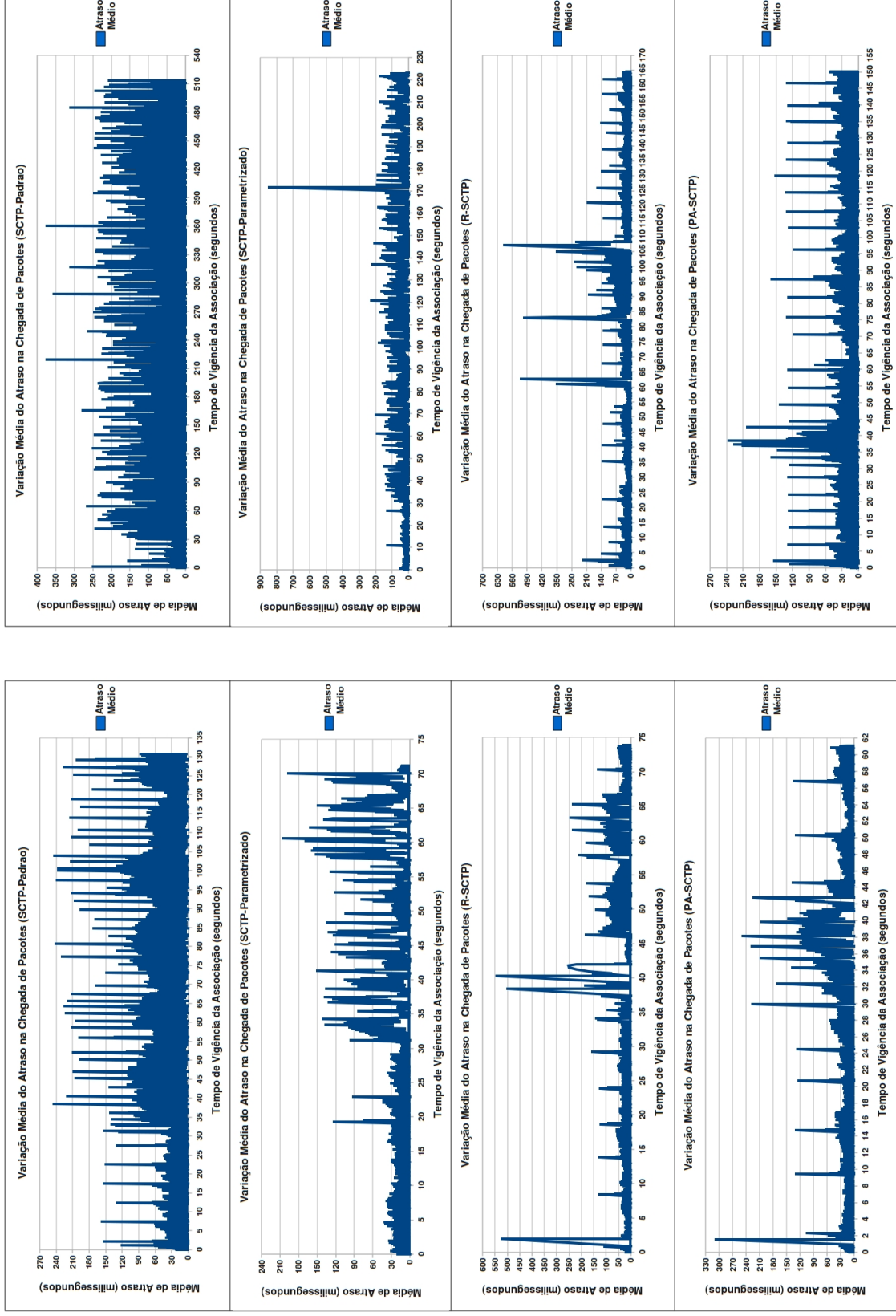


Figura 6.13: Atrasos médios durante o recebimento de arquivos de 15 e 41MB

6.8.2 Com relação à variação da janela de transmissão CWND (gráficos da figura 6.14)

Com relação ao SCTP-Padiao, é perceptível a redução significativa da janela de transmissão após 30 segundos de vigência da associação, mantendo-se baixa até o término da associação. Nesse cenário, praticamente o SCTP teve um desempenho similar ao protocolo TCP.

Analisando o comportamento do SCTP-Parametrizado, houve uma redução da janela de transmissão para um valor menor do que o SCTP-Padiao, mesmo assim ainda conseguiu obter um melhor desempenho devido à sua constante troca de caminho primário. Um efeito colateral dessa constante mudança é o uso de dois caminhos de transmissão praticamente ao mesmo tempo, ou seja, geram tráfego em dois caminhos e provocam uma concorrência desleal e danosa em ambientes onde coexistem aplicações que fazem uso de outros protocolos de transporte como TCP e UDP.

Após 30 segundos de associação, o SCTP-Parametrizado apresentou constantes trocas do caminho primário devido a interpretações equivocadas de perdas de pacotes. Isso é visível quando observado seus gráficos (figura 6.14). Essas trocas ocorreram com mais intensidade após 30 segundos do início da associação, constatado pela redução da janela de transmissão para 4.380B. Nos momentos em que houve perda de pacotes, conforme mostrado nos gráficos (figura 6.14), ocorreu a redução da janela de transmissão para 1.500B. Esse comportamento frequente, de troca do caminho primário, e perda de pacotes resultam em uso quase simultâneo dos dois caminhos disponibilizados na associação. No caso da ocorrência da perda de pacotes, houve a utilização do caminho secundário para a retransmissão do pacote perdido que sempre se realiza pelo outro caminho disponível na associação, ou seja, pelo caminho secundário.

O R-SCTP apresentou um decréscimo significativo da janela de transmissão. No caso do gráfico gerado durante a transmissão do arquivo de 15MB entre os tempos de 35 e 40 segundos e partir de 45 segundos, ele ficou preso ao uso do caminho congestionado como primário por aproximadamente 25 segundos, o que acarretou na queda do seu desempenho. No caso da transferência do arquivo de 41MB entre o tempo de 60 e 110 segundos aproximadamente, ele também passou esse período usando o caminho congestionado como primário o que também prejudicou o seu desempenho.

O R-SCTP utiliza o RTT como um indicador para determinar se o caminho está bom ou ruim. No caso dos gráficos (figura 6.14), o R-SCTP deixou de usar o novo caminho primário para utilizar o antigo caminho primário congestionado devido ao próprio controle de fluxo do protocolo, no envio de pacotes de dados, contribuir para o aumento no valor do RTT. No outro caminho, o uso de pacotes de sondagem pequenos, mesmo em um meio congestionado, produziu valores de RTT pequenos que, em alguns momentos, fez com que se alterasse o caminho primário novamente, fazendo que o R-SCTP voltasse a usar o caminho congestionado como primário.

O retorno do R-SCTP para usar o caminho congestionado como primário não ocorreu em todos os experimentos, já que foi repetido cinco vezes. Mas isso aconteceu em média de 60 a 70% das vezes.

Com relação ao comportamento do PA-SCTP, é perceptível pelos gráficos que o protocolo se comportou de forma esperada. Ao aparecimento do tráfego concorrente, houve, em um primeiro momento, a redução da janela de transmissão e, em um segundo momento, a efetivação do *handover*. Um fato curioso que se apresentou nesse experimento foi que, após realizado o *handover*, a janela de transmissão demorou mais do que era teoricamente esperado para aumentar a sua capacidade, já que migrou o seu caminho de transmissão de dados para um não congestionado. Ao fazer uma análise mais profunda dos dados, foi notado que, por algum motivo não plausível, a janela de transmissão se iniciou em 4.380B e cresceu até 6000B, permanecendo nesse tamanho por aproximadamente seis segundos e somente após esse tempo é que teve o crescimento que é normalmente esperado.

