Pós-Graduação em Ciência da Computação

“aCMT-UC: Uma proposta de distribuição de carga centrada no usuário para redes heterogêneas sem fio”

Por

Lorena Lima Marques

Dissertação de Mestrado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, ABRIL/2013

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

LORENA LIMA MARQUES

“aCMT-UC: Uma proposta de distribuição de carga centrada no usuário para redes heterogêneas sem fio"

ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADOR(A): Kelvin Lopes Dias

RECIFE, ABRIL/2013

Catalogação na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Marques, Lorena Lima
aCMT-UC: uma proposta de distribuição de carga centrada no usuário para redes heterogêneas sem fio. / Lorena Lima Marques. - Recife: O Autor, 2013.
xii, 88 folhas: fig., tab.

Orientador: Kelvin Lopes Dias.
Dissertação (mestrado) - Universidade Federal de Pernambuco. CIn, Ciência da Computação, 2013.

Inclui bibliografia e apêndice.

1. Redes de computadores. 2. Sistemas distribuídos. I. Dias, Kelvin Lopes (orientador). II. Título.

004.6 CDD (23. ed.) MEI2013 – 049

Dissertação de Mestrado apresentada por **Lorena Lima Marques** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**aCMT-UC: Um Proposta de Distribuição de Carga Centrada no Usuário para Redes Heterogêneas sem Fio**" orientada pelo **Prof. Kelvin Lopes Dias** e aprovada pela Banca Examinadora formada pelos professores:

Prof. Paulo Roberto Freire Cunha

Centro de Informática / UFPE

Prof. Marco Antonio de Oliveira Domingues
Deptº Acadêmico de Sistemas Eletro-Eletrônicos / IFPE

Prof. Kelvin Lopes Dias

Centro de Informática / UFPE

Visto e permitida a impressão.

Recife, 25 de fevereiro de 2013.

Profa. Edna Natividade da Silva Barros

Coordenadora da Pós-Graduação em Ciência da Computação do

Centro de Informática da Universidade Federal de Pernambuco.

Dedico este trabalho à minha família pelo seu amor incondicional e pelo esforço empreendido para me propiciar uma educação de qualidade. Ao meu marido, Adriano, pelo incentivo, cuidado, força e ajuda necessária para que eu conseguisse concluir este trabalho.

Agradecimentos

Para o desenvolvimento desta dissertação de mestrado foi necessário muita dedicação e trabalho intenso. Entretanto, o apoio de algumas pessoas e instituições foi fundamental para a sua viabilização e por este motivo gostaria de agradecê-las formalmente.

Para iniciar este projeto:

Inicio meus agradecimentos ao Prof. Dr. Kelvin Lopes Dias, meu professor de graduação, professor de pós-graduação e meu orientador neste trabalho. Agradeço pela confiança em mim depositada, defendendo a minha candidatura e se dispondo a me orientar. Agradeço pelo seu incentivo e por ter me apresentado às novas áreas de conhecimento relacionadas com o tema deste trabalho. Dele recebi a inspiração para a escolha do tema desta dissertação.

Para desenvolver este trabalho:

Agradeço à FACEPE – FUNDAÇÃO DE AMPARO À CIÊNCIA E TECNOLOGIA DO ESTADO DE PERNAMBUCO pelo auxílio financeiro fundamental para o desenvolvimento deste trabalho. Agradeço aos colegas de curso, pela troca de experiências, especialmente ao Adriano Avelar cuja ajuda foi de grande importância para a implementação desta proposta.

Para concluir este mestrado:

Agradeço a minha banca de defesa, ao Prof. Kelvin Lopes Dias, meu orientador, ao Prof. Paulo Roberto Freire Cunha e ao Prof. Marco Antônio Domingues que me ajudaram a cumprir mais uma etapa em direção à conclusão deste mestrado.

Resumo

A crescente disponibilidade de acesso sem fio através de diferentes tecnologias (e.g., IEEE 802.11, IEEE 802.16 e 3G/UMTS) e o aumento do número de dispositivos com suporte a múltiplas interfaces heterogêneas possibilitam uma gama de oportunidades de conectividade sem fio aos usuários. Tradicionalmente, para este cenário, tem-se proposto algoritmos, protocolos e arquiteturas para a escolha da melhor rede de acordo com o perfil do usuário, característica da aplicação e da rede. Outra possibilidade de pesquisa, ainda não tão explorada no âmbito sem fio, é a distribuição de carga e conteúdo através de múltiplos caminhos simultâneos. Esta característica é conhecida como *multihoming* e pode dispor redundância de caminhos, oferecendo, dessa forma, certo grau de confiabilidade na rede.

O grande desafio para implantação do *multihoming* em redes móveis heterogêneas decorre da instabilidade que esses ambientes possuem. O problema se agrava quando existem várias interfaces tentando se comunicar simultaneamente. O CMT (*Concurrent Multipath Transfer*) usa a característica *multihoming* do SCTP (*Stream Control Transmission Protocol*) para transmitir dados através de múltiplos caminhos fim-a-fim simultaneamente. No entanto, o CMT não possui um bom desempenho em redes com diferentes características, especialmente em se tratando de vídeo.

Esta dissertação propõe o aCMT-UC (*Adaptive* CMT-*User Centric*), uma solução baseada em *multihoming*/centrada no usuário para a distribuição de carga em redes heterogêneas sem fio, tornando o CMT adaptável à dinâmica dos caminhos. O aCMT-UC prioriza os quadros de vídeo mais importantes que são transmitidos pelo melhor caminho sem fio. A distribuição de carga e priorização de vídeo são realizadas através da atribuição dinâmica de diferentes pesos para os caminhos sem fio, de acordo com um procedimento inteligente baseado em Lógica *Fuzzy*. Foram avaliadas métricas de Qualidade de Serviço e Experiência (QoS/QoE) em uma rede formada pelas tecnologias IEEE 802.11, IEEE 802.16 e 3G/UMTS em um simulador de redes (ns-2). Os resultados obtidos demonstram a eficácia da proposta em garantir QoS, bem como, o suporte adequado aos requisitos de tráfego de vídeo.

Palavras-chave: *multihoming*, SCTP, CMT, Redes heterogêneas sem fio, IEEE 802.11, IEEE 802.16, UMTS, QoE, QoS, distribuição de carga.

Abstract

The increasing availability of wireless access through different technologies (e.g., IEEE 802.11, IEEE 802.16 and 3G/UMTS) and the growing number of devices enabled with multiple heterogeneous interfaces will improve the wireless connectivity opportunities for the users. Traditionally, in this scenario, the research has been carried out to propose novel algorithms, protocols and architectures in order to choose the best network according to the user profile, application and the network characteristics. Another research possibility, not yet well explored in the wireless domain, is the load and content distribution through multiple paths simultaneously. This feature is known as multihoming and can provide path redundancy, thus improving the network reliability.

The big challenge for the deployment of multihoming in heterogeneous mobile networks stems from the instability intrinsic to these wireless environments. The problem is even challenging when there are multiple and simultaneous connectivity opportunities. The CMT (Concurrent Multipath Transfer) uses the multihoming feature of SCTP (Stream Control Transmission Protocol) to transmit data via multiple end-to-end paths simultaneously. However, the CMT does not have a good performance in networks with different characteristics, especially for video traffic.

This dissertation proposes the aCMT-UC (Adaptive User-Centric CMT), a user-centric/multihoming based solution for load distribution in wireless heterogeneous networks, which makes CMT adaptable to the dynamic of paths. The aCMT-UC prioritizes the most important video *frames* which are transmitted through the best wireless path. The load distribution and video prioritization are carried out through the dynamic assignment of different weights to the wireless paths according to an intelligent procedure based on Fuzzy Logic. We evaluated QoS/QoE metrics technologies in a network formed by IEEE 802.11, IEEE 802.16 and 3G/UMTS and modeled in the network simulator (ns-2). The results demonstrate the effectiveness of the proposal to ensure the QoS, as well as, the adequate support to the requirements of video traffic.

Keywords: multihoming, SCTP, CMT, heterogeneous wireless networks, IEEE 802.11, IEEE 802.16, UMTS, QoE, QoS, load distribution.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

aCMT	Adaptive CMT
aCMT-UC	Adaptive CMT - User Centric
CMT	Concurrent Multipath Transfer
CN	Correspondent Node
CWND	Congestion Window
DNS	Domain Name System
DoS	Denial of Service
FER	Frame Error Rate
GSPN	Generalized Stochastic Petri Sets
HA	Home Agent
IETF	Internet Engineering Task Force
IP	Internet Protocol
LAN	Local Area Network
MN	Mobile Node
MOS	Mean Opinion Score
MPTCP	Multipath Transmission Control Protocol
MTU	Maximum Transmission Unit
NAM	Networks Animator
NS-2	Network Simulator version 2 Partial Reliability
PR-SCTP	Partial Reliability - Stream Control Transmission Protocol
PSNR	Peak Signal to Noise Ratio
P2P	Peer-to-Peer
QoE	Quality of Experience
QoS	Quality of Service
RFC	Request for Comments

RTT	Round Trip Time
SCTP	Stream Control Transmission Protocol
SPN	Stochastic Petri Sets
SSIM	Structural SIMilarity
SS7	Signalling System No. 7
TCP	Transmission Control Protocol
TSN	Transport Sequence Number
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
VQM	Video Quality Metric
VoIP	Voice over IP
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide interoperability for Microwave Access
WLAN	Wireless LAN
WM2-SCTP	Wireless Multipath Multi-flow - SCTP

1. Introdução

A crescente disponibilidade de acesso sem fio através de diferentes tecnologias (e.g., IEEE 802.11, IEEE 802.16 e 3G/UMTS) e o aumento do número de dispositivos com suporte a múltiplas interfaces heterogêneas possibilitam uma gama de oportunidades de conectividade aos usuários. Tradicionalmente, para este cenário, tem-se proposto algoritmos, protocolos e arquiteturas para a escolha da melhor rede de acordo com o perfil do usuário, característica da aplicação e da rede. No entanto, outra possibilidade, ainda não tão explorada no âmbito sem fio, é a distribuição de carga e conteúdo através de vários caminhos de forma simultânea. Esta característica é conhecida como *multihoming* e provê redundância de caminhos, oferecendo, dessa forma, certo grau de confiabilidade na rede (STEWART, 2001).

O grande desafio para implantação do *multihoming* em redes móveis heterogêneas decorre da instabilidade que esses ambientes possuem. A possibilidade de desconexões frequentes, de variações na vazão e de perdas inerentes ao caminho sem fio são fatores que dificultam a comunicação nesses tipos de redes. O desafio é ainda maior quando existem várias interfaces heterogêneas sem fio disponíveis simultaneamente (WALLACE, 2011).

Em termos de protocolo da pilha TCP/IP (*Transmission Control Protocol/Internet Protocol)*, o SCTP (*Stream Control Transmission Protocol*) pode ser utilizado para fornecer a possibilidade de transmissão via múltiplos caminhos fim-a-fim tanto em redes sem fio como em redes cabeadas. Originalmente, o SCTP foi projetado como um protocolo de uso geral para transportar mensagens de sinalização telefônica. A definição do protocolo foi feita pelo grupo de trabalho SIGTRAN do IETF (*Internet Engineering Task Force*), que emitiu um documento de padronização do SCTP em outubro de 2000 (RFC 2960).

Como o SCTP possui a característica *multihoming* de forma nativa, quando utilizado em redes sem fio, o dispositivo móvel do usuário pode se comunicar com um mesmo servidor através de várias interfaces (Wi-Fi, WiMax, etc.). No entanto, não existe um mecanismo de avaliação inicial que indique qual o melhor caminho para transmissão, entre os disponíveis, a ser utilizado para o envio dos dados. Além disso, mesmo possuindo os elementos fundamentais para dar suporte ao *multihoming*, o SCTP

não possibilita transmissões simultâneas de dados via múltiplos caminhos de forma concorrente, ou seja, em sua forma original, uma interface é utilizada como primária e as outras interfaces dos dispositivos são utilizadas somente para redundância e, apenas, em caso de falha da primária (STEWART, 2000).

Para sobrepor essa limitação, em (IYENGAR, 2006) foi proposto o SCTP-CMT ou apenas CMT (*Concurrent Multipath Transfer*). O CMT é uma extensão baseada no SCTP que torna possível o suporte completo ao *multihoming*, além de prover distribuição de carga e aumento da vazão dos dados em redes homogêneas. Redes heterogêneas utilizam diferentes larguras de banda para cada caminho, dependendo das características de rede. Estudos apontam que o desempenho do CMT, ao se utilizar caminhos com largura de banda variada, apresenta uma queda considerável (KIM, 2010).

Como o CMT não fornece um suporte adequado em redes heterogêneas, propusemos uma extensão do CMT denominada aCMT (AVELAR, 2012) ou CMT Adaptativo tentando aliviar o aumento significativo do uso da rede, sendo possível tratar a disseminação de dados de uma maneira balanceada por múltiplos caminhos, ao invés de escolher um melhor caminho para todos os pacotes de um fluxo. O esquema utiliza a Lógica *Fuzzy* para tornar o CMT sensível à dinâmica da rede. Desta forma, o aCMT pode diminuir a vazão no caminho onde o tráfego está mais congestionado ou aumentar onde o enlace estiver mais livre, fazendo com que a aplicação obtenha maior vazão, melhor qualidade de serviço (QoS – *Quality of Service*) e, consequentemente, forneça uma escolha inteligente do caminho.

A transmissão de dados pela Internet deve sofrer um aumento significativo nos próximos anos. Existe uma previsão de que em 2016, 88% do tráfego da Internet será composto de usuários domésticos e/ou universitários e somente 12% será relacionado a empresas e governo. Focando no tráfego de usuários domésticos e/ou universitários, estima-se que teremos, um misto de 54% do tráfego destinado a vídeos, 23% como sendo trafego P2P (*Peer-to-Peer*) e 23% de outros tráfegos como web. Isso torna o vídeo, o tipo de fluxo que consumirá maior banda nos próximos anos (CISCO, 2012).

Apesar das diversas arquiteturas propostas para prover QoS, a percepção do usuário para o tráfego de vídeo pode não ser satisfatória mesmo com as garantias de métricas tradicionais como vazão, atraso e perda de pacotes. Dessa forma, as novas

propostas de protocolos, mecanismos e arquiteturas para a Internet requerem suporte adequado à qualidade de experiência (QoE – *Quality of Experience*) a fim de garantir a satisfação do usuário.

Deste modo, há um desafio de se utilizar o *multihoming* para prover balanceamento de carga. Com isso, neste trabalho, propõe-se o aCMT-UC (*Adaptive CMT - User Centric*) com o objetivo de resolver este desafio e propor uma solução inteligente que permita o balanceamento de carga, além do aprovisionamento de QoS e QoE em um ambiente heterogêneo com redes de acesso Wi-Fi, WiMAX e UMTS.

1.1 Objetivos da Pesquisa

A seguir, serão apresentados os objetivos geral e específicos da pesquisa desenvolvida nesta dissertação.

1.1.1 Objetivo Geral

Esta pesquisa consiste em desenvolver uma extensão ao protocolo de transporte SCTP-CMT, que objetiva classificar e priorizar a melhor rede entre as disponíveis no ambiente, com base em técnicas de inteligência artificial e de tomada de decisão, visando manter a qualidade de serviço e experiência da transmissão de dados do dispositivo móvel.

1.1.2 Objetivos Específicos

- Caracterizar o problema do uso de diferentes tecnologias sem fio simultâneas, verificando como a distribuição de carga pode ser realizada;
- Realizar um levantamento sobre o estado da arte das diversas soluções relacionadas com o tema em questão;
- Propor e avaliar solução adaptativa e centrada no usuário para o acesso heterogêneo sem fio com suporte a *multihoming*, QoS e QoE;
- Propor mecanismo para a priorização de *frames* de vídeo mais significativos e algoritmo para a tomada de decisão na seleção de rede baseada em técnica de inteligência computacional, tornando o sistema dinâmico;
- Caracterizar e desenvolver um sistema baseado em lógica Fuzzy para a tomada de decisão sobre a distribuição do tráfego entre as redes de acessos sem fio heterogêneas;

- Realizar a avaliação de desempenho da proposta por meio de simulações no NS-2 (*Network Simulator* 2).

Esta dissertação está dividida da seguinte forma. No Capítulo 2, são apresentados os trabalhos relacionados. O Capítulo 3 traz uma breve descrição dos conceitos e protocolos relacionados à pesquisa desenvolvida, em seguida, o Capítulo 4 caracteriza o problema abordado nesta dissertação por meio de modelos baseados em redes de Petri para o SCTP e o SCTP-CMT. A proposta do trabalho é apresentada no Capítulo 5, onde é detalhada a política utilizada para tornar o protocolo SCTP-CMT adaptativo. No Capítulo 6 são apresentados os métodos de validação da proposta, bem como as modificações necessárias para seu funcionamento. A avaliação da proposta será discutida no Capítulo 7, utilizando-se tanto métricas de QoS quanto de QoE, apresentando a eficácia da proposta e, por fim, o Capítulo 8 conclui esta dissertação com as principais contribuições obtidas e a proposta dos trabalhos futuros.

2. Trabalhos Relacionados

O balanceamento de carga em vários caminhos está longe de ser um conceito novo. No entanto, a maioria das abordagens que abrangem caminhos heterogêneos que são flexíveis para mudanças nas condições de rede servem como soluções fim-a-fim, como os protocolos da camada de transporte SCTP (STEWART, 2007) e MPTCP (FORD, 2011). Em contraste, quando o balanceamento de carga é implementado dentro da infraestrutura de rede é normalmente feito de forma muito mais rígida. Na camada de rede, uma configuração muito básica (BATES, 1998) envolve o uso de políticas de roteamento para dividir o tráfego de entrada e saída entre os pontos de saída diferentes. No entanto, esta divisão, que é baseada em sub-redes, é bastante estática e só pode reagir a mudanças em padrões de tráfego com um longo atraso – geralmente necessita da intervenção do administrador. Deste modo, serão apresentados alguns trabalhos que envolvam seleção de caminho e balanceamento de carga.

O artigo (AYDIN, 2009) analisa a vazão de aplicações que utilizam SCTP-CMT sobre redes 802.11 com múltiplos saltos estáticos através do simulador QualNet. Os autores consideraram todos os nós como estáticos e sem conexão com redes cabeadas ou Internet. Eles sugeriram que a retransmissão de pacotes deveria ser feita sobre o caminho com maior cwnd e ssthresh. A análise foi feita através da comparação do SCTP CMT com três técnicas: padrão SCTP utilizando um único caminho com a melhor largura de banda, padrão SCTP utilizando um único caminho com a pior largura de banda e o padrão SCTP utilizando todos os caminhos disponíveis. Os resultados mostraram que o SCTP-CMT possuiu melhor desempenho em relação às três alternativas utilizadas como comparação. Apesar deste trabalho abordar o CMT em redes móveis, limita-se a redes homogêneas. Os autores não propuseram nenhuma política que reduzisse os impactos gerados pelo uso do CMT, já que ao se empregá-lo em redes com diferentes características, a vazão tende a degradar com o tempo.

O trabalho (ZHANG, 2009) avalia o CMT em redes sem fio 802.11 com uma abordagem *cross-layer* em função de dois parâmetros: FER (*Frame Error Rate*) na camada de enlace e RTT (*Round-Trip Time*) na camada de transporte. Porém, também analisa apenas em redes homogêneas. Esses parâmetros são utilizados para indicar as condições do caminho e enviar pacotes através do caminho mais rápido, visando reduzir o número de pacotes fora de ordem no receptor. A proposta *Cross-Layer CMT* foi

avaliada através do ns-2 e comparada com o CMT padrão, através da variação do parâmetro FER, onde o caminho 1 é fixado em 1% e o caminho 2 varia de 1% a 10%. Deste modo, os autores analisam a vazão da aplicação de acordo com condições fixadas previamente. Apesar de apresentar bons resultados, os autores se limitaram a avaliar apenas a tecnologia Wi-Fi, além da análise não ser realizada de forma dinâmica.