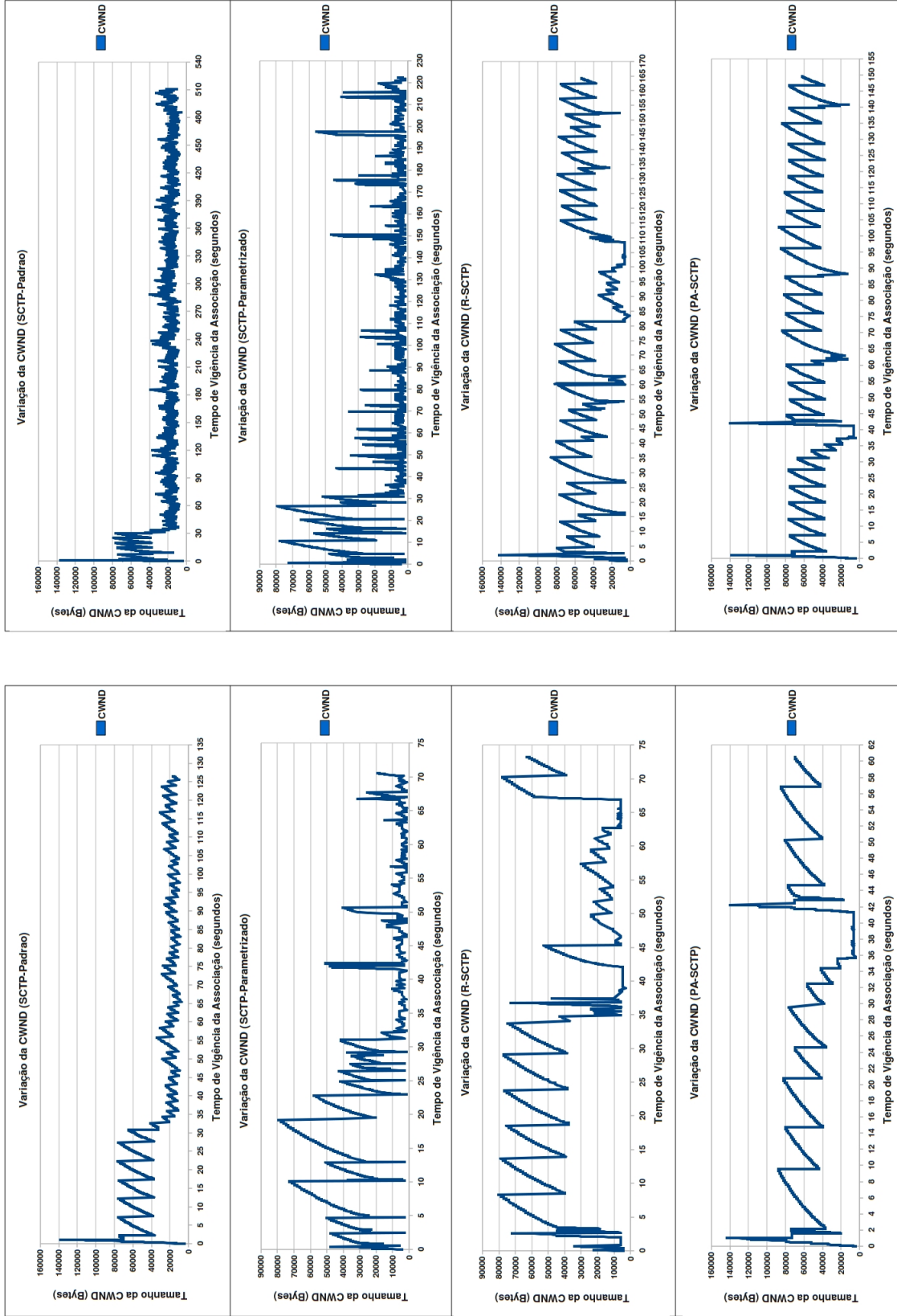


Figura 6.14: Variação da CWND no transmissor referente a arquivos de 15MB e 41MB

6.8.3 Com relação à vazão (gráficos da figura 6.15)

Quando analisados os gráficos gerados (figura 6.15) com o uso do SCTP-Padrao, é perceptível uma queda no recebimento de pacotes de dados por unidade de tempo. Isso é justificado pela insistência do protocolo SCTP-Padrao em permanecer utilizando o caminho primário, mesmo com ele se apresentando congestionado, e na presença de um caminho secundário livre de qualquer tipo de congestionamento. Isso claramente demonstra a ineficiência do SCTP-Padrao no uso do *multi-homed*.

O SCTP-Parametrizado também apresentou uma variação no recebimento de pacotes de dados após 30 segundos de estabelecimento da associação. Essa perda é justificada pelo comportamento desse protocolo em insistir de tempos em tempos a usar o caminho congestionado como primário.

O R-SCTP também demonstra nos gráficos (figura 6.15) oscilações no recebimento de pacotes de dados devido à insistência de tentar usar o caminho congestionado como primário para a transmissão de pacotes de dados.

Já o PA-SCTP apresentou uma pequena variação apenas após os primeiros segundos do aparecimento do tráfego concorrente. Após a efetivação do *handover*, não se apresentaram mais variações significativas no recebimento de pacotes de dados, como constatado com o uso dos outros protocolos.

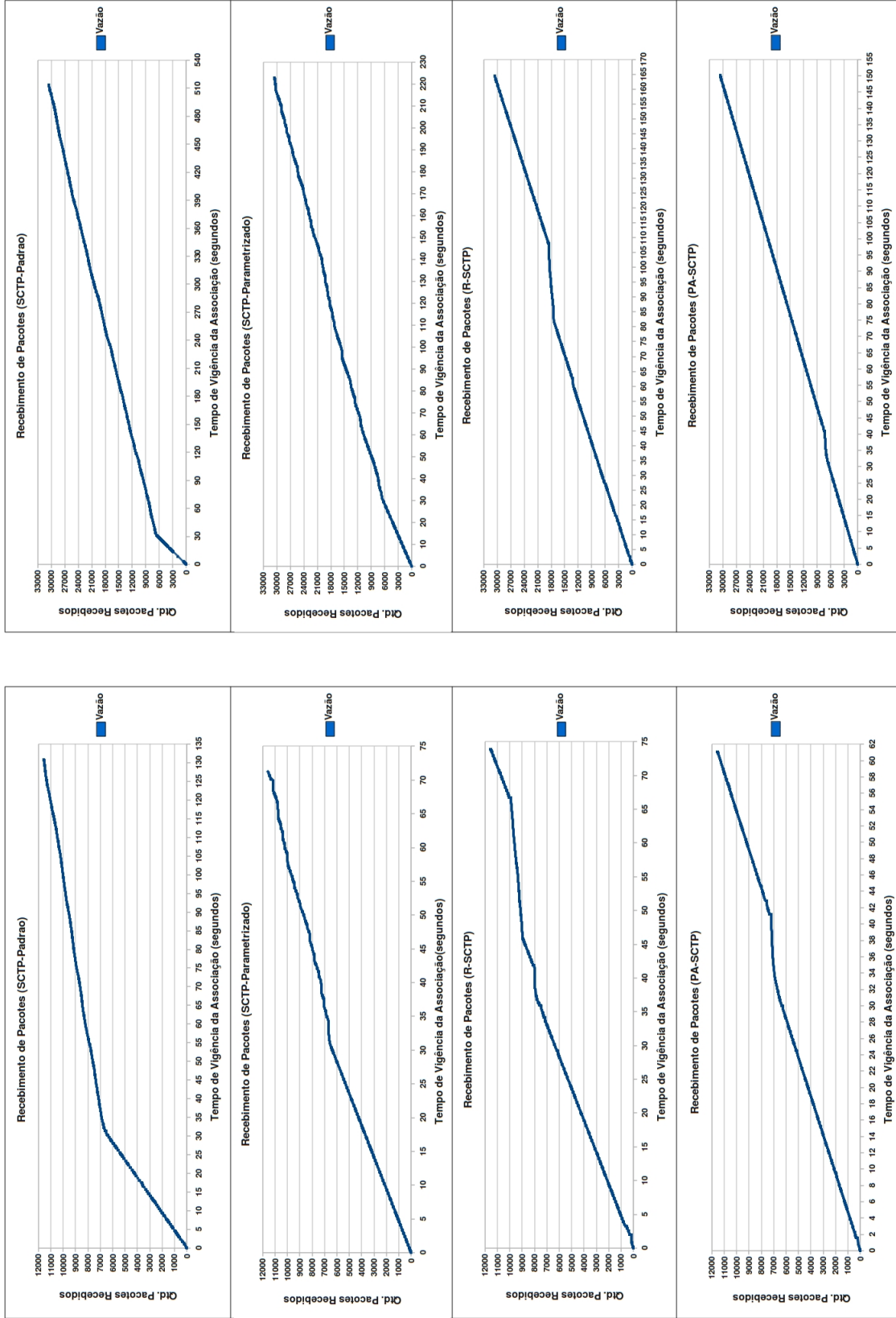


Figura 6.15: Vazão na ótica do receptor referente a arquivos de 15MB e 41MB

6.8.4 Conclusão do experimento

O SCTP-Padroao mostrou a sua ineficácia. Esse resultado já era esperado, uma vez que, em vários artigos referenciados dentro deste documento, já hávia sinalizações claras do prejuízo para as aplicações devido à onerosidade na tomada de decisão de troca do caminho primário.

O ajuste nos parâmetros configuráveis do SCTP, resultando no aqui denominado SCTP-Parametrizado, se mostrou mais uma vez ineficaz, uma vez que, mesmo tendo à disposição um caminho secundário em condições de tráfego de pacotes, o SCTP-Parametrizado de tempos em tempos fazia uso do caminho congestionado. Isso mais uma vez comprova que ajustes nos parâmetros configuráveis do protocolo SCTP não é garantia de conforto às necessidades da aplicação.

Com o SCTP-Parametrizado, mais uma vez, como já ocorrido em experimentos anteriores, houve o uso dos dois caminhos pertencentes à associação. Isso ocorreu com mais veemência a partir do momento do surgimento do tráfego concorrente em que o SCTP-Parametrizado passou praticamente a usar os dois caminhos da associação paralelamente, ora devido à ocorrência de *timeouts* que obrigava a retransmissão do pacote por um caminho secundário, ora devido à troca do caminho primário.

O R-SCTP se comportou nesse experimento similarmente ao SCTP-Parametrizado, ou seja, houve muitas trocas do caminho primário e períodos de permanência longos em que o R-SCTP passou a usar o caminho congestionado como primário. Nesse experimento, fica evidente que o uso do RTT como métrica a ser utilizado para selecionar o melhor caminho nem sempre atende de forma adequada à necessidade da aplicação.

Tendo em vista todos os pontos aqui levantados, mais uma vez se mostrou a superioridade do PA-SCTP na tomada de decisão do momento da troca do caminho primário. Apenas uma vez foi feita a troca do caminho primário, sendo logo após o surgimento do tráfego concorrente.

6.9 Experimento 4 - condição 4

Nesse experimento, é demonstrado o comportamento do SCTP-Padrão, SCTP-Parametrizado, R-SCTP e o PA-SCTP em uma situação em que o **Cliente1** está associada, apenas ao AP1 e, a partir do momento que se movimenta em direção ao AP2, também se associa a este e passa a ser uma estação *multi-homed*, entre três e cinco segundos após o estabelecimento da associação. Nesse experimento, o meio de transmissão se encontra na **Condição 4** (tabela 6.5).

Os gráficos das figuras 6.16, 6.17, 6.18 e 6.19 apresentam, respectivamente, o atraso, o atraso médio, a CWND e a vazão gerados na vigência da associação, primeiramente, na operação com um arquivo de 15MB, e em um segundo momento, no de 41MB, usando o

SCTP-Padrao, SCTP-Parametrizado, R-SCTP e PA-SCTP.

A cada arquivo transferido foram gerados gráficos referentes a atraso, atraso médio, CWND e vazão. Os gráficos referentes a atraso, atraso médio e vazão foram gerados de informações obtidas da recepção de pacote de dados. Já o gráfico CWND foi resultante do transmissor de pacotes de dados.

Para cada gráfico gerado, foi realizado o mesmo experimento cinco vezes. O fluxo de pacotes de dados SCTP sempre fluiu do servidor para o cliente, conforme os papéis sugeridos pelo *testbed* documentado na figura 6.6.

Durante o experimento, foi considerado que o **Cliente1** estava parado sobre a área de influência dos AP1, sendo que recebeu o IP 192.168.100.3 e, a partir do momento que se movimentou em direção à área de influência do AP2, recebeu na outra interface *wireless* o IP 192.168.200.3. O **Cliente1** continuou se movimentando em direção ao AP2, chegou ao ponto de perder a associação com o AP1 e ficou somente sobre influência do AP2 (isso ocorreu entre os tempos de 70 e 80 segundos).

Em todos os experimentos, na visão inicial do servidor, o destino primário inicial foi 192.168.100.3, e o seu destino secundário foi 192.168.200.3 quando esteve também sobre a área de influência do AP2.

A perda total de conectividade do caminho primário, ou seja, a ligação do **Cliente1** com o AP1, foi entre 75 e 90 segundos em todos os experimentos.

6.9.1 Com relação ao atraso e ao atraso médio (gráficos das figuras 6.16 e 6.17)

Quando analisamos o tempo total de transmissão, o R-SCTP e o PA-SCTP foram os melhores, pois concluíram o processo de transmissão do arquivo em menos tempo. Esse ganho foi obtido no caso do R-SCTP fazendo vários chaveamentos do caminho primário, enquanto que o PA-SCTP só fez uma troca do caminho primário de forma oportuna, conseguindo um bom desempenho.

Ao serem analisados o atraso e o atraso médio, é perceptível que houve picos consideráveis. O SCTP-Padrao fez a troca do caminho primário apenas uma vez, e seu maior pico de atraso e atraso médio foi decorrente da sua onerosidade em determinar a execução do *handover*. O atraso foi relativamente pequeno se comparado ao experimento 2. Isso é devido ao fato de que, quando um das interfaces *wireless* perdeu a associação com o AP, a associação SCTP tornou-se *single-homed*, o protocolo SCTP-Padrao determinou a execução imediata do procedimento de *handover*.

Com o SCTP-Padrao houve o crescimento gradativo do atraso e atraso médio a partir da movimentação em direção ao AP2. Quando houve a perda de conectividade com o AP1, surge os grandes picos de atraso, e após o procedimento de *handover* é que há a normatização

do atraso e atraso médio. Mais uma vez fica aqui comprovado, por esse experimento, o custo causado à aplicação, por essa tomada de decisão de *handover* baseada na perda de conectividade.

Com o SCTP-Parametrizado, o pico de atraso e de atraso médio apresentado foi devido ao fato de que, mesmo um caminho estando em condições desfavoráveis (para transmissão de pacotes de dados), o protocolo insistia em utilizá-lo gerando, dessa forma, os picos de atraso, fora o fato de constante redução da janela de congestionamento devido às frequentes interpretações de perda de pacote induzida muitas vezes pelos valores baixos do RTO.Min e do RTO.Max.

O R-SCTP e o PA-SCTP não apresentaram grandes picos de atraso ou atraso médio, pois fizeram um uso mais consciente dos caminhos existentes dentro da associação. A ressalva é que com o R-SCTP houve quatro trocas do caminho primário, enquanto que o PA-SCTP fez apenas uma.

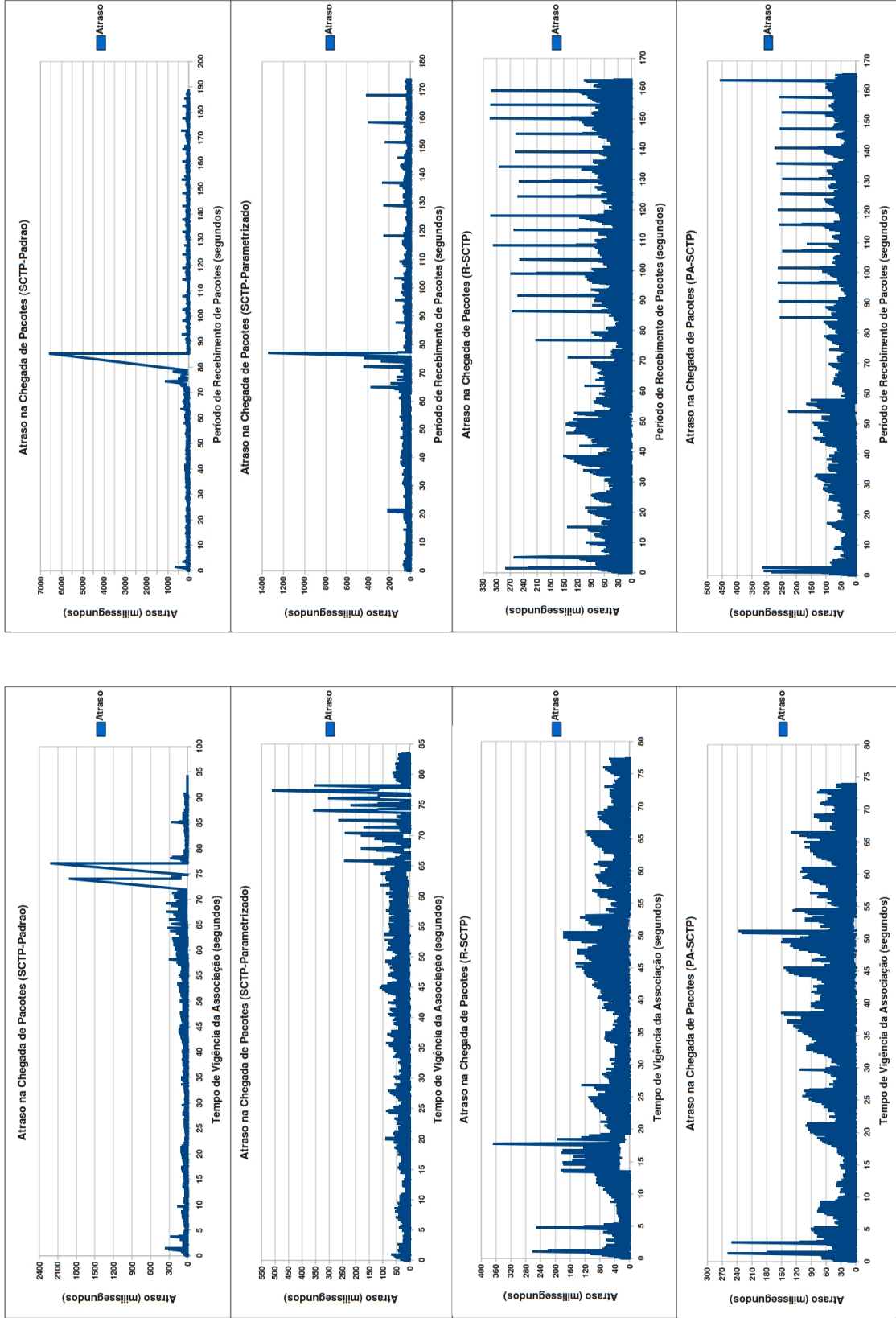


Figura 6.16: Atrasos durante o recebimento de arquivos de 15 e 41MB

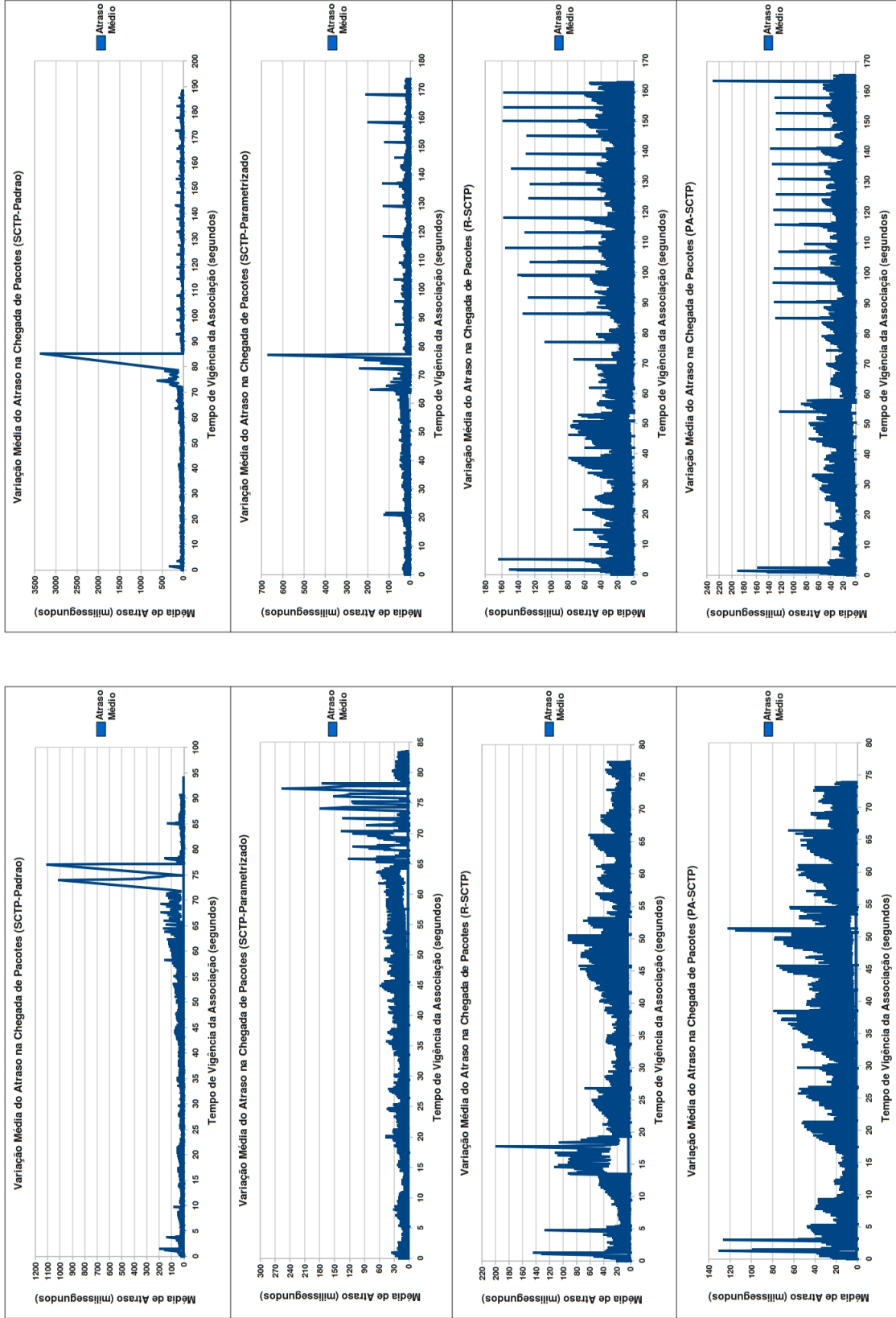


Figura 6.17: Atrasos médios durante o recebimento de arquivos de 15 e 41MB

6.9.2 Com relação à variação da janela de transmissão CWND (gráficos da figura 6.18)

Ao analisarmos a variação da CWND, é perceptível que o SCTP-Padrao e o PA-SCTP mantiveram uma variação previsível da CWND. Pelos gráficos da figura 6.18, é possível perceber que houve apenas uma alteração do caminho primário, no caso do uso do PA-SCTP. Quando ocorrem alteração do caminho primário ou perda de pacotes, a CWND teve seu tamanho reduzido, respectivamente, para 4.380B e 1MTU.