Os artigos (HUANG, 2009) e (BUDZISZ, 2009) abordam os protocolos SCTP e SCTP-CMT utilizando *multihoming* em ambientes de mobilidade. Porém os artigos que tratam mobilidade, não atentam para a distribuição de carga que pode ser feita antes e depois do processo de *handover*, ficando limitados apenas ao momento de execução do processo, para evitar perdas de pacotes.

Em (RÜNCOS, 2011) é avaliado, através de simulações no ns-2, o impacto no desempenho do SCTP para o transporte de tráfego VoIP em terminais *multihoming*, variando-se dois parâmetros: PMR (*Path.Max.Retrans*) e RTOmax (limite superior do parâmetro *Retransmission TimeOut*), juntamente com o algoritmo de seleção automática de rotas baseado no menor atraso (*delay-centric*). A qualidade da chamada de cada simulação era medida usando o MOS (*Mean Opinion Score*). Neste trabalho os autores não utilizam transmissões simultâneas e também não realizam simulações em ambientes heterogêneos.

Em (NGUYEN, 2011) é avaliado o compartilhamento de carga utilizando o MPTCP (*Multipath Transmission Control Protocol*), uma versão modificada do protocolo de transporte TCP (*Transmission Control Protocol*) que permite a transmissão de dados via múltiplos caminhos simultaneamente. Os autores montaram um *testbed* com três possíveis cenários: dois caminhos Ethernet, um caminho Ethernet e outro Wi-Fi e o terceiro cenário utilizando-se um caminho Wi-Fi e outro 3G. Os resultados das medições mostram que MPTCP com controle de congestionamento acoplado fornece um melhor desempenho em ambientes homogêneos. No entanto, os testes realizados em ambiente com redes heterogêneas geraram uma degradação na vazão, revelando a necessidade de um algoritmo inteligente para a distribuição de carga no MPTCP.

Em (ZHU, 2009) os autores consideram o problema de alocação de taxa entre múltiplos fluxos simultâneos de vídeo em redes de acesso heterogêneas. Foi desenvolvido e avaliado um *framework* analítico de alocação de taxa ideal, com base na

taxa de bits disponível e RTT sobre cada característica de acesso de rede e taxa de distorção de vídeo. A taxa de alocação é formulada como um problema de otimização convexa que minimiza a distorção total esperada de todos os fluxos de vídeo. Os esquemas de alocação de várias taxas são avaliados em simulações de múltiplos vídeos de alta definição (HD), através de fluxos TCP no ns-2. Os autores se limitam a analisar a proposta apenas em redes Wi-Fi e Ethernet.

Em (SONG, 2012) os autores propõem uma abordagem analítica para avaliar o nível de desempenho de uma transmissão que envia rajadas de tráfego de vídeo através de múltiplos canais disponíveis de forma probabilística, podendo obter a média de atraso, jitter e a probabilidade de falha por atraso. Para permitir a transmissão via múltiplos caminhos através de dispositivos sem fio, cada rajada de vídeo pode ser enviada para uma rede sem fio disponível de acordo com uma probabilidade de divisão de fluxo, no entanto os autores não realizam uma análise dinâmica para determinar a probabilidade de divisão dos fluxos.

Em (WALLACE, 2012) são abordadas as principais pesquisas envolvendo a utilização do *multihoming*, como gerenciamento de *handover*, CMT e atividade *cross-layer*. Os autores citam uma série de contribuições que melhoraram a eficácia operacional do CMT. No entanto, concluem que ainda é necessário realizar mais pesquisas que envolvam seleção de caminhos de transmissão para redes sem fio.

Em (NIGHTINGALE, 2012) os autores propõe o CMT-NEMO, um esquema de transferência simultânea via múltiplos caminhos de conteúdo de vídeo escalável para os usuários em diversas bases de redes móveis. A proposta divide um fluxo de vídeo SVC (H.264 *Scalable Video Coding*) em um número de sub-fluxos (um para cada caminho de rede disponível) e associa cada sub-fluxo a uma BID (*Binding ID*), que identifica unicamente um caminho do HA (*Home Agent*) para o MR (*Mobile Router*). No entanto, a ideia se limita a empregar o esquema em um testbed no núcleo da rede, ou seja, dividir o fluxo na parte cabeada.

Em (TU, 2012) os autores estudaram a melhoria da capacidade das redes sem fio em admitir um número maior de fluxos multimídia simultâneos com desempenho garantido. Uma política de escalonamento de fluxo e de agregação de canal foram teoricamente analisadas e apresentadas como uma forma de utilizar estrategicamente os recursos da rede para a transmissão de fluxos multimídias concorrentes. Deste modo, os

autores propõem um algoritmo eficiente de transmissão multi-fluxos *multicast*. As políticas estudadas e o algoritmo proposto foram avaliados através do ns-2 e se limitaram a redes homogêneas.

Em (YUAN, 2010) os autores apresentam o WM2-SCTP (*Wireless Multipath Multi-flow - Stream Control Transmission Protocol*), uma solução na camada de transporte para a transferência via múltiplos caminhos, concomitante com sub-fluxos paralelos. O WM2-SCTP visa explorar as características *multihoming* e *multi-streaming* do SCTP através do agrupamento de fluxos SCTP em sub-fluxos, com base em seus requisitos de QoS. Deste modo, selecionam-se os melhores caminhos para cada sub-fluxo com o objetivo de melhorar as taxas de transferência de dados. Os autores fizeram uma análise comparativa entre o WM2-SCTP e o SCTP-CMT através do ns-2, limitando-se a tecnologia Wi-Fi.

A Tabela 1 apresenta uma síntese dos trabalhos relacionados. Os trabalhos que se aproximam do aCMT-UC são avaliados em apenas uma tecnologia. O único trabalho (NGUYEN, 2011) que utilizou três tecnologias para avaliação apresentou baixo desempenho.

Tabela 1. Resumo dos Trabalhos Relacionados.

Artigo	Tipo de Análise	Tecnologias Avaliadas	Objetivo
Aydin, I. et al. 2009	Simulação (QualNet)	IEEE 802.11	Sugerem que a retransmissão de pacotes deve ser feita sobre o caminho com maior cwnd e ssthresh.
Zhang, X. et al 2009	Simulação (ns-2)	IEEE 802.11	Utiliza os parâmetros FER e RTT para indicar as condições do caminho e enviar pacotes através do caminho mais rápido.
Rüncos, R. et al. 2011	Simulação (ns-2)	Ethernet	Avaliam o desempenho de um tráfego VoIP em terminais *multihoming*, variando-se dois parâmetros: PMR e RTOmax, juntamente com o algoritmo de seleção automática de rotas baseado no menor atraso (*delay-centric*).
Nguyen, S. et al. 2011	Testbed	Ethernet Wi-Fi 3G	Inserem um controle de congestionamento no MPTCP para avaliar a vazão em diferentes cenários.

Zhu, X. et al. 2009	Simulação (ns-2)	Ethernet IEEE 802.11	Foi desenvolvido e avaliado um *framework* analítico de alocação de taxa ideal para vídeo, com base na taxa de bits disponível e RTT.
Song, W. et al. 2012	Modelagem (Markov)	IEEE 802.11 IEEE 802.16	Propõem uma abordagem analítica para avaliar o nível de desempenho de uma transmissão que envia rajadas de tráfego de vídeo através de múltiplos canais disponíveis de forma probabilística
Nightingale, J. et al. 2012	Testbed	Ethernet	Propõe o CMT-NEMO, um esquema de transferência simultânea via múltiplos caminhos de conteúdo de vídeo escalável.
Tu, W. et al. 2012	Simulação (ns-2)	IEEE 802.11	Uma política de escalonamento de fluxo e de agregação de canal foram teoricamente analisadas e apresentadas como uma forma de utilizar estrategicamente os recursos da rede para a transmissão de fluxos multimídias concorrentes.
Yuan, Y. et al 2010	Simulação (ns-2)	IEEE 802.11	O WM2-SCTP visa explorar as características *multihoming* e *multi-streaming* do SCTP através do agrupamento de fluxos SCTP em sub-fluxos, com base em seus requisitos de QoS.

3. Fundamentação teórica

A Internet consiste em um conjunto de padrões (protocolos) que definem como as informações são transmitidas pela rede. Sendo um conjunto de protocolos, ou pilha de protocolos, a Internet oferece serviços de comunicação de forma modular, tratando problemas específicos em etapas bem definidas. Devido a essa característica multiprotocolar, a Internet é usualmente denominada de pilha TCP/IP, como referência a dois protocolos: o IP (*Internet Protocol*) e o TCP (*Transmission Control Protocol*). Quando uma aplicação deseja utilizar os recursos de comunicação da Internet, os pontos de acesso a esses serviços são os protocolos de transporte, cuja função é promover uma transferência de dados confiável e eficiente entre a máquina de origem e a máquina de destino, independente das redes físicas em uso no momento (TANENBAUM, 2003).

Os dados transmitidos da camada de aplicação para a camada de transporte são chamados de dados de usuário, tendo significado apenas para o emissor e o receptor dos datagramas IP, deste modo, costuma-se dizer que os protocolos de transporte são fim-a-fim.

Dentre os diversos mecanismos para garantir confiabilidade, o mais comum é oferecer operações de notificação de erro, perda ou duplicação de pacotes. Através de mensagens de controle dos protocolos de transporte, o destino dos dados perdidos pode informar a falta de informações necessárias, requisitando uma ação de retransmissão desses dados. Se essas transmissões são feitas e de que modo, isso varia de acordo com o protocolo de transporte utilizado.

Outra tarefa dos protocolos de transporte é o controle de fluxo, cuja função é evitar, entre outras coisas, que um emissor muito rápido sobrecarregue um receptor muito lento, ou seja, a ideia por trás de um bom fluxo é que a taxa de transmissão, em uma determinada comunicação de transporte, deve variar dinamicamente, adequando-se às situações da rede e às condições de processamento das estações envolvidas.

No contexto dos programas aplicativos, o protocolo responsável diretamente pela abstração das particularidades da interligação em rede é o protocolo de transporte. Orginalmente, a Internet definiu dois protocolos padrões de transporte: o UDP (*User Datagram Protocol*) e o TCP. As aplicações que não necessitam de uma transmissão confiável, porém, desejam uma comunicação rápida e flexível, recorrem ao protocolo

UDP, enquanto o TCP é destinado às comunicações confiáveis. No entanto, o SCTP (*Stream Control Transmission Protocol*) (RFC 2960) (STEWART, 2000) fornece um número de funções robustas e flexíveis a diversas aplicações, funções essas não presentes nos protocolos UDP ou TCP.

Este capítulo está organizado da seguinte forma. A seção 3.1 aborda as principais características do protocolo SCTP. A seção 3.2 apresenta a extensão utilizada como base para a proposta deste trabalho.

3.1 Stream Control Transmission Protocol (SCTP)

Inicialmente, o SCTP foi desenvolvido para troca de mensagens telefônicas na Internet, facilitando o transporte de protocolos como o SS7 (*Signalling System No. 7*). Contudo, as características do SCTP tornam seu uso atraente em aplicações comuns da Internet, pois utilizas as principais características encontradas tanto no protocolo TCP quanto no UDP, o que resulta em sua potencial incorporação ao grupo dos tradicionais protocolos de transporte da pilha TCP/IP.

A definição formal do protocolo está na RFC 2960 (STEWART, 2000), sendo corrigido na listagem contida no *SCTP Implementer's Guide* (STEWART, 2003). As características do SCTP podem ser facilmente utilizadas por praticamente qualquer aplicação que utilize o TCP.

O SCTP possui transmissão confiável por ser orientado à conexão. Contudo, embora o SCTP seja orientado à conexão, como o TCP, a conexão SCTP é definida como uma *associação* que possui um contexto e significado mais amplo que a conexão TCP.

No contexto do SCTP, dois terminais estabelecem uma associação SCTP para se comunicarem através deste protocolo. Uma associação, como uma conexão TCP, possui um caráter lógico *fim-a-fim*, sendo controlado por variáveis e protocolos da camada de transporte. A principal diferença entre uma associação SCTP e uma conexão TCP é que a primeira corresponde a um número arbitrário de fluxos *simplex* (unidirecionais) acertados durante o início da associação, enquanto a última é formada apenas por um único fluxo *full-duplex* (bidirecional) (COSTA, 2005). Outra característica que os diferenciam diz respeito à como os dados de usuários são tratados pelo protocolo de transporte, já que o SCTP é orientado à mensagem e o TCP é orientado a octetos, deste modo os dados de usuários a serem transmitidos pelo SCTP são tratados como blocos

independentes, com significados próprios, lembrando um serviço de datagrama confiável.

Em uma associação, os pacotes SCTP transmitem as mensagens de usuários e de controle. O pacote SCTP consiste de um cabeçalho comum, seguido por uma ou mais mensagens (ou blocos de informações), como apresentado na Figura 1. Cada campo deste cabeçalho, que somados correspondem a um tamanho de doze octetos, é comum a todas as mensagens SCTP.

Figura 1. Formato dos pacotes SCTP.

Os quatro octetos iniciais correspondem aos campos PORTA DE ORIGEM e PORTA DE DESTINO, que possuem valores utilizados para multiplexação de comunicações diferentes. Os números de porta, juntamente com os endereços IP de origem e de destino, que podem ser mais de um por estação (*multihoming*), são utilizados para identificar as associações. No caso de conexões TCP, apenas dois endereços IP são utilizados na identificação (*single-homed*). Contudo, assim como no TCP, apenas um par de portas é empregado para identificação das associações SCTP.

No desenvolvimento do protocolo SCTP, um dos pontos mais importantes a ser considerado foi a questão da segurança. Uma das medidas adotadas foi a adoção do campo ETIQUETA DE IDENTIFICAÇÃO no cabeçalho comum dos pacotes SCTP, com a finalidade de verificar a autenticidade de determinado pacote, validando seu

emissor através de uma troca de valores de identificação, que são gerados aleatoriamente pelas estações no estabelecimento da associação, de modo que estes valores permaneçam inalterados durante todo o tempo de vida da associação, não havendo qualquer mecanismo de atualização dinâmica entre eles.

Cada associação possui dois valores para o campo ETIQUETA DE IDENTIFICAÇÃO, sendo um para cada direção de comunicação. Cada terminal calcula sua própria etiqueta, de modo aleatório, e a informa ao terminal par da associação. Deste modo, é possível evitar que o valor desse campo seja previamente conhecido por uma estação que deseja enviar mensagens forjadas para a associação.

Outro ponto a ser considerado na implementação do protocolo SCTP são os mecanismos de controle de erro, através da adoção de uma SOMA DE VERIFICAÇÃO de quatro octetos, sendo mais robusta que o algoritmo adotado pelo TCP, que possui apenas dois octetos.

Múltiplas mensagens de usuário e controle podem ser encapsuladas em um mesmo pacote com o objetivo de aumentar a eficiência de comunicação, porém isso não deve ocorrer para as mensagens de controle INIT, INIT ACK e SHUTDOWN COMPLETE.

Quando o pacote é processado, as mensagens encapsuladas são separadas na ordem em que são recebidas. Deste modo, em um pacote SCTP algumas mensagens devem aparecer, obrigatoriamente, antes de outras mensagens em determinados momentos da associação.

Uma implementação SCTP pode empacotar várias mensagens em um único pacote, durante congestionamentos, mesmo que o usuário tenha requerido o não-empacotamento.

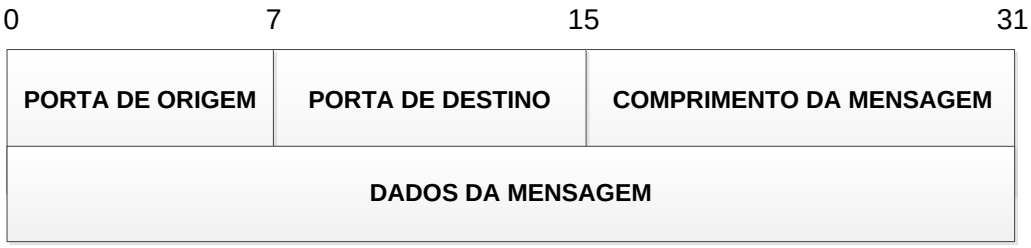
O protocolo SCTP oferece os seguintes serviços a seus usuários:

- Entrega confirmada de dados de usuários, livre de erros e não duplicados;
- Fragmentação de dados em conformidade com o MTU (*Maximum Transmission Unit*) descoberto no caminho;
- Entrega sequencial de dados de usuário em múltiplos fluxos (*multi-streaming*), com opção para entrega por ordem de chegada de mensagens de usuário individuais;
- Empacotamento opcional de múltiplas mensagens de usuário num único pacote SCTP;

- E tolerância a falhas de rede através do suporte a caminhos múltiplos (*multihoming*) em qualquer ou ambas as extremidades de uma associação.

3.1.1 Formato das Mensagens SCTP

Todo pacote SCTP típico possui uma ou mais mensagens, de usuário ou de controle, organizadas após o cabeçalho do pacote, com o tamanho de doze octetos. Independente da funcionalidade, toda mensagem segue um formato padrão, possuindo uma parte fixa de quatro octetos e outra variável, que depende do tipo de mensagem. No caso da parte variável, existe uma série de parâmetros divididos em: obrigatórios e opcionais. O formato das mensagens SCTP segue o modelo apresentado na Figura 2.



O campo TIPO, com um octeto, é utilizado para diferenciar os tipos de mensagens possíveis, podendo assumir valores que variam de 0 a 254, sendo o valor 255 reservado para usos futuros. A Tabela 2 apresenta os valores mais utilizados nesse campo.

Tabela 2. Tipos de mensagens SCTP.

Valor de Tipo	Mensagem Correspondente
0	Dados de usuário
1	Initiation (INIT)
2	Initiation Ack (INIT ACK)
3	Selective Ack (SACK)
4	Heartbeat Request (HEARTBEAT)
5	Heartbeat Ack (HEARTBEAT ACK)
6	Abort (ABORT)
7	Shutdown (SHUTDOWN)
8	Shutdown Ack (SHUTDOWN ACK)
9	Operation Error (ERROR)
10	State Cookie (COOKIE ECHO)
11	Cookie Ack (COOKIE ACK)

12	Explict Congestion Notification Echo (ECNE)
13	Congestion Window Reduced (CWR)
14	Shutdown Complete (SHUTDOWN COMPLETE)
15 ao 62	Reservado
63	Extensão de Mensagem
64 ao 126	Reservado
127	Extensão de Mensagem
128 ao 190	Reservado
191	Extensão de Mensagem
192 a 254	Reservado
255	Extensão de Mensagem

A codificação dos valores do campo TIPO tem como base a idéia de que os dois bits mais significativos (mais à esquerda) indicam o que uma estação deve fazer caso não reconheça o tipo de mensagem especificado (mensagem não suportada). O procedimento a ser tomado é apresentado a seguir:

- 00 – Pare o processamento do pacote SCTP, descarte esse pacote e não processe mais qualquer mensagem adicional desse pacote.
- 01 – Pare o processamento do pacote SCTP, descarte esse pacote e não processe mais qualquer mensagem adicional desse pacote. Notifique a origem do pacote sobre o erro encontrado.
- 10 – Ignore a mensagem com erro e continue processando as demais.
- 11 – Ignore a mensagem com erro e continue processando as demais. Notifique a origem do pacote sobre o erro encontrado.

As mensagens SCTP padrão pertencem à primeira das quatro categorias apresentadas acima.

Outra opção de controle pode ser utilizada com o campo FLAGS DE CONTROLE, de um octeto, que tem como objetivo utilizar seus bits como indicadores de alguma opção, podendo estar marcados ou desmarcados. O uso desse campo é opcional, bem como o número de bits utilizados, dos oito disponíveis, pode variar de mensagem para mensagem.

Para indicar o tamanho total da mensagem, é empregado o campo COMPRIMENTO DA MENSAGEM. O valor desse campo indica o comprimento total, em octetos, da mensagem em questão, incluindo os quatro octetos do cabeçalho padrão

de mensagens. A finalidade desse campo é permitir que as mensagens possuam tamanho variável.

Deve-se notar que o cabeçalho comum do pacote SCTP não contém nenhum tipo de indicação de seu comprimento. Quando a informação, no nível de transporte, é entregue da camada de rede para o SCTP, esse protocolo precisa separar as mensagens que estão encapsuladas nesse pacote. Após os passos operacionais padrão do SCTP, como checagem do campo ETIQUETA DE IDENTIFICAÇÃO, o protocolo SCTP começa a separar as mensagens recebidas. Primeiramente, sabe-se que os doze primeiros octetos estão sempre presentes (em mensagens sem erro). Após esses campos, que correspondem ao cabeçalho comum do pacote SCTP, segue uma ou mais mensagens. Cada mensagem tem quatro octetos fixos, e os dois últimos indicam o comprimento da mensagem. Examinando esse campo, é possível separar as mensagens e analisá-las de forma independente. A Figura 3 apresenta um exemplo de encapsulamento de mensagens em um mesmo pacote.

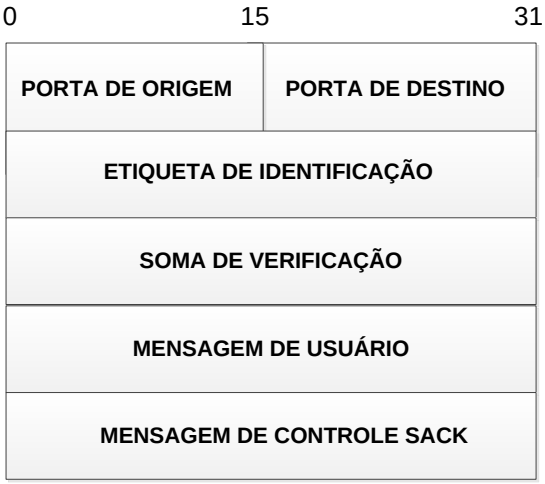


Figura 3. Exemplo de encapsulamento de mensagens em um mesmo pacote.

As Mensagens de Usuário (DATA) contém os dados do usuário a serem transmitidos em uma associação SCTP. Apenas as mensagens de usuário estão associadas a fluxos de comunicação na associação, ou seja, aos mecanismos de confiabilidade a eles associados. A Figura 4 apresenta o formato das mensagens de usuário.

A análise da mensagem de dados de usuários inicia com a verificação do campo FLAGS DE CONTROLE. Nesse campo, três bits (*flags*) são utilizados: U, B e E. Os cinco bits restantes são *reservados,* não sendo utilizados. Os bits reservados não devem ser marcados, ficando seus valores estabelecidos com zero.

O bit U, quando marcado, indica que a mensagem sendo transmitida não está ordenada e não há nenhum *número de sequência no fluxo* atribuído a essa mensagem. As mensagens com bit U marcado são as mensagens urgentes. Se uma mensagem urgente é fragmentada, para atender o MTU do caminho, os fragmentos da mensagem devem ter também o bit U marcado, indicando que os fragmentos de mensagens urgentes são também urgentes.

Eventualmente, as mensagens de usuário devem ser fragmentadas para atender as exigências do MTU descoberto no caminho. O cabeçalho dos pacotes SCTP e as mensagens de controle não são fragmentados. A fragmentação deve ocorrer apenas com as mensagens de usuário, que são, de fato, as maiores mensagens transportadas em uma associação.

Deve-se ter em mente que, apesar da fragmentação, os dados de usuários continuam sendo tratados de forma “atômica”. Caso haja algum enlace, no caminho entre o emissor e o receptor do pacote SCTP, com MTU menor que o tamanho do pacote de rede enviado, a fragmentação desse pacote segue os procedimentos de fragmentação do protocolo IP.

O bit B, quando marcado, indica que a mensagem é o primeiro fragmento de uma mensagem maior. Já o bit E indica que a mensagem é o último fragmento de uma mensagem também maior. Dessa forma, uma mensagem de usuário não-fragmentada

deve ter dois bits B e E marcadas indicando que, ao mesmo tempo, essa mensagem é o primeiro e último fragmento de uma mensagem. Se uma mensagem engloba o primeiro e último fragmento de uma mensagem maior, isto significa que esse fragmento, na realidade, é a própria mensagem. A Tabela 3 apresenta os estados possíveis desses dois bits e seus significados associados.

Tabela 3. Significado dos bits B e E.

BIT B	BIT E	SIGNIFICADO
1	1	Mensagem não-fragmentada.
0	0	Parte "mediana" de uma mensagem fragmentada.
1	0	Primeira parte de uma mensagem fragmentada.
0	1	Última parte de uma mensagem fragmentada.

Caso um fragmento seja uma parte "mediada" de uma mensagem, isto é, não seja nem o primeiro nem o último fragmento, outras informações são utilizadas para fazer a remontagem dos fragmentos de forma correta. Essas informações adicionais estão sempre presentes na própria mensagem de usuário.

Como já foi visto, o campo COMPRIMENTO DA MENSAGEM contém o valor do tamanho da mensagem de usuário em octetos, incluindo o cabeçalho dessa mensagem.

Para indicar unicamente uma mensagem, permitindo, entre outras coisas, a remontagem correta de fragmentos de mensagens no destino, é utilizado o campo TSN (*Transport Sequence Number* – Número de Sequência de Transporte). Os valores para o campo TSN possuem uma distribuição sequencial. Dessa forma, uma mensagem possui valor do campo TSN uma unidade maior que o valor do mesmo campo da mensagem anteriormente marcada. O primeiro valor TSN, utilizado como base para a marcação das mensagens, é estabelecido aleatoriamente no início da associação SCTP.

O campo NÚMERO DO FLUXO contém a identificação do fluxo associado à mensagem. Já o campo NÚMERO DE SEQUÊNCIA NO FLUXO indica a posição da mensagem relativa ao seu fluxo associado. Os fragmentos das mensagens devem ter o mesmo valor para NÚMERO DE SEQUÊNCIA DE FLUXO e NÚMERO DE FLUXO, o que permite a associação dos fragmentos como partes da mesma mensagem. Para se saber a ordem em que os fragmentos devem ser considerados para remontagem, é preciso utilizar o campo TSN, uma vez que toda mensagem, fragmentada ou não, tem um valor TSN único.

O campo IDENTIFICADOR DE PROTOCOLO representa um identificador específico de um protocolo de aplicação. Esse valor não é utilizado pelo SCTP, porém, pode ser usado por algumas entidades de rede, como servidores *proxies*, para identificar o protocolo usuário do SCTP.

Caso os dados de usuário não sejam múltiplos de quatro octetos, deve haver preenchimento, com octetos formados por bits zero, ao final da mensagem. É preciso notar que os octetos de preenchimento não são inseridos no cálculo total do comprimento do bloco. Dessa forma, permite-se a separação dos dados de usuário dos valores de preenchimento, no processo de desmontagem das mensagens contidas nos pacotes SCTP.

A mensagem de controle INIT é responsável pelo início do processo de estabelecimento de associações SCTP. Ela é a primeira mensagem enviada numa associação, proveniente de um cliente e destinada a um servidor.

A mensagem de controle SACK é enviada para reconhecer a correta recepção de dados de usuários. Além disso, essa mensagem indica também o valor atual da *janela deslizante*. Realizam-se, assim, controle de erro e de fluxo com uma mesma mensagem de controle.

No procedimento de notificação por mensagens SACK, a falta de mensagens de dados de usuários, que deveriam ter sido recebidas, é informada à origem dessas mensagens. Através dessa notificação, é requisitada a retransmissão das mensagens de usuário faltosas. Deve-se lembrar que a eventual retransmissão de mensagens de dados de usuário segue as regras de escopo de ordenação dos fluxos da associação. A Figura 5 apresenta o formato da mensagem de controle SACK. Assim como ocorre com as mensagens de controle INIT e INIT ACK, os FLAGS DE CONTROLE não são utilizados nas mensagens SACK.

O campo TSN CUMULATIVO desempenha um importante papel na tarefa de requisição de retransmissão de mensagens de dados faltosas. Esse campo indica que todas as mensagens de usuário com valor TSN menor ou igual ao valor contido no campo TSN CUMULATIVO das mensagens SACK foram corretamente recebidas. Nesse contexto, deve-se lembrar que as mensagens de controle não sofrem qualquer tipo de controle de erro por parte dos campos TSN. O seu reconhecimento é realizado por mensagens de controle específicas, como as mensagens INIT ACK, que além de passarem informações para o estabelecimento de uma associação SCTP, reconhecem o correto recebimento de mensagens INIT.

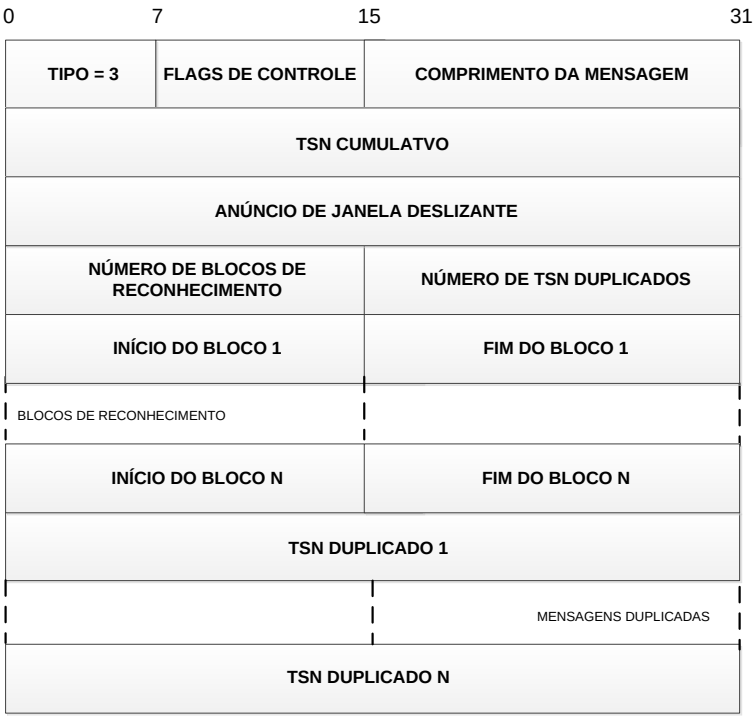


Figura 5. Mensagem de controle SACK.

Nas mensagens INIT e INIT ACK existe um valor que indica o tamanho inicial da *janela deslizante* a ser utilizada no controle de fluxo. Contudo, para esse controle, é utilizado o valor do campo ANÚNCIO DE JANELA DESLIZANTE, presente nas mensagens SACK. As mensagens SACK são constantemente enviadas em uma associação SCTP para indicar a correta recepção de mensagens de dados de usuário. Portanto, a atualização da janela deslizante ocorre de forma eficiente através do uso desse tipo de mensagem. As mensagens INIT E INIT ACK são enviadas apenas no estabelecimento da associação.

As mensagens SACK informam as mensagens de usuário que foram corretamente recebidas, utilizando, para essa finalidade, o campo TSN CUMULATIVO. Além disso, as mensagens SACK podem informar intervalos de mensagens faltosas numa sequência de mensagens recebidas. A idéia é evitar a retransmissão de mensagens corretamente recebidas devido à perda de mensagens anteriormente enviadas, como ocorre com o mecanismo padrão de retransmissão do TCP.

Para indicar o número de intervalos de mensagens faltosas na sequência de mensagens recebidas, utiliza-se o campo NÚMERO DE BLOCOS DE RECONHECIMENTO. O número de intervalos de mensagens corretamente recebidas, com TSN superior ao TSN CUMULATIVO, e, portanto, após a ocorrência de algum intervalo de mensagens

faltosas, é indicado por esse campo. Todas as mensagens de dados com TSN maior ou igual ao campo TSN CUMULATIVO adicionado ao INÍCIO DO BLOCO N, e menor ou igual ao TSN CUMULATIVO adicionado ao FIM DO BLOCO N, foram recebidas corretamente.

A mensagem de controle ABORT é utilizada para encerrar uma associação de forma não-coordenada, onde dados de usuário transmitidos numa associação podem ser perdidos e não recebidos pela estação destino.

A mensagem de controle SHUTDOWN é utilizada para encerrar uma associação de forma "coordenada". Por coordenada entende-se que procedimentos especiais são executados antes do encerramento da associação, evitando-se, por exemplo, perdas não desejadas de dados.

A mensagem SHUTDOWN ACK é utilizada para indicar o correto recebimento de uma mensagem de controle SHUTDOWN. Em termos práticos, esse tipo de mensagem indica, além de recebimento correto de um shutdown, que a estação emissora de um SHUTDOWN ACK está pronta para o encerramento da associação e que o encerramento da associação não resultará em perda de dados não esperada.

Após o recebimento de uma mensagem SHUTDOWN ACK, devem-se executar dois procedimentos. Primeiro, é necessário enviar a mensagem de controle SHUTDOWN COMPLETE, para indicar, ao par da associação, que essa estação encerrou sua participação na associação e que, agora, a estação receptora dessa mensagem deve fazer o mesmo. Em seguida, após o envio dessa mensagem, todas as variáveis e posições de memória alocadas (*buffer*) para a associação em questão são descartadas por essa estação.

A mensagem de controle ERROR é utilizada para informar a ocorrência de erros em algum processamento na associação. A ocorrência de erros é notificada pelos campos *indicação de erro*. Valores para o campo *tipo*, também chamado de *código de erro*, são visualizados na Tabela 4.

Tabela 4. Tipos de indicação de erro.

Código de Erro	Interpretação
1	Identificador de fluxo inválido
2	Parâmetro obrigatório faltando
3	Erro de cookie
4	Sem recursos de processamento
5	Endereço não-resolvível
6	Tipo de mensagem não reconhecido

7	Parâmetro obrigatório inválido
8	Parâmetro irreconhecível
9	Nenhum dado numa mensagem de usuário
10	Cookie recebido enquanto *shutting down*

3.1.2 Estabelecimento de associações SCTP

O estabelecimento de uma associação SCTP é realizado pela troca de quatro mensagens, num processo conhecido como *four-way handshake*. Nesse procedimento, a parte passiva da comunicação, o servidor, apenas aloca recursos após o recebimento da terceira mensagem do *handshake*, devendo ela ser validada corretamente. Dessa forma, assegura-se que uma requisição de abertura de associação origina-se da parte correta, evitando ataques de seguranças ao SCTP, que são comuns ao TCP.

Apenas como comparação, o estabelecimento de conexões TCP ocorre por um processo de troca de três mensagens, conhecido como *three-way handshake*. O TCP aloca recursos no recebimento da primeira e segunda mensagem do *handshake*, o que resulta numa alta sensibilidade a ataques do tipo DoS (*Denial of Service* – Negação de Serviço).

A Figura 6 apresenta a troca de mensagens para o estabelecimento de uma associação SCTP. A estação que envia um pedido INIT é considerada como *cliente*, enquanto o emissor de uma mensagem de controle INIT ACK é considerado como *servidor*. Para que a associação esteja corretamente estabelecida, as duas pontas da comunicação devem alocar todos os recursos necessários. Portanto, apenas após o envio e recebimento da quarta mensagem de controle, a associação estará completamente definida, embora mensagens de usuário sejam enviadas no mesmo pacote SCTP que contenha mensagens de controle COOKIE ECHO e COOKIE ACK.

Mensagens de dados de usuário estão contidas no mesmo pacote que encapsule mensagens de controle COOKIE ECHO e COOKIE ACK, pois quando uma estação recebe uma dessas mensagens, os recursos necessários à correta operação da associação são alocados pela estação. Com os recursos devidamente alocados, as mensagens de usuário podem ser devidamente processadas sem qualquer problema. É claro que, se a mensagem COOKIE ACK não é recebida e devidamente processada pela estação remota da associação, esta última não é corretamente estabelecida e os dados de usuário recebidos na terceira e quarta mensagem do *handshake* são devem ser considerados.

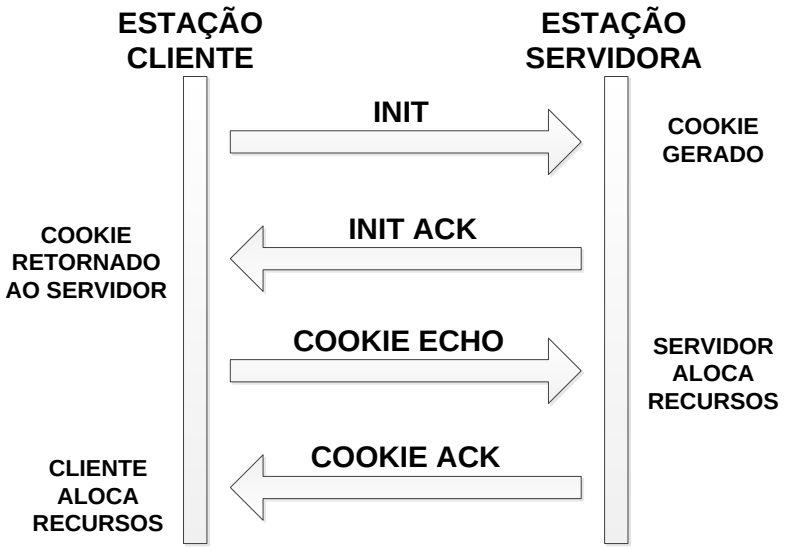


Figura 6. Troca de mensagens para o estabelecimento de uma associação.

Na análise do estabelecimento de associações SCTP, podem-se identificar dois comportamentos básicos: o do *cliente* e o do *servidor*. O SCTP define os estados lógicos possíveis que clientes e servidores podem ter comunicações que utilizam esse protocolo. Os estados possíveis são diferentes para esses dois papéis que as estações podem ter em uma associação.

No lado do servidor, o terminal está no estado *closed* quando recebe um pedido de estabelecimento de associação. A mensagem utilizada para o cliente requisitar ao servidor a abertura de uma associação é a mensagem de controle INIT. A partir de então, o servidor gera todas as variáveis que ele precisa para estabelecer uma associação (e alocar recursos), criando, em seguida, uma tabela *hash* segura para referenciar essas variáveis com uma chave de segurança. Esses valores são, então, colocados em um *cookie*, juntamente com um valor de autenticação (conhecida como MAC). O *cookie* criado pelo servidor é retornado ao emissor do INIT numa mensagem INIT ACK, utilizando um parâmetro específico contido nessa mensagem. Até então, o servidor permanece no estado *closed*. Para garantir que a mensagem enviada chegue ao destino, um *temporizador* pode ser utilizado.

A transmissão de mensagens de controle SCTP utiliza temporizadores para assegurar a retransmissão de mensagens não recebidas. Quando uma mensagem de controle é transmitida, um contador de tempo (temporizador) é iniciado. Caso o limite de tempo termine e nenhuma mensagem de controle de resposta seja recebida, a

mensagem não reconhecida é reenviada. Contudo, se uma resposta apropriada chegar em tempo hábil, a contagem do temporizador é encerrada e nenhuma ação de retransmissão é tomada. Para evitar retransmissões indefinidas, o protocolo SCTP considera um limite máximo de retransmissões para mensagens de controle perdidas; quando o limite é esgotado, o SCTP considera o caminho *primário* como não-alcançável, devendo-se iniciar os procedimentos para utilização dos endereços alternativos (caso existam) ou para encerramento da associação.

Ao receber uma mensagem COOKIE ECHO, contendo uma variável *cookie* como parâmetro, o servidor verifica o valor da variável MAC para certificar a validade da mensagem recebida. A idéia dessa verificação é assegurar que o *cookie* recebido é realmente o que foi criado pelo servidor.

Caso o valor da variável MAC seja equivalente ao valor esperado, o *cookie* presente na mensagem COOKIE ECHO é o *cookie* que o servidor criou previamente. A partir dos valores presentes nesse *cookie*, o servidor aloca os recursos necessários para a associação SCTP. Em seguida, o servidor envia uma mensagem COOKIE ACK ao cliente, entrando no estado *established*. A partir desse ponto, o servidor está pronto para receber e emitir dados, porém, a associação ainda não está completamente definida, pois a parte cliente ainda não alocou os recursos necessários para a associação. Um diagrama com os estados lógicos possíveis para uma estação servidora numa associação é apresentado na Figura 7.