Em relação aos gráficos (figura 6.18) referentes ao SCTP-Padrao, é notório que a CWND diminuiu de acordo com o afastamento do cliente do AP1 (caminho primário). Afastou tanto que a interface de rede *wireless* perdeu a associação com o AP1 e logo, em seguida, ocorreu a troca do caminho primário, ou seja, passou a utilizar a associação com o AP2 como caminho primário. Um problema a ser notado, com essa onerosidade em fazer o *handover*, é a queda no recebimento de pacotes de dados, devido ao uso de um controle de fluxo mais conservador com relação ao uso dos recursos da rede, no caso o *slow start*.

Quando são observados os gráficos (figura 6.18) referentes ao SCTP-Parametrizado, é perceptível que a CWND diminuiu em conformidade com o afastamento do **Cliente1** do AP1. A frequente troca do caminho primário e a interpretação de perda de pacotes por parte do protocolo durante o período de afastamento do AP1 e proximidade do AP2 contribuíram para que o protocolo trabalhasse durante muito tempo no modo *slow start*. Durante esse experimento, também foi visível o uso praticamente paralelo dos dois caminhos da associação para transmissão de pacotes de dados, ora devido a troca do caminho primário e, em outros momentos, por retransmissão de pacotes perdidos. Como já comentado anteriormente, isso traz uma competição desleal para outras aplicações que não fazem uso desse protocolo.

Através dos gráficos (figura 6.18), é visível também a instabilidade do SCTP-Parametrizado em relação ao SCTP-Padrao e ao PA-SCTP, por não conseguir manter uma regularidade de fluxo de pacotes de dados.

Com relação ao R-SCTP, é perceptível de duas a três trocas do caminho primário do início da associação até o momento, em que o **Cliente1** perdeu a conexão com o AP1. Mesmo com essas trocas desnecessárias, obteve um bom desempenho se comparado ao SCTP-Padrao e ao SCTP-Parametrizado.

Já o PA-SCTP fez apenas uma troca do caminho primário. Isso é perceptível no momento entre 50 e 60 segundos. A partir desse período, o **Servidor1** começou a utilizar o enlace que envolveu o AP2 como destino primário para a transmissão dos seus pacotes de dados.

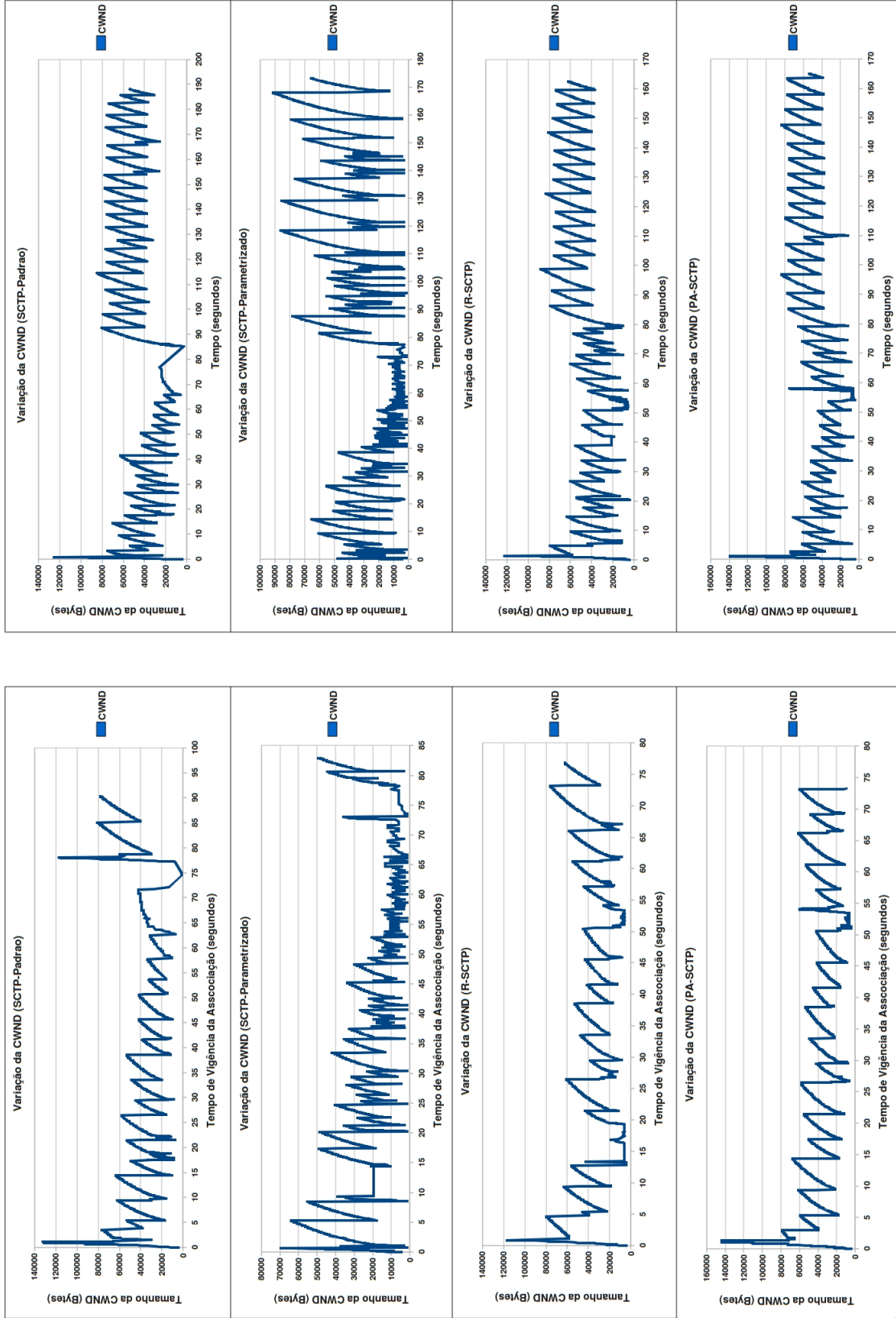


Figura 6.18: Variação da CWND no transmissor referente a arquivos de 15MB e 41MB

6.9.3 Com relação à vazão (gráficos da figura 6.19)

No gráfico referente ao arquivo de 15MB, o SCTP-Padrao mostrou claramente que, no início da associação, a qual usava o AP1 como um enlace do caminho primário, houve um crescimento inicial no fluxo de recebimento de pacotes de dados e, em seguida, começou a ocorrer uma degradação suave na taxa de recebimento de pacotes acompanhando o afastamento do **Cliente1** do AP1. Entre 70s e 90s, praticamente houve uma estagnação e, após 90s, começou um crescimento vertiginoso na taxa de recebimento de pacotes, ou seja, nesse ponto, houve a troca do caminho primário. O mesmo entendimento pode ser percebido em relação à transferência do arquivo de 41MB.

Quanto ao SCTP-Parametrizado, é perceptível que obteve um desempenho bem parecido com o SCTP-Padrao devido à sua constante interpretação de perdas de pacotes e consequente retransmissão, como também a troca do caminho primário em períodos desnecessários. Nos gráficos (figura 6.19), também ficou claro que nem sempre ajustes nos parâmetros configuráveis do protocolo podem trazer melhorias no seu desempenho, fora ao fato de poder trazer prejuízos as aplicações subjacentes.

Com relação ao R-SCTP (15MB), é perceptível, entre o tempo 10s e 20s, houve redução grande no recebimento de pacotes devido a um período intenso de troca do caminho primário.

Ao analisarmos os gráficos do PA-SCTP, é nítido que houve uma atenuação no recebimento de pacotes de dados, devido ao afastamento do AP1 (caminho primário) e, após à realização do procedimento de *handover*, obteve-se um crescimento íngreme no recebimento de pacotes de dados.

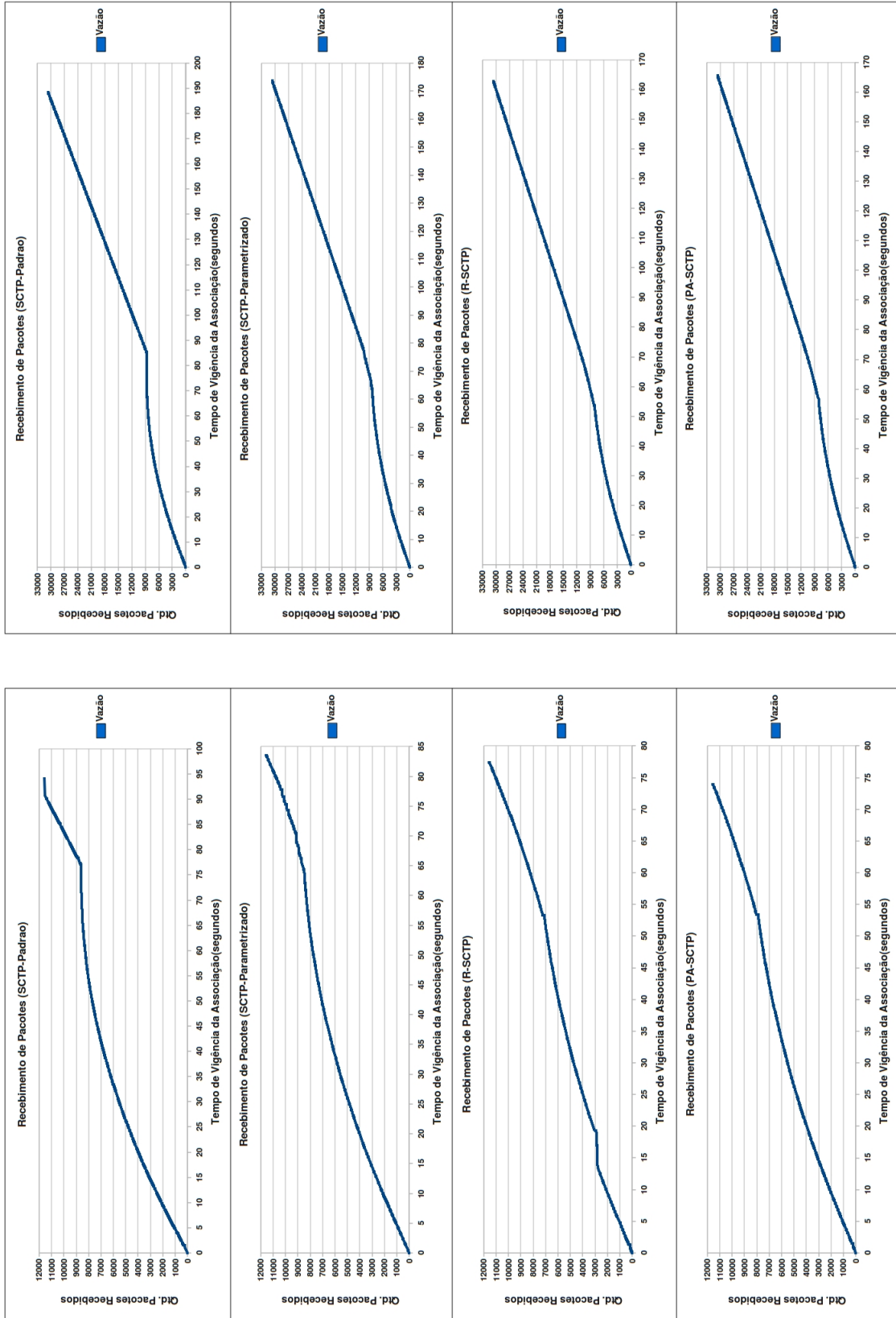


Figura 6.19: Vazão na ótica do receptor referente a arquivos de 15MB e 41MB

6.9.4 Conclusão do experimento

Mais uma vez se mostrou a constância do PA-SCTP em relação ao SCTP e suas propostas de melhoria. Nesse experimento, o R-SCTP mostrou um desempenho similar ao PA-SCTP. Mas, ao analisarmos todos os experimentos, o melhor desempenho, sem dúvida, é alcançado com o PA-SCTP.

A desvinculação do próprio protocolo SCTP e das outras propostas, em relação às necessidades da aplicação, mostrou que o seu procedimento para tomada de decisão de *handover*, em muitas das ocasiões, não são benéficos para a aplicação.

O SCTP-Padrao se mostrou mais uma vez oneroso para aplicações. O SCTP-Parametrizado, além de oneroso, apresentou uma disputa por recurso desleal para com outros tipos de protocolos. E no caso do R-SCTP, existe a falta de uniformidade dos seus resultados em todos os cenários, além do fato de que, nesse experimento, houve praticamente momentos de paradas no recebimento de dados.

Devido aos fatores expostos, mais uma vez é destacada a relevância da pesquisa que culminou a criação do PA-SCTP, que incorpora um mecanismo mais apropriado para gerir o recurso de *multi-homing* do SCTP.

6.10 Experimento 5 - condição 5

Nesse experimento, foi demonstrado o comportamento do SCTP-Padrão, do SCTP-Parametrizado, do R-SCTP e do PA-SCTP quando submetidos a uma situação em que o **Cliente1** estava enlaçado apenas ao AP1, e abriu uma associação SCTP com o **Servidor1**. Após a associação SCTP estabelecida o **Cliente1** se movimenta em direção ao AP2, e também se enlaça a este e passa a ser uma estação *multi-homed*, entre três e cinco segundos após o estabelecimento da associação SCTP. Nesse experimento, o meio de transmissão se encontra na **Condição 5** (tabela 6.5).

O diferencial desse experimento para o anterior é que, após 30 segundos do início da associação, um fluxo de dados concorrentes foi inserido no caminho primário nos mesmos moldes que ocorreu no experimento 3.

Os gráficos das figuras 6.20, 6.21, 6.22 e 6.23 apresentam, respectivamente, o atraso, o atraso médio, a CWND e a vazão gerados na vigência da associação primeiramente na operação com um arquivo de 15MB e, em um segundo momento, no de 41MB, usando o SCTP-Padrao, SCTP-Parametrizado, R-SCTP e PA-SCTP.

A cada arquivo transferido, foram gerados gráficos referentes a atraso, atraso médio, CWND e vazão. Os gráficos referentes a atraso, atraso médio e vazão são oriundos de informações obtidas a cada recepção de pacote de dados. Já os dados usados para gerar o gráfico CWND foram obtidos no lado transmissor a cada pacote de dado transmitido.

Para cada gráfico gerado, foi realizado o mesmo experimento cinco vezes. O fluxo de pacotes de dados SCTP sempre fluiu do servidor para o cliente, conforme os papéis sugeridos pelo *testbed* documentado na figura 6.6.

Durante o experimento, foi considerado que o **Cliente1** estava parado sobre a área de influência dos AP1, sendo que recebeu o IP 192.168.100.3 e, a partir do momento que se movimentou, entrou na área de influência do AP2 e recebeu o IP 192.168.200.3. O **Cliente1** continuou se movimentado em direção ao AP2, chegou ao ponto de perder a associação com o AP1 e ficou somente sobre influência do AP2 (isso ocorre entre os tempos de 70 e 80 segundos).

Em todos os experimentos, na visão inicial do servidor, o destino primário inicial foi 192.168.100.3, e o seu destino secundário foi 192.168.200.3 quando esteve também sobre a área de influência do AP2. O destino primário inicial só foi diferente quando mencionado no texto.

6.10.1 Com relação ao atraso e ao atraso médio (gráficos das figuras 6.20 e 6.21)

Quando analisamos o tempo total de transmissão, o SCTP-Parametrizado e o PA-SCTP conseguiram concluir o processo de transmissão do arquivo em menor tempo. Durante todo o tempo de vigência da associação, o SCTP-Parametrizado trocou o caminho primário por diversas vezes, enquanto que mais uma vez o PA-SCTP esperou o momento mais oportuno para fazer o *handover*.

Ao analisarmos o atraso e o atraso médio no SCTP-Padrao, notamos claramente o aumento dessas medidas causado pelo afastamento do AP1 (caminho primário inicial) e pela geração de fluxo concorrente. Se comparamos esses mesmos gráficos com o do experimento anterior, percebemos uma piora no desempenho do protocolo, uma vez que ele é relutante em trocar o caminho primário, ou seja, só o troca quando não houver mais conectividade.

Em se tratando do SCTP-Parametrizado, o fluxo concorrente não causou atrasos significativos. O tráfego concorrente presente no AP1 fez com que o protocolo permanecesse por mais tempo usando o outro caminho (que envolve o AP2), fazendo com que tivesse um desempenho um pouco melhor. Mesmo assim ainda é perceptível a troca exagerada do caminho primário, a grande sensação de perda de dados por parte do protocolo e o múltiplo uso dos dois caminhos. Um adendo sobre o uso dos múltiplos caminhos é que a retransmissão de pacotes perdidos sempre é feita pelo outro caminho, e isso agrava ainda mais a sensação de uso dos múltiplos caminhos da associação. Se fosse utilizado apenas um caminho, seria possível realizar o mesmo trabalho não sendo danoso para outras aplicações.

Com a presença do fluxo concorrente, o R-SCTP se comportou de forma pior se comparado ao experimento 4. A presença do fluxo concorrente e a deterioração natural do caminho primário inicial não foram suficientes para fazer com que o R-SCTP deixasse de usar

o caminho deteriorado em função da presença de um caminho secundário, que, a cada momento que passa, só melhora a sua aptidão no transporte de pacotes de dados. Isso é perceptível nos dois casos, na transmissão do arquivo de 15MB e 45MB.

O PA-SCTP mais uma vez soube oportunamente trabalhar com um ambiente *multi-homing*. Como no uso do SCTP-Padrao, o PA-SCTP fez apenas uma troca do caminho primário, não se comportou em nenhum momento de forma lesiva a outras aplicações. Uma observação apenas em relação ao pico de atraso encontrado na transferência do arquivo de 15MB provavelmente provocado por fatores referentes à movimentação e ao controle de congestionamento.