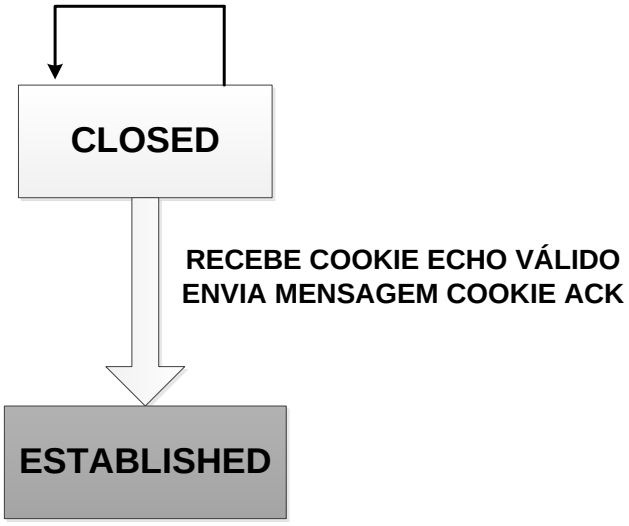
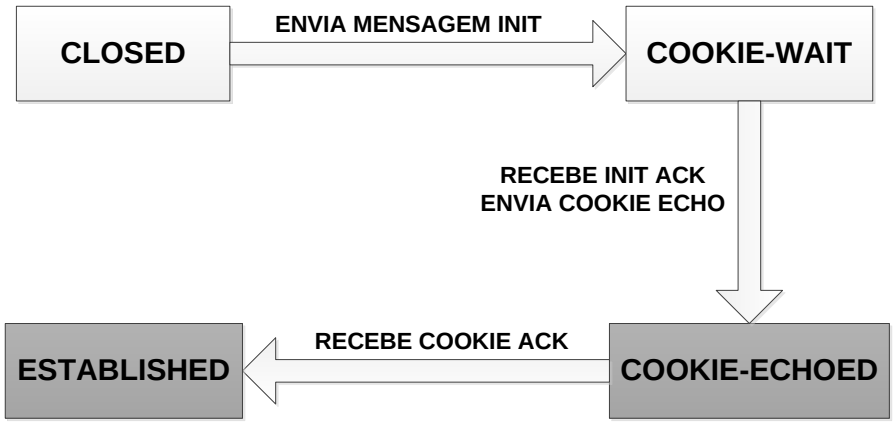


Figura 7. Estados lógicos para um servidor numa associação.

No lado cliente da associação, uma mensagem INIT é enviada para um endereço de transporte (endereço de rede e porta) do servidor. Em seguida, um temporizador é iniciado, o que garante retransmissão da mensagem INIT após um certo tempo sem recebimento de uma confirmação INIT ACK. Após o envio da mensagem INIT, o cliente entra no estado *cookie wait*.

Quando um cliente no estado *cookie wait* recebe uma mensagem de controle INIT ACK do servidor, ele interrompe o temporizador e monta uma mensagem COOKIE ECHO a partir dos dados (inclusive o *cookie*) retornados na mensagem INIT ACK. Após o envio da mensagem COOKIE ECHO previamente montada, um temporizador é novamente iniciado, o que garante retransmissão dessa mensagem caso nenhum COOKIE ACK seja recebido pelo cliente. Agora, o cliente entra no estado *cookie echoed*. Após o recebimento de uma mensagem de controle COOKIE ACK do servidor, o cliente entra no estado *established*, no qual a associação, agora, está completamente definida. Os estados lógicos possíveis para um cliente numa associação são apresentados na Figura 8.



A fim de reduzir problemas de segurança, algumas mensagens de controle utilizadas para reconhecimento devem ser enviadas especificamente para o *endereço de origem* contido no *datagrama* IP que carrega uma mensagem de requisição. Por exemplo, quando um servidor receber uma mensagem INIT de um cliente, para iniciar uma associação SCTP, o servidor sempre envia uma resposta INIT ACK para o endereço de origem contido no cabeçalho do *datagrama* IP que encapsula a mensagem

INIT. Nesses casos, não se devem utilizar os endereços alternativos para transmitir essas mensagens, mesmo em caso de não-operabilidade do caminho *primário*.

3.1.3 Encerramento de associações SCTP

Quando uma estação (cliente ou servidor) SCTP se encontra em um estado que não é *closed*, pode-se fazer a requisição do encerramento da associação. Há dois tipos de abordagens no encerramento de uma associação: uma "coordenada", que garante a não perda de dados, e, outra, forçada, onde a associação é encerrada sem garantia em relação aos dados que estão sendo transmitidos ou mantidos em *buffers* da associação.

O encerramento coordenado de uma associação é baseado em mensagens SHUTDOWN. Nesse procedimento, um terminal (cliente ou servidor) que deseje encerrar uma comunicação pára de receber dados da camada de aplicação, e após a confirmação dos dados já enviados, transmite uma mensagem SHUTDOWN à outra parte. Um temporizador é utilizado para retransmitir mensagens SHUTDOWN sem respostas.

O receptor de uma mensagem SHUTDOWN responde com um SHUTDOWN ACK assim que todos os dados presentes nos seus *buffers* da associação tiverem sido confirmados. Nesse caso, um temporizador também é utilizado para garantir o correto envio dessa mensagem.

Quando a estação requisitante do encerramento da associação recebe um SHUTDOWN ACK, seu temporizador é interrompido e esse terminal envia um SHUTDOWN COMPLETE ao terminal na outra extremidade. Todos os dados pertencentes à associação são excluídos, entrando essa estação em estado *closed*.

Por fim, o receptor de um SHUTDOW COMPLETE remove todas as suas variáveis alocadas para a associação, entrando também no estado *closed*. A associação está, agora, completamente encerrada pelas duas estações. A Figura 9 apresenta a troca de mensagens envolvidas nesse processo. Nesse exemplo, o pedido de encerramento da associação parte da estação cliente, apesar de não ser obrigatória essa condição.

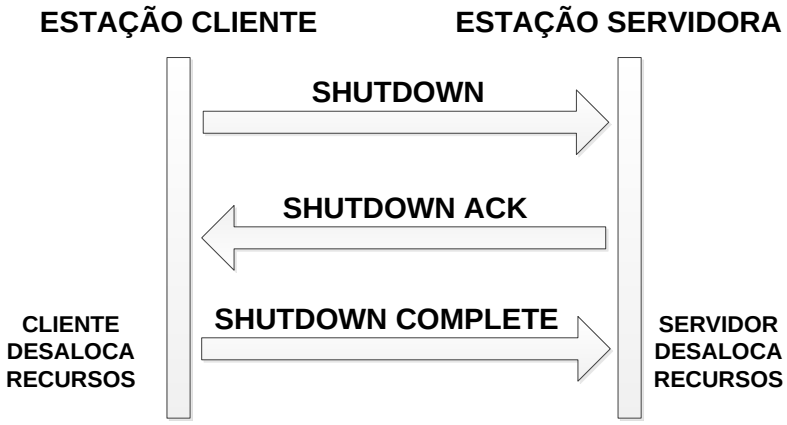


Figura 9. Encerramento coordenado de associações SCTP.

Para tornar o protocolo SCTP mais flexível em ambientes onde a confiabilidade dos dados não é requisito primário, como em comunicações em tempo real, foi especificada uma extensão denominada como PR-SCTP (*Partial Reliability SCTP* – SCTP Parcialmente Confiável) (STEWART, 2003). Nessa extensão, um terminal SCTP pode informar ao seu par da comunicação que mensagens de usuários perdidas e provenientes dessa estação não serão retransmitidas. As principais características utilizadas nesta proposta são o *multihoming* e o CMT (*Concurrent Multipath Transfer*). Seus funcionamentos serão detalhados na seção a seguir.

3.2 Multihoming

A característica *multihoming* (múltiplos caminhos) possibilita que os sistemas que possuem múltiplas interfaces para redundância usem uma interface em lugar de outra, possibilitando maior estabilidade na comunicação caso haja problemas na rede, como desconexões. No SCTP, uma interface é estabelecida como primária e as outras se tornam secundárias. Se a primeira falhar por qualquer motivo, uma secundária é escolhida e utilizada. Quando a interface primária estiver disponível novamente, as comunicações podem ser transferidas de volta sem que o aplicativo sequer saiba que houve um problema. Por exemplo, em um terminal convencional, a falha de um acesso LAN (*Local Area Network*) pode isolar a estação destino, porém, com o uso de mais de um endereço IP numa associação SCTP, a comunicação será reforçada através de acessos redundantes.

O SCTP permite inclusive que uma mesma associação utilize simultaneamente caminhos IPv4 e IPv6 (COSTA, 2005). Também permite que o endereço seja informado para o participante remoto como um nome DNS (*Domain Name System).*

Os terminais SCTP trocam listas de seus endereços (primários e alternativos) durante o início da associação para suportar os múltiplos caminhos. Um terminal SCTP é capaz de fazer deduções, como considerar uma estação destino indisponível, em vez de considerar problemas na rede, caso todos os endereços IP (primários e alternativos) desse terminal remoto estejam inalcançáveis. A Figura 10 ilustra um típico par de remetente/receptor em um ambiente *multihoming* utilizando duas tecnologias de redes diferentes (por exemplo, Wi-Fi e Ethernet).

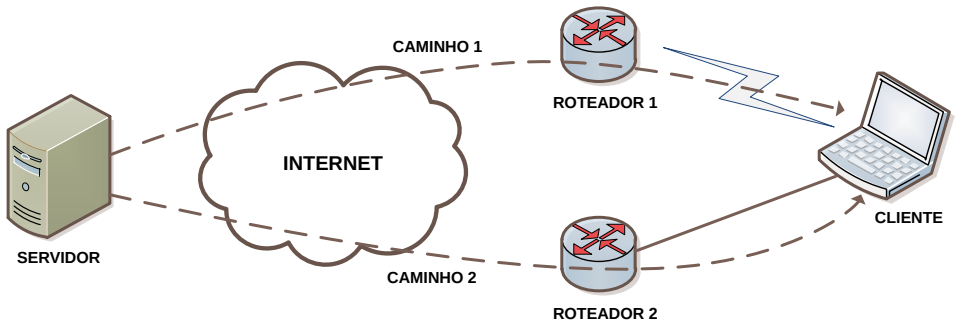


Figura 10. Um Cenário *Multihoming*.

A característica *multihoming* não foi projetada para realizar balanceamento de carga entre os caminhos redundantes. A função do *multihoming* é apenas fornecer redundância em caso de problemas na rede. Deste modo, foi necessário utilizar na proposta uma extensão denominada CMT (*Concurrent Multipath Transfer*) (IYENGAR, 2006). Suas funcionalidades serão detalhadas na seção a seguir.

3.3 Concurrent Multipath Transfer (CMT)

De acordo com a RFC 4960, a transferência de dados normalmente utiliza apenas o caminho primário para transmissão e os caminhos restantes são utilizados como *backup*. Utilizando apenas um caminho e mantendo as outras rotas ociosas não é possível obter a melhor utilização dos recursos da rede. O problema da subutilização em qualquer rede pode ser resolvido através da distribuição de carga. Deste modo, é desejável que haja compartilhamento de carga entre todos os caminhos disponíveis para alcançar os benefícios de uma melhor utilização da largura de banda da rede.

O CMT (*Concurrent Multipath Transfer*) (IYENGAR, 2006) é uma das extensões para o SCTP, que foi proposta para alcançar balanceamento de carga fim-a-fim através de vários caminhos disponíveis. Deste modo, pode oferecer suporte completo ao *multihoming* com a utilização simultânea de todos os caminhos disponíveis, atingindo o compartilhamento de carga e aumentando a produtividade de um aplicativo.

Embora o CMT proporcione uma melhor utilização dos recursos de rede, ele falha em relação ao desempenho. Assim, as técnicas de otimização são cada vez mais necessárias para obter um bom desempenho em qualquer rede. Tais técnicas incluem ajustes da rede e modificações no modo de funcionamento. A abordagem CMT resultou em um melhor desempenho em caminhos homogêneos, enquanto que para os caminhos heterogêneos com diferentes larguras de banda e/ou redes de acesso sem fio, não há garantias na sua eficácia. O fraco desempenho observado está relacionado com as diferenças na largura de banda, capacidade de diferentes enlaces e mecanismos de detecção de perdas e recuperação (NATARAJAN, 2008), (DREIBHOLZ, 2011).

Atualmente, alguns problemas foram encontrados durante a utilização do SCTP-CMT para compartilhamento de carga sobre caminhos assimétricos, o que resulta na taxa de transferência reduzida. Em (DREIBHOLZ, 2010), os autores propuseram soluções através de técnicas de otimização para melhorar esse rendimento.

A crescente disponibilidade de diferentes tecnologias de acesso sem fio oferece a oportunidade para a distribuição, em tempo real, de conteúdo multimídia usando o mecanismo de transferência paralela em particular. Atraso fim-a-fim e perda de pacotes são requisitos vitais de QoS e QoE para aplicações multimídia. Deste modo, este trabalho apresenta uma proposta de balanceamento de carga baseada nas condições da rede. O detalhamento da proposta será abordado no capítulo a seguir.

4. Avaliação dos Protocolos SCTP e CMT

Uma Rede de Petri é uma abstração de um sistema real. Ela é um modelo formal do fluxo de dados do sistema modelado em questão. As propriedades, conceitos e técnicas para modelagem de uma Rede de Petri foram desenvolvidos utilizando métodos simples e poderosos para descrição, análise do fluxo de dados e controle de sistema.

O formalismo de Redes de Petri é utilizado principalmente em sistemas que possam apresentar atividades assíncronas e concorrentes. Redes de Petri têm sido utilizadas principalmente para a modelagem de sistemas de eventos em que estes possam ocorrer concorrentemente, havendo obstáculos na concorrência, precedência ou frequência desses eventos (PETERSON, 1977).

A estrutura de uma Rede de Petri é um grafo bipartido que compreende um conjunto de lugares, um conjunto de transições e um conjunto de arcos direcionados. Os lugares são representados graficamente por círculos, as transições por barras e os arcos direcionados por setas. Os arcos direcionados conectam os lugares às transições e as transições aos lugares.

Uma Rede de Petri pode ser considerada marcada quando ela possuir *tokens* ou marcas. *Tokens* encontram-se nos lugares. As transições, quando disparadas, consomem *tokens* dos lugares que as alimentam e geram marcas nos lugares por elas alimentadas. *Tokens* são sinais em uma Rede de Petri representados graficamente por um ponto preto. A quantidade e a posição dos *tokens* em uma Rede de Petri podem variar durante o funcionamento da mesma.

Uma Rede de Petri é executada através do disparo de transições. Para ocorrer o disparo de uma transição, é necessário que esta transição esteja habilitada para isto. Uma transição é considerada habilitada para disparar quando todos os lugares de entrada dela contiverem pelo menos um *token*.

Usando o formalismo de Redes de Petri, é possível descrever somente a estrutura lógica de sistemas, pois tal formalismo não inclui nenhum conceito de tempo. Entretanto, frequentemente, o conceito de tempo tem um papel importante na descrição do comportamento de sistemas. A introdução do conceito de tempo no formalismo de

Redes de Petri permite a descrição de um comportamento dinâmico de sistemas (MARSAN, 1986).

O formalismo de Redes de Petri Temporizadas tem como sua principal característica a associação de um atraso fixo para cada transição do modelo. Redes de Petri Temporizadas foram o passo inicial para a criação do formalismo de Redes de Petri Estocásticas (SPN). A possibilidade de unir a habilidade do formalismo de Redes de Petri para descrever sincronização e concorrência com um modelo estocástico é o principal atrativo para obter-se uma avaliação quantitativa de sistemas computacionais complexos. As Redes de Petri Estocásticas são obtidas através da associação de um tempo distribuído exponencialmente com o disparo de cada transição da rede (BALBO, 1994).

Dada uma Rede de Petri Estocástica com uma marcação que possua diversas transições habilitadas a serem disparas, uma das transições ocorre. Quando uma transição de uma Rede de Petri Estocástica é disparada, assim como no formalismo de Redes de Petri, uma nova marcação pode ser gerada. Esta nova marcação pode conter transições que já encontravam-se habilitadas na marcação anterior, mas não foram disparadas.

Uma característica importante do formalismo de Redes de Petri Estocásticas é que ele pode ser facilmente compreendido por pessoas que não tenham familiaridade com métodos de modelagem probabilística. Entretanto, a representação gráfica de sistemas utilizando o formalismo de Redes de Petri Estocásticas encontra-se limitada à medida que aumenta-se o tamanho e a complexidade do sistema em questão. Além disso, o número de estados da Cadeia de Markov equivalente à Rede de Petri cresce muito rapidamente conforme a dimensão do gráfico cresce. Portanto, o formalismo de Redes de Petri Estocásticas pode ser usado para modelar somente sistemas de tamanho limitado.

Geralmente, não é desejável associar uma variável aleatória distribuída exponencialmente para cada transição do modelo, conforme descrito no formalismo de Redes de Petri Estocásticas. O formalismo de Redes de Petri Estocásticas Generalizadas (GSPN) associa tempo somente para alguns eventos que acredita-se ter um grande impacto na avaliação do sistema.

Um típico exemplo disso pode ser o caso em que a sequência de operações de um sistema compreende atividades cujas durações diferem bruscamente. Logo, é conveniente que atividades de duração desprezível sejam modeladas somente do ponto de vista lógico, enquanto que atividades de duração mais longa seriam associadas a uma variável aleatória. Através desta modelagem, reduz-se o número de estados da Cadeia de Markov equivalente ao modelo, consequentemente reduzindo a complexidade da solução do mesmo. Além disso, a utilidade de uma estrutura lógica que pode ser usada em conjunto com uma estrutura temporizada permite a construção de um modelo compacto de sistemas computacionais complexos (MARSAN, 1986).

O formalismo de Redes de Petri Estocásticas Generalizadas (MARSAN, 1984) permite duas classes diferentes de transições no modelo: transições imediatas e transições temporizadas. A transição imediata dispara em tempo zero assim que encontra-se habilitada. A transição temporizada, assim como no formalismo SPN, dispara após um tempo aleatório, distribuído exponencialmente, associado à mesma quando habilitada.

As transições temporizadas são representadas graficamente no modelo por barras brancas, e as transições imediatas por barras pretas. Obviamente, as taxas de disparos estão associadas somente às transições temporizadas, e estas podem ser dependentes da marcação da GSPN.

Deste modo, para mostrar os pontos fortes e fracos do SCTP-CMT em relação ao SCTP foi realizada uma comparação entre os dois esquemas utilizando o formalismo de Redes de Petri. O resultado desta comparação ajudou na formulação e desenvolvimento de uma nova proposta para o CMT em ambiente de redes heterogêneas (IEEE 802.11/Wi-Fi, IEEE 802.16/WiMAX e 3G/UMTS). As modelagens das Redes de Petri foram realizadas utilizando o *software* TimeNET (TimeNET 2011).

TimeNET é uma ferramenta de modelagem e simulação de sistemas que utiliza o formalismo de Redes de Petri Estocásticas, disponibilizada gratuitamente, e está sendo utilizada em diversas instituições de pesquisa para modelagens dos mais diversos sistemas.

Assim como muitas ferramentas de modelagem de Redes de Petri, o TimeNET foi desenvolvido utilizando uma estrutura composta por módulos, que são divididos basicamente em: Módulo de Edição, de Simulação e de Análise de Desempenho.

O *software* de modelagem TimeNET e consequentemente a utilização da extensão de Rede de Petri Estocástica, fornecem maiores recursos de modelagem, como a utilização de transições temporizadas capazes de modelar e produzir atrasos em seus disparos (GERMAN, 1995). As transições temporizadas são classificadas como: exponenciais ou determinísticas. Os atrasos nas transições exponenciais muitas vezes não representam a realidade em muitos sistemas e podem dificultar as análises de desempenho do sistema que está sendo modelado. As transições determinísticas são utilizadas para modelar e simular operações com frequências definidas. Essas transições são bastante utilizadas em simulações de sistemas de comunicação onde o tempo e as frequências são atributos importantes para a simulação, principalmente para a realização das análises de desempenho.

A Figura 11 mostra o modelo GSPN (*Generalized Stochastic Petri Net*) (MARSAN, 1995) do SCTP padrão. O foco da modelagem está na característica *multihoming* do protocolo. Com isso, considera-se que as associações SCTP já foram estabelecidas e o protocolo está pronto para transmitir.

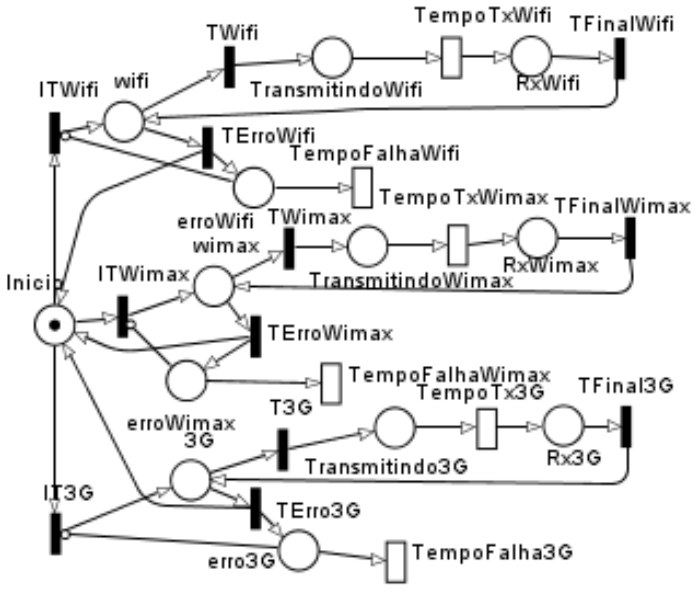


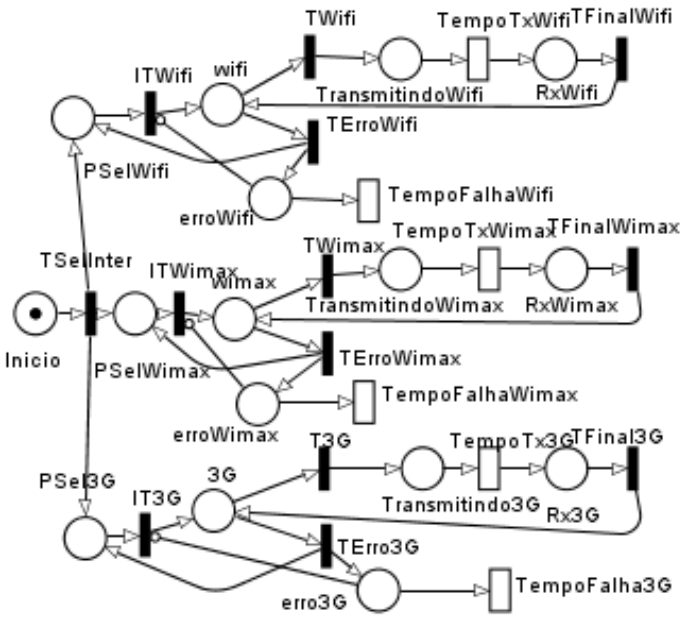
Figura 11. Modelo GSPN do SCTP padrão.

O lugar #Inicio indica o início da seleção de interface do protocolo. Um *token* presente neste lugar sugere que o protocolo deve decidir qual será a interface primária e quais serão as secundárias a serem consideradas na transmissão. É interessante lembrar que, no SCTP, apenas uma interface pode transmitir dados por vez. Enquanto uma interface está transmitindo (primária), as outras (secundárias) ficam em estado de espera. Após a escolha da interface primária, inicia-se o processo de transmissão de dados. Sempre que houver um *token* no lugar de uma interface (#wifi, #wimax, #3G) haverá um conflito, pois apenas uma interface pode estar ativa para transmissão.

O disparo da transição TWifi indica a operação normal de envio de pacotes para o receptor através da interface Wi-Fi, enquanto que a ativação de TErroWifi indica a ocorrência de um erro e a impossibilidade de transmissão de pacotes através da interface Wi-Fi.

Toda vez que ocorre um erro em uma das interfaces, esta não poderá ser mais escolhida por um dado período de tempo (simulando o pacote *Heartbeat* do SCTP). Esta operação é modelada pelo arco inibidor que se origina de um lugar de erro (#erroWifi, #erroWimax, #erro3G) para a transição que habilita a respectiva interface. A interface em estado de erro permanece assim por um período de tempo exponencialmente distribuído com média igual às variáveis TempoFalhaWifi, TempoFalhaWimax ou TempoFalha3G, dependendo da interface em questão. A rede irá transmitir por uma determinada interface até ocorrer um erro que a torne inoperante, como em caso de desconexões frequentes em redes sem fio.

O modelo da Figura 12 é semelhante ao da Figura 11. A diferença entre os modelos é a mesma diferença entre os protocolos SCTP padrão e CMT. Enquanto o SCTP habilita apenas uma interface para o tráfego de dados, o CMT utiliza todas as interfaces de um dispositivo para enviar dados em uma mesma associação. Como apresentado na Figura 12, após a transição de início, cada interface recebe um *token* de marcação. Desse modo, todas as interfaces ficam habilitadas para o tráfego de dados.



A Tabela 5 apresenta as métricas utilizadas no TimeNET para a avaliação dos modelos. Basicamente, têm-se duas métricas: a probabilidade do sistema estar transmitindo por uma das três interfaces e a probabilidade de uma interface entrar em estado de erro, ou seja, ficar inoperante.

Tabela 5. Métricas de avaliação no TimeNET

Métrica/Expressão (TimeNet)	Descrição
probWifi / P{#transmitindoWifi > 0}	Probabilidade da Interface Wi-Fi estar ativa e transmitindo.
probWimax / P{#transmitindoWimax >0}	Probabilidade da Interface Wimax estar ativa e transmitindo.
prob3G / P{#transmitindo3G > 0}	Probabilidade da Interface 3G estar ativa e transmitindo.
probErroWifi / P{#erroWifi > 0}	Probabilidade da interface Wi-Fi estar em estado de erro
probErroWimax / P{#erroWimax > 0}	Probabilidade da interface Wimax estar em estado de erro
probErro3G / P{#erro3G > 0}	Probabilidade da interface 3G estar em estado de erro

Para avaliar os dois protocolos na mesma condição, os atrasos (*delays*) das transições temporais foram configurados com valores médios iguais em ambos os modelos. Os tempos das transições “TempoFalhaWifi”, “TempoFalhaWimax” e “TempoFalha3G”, observados na Figura 3, foram definidos de acordo com uma

distribuição exponencial com média igual a 3s nos dois modelos (SCTP e CMT). O mesmo ocorreu com "TErroWifi", "TErroWimax" e "TErro3G", porém a média adotada foi igual a 1s. Neste experimento, o tempo que o protocolo leva transmitindo é de 20s, sendo o mesmo para as três interfaces. A Tabela 6 sumariza os valores configurados e obtidos empiricamente.

Tabela 6. Valores de tempo ou peso das transições.

Métrica	Valor
TempoTxWifi, TempoTxWimax, TempoTx3G	20.0s
TempoFalhaWifi, TempoFalhaWimax, TempoFalha3G	3.0s
TWifi, TWimax, T3G	0.9 (90%)
TErroWifi, TErroWimax, Terro3G	0.1 (10%)

Toda ocorrência de conflito em um modelo é resolvida através dos pesos associados às transições conflitantes. Com isso, os pesos das transições TWifi, TWimax e T3G são iguais entre si e definidos como 0.9. Já os pesos de TErroWifi, TErroWimax e TErro3G são todos iguais a 0.1. Deste modo, a probabilidade de ocorrer um erro em uma dada interface é de 10%, por outro lado o protocolo transmite normalmente em 90% dos casos.

Após as devidas configurações dos parâmetros no TimeNET, iniciou-se a fase de coleta e análise dos dados. A Tabela 7 mostra os resultados obtidos na simulação estacionária dos protocolos no TimeNET.

Tabela 7. Resultados da simulação no TimeNET.

	SCTP	CMT
probWifi	0,32293 (32,29%)	0,98447 (98,44%)
probWimax	0,32846 (32,84%)	0,93723 (93,72%)
prob3G	0,31066 (31,06%)	0,91541 (91,54%)
probErroWifi	0,0055 (0,55%)	0,01570 (1,57%)
probErroWimax	0,0056 (0,56%)	0,01644 (1,64%)
probErro3G	0,0057 (0,57%)	0,01607 (1,67%)

A Tabela 7 mostra que a probabilidade de uma interface estar transmitindo no CMT é bem maior que a mesma interface estar transmitindo no SCTP. Por outro lado, a probabilidade do sistema estar em estado de erro é maior no CMT em comparação com o SCTP. Estas conclusões mostram que apesar de o CMT fornecer maior vazão ele é mais sensível aos erros que ocorrem na rede.

Com o intuito de quantificar a comparação, a vazão total foi calculada a partir dos modelos utilizando os dados da Tabela 8 e a Equação 1. A vazão total é definida como o número de bytes trafegados durante um período de tempo. Esta é obtida através do produto do tempo de simulação com a soma das probabilidades de um *token* estar na marcação de uma tecnologia multiplicado pela vazão teórica máxima daquela tecnologia. Para apresentar os resultados na escala de GBytes, dividiu-se o resultado obtido pelo produto 8 x 1024, conforme denotado na Equação 1. Onde T é o tempo de simulação.

$$\begin{aligned} Vazão_{Total} = T * [& (P\{\#\text{transmitindoWifi} > 0\} * \text{vazãoWifi}) + \\ & (P\{\#\text{transmitindoWiMax} > 0\} * \text{vazãoWiMax}) + \\ & (P\{\#\text{transmitindo3G} > 0\} * \text{vazão3G})] * [\frac{1}{8*1024}] \end{aligned} \quad (1)$$

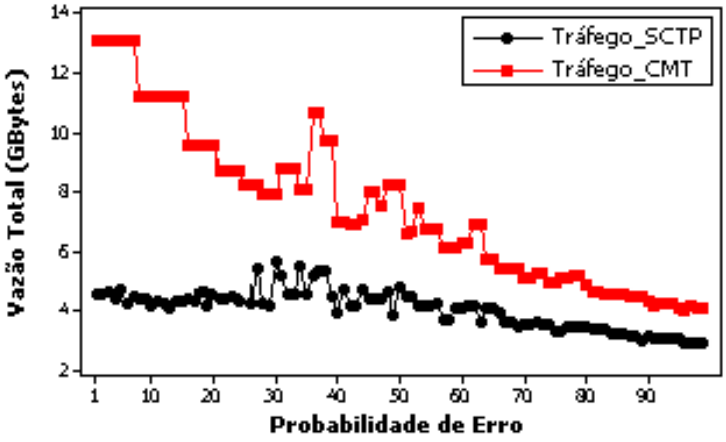
Além da vazão, utilizou-se para comparação de desempenho, entre os protocolos, a métrica de perda de pacotes. A Equação 2 denota o cálculo para a perda de pacotes e utiliza os parâmetros apresentados na Tabela 8.

Tabela 8. Valores para o cálculo da vazão total da rede.

Variáveis	Valores
vazaoWifi	54 Mbit/s
vazaoWimax	1 Gbit/s
vazao3G	10 Mbit/s
tempoSimulacao(T)	100 segundos

$$\begin{aligned} Perda_{\text{pacotes}} = T * [& (P\{\#erro\text{Wifi} > 0\} * \text{vazãoWifi}) + \\ & (P\{\#erroWiMax > 0\} * \text{vazãoWiMax}) + \\ & (P\{\#erro3G > 0\} * \text{vazão3G})] * [\frac{1}{8} * \frac{1}{1024}] \end{aligned} \quad (2)$$

É importante ressaltar que as simulações de desempenho foram feitas de forma estacionária para cada probabilidade de erro, variando de 1% a 50% através de configurações no TimeNET. Os resultados apresentados nas Figuras 13 e 14 evidenciam a média para cada simulação com 95% de nível de confiança.

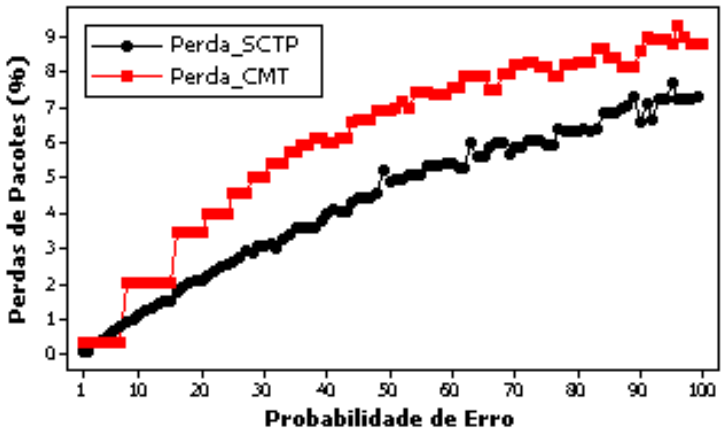


A Figura 13 mostra o total de tráfego do SCTP e do CMT quando se varia a probabilidade de erro nas interfaces (em termos percentuais). Nota-se, nesta figura, que inicialmente o volume total da rede no modelo CMT é bem maior (próximo de 13 GBytes) comparado com aproximadamente 4,3 GBytes do SCTP, representando uma superioridade, em termos de vazão total, do protocolo CMT sobre o SCTP.

Os conflitos de transições nas Redes de Petri são resolvidos pelos pesos destas transições. Para aumentar a probabilidade de erro, aumentaram-se os pesos das transições de erro nos modelos.

A Figura 14 apresenta os resultados das perdas de pacotes ocorridas na rede variando-se a probabilidade de erro. Nesta figura, percebe-se que quanto maior a probabilidade de erro, maior serão as perdas em ambos os protocolos. Nota-se que o desempenho do protocolo CMT é inferior ao SCTP. As perdas com o CMT são maiores, pois quanto maior o número de interfaces transmitindo simultaneamente, maior será a probabilidade de ocorrência de erros.

Deste modo, através da análise conjunta dos resultados apresentados nas Figuras 13 e 14 nota-se que apesar do CMT fornecer maior tráfego de dados, ele é mais sensível aos erros que ocorrem na rede se comparado com o SCTP. Este problema pode se tornar ainda mais evidente quando se trata de redes sem fio, onde ocorrem mais erros e perdas devido aos problemas inerentes ao caminho sem fio.



Com isso, conclui-se que o CMT precisa de políticas robustas de detecção, correção e prevenção de falhas, através de uma política inteligente de distribuição do tráfego. Um número de contribuições sólidas elevou a eficácia do CMT. No entanto, há uma necessidade de realizar pesquisar que envolvam a escolha do melhor caminho ou a possibilidade de realizar distribuição de carga, principalmente no âmbito de redes heterogêneas sem fio. Sendo assim, este trabalho propõe o aCMT-UC (Adaptive CMT – User Centric) para realizar distribuição de carga através de uma análise das condições dos enlaces existentes, bem como a priorização de *frames* de vídeos importantes.

5. A Proposta

Neste capítulo, será apresentada a proposta deste trabalho que visa a utilização de um Sistema *Fuzzy* para a composição de uma arquitetura de distribuição de carga dinâmica em ambiente de redes sem fio heterogêneas.

Para tornar o CMT um protocolo com melhor desempenho em redes sem fio heterogêneas, propõe-se que os caminhos de uma associação SCTP-CMT sejam monitorados e gerenciados por um arcabouço denominado CMT Adaptativo ou aCMT, definido a seguir. Para montar o arcabouço aCMT foi necessário incluir na proposta dois componentes: Sistema de Pesos e Sistema *Fuzzy*.

A função do Sistema de Pesos é diferenciar os vários caminhos de uma associação SCTP. Os pesos são responsáveis por rotular um determinado caminho fim-a-fim. Se o caminho tiver um peso alto, quer dizer que ele está com qualidade e é seguro para a transmissão, ao passo que se o peso está baixo infere-se que aquele caminho está impróprio para a comunicação. Desta forma, o Sistema de Pesos transforma o CMT em um protocolo adaptativo.

Por sua vez, o Sistema *Fuzzy* é utilizado para escolha dos pesos que serão atribuídos nos caminhos das associações SCTP. A vazão será proporcional aos pesos de cada caminho. A lógica *fuzzy* foi usada porque é importante que a escolha dos pesos seja feita através de um arcabouço inteligente, isso aumenta a robustez da proposta frente à instabilidade de cenários sem fio, além do que, a Lógica *Fuzzy* é ideal para tomada de decisões onde as variáveis podem assumir valores imprecisos.

No caso de tráfego em tempo real/multimídia, propõem-se neste trabalho o aCMT-UC que foca em tráfego de vídeo. A ideia por trás do aCMT-UC consiste na priorização do envio dos *frames* de vídeo por uma determinada interface sem fio, dependendo do tipo desse frame. As seções, a seguir, detalham o funcionamento de cada extensão proposta.

5.1 Sistema de Pesos do aCMT

Para diferenciar os caminhos de uma associação SCTP-CMT é atribuído o conceito de *peso* para cada caminho. A Figura 15 ilustra que para o CMT (padrão), as cargas são distribuídas igualmente entre os caminhos. Enquanto que no aCMT, o Sistema *Fuzzy* detecta que o caminho WiMAX, por exemplo, está mais congestionado que o caminho Wi-Fi, por este motivo, o peso da interface Wi-Fi é configurado, neste exemplo, 3 vezes maior (3N) que o da interface WiMAX (N), como mostrado na Figura 6. Deste modo, são distribuídos o triplo de carga pelo caminho Wi-Fi em relação ao caminho WiMAX.

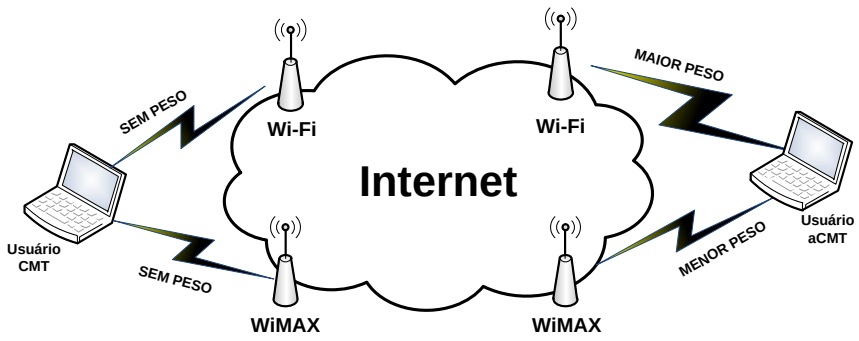


Figura 15. Comparação ilustrativa da comunicação *multihoming* utilizando CMT e aCMT.

Para definir o peso de cada caminho foi necessário implementar um Sistema Fuzzy, que avalia as condições da rede para definir quando o peso do caminho deve aumentar ou diminuir. Sendo assim, a seção, a seguir, detalha o funcionamento do Sistema *Fuzzy* em questão.

5.2 Sistema *Fuzzy* do aCMT

Diferentemente da lógica tradicional, que utiliza valores exatos, a Lógica *Fuzzy* é um sistema capaz de trabalhar com informações imprecisas e transformá-las em uma linguagem matemática de fácil implementação computacional. A Lógica *Fuzzy* é uma ferramenta muito utilizada para tomada de decisão possibilitando, assim, o balanceamento de carga em redes.