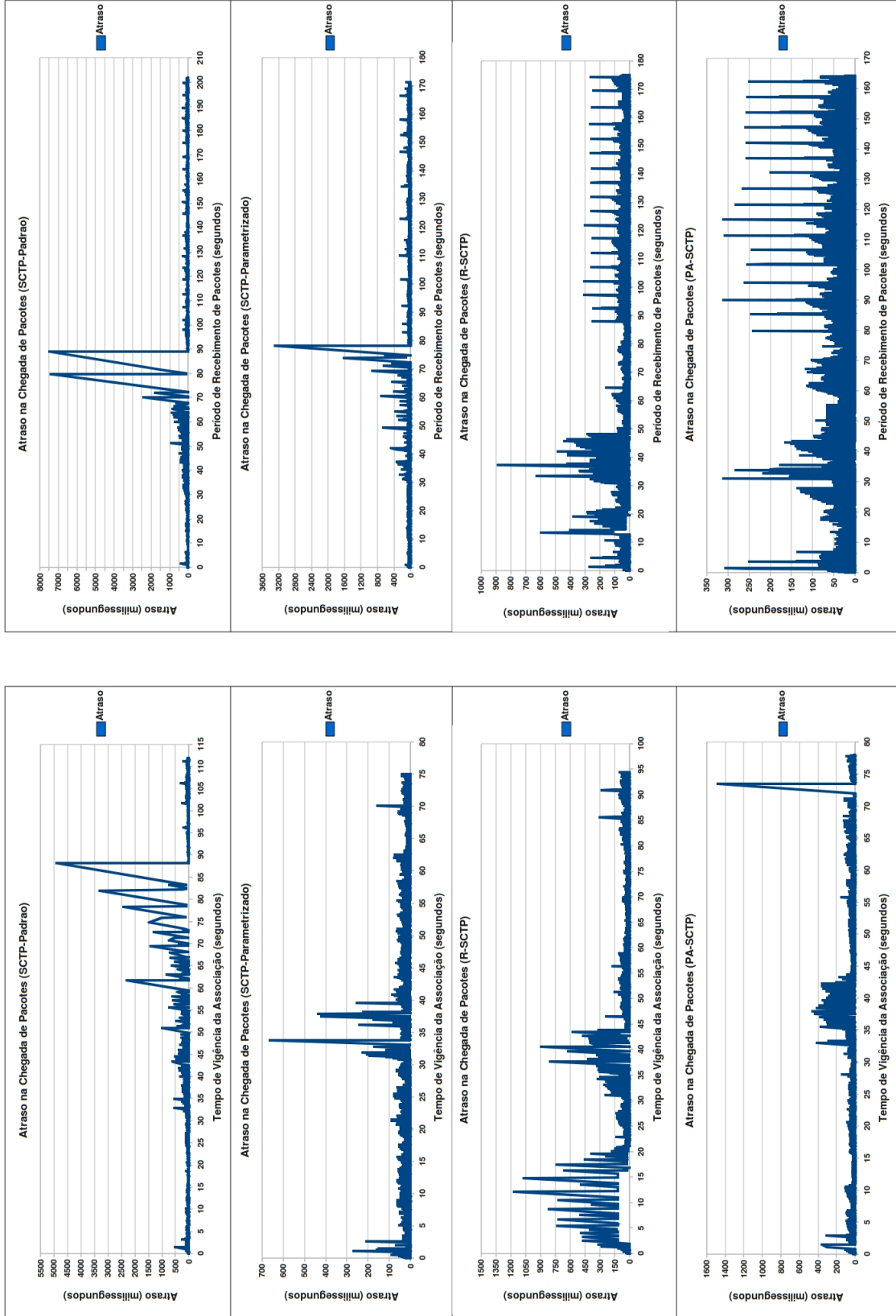


Figura 6.20: Atrasos durante o recebimento de arquivos de 15 e 41MB

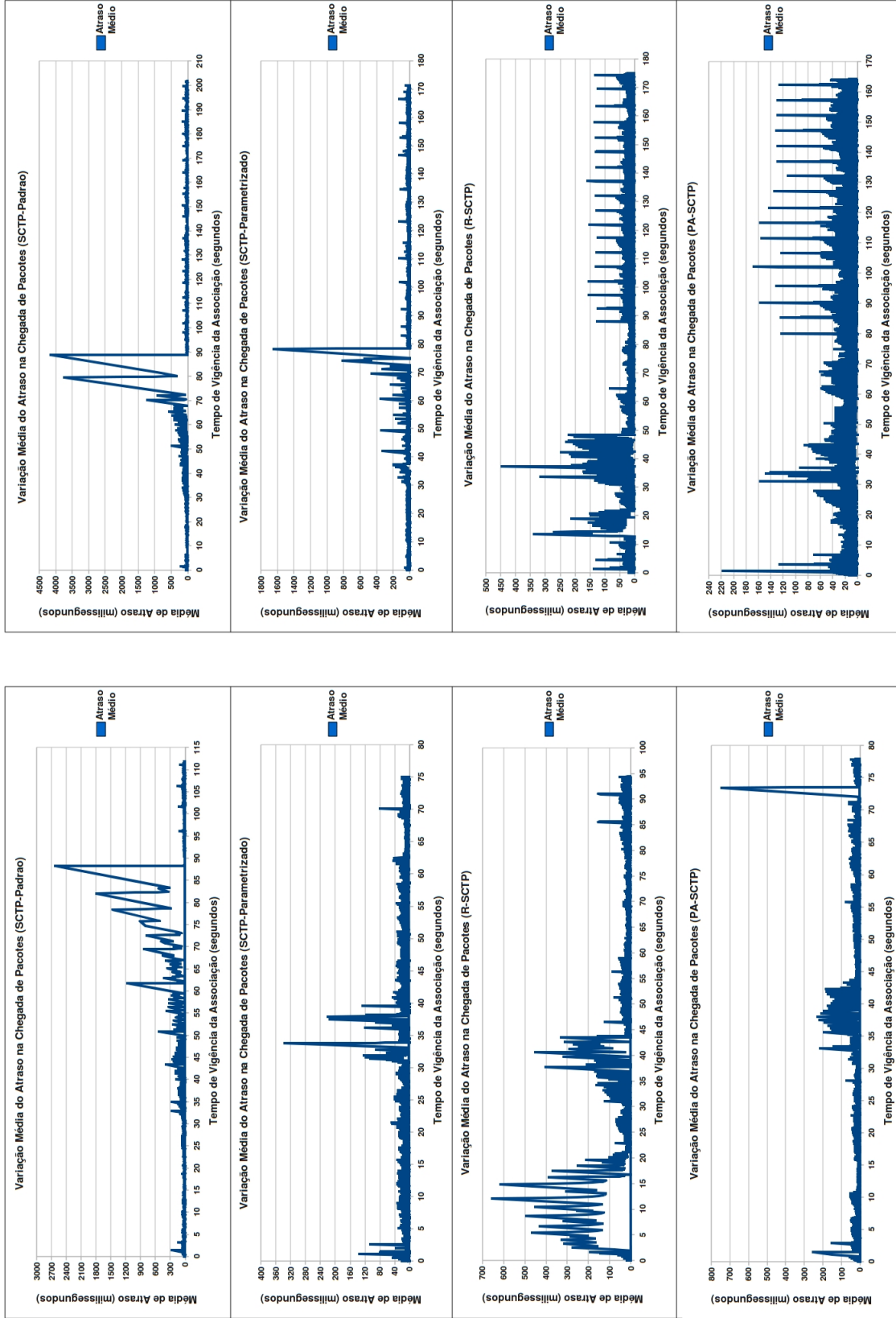


Figura 6.21: Atrasos médios durante o recebimento de arquivos de 15 e 41MB

6.10.2 Com relação à variação da janela de transmissão CWND (gráficos da figura 6.22)

Ao analisarmos a variação da CWND, é perceptível que o SCTP-Padiao e o PA-SCTP mantiveram uma variação previsível da CWND. Nesses gráficos (figura 6.22), é perfeitamente perceptível que houve apenas uma alteração do caminho primário, quando utilizado o PA-SCTP. Quando há alteração do caminho primário ou perda de pacotes, a CWND teve seu tamanho reduzido, respectivamente, para 4380B e 1MTU.

Em relação aos gráficos referentes ao SCTP-Padiao, é notório que a CWND diminuía à medida que o cliente se afastava do AP1 (caminho primário). O **Cliente1** se afastou tanto que a interface de rede *wireless* perdeu a associação com o AP1 e logo, em seguida, ocorreu a troca do caminho primário. O fluxo concorrente presente após 30 segundos de associação foi um agravante para o SCTP-Padiao, que, mesmo assim, só fez a troca do caminho primário quando este não apresentou mais conectividade. Também após a troca do caminho primário, o SCTP-Padiao comportou-se da forma esperada.

Nos gráficos referentes ao SCTP-Parametrizado, é perceptível que a CWND diminuía em conformidade com o afastamento do cliente1 do AP1. A frequente troca do caminho primário e a sensação de perda de pacotes por parte do protocolo durante o período de afastamento do AP1 e proximidade do AP2 contribuíram para que o protocolo trabalhasse durante muito tempo no modo *slow start*. Durante esse experimento, também foi visível o uso praticamente paralelo dos dois caminhos de comunicação da associação, podendo prejudicar, dessa forma, alguma aplicação TCP/UDP que pudesse estar usando enlaces que compunham a associação do SCTP-Parametrizado.

Mesmo após a associação ter ficado *single-homed*, o protocolo ainda teve várias sinalizações de perda de pacote, o que contribuiu para a constante redução da CWND e, nesse caso, uma maior lentidão na chegada de pacotes.

Com relação ao R-SCTP, é perceptível que, logo no início da associação, o protocolo, mesmo estando em um caminho melhor naquele momento, optou ligeiramente por trocar o caminho primário, permanecendo períodos relativamente longos em modo *slow start*, o que prejudicou significativamente o desempenho desse protocolo. Apesar de ter tido um pior desempenho que o SCTP-Parametrizado, é visível que o fluxo de pacotes de dados é constante enquanto que no SCTP-Parametrizado oscilou muito.

Já o PA-SCTP fez apenas uma troca do caminho primário. Isso é perceptível no momento entre 35 e 45 segundos. A perda de conectividade do atual caminho primário se deu entre 70 e 80 segundos, isso mostra que o PA-SCTP foi relativamente eficaz mais uma vez em alterar o caminho primário antes que ele ficasse crítico para atender às necessidades da aplicação.