Um sistema fuzzy é constituído pela fuzzificação, que traduz as variáveis de entrada em conjuntos fuzzy, pela inferência, o qual realiza o raciocínio fuzzy com base em um sistema de regras relacionada às variáveis de entrada com as de saída, e pela

defuzzificação, que é a tradução da saída num valor numérico (ESPINOSA, 2004). A Figura 16 apresenta o esquema ilustrativo de um sistema fuzzy.

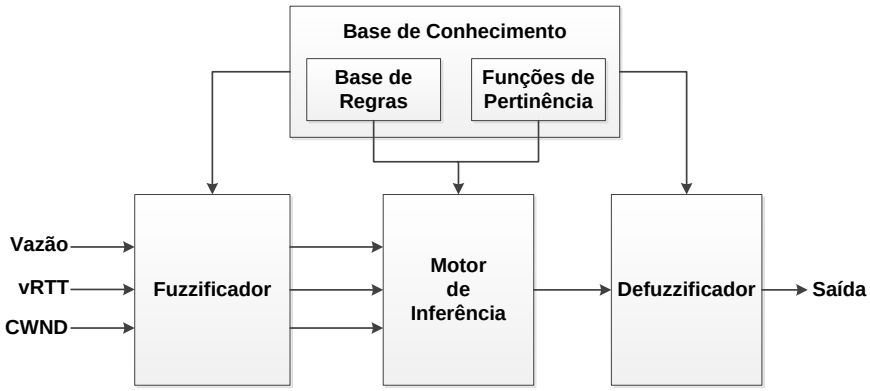


Figura 16. Representação do Sistema Fuzzy.

- **Entrada:** são os dados numéricos que irão alimentar o sistema. Através desses valores que o sistema tomará as decisões.
- **Fuzzificador:** Nessa etapa as entradas não *fuzzy* ou precisas são apresentadas ao sistema por intermédio de medições ou observações de dados, os quais são considerados como sendo o conjunto de dados de entrada do sistema. Deste modo, é necessário efetuar um mapeamento desses dados de entrada para o conjunto *fuzzy*, de tal forma, que o sistema possa identificar a quais variáveis linguísticas esses dados pertencem e o quanto os mesmos são pertinentes a essas variáveis. Nesta fase, também ocorre à ativação das regras *fuzzy* relevantes para um dado sistema;
- **Base de Conhecimento:** Formada por uma Base de Regras e de Dados (Funções de Pertinência). As regras podem ser fornecidas por especialistas, com base em seu conhecimento a respeito do processo que se deseja analisar, em forma de sentenças linguísticas, e se constituem em um aspecto fundamental no desempenho de um sistema de inferência *fuzzy*. Desta forma, o sistema de inferência *fuzzy* terá um desempenho confiável e satisfatório, somente se, as regras expressarem o comportamento do sistema de forma fiel e consistente. Tal método foi utilizado neste trabalho. Entretanto, a extração de um conjunto de regras advindas do conhecimento desses especialistas pode não ser uma tarefa fácil, por mais que os mesmos conheçam profundamente o problema que se deseja analisar. Portanto, existem alternativas ao uso do

conhecimento dos especialistas para a definição da base de regras, tais como os métodos de extração de regras a partir de dados numéricos. Esses métodos são particularmente úteis em aplicações onde haja disponível um conjunto de dados numéricos que refletem o comportamento entrada/saída do sistema;

- **Motor de Inferência:** No processo de inferência ocorrem as operações com os conjuntos *fuzzy*. Um aspecto importante é a definição dos conjuntos *fuzzy* correspondentes às variáveis de entrada e às de saída, pois o desempenho do sistema de inferência dependerá do número de conjuntos e de sua forma adotada. É possível efetuar uma sintonia manual das funções de pertinências dos conjuntos, mas é mais comum empregarem-se métodos automáticos. A integração entre sistemas de inferências *fuzzy* e redes neurais artificiais tem se mostrado adequadas para a sintonização das funções de pertinências, assim como para a geração automática de regras;
- **Defuzzificador:** Após o processo de Inferência, tem-se o processo de defuzzificação que, de posse do conjunto *fuzzy* de saída adquirido através do processo de inferência, é responsável pela interpretação dessa informação para saídas precisas (dados não *fuzzy*). Isto se faz necessário, já que, em aplicações práticas são requeridos valores não *fuzzy*.

Cada objeto tem um grau de pertinência em relação a um dado conjunto *fuzzy*, sendo este definido por uma função chamada de função de pertinência (3), ou seja:

$$\mu_A(x): X \rightarrow [0,1]; x \in X \tag{3}$$

onde $\mu_A(x)$ retorna o grau de pertinência do objeto x, pertencente ao universo de discurso X, em relação ao conjunto *fuzzy* A, sendo que o grau de pertinência é um valor normalizado pertencente (localizado) entre 0 (zero) e 1(um), onde tais valores limites indicam exclusão ou inclusão totais ao conjunto. Os principais tipos de função de pertinência são as triangulares, trapezoidais, gaussianas e sigmoides.

Um dos métodos de inferência *fuzzy* mais utilizados é o tipo Mamdani (MAMDANI, 1974), que aborda em cada regra o operador lógico "E" e agrega as regras por meio do operador lógico "OU" tendo como formato geral (4):

Se $X_1 \in A_1$ e $X_2 \in A_2$ e... $X_n \in A_n$ então $Y \in B_j$ (4)

onde, X_1,..., X_n são variáveis de entrada; A_i representa um dos conjuntos *fuzzy* definidos sobre o domínio da variável de entrada X_1; Y é a variável *fuzzy* de saída; B_j é um dos conjuntos *fuzzy* definidos sobre o domínio da variável Y.

É empregado nesta proposta um sistema *fuzzy* do tipo Mamdani cuja intensidade de disparo de uma regra de inferência será uma média aritmética dos graus de pertinência obtidos. Os conjuntos *fuzzy* são combinados pela escolha do valor máximo e o defuzzificador utiliza o chamado método centróide sem sobreposição para fornecer a resposta do sistema *fuzzy*, sendo este o mais utilizado dentre todos os métodos de defuzzificação por considerar toda a distribuição de possibilidade do conjunto *fuzzy* de saída (ORTEGA, 2001).

No formato dos conjuntos *fuzzy* sobre cada uma das variáveis de entrada, optou-se por usar funções de pertinência trapezoidais devido a sua fácil compreensão. Cada variável de entrada possui três conjuntos *fuzzy* definidos sobre ela. A função trapezoidal possui 4 parâmetros: a, b, m_1 e m_2. Sendo "a" o primeiro ponto e "b" o último ponto onde μ(x) é 0 (zero). Os parâmetros "m_1" e "m_2" representam o intervalo de pontos onde o μ(x) possui valor 1 (um), ou seja, se $x \in [m_1, m_2] \rightarrow \mu(x) = 1$. O grau de pertinência de uma função trapezoidal (Equação 5) é determinado por (ADELI, 2006):

$$\mu(x) = 0, \begin{cases} 0, se\ x\ \leq a \\ \frac{x-a}{m-a}, se\ x \in [a, m_1] \\ 1, se\ x \in [m_1, m_2] \\ \frac{b-x}{b-m}, se\ x \in [m_2, b] \\ 0, se\ x\ \geq b \end{cases} \quad (5)$$

Uma das variáveis de entrada utilizada no sistema *fuzzy* é o RTT (*Round Trip Time*), que é o tempo medido entre a transmissão de um segmento e a recepção de um *ack*. Normalmente, as medições de RTT são utilizadas como um indicador do estado interno da rede. Através da lógica fuzzy, o comportamento contínuo e impreciso das informações podem ser tratadas sem a necessidade de valores rígidos. Além disso, exige baixo processamento. Isto a torna bastante adequada para avaliar os valores de RTT, janela de congestionamento (cwnd) e vazão, onde imprecisão e incertezas são efetivamente presentes (CHENG, 2001).

Os valores de vRTT (Figura 17) e janela de congestionamento (Figura 18) não diferem entre as diferentes tecnologias, consequentemente, o conjunto de pertinência destas variáveis serve para todos os caminhos.

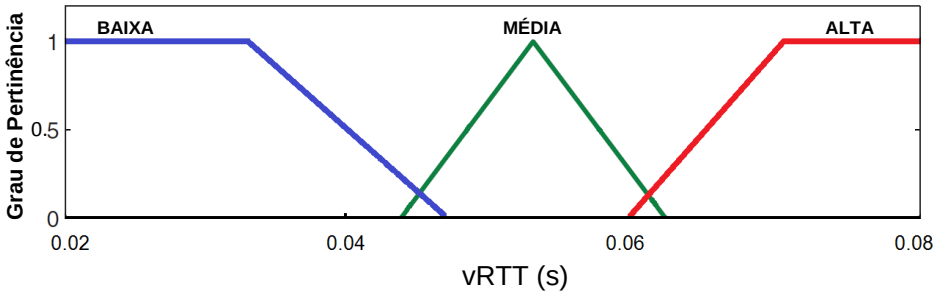


Figura 17. Grau de Pertinência para a entrada vRTT.

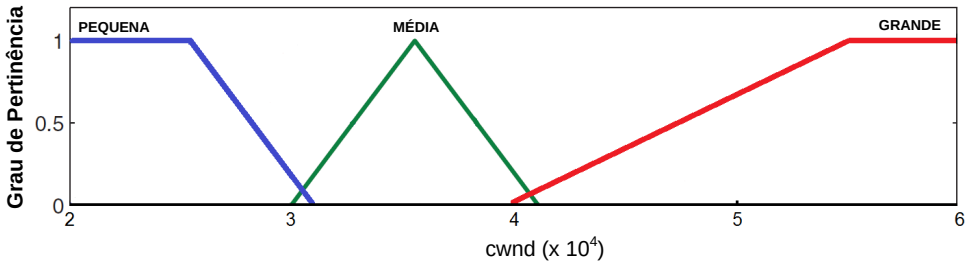


Figura 18. Grau de Pertinência para a entrada CWND.

No entanto, os valores de vazão são dependentes da tecnologia utilizada, com isso, houve a necessidade de criar conjuntos específicos para cada tecnologia utilizada no cenário de testes.

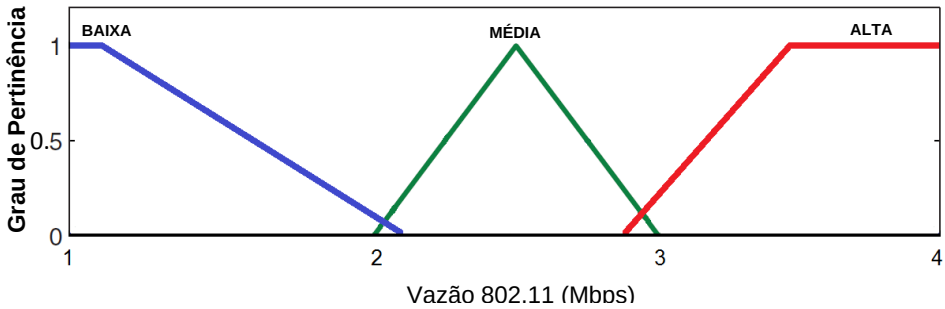


Figura 19. Grau de Pertinência para a vazão do 802.11.

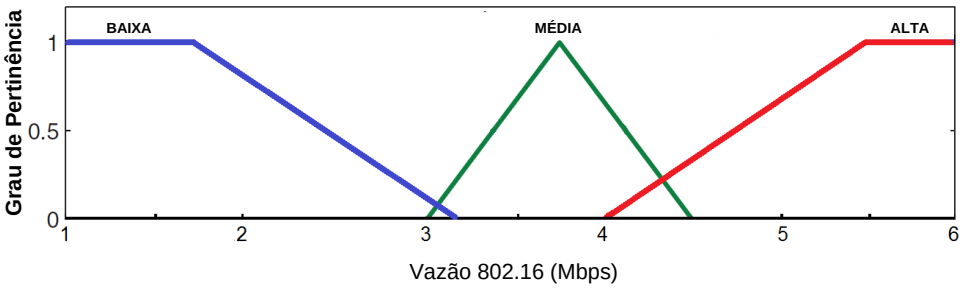


Figura 20. Grau de Pertinência para a vazão do 802.16.

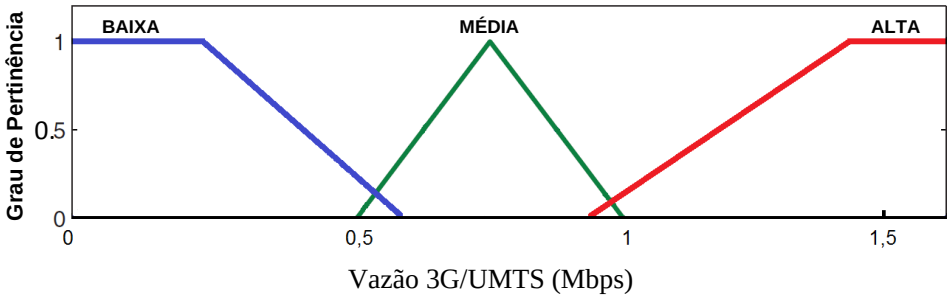


Figura 21. Grau de Pertinência para a vazão do 3G/UMTS.

A variável de saída (s), Figura 22, possui três conjuntos *fuzzy* que indicam como os pesos serão modificados. Em termos linguísticos, são definidos da seguinte maneira: DIMINUI (Ds), MANTÉM (Ms), AUMENTA (As), referindo-se ao peso do caminho aCMT.

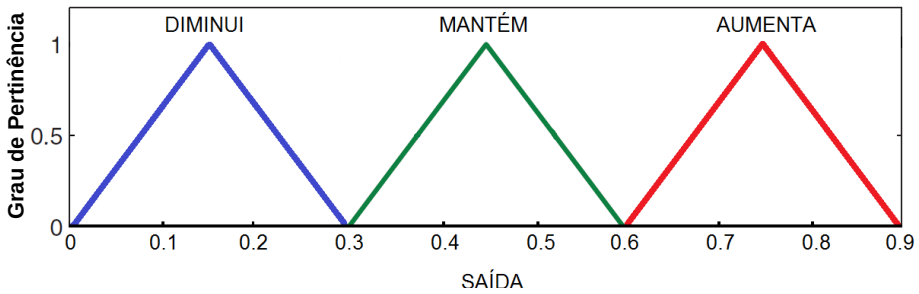


Figura 22. Saída do Sistema Fuzzy.

Optou-se por utilizar apenas estas três variáveis (vRTT, CWND e Vazão) porque quanto maior o número de variáveis mais complexo fica o sistema *fuzzy*. Se este sistema se tornar complexo, pode haver um grande *overhead* na rede, devido à demora no cálculo de atualização dos pesos, tornando o sistema inviável. Vale salientar que a

quantidade de regras é diretamente proporcional a quantidade de variáveis linguísticas e termos linguísticos. Se um sistema possuir três variáveis linguísticas, como vRTT, CWND e vazão, e três termos linguísticos, como baixa, média e alta, tem-se portanto 3^3 = 27 regras na base de conhecimento, como demonstrado na Tabela 9.

Tabela 9. Regras de Inferência.

Regra	Cwnd	vRTT	Vazão	Saída
1	PEQUENA	BAIXA	BAIXA	DIMINUI
2	PEQUENA	BAIXA	MÉDIA	MANTÉM
3	PEQUENA	BAIXA	ALTA	AUMENTA
4	PEQUENA	MÉDIA	BAIXA	DIMINUI
5	PEQUENA	MÉDIA	MÉDIA	DIMINUI
6	PEQUENA	MÉDIA	ALTA	AUMENTA
7	PEQUENA	ALTA	BAIXA	DIMINUI
8	PEQUENA	ALTA	MÉDIA	DIMINUI
9	PEQUENA	ALTA	ALTA	AUMENTA
10	MÉDIA	BAIXA	BAIXA	DIMINUI
11	MÉDIA	BAIXA	MÉDIA	AUMENTA
12	MÉDIA	BAIXA	ALTA	AUMENTA
13	MÉDIA	MÉDIA	BAIXA	DIMINUI
14	MÉDIA	MÉDIA	MÉDIA	AUMENTA
15	MÉDIA	MÉDIA	ALTA	AUMENTA
16	MÉDIA	ALTA	BAIXA	DIMINUI
17	MÉDIA	ALTA	MÉDIA	DIMINUI
18	MÉDIA	ALTA	ALTA	MANTÉM
19	GRANDE	BAIXA	BAIXA	DIMINUI
20	GRANDE	BAIXA	MÉDIA	AUMENTA
21	GRANDE	BAIXA	ALTA	AUMENTA
22	GRANDE	MÉDIA	BAIXA	DIMINUI
23	GRANDE	MÉDIA	MÉDIA	DIMINUI
24	GRANDE	MÉDIA	ALTA	AUMENTA
25	GRANDE	ALTA	BAIXA	DIMINUI
26	GRANDE	ALTA	MÉDIA	DIMINUI
27	GRANDE	ALTA	ALTA	AUMENTA

A Figura 23 mostra o fluxograma do funcionamento do CMT padrão, o protocolo inicia a transferência através de um dos caminhos selecionados e, enquanto existir dados, o CMT prossegue transmitindo, sempre selecionando outra interface para a transmissão no final de cada iteração. Deste modo, os dados são enviados através de todas as interfaces disponíveis de maneira equivalente.

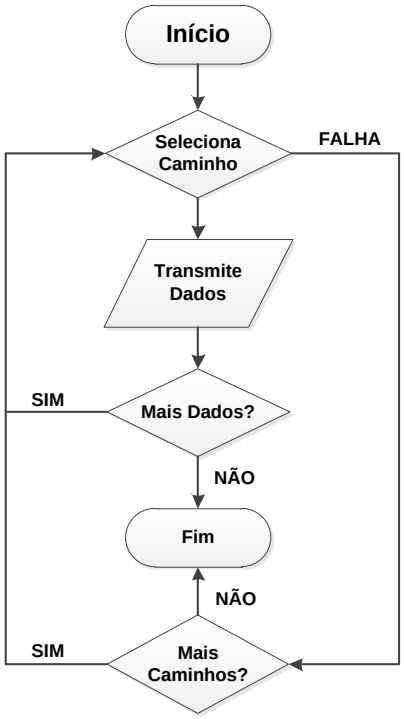


Figura 23. Fluxograma do CMT.

A Figura 24 exibe o fluxograma do protocolo aCMT, em tracejado estão destacados os módulos inseridos no CMT para torná-lo adaptativo. Após o início da transmissão, o sistema seleciona um caminho, e monitora, em um intervalo Δt (onde Δt = 1s), as variáveis vRTT (variação do RTT), vazão e janela de congestionamento (cwnd) atuais de cada caminho. Através de experimentos, observou-se que o sistema obteve melhor desempenho quando o caminho era monitorado em intervalos de 1 segundo.

Estas informações são inseridas no sistema fuzzy e os pesos são atualizados de acordo com a resposta do sistema. O número de pacotes (npkts) enviados é proporcional ao peso do caminho em questão, quanto maior o peso do caminho, mais pacotes serão enviados por este caminho.

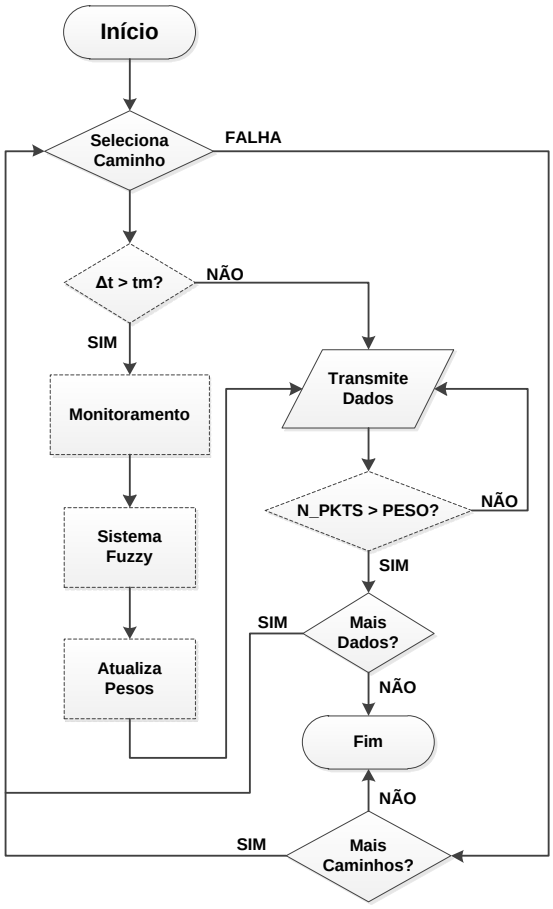


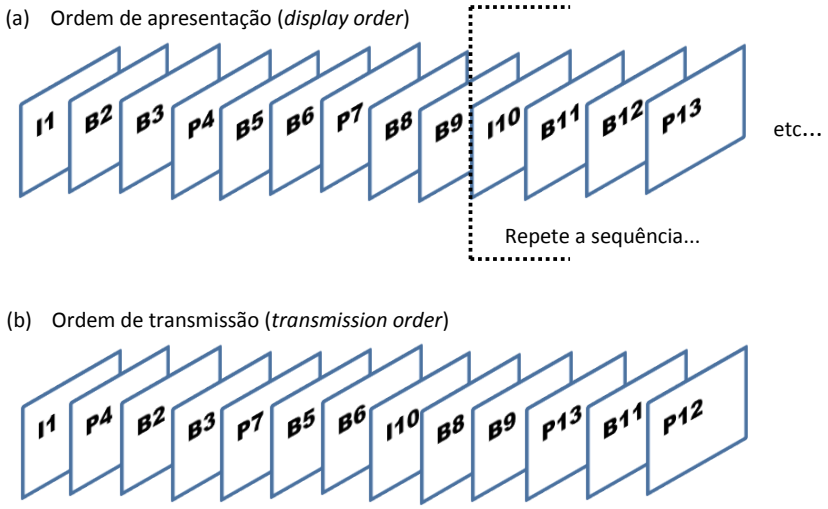
Figura 24. Fluxograma do aCMT.

5.3 aCMT-UC (Adaptive CMT – User Centric)

Uma vez que o aCMT considera apenas tráfego de melhor esforço, propõem-se neste trabalho o aCMT-UC que foca em tráfego de vídeo. A fim de atender as necessidades dos usuários e maximizar o seu QoE, a solução de comunicação perfeita tem de ser centrada no usuário (SCHUMACHER, 2010), deste modo, a ideia por trás do aCMT-UC consiste na priorização do envio do frame de vídeo por uma determinada interface sem fio, dependendo do tipo desse frame. O padrão MPEG (*Moving Picture Experts Group*) codifica a informação de uma cena em múltiplos grupos de imagens (GOP – *Group of Pictures*). O GOP é sempre iniciado por um quadro I (*Intra Coded Frame*), que é decodificado sem necessidade de informações contidas em outros quadros, seguido pelos quadros P (*Predictive-Frame*) e quadros B (*Bidirectional-Frame*) (WIEGAND, 2003).

Os quadros P, para serem decodificados, dependem das informações dos quadros I ou P anteriores mais próximos e os quadros B usam informações dos quadros P e I mais próximos, tanto os passados quanto os futuros, como referência para a decodificação da imagem. Devido a estas interdependências, a perda de um frame "I" resulta na perda do GOP inteiro, isto gera um impacto muito pior na qualidade do vídeo do que a perda de um frame "B" que não tem quadros dependentes.

Um típico GOP é ilustrado em sua ordem de apresentação (*display order*) na Figura 16(a). O mesmo GOP é apresentado na ordem de transmissão (*Transmission order*) na Figura 16(b). A estrutura de um GOP pode ser descrita por dois parâmetros: o **N**, número de imagens no GOP, ou seja, a distância entre dois *frames* I, e o **M**, número de imagens entre duas imagens P incluindo uma delas. A estrutura do GOP ilustrado na Figura 25 é descrita como **N** =9 e **M** = 3.



Em (GREENGRASS, 2009), os autores estudaram o impacto da perda de pacotes na QoE. O trabalho mostra que o descarte de pacotes que transportam quadros I pode resultar em distorções na imagem, sendo estas propagadas por todos os quadros ao longo do mesmo GOP. Deste modo, a qualidade do vídeo será recuperada apenas quando o decodificador receber um novo quadro I intacto.

No aCMT-UC, primeiramente é verificado qual tipo de pacote/frame será enviado (I ou P ou B), depois é escolhida a interface de acordo com o tipo de frame, ou seja, os *frames* mais importantes serão enviados pelo melhor caminho (Figura 26). O

mecanismo de envio de *frames* do aCMT-UC leva em consideração a importância de cada frame. O frame “I” é enviado pela interface de maior peso, o frame “P” pela segunda melhor interface e o frame “B” é enviado pela interface de menor peso (Apêndice A). Desta forma, a probabilidade de se perder um frame I e P são menores do que se perder um frame B.

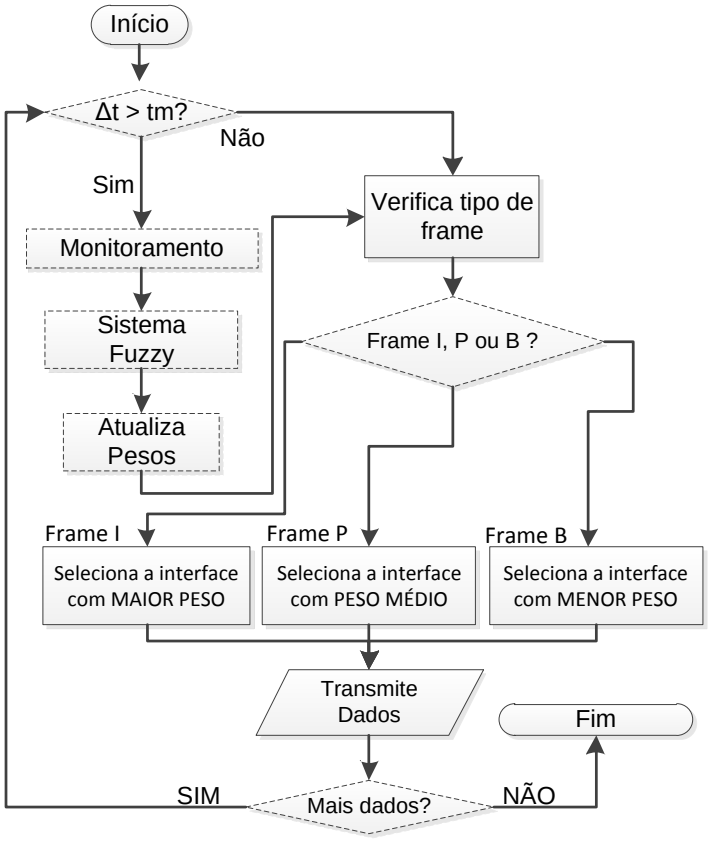


Figura 26. Fluxograma do aCMT-UC.

O capítulo, a seguir, apresenta a metodologia utilizada para avaliar a proposta, além de detalhar o cenário utilizado nas simulações. A proposta foi implementada no ns-2 e foi utilizado o *framework* EvalVid-aCMT-UC para gerar tráfego de vídeo na simulação.

6. Metodologia e Cenário de Avaliação

Neste capítulo, serão tratados os procedimentos necessários para implementar as propostas apresentadas conforme descritas no capítulo 3. Deste modo, serão descritos o cenário e as especificações para o funcionamento da proposta, além do cenário de avaliação. O capítulo inicia com a descrição do ambiente de simulação, ou seja, onde o sistema será validado. A seção seguinte, 6.2, descreve o tipo de tráfego utilizado para avaliar a proposta, bem como o *framework* utilizado para gerá-lo. Por fim, a seção 6.3 descreve as modificações necessárias para a implementação da proposta no ns-2.

6.1 O ambiente de simulação

No estudo das comunicações, a simulação aparece muitas vezes como a única ferramenta que se pode usar com relativa facilidade, dada a dificuldade em reunir equipamentos suficientes para experimentar laboratorialmente topologias complexas e determinados cenários hipotéticos de tráfego. No entanto, mesmo em cenários e topologias mais simples o seu uso é ainda atrativo quer pela facilidade na coleta de dados e geração de gráficos, quer também pela possibilidade de análise passo-a-passo das diferentes máquinas protocolares.

Em particular, o simulador de redes NS-2 (NS-2, 2011), juntamente com o animador NAM (*Networks Animator*) (NAM, 2011), constitui um conjunto adequado de ferramentas para o ensino de comunicações por computador. O NS-2 inclui praticamente todos os protocolos IP que são ensinados ao nível de uma licenciatura, bem como algumas versões experimentais de outros protocolos. Inclui ainda um bom suporte multicast e de QoS, nomeadamente DiffServ e IntServ, uma vez que é também o simulador mais popular entre os pesquisadores da área. Tem ainda a seu favor uma enorme comunidade de utilizadores e possui boa manutenção por parte da equipe de desenvolvimento. Implementado num misto de duas linguagens (C++ e Otcl), consegue um bom compromisso entre a rapidez dos executáveis e a versatilidade das linguagens de *scripting*.

O ns-2 é um interpretador de script Tcl orientado a objeto (OTcl). Esta biblioteca contém objetos de escalonamento de eventos, objetos de componentes de rede e módulos de ajuda de configuração de rede (CHUNG, 2003). Em outras palavras, para

utilizar o ns-2, se programa em Otcl. Para configurar e rodar um simulador de rede, o usuário deve escrever um script Otcl que inicia um escalonador de eventos, configura a topologia da rede utilizando os objetos de rede e funções das bibliotecas, e informam às fontes de tráfego quando devem começar a parar de transmitir pacotes através do armazenador de eventos.

Os objetos compilados são disponibilizados para o interpretador de Otcl por uma ligação Otcl que cria uma correspondência do objeto Otcl para cada um dos objetos C++. Também fazem com que as funções de controle e as variáveis de configuração especificadas pelo objeto C++ ajam como funções de membros e variáveis de membros de objetos Otcl correspondentes. Desta forma, os controles de objetos C++ são dados pelo Otcl. Isto também é possível para adicionar funções de membros e variáveis para o C++ ligados aos objetos Otcl. O objeto em C++ que não é necessário ser controlado na simulação ou internamente utilizado por outro objeto, não precisa ser ligado ao Otcl.

Quando a simulação é feita, o ns-2 produz um ou mais arquivos de saída baseados em texto que contém dados da simulação detalhados. Os dados podem ser utilizados para análise de simulações ou como entrada para uma ferramenta de simulação gráfica chamada *Network Animator* (NAM). O NAM tem uma interface de usuário gráfica bastante amigável e tem um controlador de velocidade. Pode ainda apresentar, graficamente, informações presentes como vazão e números de pacotes emitidos para cada *link*. Contudo as informações gráficas não podem ser utilizadas para análises profundas de simulações.

A ferramenta *TraceGraph* utiliza arquivos de saída, em formato de texto, para obtenção de inúmeras informações, tais como atraso, jitter, tempo de processamento, número de nós intermediários e estatísticas (TRACEGRAPH 2000).

A avaliação da proposta foi realizada no ns-2. Integramos o módulo SCTP/CMT desenvolvido pela Universidade de Delaware e o módulo *NIST Mobility* (NIST, 2011) em um novo *patch*. As tecnologias desenvolvidas para o ns-2, como os padrões IEEE 802.3, IEEE 802.11, IEEE 802.16 e 3G/UMTS, não foram projetadas para executar simultaneamente em um mesmo ambiente. O pacote *NIST Mobility* foi criado para resolver este problema e, deste modo, é possível simular ambientes heterogêneos que

utilizam diversas tecnologias de redes de acesso. Este módulo para o ns-2 possui as seguintes características:

- Suporte para descoberta de redes e modificações de endereços;
- Permite *handover* heterogêneo entre variadas tecnologias;
- Modifica a implementação padrão do 802.11 para suportar *handover*;
- E define um projeto genérico de nós com múltiplas interfaces.

Um nó com múltiplas interfaces consiste em um nó virtual que liga vários outros nós de tecnologias diferentes ou similares. É através dessa característica que é possível simular um nó SCTP-CMT com múltiplas interfaces, tornando viável a distribuição de carga através de diferentes tecnologias sem fio. Foi necessário, ainda, modificar o núcleo do ns-2 para inserir a funcionalidade do Sistema *Fuzzy* em uma simulação de redes heterogêneas.

Os experimentos visaram observar os benefícios da proposta de tornar o CMT adaptativo e melhorar a distribuição de carga nas associações SCTP/CMT em cenários heterogêneos. Os resultados coletados mediram o impacto da distribuição de carga sob o ponto de vista do usuário por meio de análise de parâmetros de QoS e QoE de vídeos reais, como vazão, PSNR (*Peak Signal to Noise Ratio*), SSIM (*Structural Similarity Index*), VQM (*Video Quality Metric*) e MOS (*Mean Opinion Score*). O caminho fim-a-fim é pré-estabelecido no inicio da simulação. O usuário só utiliza as interfaces que possuem cobertura.

A Figura 27 mostra o cenário utilizado nas simulações. Este cenário ilustra a sobreposição de redes heterogêneas composta pelas tecnologias 802.11 (Wi-Fi), 802.16 (WiMAX) e 3G/UMTS. Este ambiente será bastante comum nas redes da nova geração ou NGN (*Next Generation Network*). O aumento de estações base de celular e a difusão da tecnologia como o 802.16 (WiMAX) e 3G sugere esta tendência.

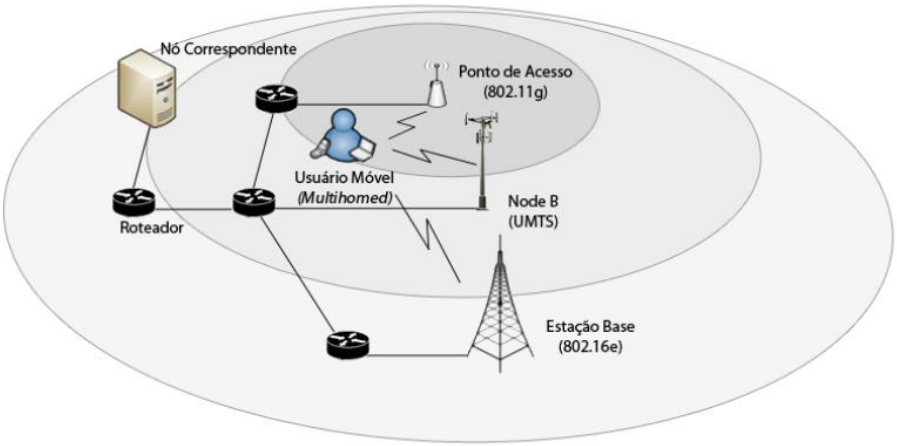


Figura 27. Cenário de Simulação no ns-2.

O cenário da Figura 27 possui, ainda, 4 roteadores cabeados conectando as redes à um nó correspondente (CN ou *Correspondent Node*), que pode ser considerado a Internet, outra rede, um Servidor de Streaming ou outro usuário qualquer. O dispositivo do usuário móvel (MN ou *Mobile Node*) possui várias interfaces (*multihoming*) que permitem a ele conectar-se com várias redes ao seu redor simultaneamente.

Para analisar a viabilidade da proposta foi necessário modificar o EvalVid padrão (KLAUE, 2003), pois só há interface de comunicação com o protocolo de transporte UDP. O EvalVid é um conjunto de ferramentas desenvolvido para fins de avaliação da qualidade de um vídeo transmitido sobre uma rede de comunicação real ou virtual. Destina-se à pesquisadores que desejem avaliar modelagens e/ou configurações de rede com relação à qualidade de vídeo percebida por um usuário (EVALVID, 2011).

Sendo assim, desenvolveu-se um *framework* EvalVid-aCMT-UC, com o objetivo de gerar tráfego de vídeo através de múltiplos caminhos simultâneos. O SCTP original, essencialmente não suporta o modo de transmissão parcialmente confiável de dados. Assim, a extensão PR-SCTP (*Partial Reliable* – SCTP) (Stewart, 2004) foi proposta para aplicações em tempo real. Para o nosso trabalho é necessário, ainda, utilizar a transmissão via múltiplos caminhos, deste modo utilizou-se a extensão PR-CMT (*Partial Reliability - Concurrent Multipath Transfer*) (HUANG, 2008), que possibilita o PR-SCTP transmitir simultaneamente por todos os caminhos disponíveis.

Os arquivos *traces* consistem em características reais do vídeo comprimido, incluindo número de *frames*, tipo de *frames*, tamanho, fragmentação em segmentos e o

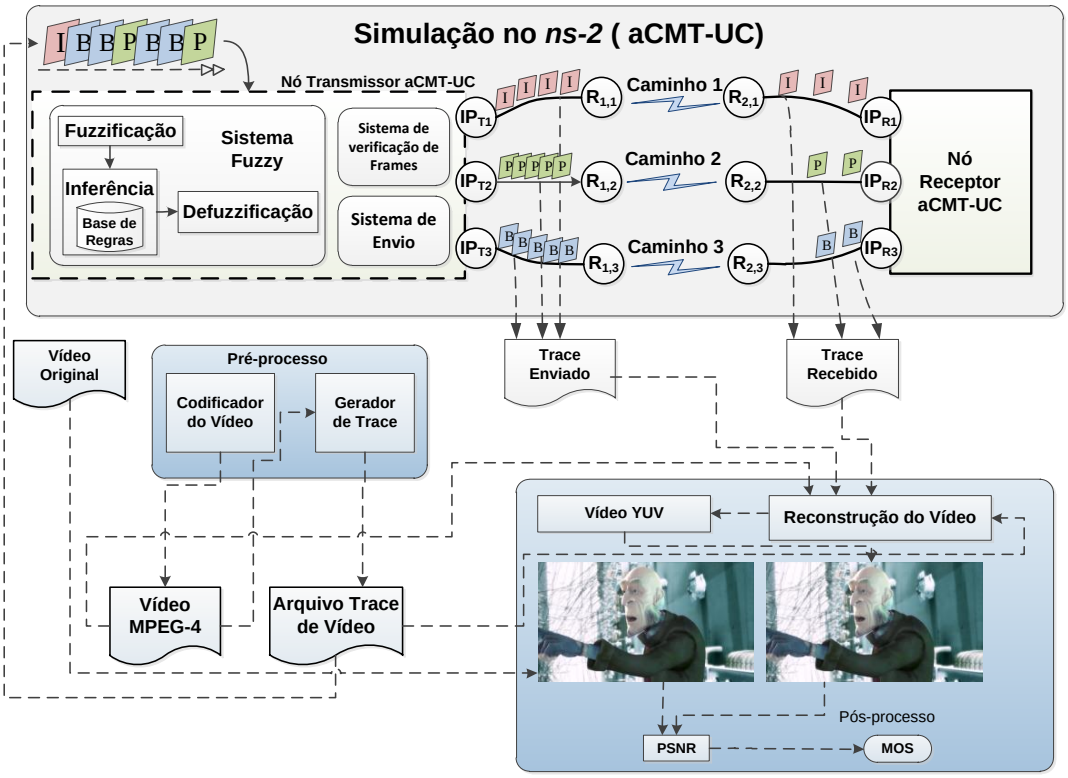
tempo para cada *frame* de vídeo. Essas características podem ser utilizadas para a construção de modelos de tráfego matemáticos e geradores de tráfego para simuladores de rede, desde que seja determinado o tamanho dos pacotes e os tempos escalonados.

Quando as áreas de cobertura das diferentes tecnologias sem fios se sobrepõem, não há necessidade de restringir-se a uma única interface. Particularmente, em cenários onde há uma variação na largura de banda e perdas inerentes ao caminho sem fio, faz sentido aumentar a probabilidade de que os pacotes perceptivelmente importantes cheguem ao receptor, de preferência descartando os pacotes menos importantes. No nível de frame, um esquema de classificação bem conhecido para vídeos com compensação de movimento (por exemplo, MPEG-*, * H.26) explora o fato de que, em geral, os I-*frames* são mais importantes que os P-*frames*, que são mais importantes do que os B-*frames*.