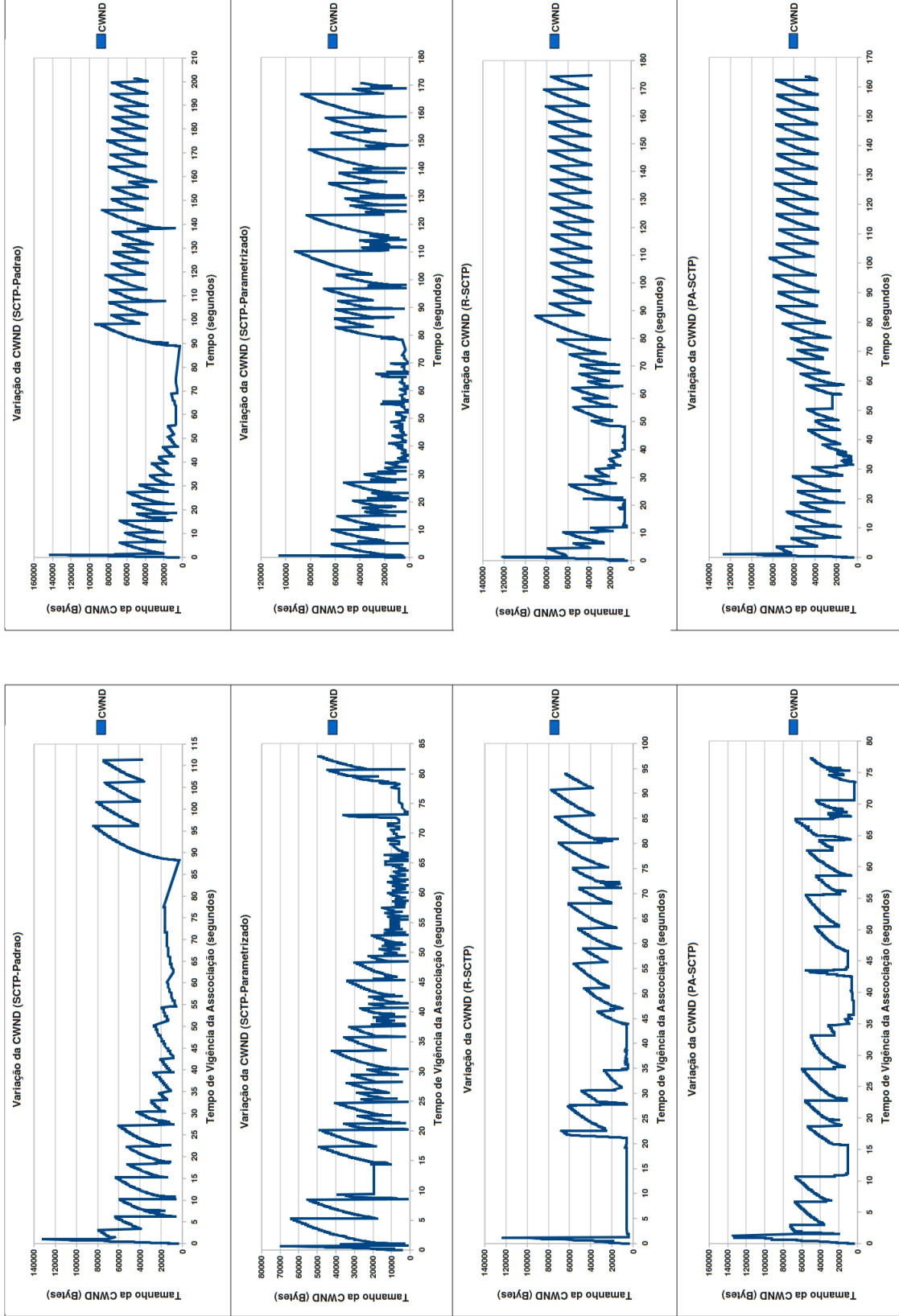


Figura 6.22: Variação da CWND no transmissor referente a arquivos de 15MB e 41MB

6.10.3 Com relação à vazão (gráficos da figura 6.23)

Como é fácil de perceber, o PA-SCTP foi o que menos causou impacto na taxa de recebimento de pacotes de dados no decorrer da associação.

O SCTP-Padrao, em alguns momentos, praticamente parou de receber pacotes de dados, ou passou a receber um fluxo muito menor que o ideal.

O SCTP-Parametrizado se saiu um pouco melhor do que os outros, mas a sua frequente interpretação precipitada de perda de pacotes forçou a retransmissão de pacotes de dados pelo caminho secundário. Junte isso ao fato da sua constante oscilação na janela de transmissão, uma vez que, a cada pacote perdido, a CWND foi reduzida para 1MTU.

Já o R-SCTP se portou de forma insensata ao insistir, em usar o caminho congestionado, tanto é, que isso é visível no gráfico (figura 6.23) referente à vazão. Após a associação ter se tornado *single-homed*, o R-SCTP se comportou normalmente, mas a perda ocasionada anteriormente pelo seu comportamento inapropriado, quando estava sob influência de uma associação *multi-homed*, deixou esse protocolo a desejar no fluxo de pacotes recebidos.

Quando é observado os gráficos (figura 6.23) referentes ao PA-SCTP, não ocorreu a interrupção, mesmo que momentânea de recebimento de pacotes de dado. Existiu períodos de declínio, mas tudo o que faz é em conformidade com a aplicação, obtendo, dessa forma, um melhor custo benefício.

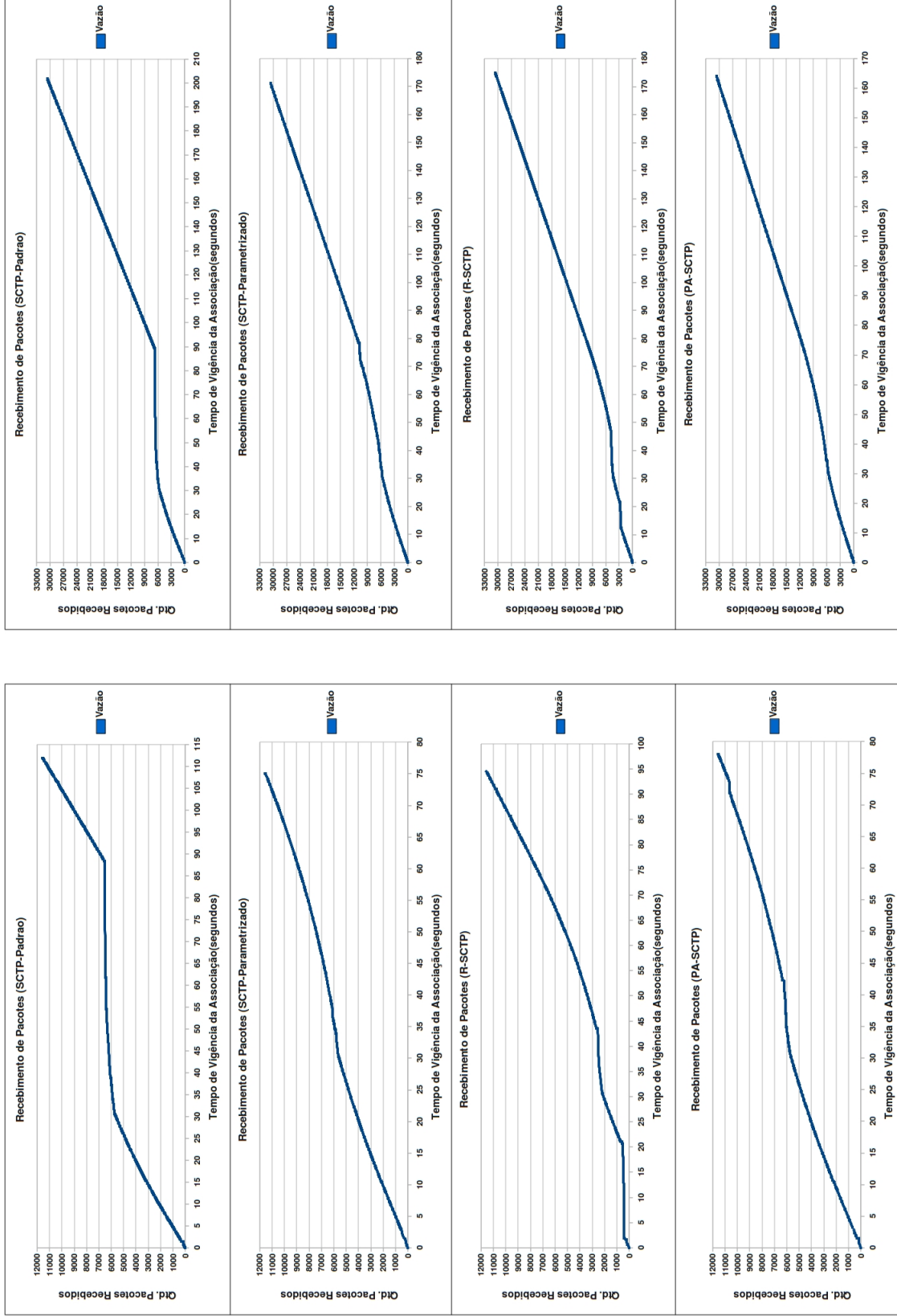


Figura 6.23: Vazão na ótica do receptor referente a arquivos de 15MB e 41MB

6.10.4 Conclusão do experimento

Nesse experimento, o PA-SCTP se mostrou mais uma vez de forma eficaz, conseguiu um desempenho melhor devido ao seu mecanismo ser mais eficaz para tomar decisão do momento de *handover*.

No experimento, existiram dois fatores para a deterioração da qualidade do meio, o afastamento do AP1 e o fluxo concorrente. Devido à onerosidade apresentada nos gráficos em tomar a decisão *handover*, é previsível a conclusão mais uma vez da ineficácia do SCTP-Padrao em gerir o *multi-homing*. Resumindo o SCTP-Padrao visa à conectividade que nem sempre anda lado a lado com a vazão ofertada, ou seja, enquanto houver conectividade para o SCTP-Padrao, o caminho primário é válido, não importando a taxa de fluxo de dados sobre o meio.

O SCTP-Parametrizado continuou os mesmos problemas já relacionados em experimentos anteriores, troca constante do caminho primário e interpretações errôneas de perda de dados. Ajustes nos parâmetros configuráveis do SCTP-Padrao mais uma vez comprovam que são estagnados a um determinado cenário, ou seja, se mudou o cenário, é preciso fazer uma nova pesquisa sobre os melhores valores para os parâmetros configuráveis do SCTP.

O R-SCTP utiliza métricas para tomada de decisão de *handover* que também não prioriza a necessidade da aplicação e podem, dependendo do cenário, conduzir a constantes trocas do caminho primário.

Mais uma vez o PA-SCTP mostrou a sua estabilidade e produziu um cenário de transição que menos impactasse na aplicação com relação ao atraso médio no recebimento de pacotes de dados.

Capítulo 7

Conclusão

Este trabalho, em um primeiro momento, caracterizou a forma de gerenciamento do *multi-homing* no SCTP como prejudicial, do ponto de vista de algumas aplicações, devido ao *handover* basear-se na perda de conectividade do caminho primário que leva a paradas significativas na transmissão de pacotes de dados. Em um segundo momento foi proposto um mecanismo para determinar quando trocar o caminho primário e que seja menos prejudicial a aplicações que façam uso do protocolo SCTP para o transpote de pacotes de dados.

O procedimento de alterar os parâmetros configuráveis do SCTP, previsto pela própria RFC 4960, para forçar a execução do *handover* mais rapidamente, realmente trouxe um ganho de desempenho significativo mas com resultados instáveis já que o ganho depende de fatores, como por exemplo, largura de banda equivalente dos caminhos de transmissão de pacotes de dados, englobados dentro da associação.

Uma importante conclusão obtida a partir dos experimentos é que, quando há uma redução significativa dos valores dos parâmetros configuráveis do protocolo SCTP, existe, em alguns casos, o uso quase paralelo do caminho primário e secundário para a transmissão de pacotes de dados. O caminho secundário, foi usado na retransmissão de pacotes de dados, devido à interpretação equivocada de perda do pacote, quando enviado pelo caminho primário. A constante troca do caminho primário também leva à impressão de uso paralelo dos dois caminhos disponibilizados na associação. O grande problema do uso paralelo dos dois caminhos é a concorrência desleal, para com outros protocolos de transporte como o UDP e TCP, pela competição dos recursos compartilhados da rede. Se o UDP ou TCP estiverem usando um enlace que está participando de uma associação fim-a-fim SCTP, na presença de perda de pacotes de dados, retransmissões e troca frequentes do caminho primário, seriam interpretados pelos protocolos UDP e TCP como um caminho parcialmente congestionado, ou mesmo congestionado.

Por fim, os resultados mostraram que o PA-SCTP gerencia o *multi-homing* em prol da demanda da aplicação. O ganho de desempenho foi significativo, sem haver comprometimento da harmonia na disputa pelos recursos compartilhado da rede junto a outros protocolos.

A demanda da dissertação foi atendida, ou seja, fazer uso do recurso de *multi-homing* SCTP em prol das necessidades da aplicação, situação que era almejada em várias publicações inclusive com vários tipos de propostas.

Cabe ainda ressaltar que tanto o mSCTP como o PR-SCTP, apesar de serem extensões do SCTP, herdam seu algoritmo para determinação de quando realizar o *handover*, ou seja, baseada na perda de conectividade. O PA-SCTP pode ser usado juntamente com essas extensões. No caso do mSCTP, diminuirá de forma significativa a latência do gerenciamento de mobilidade e com o PR-SCTP proporcionará um melhor atendimento a aplicações de tempo-real.

Por fim, em trabalhos futuros, podem ser estudados os protocolos DCCP e SCTP no gerenciamento de mobilidade em ambiente heterogêneos, para alcançar plenamente uma situação de *soft-handover* na transição inter-redes.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] KIM, D. et al. Adaptive Primary Path Switching for SCTP Handover. In: *Proc. of ICACT*. [S.l.: s.n.], 2008. v. 2, p. 900–902.
- [2] COSTA, D. G. *SCTP - Uma Alternativa aos Tradicionais Protocolos de Transporte da Internet*. [S.l.: s.n.], 2005.
- [3] STEWART, R. *RFC 4960: Stream Control Transmission Protocol*. [S.l.]: RFC4960, 2007.
- [4] STALVIG, P. *Introdução ao Protocolo de Controle de Fluxo de Transmissão (SCTP), A próxima geração do Protocolo de Controle de Transmissão (TCP)*. Site Web - <http://www.f5networks.com.br/pdf/white-papers/introducao-ao-stream-control-transmission-protocol-sctp-wp.pdf> Acessado em 10/04/2009.
- [5] FALLON, S. et al. SCTP Switchover Performance Issues in WLAN Environments. In: *5 th IEEE Consumer Communications & Networking Conference (CCNC) January*. [S.l.: s.n.], 2008.
- [6] FUNASAKA, J. et al. A study on primary path switching strategy of SCTP. In: *Autonomous Decentralized Systems, 2005. ISADS 2005. Proceedings*. [S.l.: s.n.], 2005. p. 536–541.
- [7] GRACE, K.; PECELLI, D.; D'AMELIA, J. Improving Multi-homed SCTP Mobile Communication Performance. *Technical Papers, The MITRE Corporation*, 2006.
- [8] STEWART, R. et al. Sockets API extensions for stream control transmission protocol (SCTP). *draft-ietf-tsvwg-sctpsocket-19 (work in progress)*, 2009.
- [9] UFRS. *Protocolo SCTP*. Site Web - <http://www.inf.ufrgs.br/~cechin/Net/sctp/sctp.html> Acessado em 05/05/2009.
- [10] BUDZISZ, L. et al. An analytical estimation of the failover time in SCTP multihoming scenarios. In: *Wireless Communications and Networking Conference (WCNC)*. [S.l.: s.n.], 2007.

- [11] JR, A. C.; AMER, P.; STEWART, R. End-to-end failover thresholds for transport layer multihoming. In: *IEEE Military Communications Conference, 2004. MILCOM 2004*. [S.l.: s.n.], 2004. v. 1.
- [12] KELLY, A. et al. Delay-centric handover in SCTP over WLAN. *Transactions on Automatic Control and Computer Science*, v. 49, n. 63, p. 1–6, 2004.
- [13] SIDDIQUI, F.; ZEADALLY, S. SCTP multihoming support for handoffs across heterogeneous networks. In: *Fourth Annual Conference on Communication Networks and Services Research*. [S.l.: s.n.], 2006. p. 243–250.
- [14] CARO, A.; AMER, P.; STEWART, R. Rethinking end-to-end failover with transport layer multihoming. In: PRESSES POLYTECHNIQUES ROMANDES. *Annales des télécommunications*. [S.l.], 2006. v. 61, n. 1/2, p. 92.
- [15] JR, A. C.; AMER, P.; STEWART, R. Retransmission schemes for end-to-end failover with transport layer multihoming. In: *IEEE Global Telecommunications Conference, 2004. GLOBECOM'04*. [S.l.: s.n.], 2004. v. 3.
- [16] JR, A. C. et al. Using SCTP multihoming for fault tolerance and load balancing. *ACM SIGCOMM Computer Communication Review*, Association for Computing Machinery, Inc, One Astor Plaza, 1515 Broadway, New York, NY, 10036-5701, USA., v. 32, n. 3, p. 23–23, 2002.
- [17] IYENGAR, J.; AMER, P.; STEWART, R. Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths. *IEEE/ACM Transactions on Networking (TON)*, IEEE Press Piscataway, NJ, USA, v. 14, n. 5, p. 951–964, 2006.
- [18] FITZPATRICK, J.; MURPHY, S.; MURPHY, J. An approach to transport layer handover of VoIP over WLAN. In: *3rd IEEE Consumer Communications and Networking Conference, 2006. CCNC 2006*. [S.l.: s.n.], 2006. v. 2.
- [19] CARO, A. et al. Retransmission policies with transport layer multihoming. In: *ICON 2003*. [S.l.: s.n.], 2003.
- [20] SEO, D.; KIM, H.; KIM, J. Analysis of Dynamic Congestion Control of SCTP. In: *Convergence and Hybrid Information Technology, 2008. ICHIT'08. International Conference on*. [S.l.: s.n.], 2008. p. 687–693.
- [21] SHI, J. et al. Experimental performance studies of SCTP in wireless access networks. *Proc. Of ICCT'03*, v. 1, p. 392–395, 2003.
- [22] JUNGMAIER, A.; RATHGEB, E.; TUXEN, M. On the use of SCTP in failover-scenarios. In: *International Conference on Information Systems, Analysis and Synthesis*. [S.l.: s.n.]. p. 363–368.

- [23] NOONAN, J. et al. Stall and path monitoring issues in SCTP. In: *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*. [S.l.: s.n.], 2006. p. 1–9.
- [24] WANG, X.; LEUNG, V. Applying PR-SCTP to transport SIP traffic. In: *IEEE Global Telecommunications Conference, 2005. GLOBECOM'05*. [S.l.: s.n.], 2005. v. 2.
- [25] BRENNAN, R.; CURRAN, T. SCTP congestion control: Initial simulation studies. In: CITeseer. *Proc. 17th Int'l Teletraffic Congress*. [S.l.], 2001.
- [26] FU, S.; ATIQUZZAMAN, M. SCTP: State of the art in research, products, and technical challenges. *IEEE Communications Magazine*, Citeseer, v. 42, n. 4, p. 64–76, 2004.
- [27] STEWART, R.; T
UXEN, M.; LEI, P. SCTP: What is it, and how to use it? 2008.
- [28] WANG, Q.; ABU-RGHEFF, M. Mobility management architectures based on joint mobile IP and SIP protocols. *IEEE Wireless Communications*, v. 13, n. 6, p. 68–76, 2006.
- [29] RANDHAWA, T. A New METHOD TO SUPPORT UMTS/WLAN VERTICAL HANDOVER USING SCTP. *IEEE Wireless Communications*, p. 45, 2004.
- [30] EZZOUHAIRI, A.; QUINTERO, A.; PIERRE, S. A New SCTP mobility scheme supporting vertical handover. In: *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2006.(WiMob'2006)*. [S.l.: s.n.], 2006. p. 205–211.
- [31] STEWART, R.; RAMALHO, M. et al. *SCTP Dynamic Address Reconfiguration*. [S.l.], 2002.
- [32] STEWART, R. et al. *RFC 5061: Stream control transmission protocol (SCTP) dynamic address reconfiguration*. [S.l.]: September, 2007.
- [33] TUEXEN, M. et al. *RFC 4895, Authenticated chunks for the Stream Control Transmission Protocol (SCTP)(August 2007)*.
- [34] FRACCHIA, R. et al. A WiSE extension of SCTP for wireless networks. In: *2005 IEEE International Conference on Communications, 2005. ICC 2005*. [S.l.: s.n.], 2005. v. 3.
- [35] KASHIHARA, S. et al. Path selection using active measurement in multi-homed wireless networks. In: *Applications and the Internet, 2004. Proceedings. 2004 International Symposium on*. [S.l.: s.n.], 2004. p. 273–276.
- [36] FRACCHIA, R. et al. WiSE: Best-path selection in wireless multihoming environments. *IEEE Transactions on Mobile Computing*, v. 6, n. 10, p. 1130–1141, 2007.
- [37] NOONAN, J. et al. Simulations of multimedia traffic over SCTP modified for delay-centric handover. In: *World Wireless Congress*. [S.l.: s.n.], 2004.

- [38] HYNES, M.; KILMARTIN, L. Optimising transmission path selection in SCTP based wireless networking. In: IEEE PRESS. *Proceedings of the 6th international conference on Symposium on Wireless Communication Systems*. [S.l.], 2009. p. 298–302.
- [39] KIM, D.; KOH, S.; KIM, Y. A scheme of primary path switching for mobile terminals using SCTP handover. *WSEAS Transactions on Communications*, WSEAS PRESS, v. 6, n. 3, p. 432, 2007.
- [40] LIU, S.; YANG, S.; SUN, W. Collaborative SCTP: a collaborative approach to improve the performance of SCTP over wired-cum-wireless networks. In: *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. [S.l.: s.n.], 2004. p. 276–283.
- [41] YE, G.; SAADAWI, T.; LEE, M. SCTP congestion control performance in wireless multi-hop networks. *MILCOM*, v. 2, p. 934–939, 2002.