Os *frames* "I" agem como uma linha de base e são codificados como uma única imagem com a menor quantidade de compressão e maior quantidade de informações, sendo especialmente importantes, pois formam a base para a previsão de outros *frames*, usando vetores de movimento. Os *frames* "B" e "P" contêm menos informações e são codificados com um maior nível de compressão que os *frames* "I". Os *frames* "P" referenciam os quadros do passado e os *frames* "B" são capazes de referenciar quadros do passado e do futuro. Deste modo, optou-se por enviar os *frames* “I” através do caminho que possua maior peso, de acordo com o Sistema *Fuzzy* (AVELAR, 2012).

Os experimentos visaram observar os benefícios da proposta de tornar o CMT adaptativo e melhorar a distribuição de carga nas associações SCTP/CMT em cenários heterogêneos. Os resultados coletados mediram o impacto da distribuição de carga sob o ponto de vista dos usuários, por meio de análises de QoS e QoE. O caminho fim-a-fim foi pré-estabelecido no início da simulação e o usuário só utiliza as interfaces que possuem cobertura.

Usando as informações geradas durante a simulação e o arquivo de vídeo original comprimido, o Evalvid-aCMT-UC reconstrói o vídeo como se ele fosse recebido de uma rede real. Como mostrado na Figura 28, o framework Evalvid-aCMT-UC possui três processos nomeados como pré-processo, simulação no ns-2 e pós-processo. No pré-processo, o vídeo no formato YUV original é comprimido para o formato de vídeo MPEG-4 (*Moving Picture Experts Group - 4*), em seguida, um arquivo trace de vídeo é gerado, incluindo informações sobre cada frame (I / P / B-frame) de vídeo.



O processo de simulação do ns-2 identificará o tipo de frame e de acordo com a sua importância e enviará para o caminho definido pelo Sistema *Fuzzy*, ou seja, o frame mais importante será enviado pelo caminho que obtiver maior peso. Na fase de pós-processo, o vídeo pode ser reconstruído e convertido para o formato YUV. Esta reconstrução permite que o vídeo seja inspecionado visualmente, bem como possibilita o cálculo do PSNR (*Peak Signal to Noise Ratio*) e MOS (*Mean Opinion Score*) para o vídeo transmitido.

Uma das métricas de Qualidade de Experiência é aquela que utiliza a impressão de um observador humano ao receber o vídeo. As medidas de qualidade podem ser divididas em duas categorias. A primeira categoria contém medidas subjetivas, onde os indivíduos avaliam a qualidade do vídeo de acordo com sua experiência pessoal. A segunda categoria de medidas de qualidade contém medidas objetivas, em que um algoritmo é utilizado para calcular o valor da medida. A base das medidas objetivas é o PSNR. Na fase de pós-processamento, o vídeo pode ser reconstruído e convertido para o formato de vídeo YUV, em seguida, a avaliação da qualidade de vídeo pode ser

determinada através do cálculo de PSNR, SSIM (*Structural SIMilarity*), VQM (*Video Quality Metric*) e MOS, em comparação com o vídeo original.

6.2 Avaliação da Qualidade do Vídeo

Medições de qualidade de vídeo digital devem ser baseadas na qualidade percebida do vídeo real que está sendo recebida pelos usuários do sistema de vídeo digital, porque a impressão do usuário é o que conta no final. Existem basicamente duas abordagens para medir a qualidade de vídeo digital, nomeadas como métricas subjetivas de qualidade (QoE) e métricas objetivas de qualidade (QoS). Métricas subjetivas de qualidade compreendem um fator crucial, a percepção do usuário que está assistindo o vídeo. Já as métricas objetivas são descritas em detalhe por ITU (ITU, 1996), ANSI (ANSI, 1996) e MPEG (ISO, 1996).

Usando EvalVid é possível obter métricas de QoS (jitter, atraso, vazão e perdas de pacotes), MOS e PSNR, mas não é possível obter métricas mais robustas como SSIM e VQM. Usando a ferramenta MSU VQMT é possível obter todas as métricas de QoE utilizadas neste trabalho. Contudo, combinando-o com EvalVid, será possível transmitir os pacotes e obter métricas de QoS também. A compreensão de qualidade humana geralmente é dada em uma escala de 5 (melhor) a 1 (pior) como na Tabela 10. Esta escala é chamada MOS (*Mean Opinion Score*).

Tabela 10. Qualidade ITU-R.

Escala	Qualidade	Prejuízo
5	Excelente	Imperceptível
4	Bom	Perceptível, mas não irritante.
3	Aceitável	Ligeiramente irritante.
2	Pobre	Irritante.
1	Ruim	Muito Irritante.

Muitas pesquisas exigem métodos automatizados para avaliar a qualidade de vídeo. Testes subjetivos são caros e complexos, podendo muitas vezes não ser viável. Portanto, as métricas objetivas foram desenvolvidas para emular a percepção de qualidade do sistema visual humano (HVS - *Human Visual System*). Em (WU, 2000), há uma discussão exaustiva de várias métricas objetivas e seu desempenho em comparação com testes subjetivos.

No entanto, o método mais comum é o cálculo do PSNR (*Peak Signal to Noise Ratio*), imagem por imagem. É a derivada do SNR (*Signal to Noise Ratio*), que compara a energia de sinal com a energia de erro. O PSNR compara o máximo possível da energia do sinal com a energia de ruído, o que tem como resultado uma maior correlação com a percepção de qualidade subjetiva do que a SNR convencional (HANZO, 2001). A Equação 6 é a definição do PSNR entre o componente de luminância Y da imagem de fonte S e da imagem de destino D.

$$PSNR(n)_{dB} = 20\ log_{10} \frac{V_{peak}}{\sqrt{\frac{1}{N_{col}N_{row}} \sum_{i=0}^{N_{col}} \sum_{j=0}^{N_{row}} [Y_S(n,i,j) - Y_D(n,i,j)]^2}} \quad (6)$$

$$V_{peak} = 2^K - 1$$

K = número de bits por pixel (componente de luminância).

Uma vez que o PSNR é calculado quadro a quadro, pode ser inconveniente, quando aplicada a vídeos que possuem de várias centenas ou milhares de quadros. Além disso, é interessante analisar a distorção introduzida pela rede. Deste modo, deve-se comparar o vídeo recebido (possivelmente distorcido) com o vídeo sem distorções (vídeo enviado). Isto pode ser feito através da comparação do PSNR do vídeo codificado com o vídeo recebido, *frame* por *frame*, ou comparando suas médias e desvios-padrão.

Outra possibilidade é a de calcular primeiro o MOS (Tabela 11) e calcular a percentagem de *frames* com um MOS pior do que o do vídeo enviado (não distorcido). Este método tem a vantagem de mostrar claramente a distorção causada durante a transmissão. No capítulo 7 serão exibidos os resultados obtidos através do uso das ferramentas do EvalVid.

Tabela 11. Escala MOS para PSNR.

PSNR [dB]	MOS
> 37	5 (Excelente)
31 – 37	4 (Bom)
25 – 31	3 (Aceitável)
20 – 25	2 (Pobre)
< 20	1 (Ruim)

Em sistemas de transmissão de vídeo não só a perda real é importante para a qualidade do vídeo percebido, mas também o atraso dos *frames* e a variação do atraso, normalmente referido como *jitter*. Vídeos digitais sempre consistem em *frames* que precisam ser exibidos em uma taxa constante. Exibir um *frame* antes ou depois do tempo definido resulta na distorção do vídeo (WOLF, 2002).

As perdas de pacotes são normalmente calculadas com base nos identificadores dos pacotes. No contexto da transmissão de vídeo, não é importante apenas a quantidade de pacotes perdidos, mas também o tipo dos dados em cada pacote. Por exemplo, o codec MPEG-4 define quatro tipos diferentes de quadros (I, P, B, S) e também alguns cabeçalhos genéricos. Uma vez que é muito importante para as transmissões de vídeo, que tipo dos dados que se perde (ou não), é necessário fazer a distinção entre os diferentes tipos de pacotes.

No SSIM (*Structural SIMilarity*), o comportamento do vídeo é baseado na medição quadro a quadro comparado ao vídeo original, analisando os seguintes itens: contraste, luminosidade e estrutura, onde é gerado um valor decimal entre 0 e 1, no qual quanto mais próximo do 1 melhor será a qualidade do vídeo analisado (WANG, 2004). O VQM analisa os aspectos de ruído, distorção dos *frames*, cor e do quão embaçado está o vídeo, onde quanto mais próximo o valor se aproximar de 0 melhor será a qualidade do vídeo (XIAO, 2000).

Para incluir todas essas funcionalidades na simulação foi necessário realizar algumas mudanças no *patch* utilizado no *ns-2*. A seção seguinte contém todo o detalhamento das mudanças feitas no código do *ns-2*. Tais mudanças possibilitaram o uso do CMT em conjunto com o Sistema *Fuzzy*.

6.3 Modificações no ns-2

Várias modificações foram necessárias para implementar a proposta no ns-2. A avaliação da proposta foi conduzida no ns-2.29 (NS-2, 2011). Foram encontrados vários desafios para a conclusão deste trabalho, um deles é o fato do módulo SCTP-CMT (CARO, 2002) e *NIST Mobility* (NIST, 2011) serem de versões diferentes no ns-2. O primeiro foi implementado para a versão ns-2.34 e o segundo para a versão ns-2.29. Deste modo, foi necessário incluir o módulo SCTP-CMT no núcleo do ns-2.29. Outro problema é que o módulo CMT envia pacotes para as interfaces de forma equivalente.

Ou seja, todas as interfaces recebem os mesmos números de pacotes. Para que haja balanceamento de carga, esta funcionalidade precisava ser modificada de forma que algumas interfaces, de acordo com alguma política, recebam mais pacotes do que as demais. Por exemplo, se uma das conexões estiver muito congestionada, o fluxo parcial pode ser enviado por outra conexão com menor congestionamento.

Para portar, apenas copiaram-se os arquivos (sctp.cc, sctp.h, sctp-cmt.cc e sctp-cmt.h) do ns-2.34 para a pasta correspondente do ns-2.29. Acrescentando-se a linha "sctp/sctp-cmt.o \" ao arquivo *Makefile* no final da variável "OBJ_CC", para que o novo arquivo inserido sctp-cmt fosse reconhecido na compilação. Não foi preciso colocar o "sctp.o" pois este já está inserido na variável. Por fim compilou-se o ns-2.29. No caso do *NIST Mobility*, nada precisou ser feito.

O código fonte chave do CMT é o "sctp-cmt {cc,h}". A classe chave deste arquivo é o "SctpCMTAgent" que herda as funcionalidades da classe "SctpAgent" do arquivo "sctp {cc,h}". A classe "SctpCMTAgent" possui uma estrutura que guarda as interfaces de um nó *multihoming*, chamada "Node_S". A função que envia os pacotes para as interfaces na classe "SctpCMTAgent" é chamada de "SendMuch( )" . Esta função possui um *loop* que percorre toda a extensão da estrutura "Node_S" e a cada passada, envia pacotes para a interface corrente. Por este motivo o envio dos pacotes é dito equivalente, pois se houver 3 conexões SCTP ativas, a cada *loop* da função "SendMuch", serão enviados 3 grupos de pacotes, um para cada interface. Neste esquema, se forem enviados 9 pacotes ao todo, cada interface enviará 3 pacotes. Supõe-se que são enviados dez mil pacotes para um nó *multihomed* por meio de 3 interfaces (Wi-Fi, UMTS e WiMAX), a Tabela 12 mostra as prioridades configuradas para cada interface e o número de pacotes enviados por elas.

Tabela 12. Distribuição individual de 10.000 pacotes.

Interface	Prioridade	Número de Pacotes Enviados
WiFi	1	90
UMTS	10	900
WiMAX	100	9000

Ao analisar a Tabela 12 percebe-se que a cada pacote enviado pela interface WiFi, serão enviados 10 pela UMTS e 100 pela WiMAX. Se a vazão for levada em consideração, este método é mais eficaz que o método tradicional do CMT, pois neste

caso, mais pacotes seriam enviados por uma interface com maior vazão. A proporção 1:10:100 de pacotes por interface é apenas para exemplificar como a mudança ,da forma de envio padrão do CMT, pode tornar este método mais flexível e robusto. O foco deste trabalho é apresentar a proposta de modificação do padrão CMT para facilitar a aplicação de políticas que melhorem o desempenho da rede, como mostrado, de forma fictícia, na Tabela 12.

A chave da implementação do balanceamento de carga está na modificação do envio de pacotes dentro do *loop* que a função "SendMuch" faz na estrutura "Node_S". Antes, os pacotes eram enviados de forma indiscriminada por todas as interfaces. Agora os pacotes são enviados apenas pela interface que estiver ativa. Para que o programa pudesse identificar uma interface ativa, criaram-se outros campos dentro da estrutura "Node_S". A Tabela 13 mostra o antes e o depois da modificação da estrutura "Node_S" e em negrito o que foi acrescentado.

Tabela 13. Modificação da estrutura "Node_S".

Antes	Depois

struct Node_S {
NodeType_E eType;
void *vpData; Node_S *spNext;
Node_S *spPrev;
};
	struct Node_S {
	NodeType_E eType;
	void *vpData
	Node_S *spNext;
	Node_S *spPrev;
	int cmt_priority_;
	int cmt_count_;
	CurrentStatus status_;
	};

Na Tabela 13, a variável "cmt_priority_" armazena o valor da prioridade para cada interface. A variável "cmt_count" é um contador especial que conta o número de pacotes enviados pela interface. Ele conta de 0 até "cmt_priority_" . A interface atual fica ativa enquanto o "cmt_count_" não atinge o valor "cmt_priority_". Quando o "cmt_count_" atinge o valor "cmt_priority_" quer dizer que aquela interface já enviou o número de pacotes destinados a ela, neste momento a interface torna-se inativa e outra interface é ativada.

Para integrar o Sistema Fuzzy com o ns-2, colocou-se todo o código fonte *fuzzy* dentro no núcleo do ns-2 (*/usr/local/ns-allinone-2.29/ns-2.29-cmt/fuzzy*). A compilação

do ns-2 foi feita com o GCC/G++ 4.4 no Ubuntu 11.10. Devido ao ns-2.29 ser antigo foi necessário fazer várias correções para que a compilação pudesse ser concluída. A maioria delas são erros de conversão "*char" para "const char*" que não são feitas mais implicitamente. Para compilar o sistema fuzzy colocaram-se no *Makefile* do ns-2.29-cmt as seguintes informações em negrito:

MAKEFILE

```
INCLUDES = \
        -I. \
        -I/usr/local/ns-allinone-2.29/tclcl-1.17 -I/usr/local/ns-allinone-2.29/otcl-1.11 -
I/usr/local/ns-allinone-2.29/include -I/usr/local/ns-allinone-2.29/include -
I/usr/include/pcap \
        -I./tcp -I./sctp -I./common -I./link -I./queue \
        -I./adc -I./apps -I./mac -I./mobile -I./trace \
        -I./routing -I./tools -I./classifier -I./mcast \
        -I./diffusion3/lib/main -I./diffusion3/lib \
        -I./diffusion3/lib/nr -I./diffusion3/ns \
        -I./diffusion3/filter_core -I./asim/ -I./qs \
        -I./diffserv -I./satellite \
      -I./bluetooth \
      -I./hsntg \
        -I./hsntg/802_21 \
      -I./wimax \
      -I./interference \
      -I./myevalvid \
        -I./wpan \
        -I./fuzzy


OBJ_CC = \
        tools/random.o tools/rng.o tools/ranvar.o common/misc.o common/timer-
handler.o \
        common/scheduler.o common/object.o common/packet.o \
        common/ip.o routing/route.o common/connector.o common/ttl.o \
        trace/trace.o trace/trace-ip.o \
        classifier/classifier.o classifier/classifier-addr.o \
        [...]
        myevalvid/myevalvid.o \
        sctp/sctp-cmt.o \
        fuzzy/memberf.o fuzzy/variablf.o fuzzy/fuzzyrul.o fuzzy/ruleset.o \
        fuzzy/fuzzysystem.o fuzzy/mystring.o fuzzy/fl.o \
        $(OBJ_STL)
```

A Figura 29 exibe o Sistema Fuzzy instalado dentro do núcleo do ns-2. Após a compilação, o novo módulo *fuzzy* estará disponível para ser utilizado nas simulações do

ns-2. Para configurar o Sistema Fuzzy foi necessário gerar três arquivos de configuração (Apêndice B), sendo um para cada tipo de interface. Através dessa configuração é possível determinar os valores dos gráficos de pertinência, além das regras de inferência. Como as interfaces possuem características de vazão diferentes, foi preciso gerar um arquivo de configuração para as três interfaces utilizadas na pesquisa: WiFi, 3G/UMTS e WiMAX.

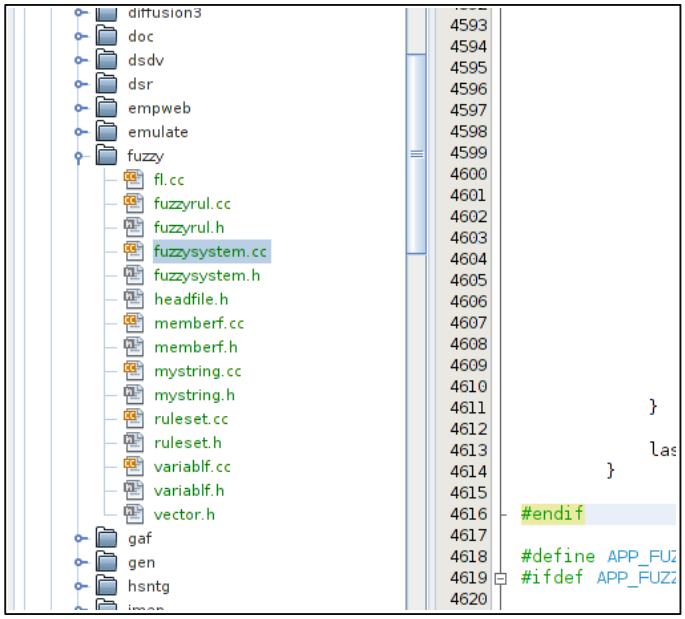


Figura 29. Sistema Fuzzy instalado no núcleo do ns-2.

Para executar a simulação do tráfego multimídia utilizou-se o vídeo “Elephants dream” no formato “.yuv”. Toda a análise foi realizada através do script “acmt_uc.sh” (Apêndice C). Este script possui 4 passos. O primeiro passo é composto pelo tratamento do vídeo, ou seja, prepara o vídeo para ser usado na simulação. Nesta etapa é criado um arquivo de trace (st_a01) contendo todas as informações importantes do vídeo que serão utilizados na simulação, incluindo o número de *frames*, tipos (I, P ou B) e tempo de envio de cada frame. O segundo passo executa a simulação no ns-2 (Apêndice D). Nesse passo, são gerados dois arquivos, sendo um de envio (sd_a01) e outro de recepção (rd_a01). Esses dois arquivos em conjunto com o “st_a01” serão utilizados no terceiro passo para a reconstrução do vídeo. O quarto passo consiste na avaliação do vídeo, sendo possível calcular o PSNR e o MOS.

O aCMT-UC é um protocolo de transporte, então para que seja utilizado no ns-2 é necessário que uma aplicação o utilize. Deste modo, foi necessário implementar uma

aplicação baseada na extensão “myevalvid” (KE, 2008) denominada como “mycmtevalvid”. Originalmente, o “myevalvid” foi desenvolvido apenas para gerar tráfego de vídeo utilizando UDP, sendo assim foi necessário incluir as funcionalidades do CMT para gerar o “mycmtevalvid”.

Com a simulação no ns-2, outra possibilidade interessante é visualizar a simulação utilizando o NAM. A Figura 30 apresenta uma captura de tela da simulação do aCMT-UC em execução no NAM. Nesta figura é possível visualizar o nó móvel com três interfaces de comunicação.

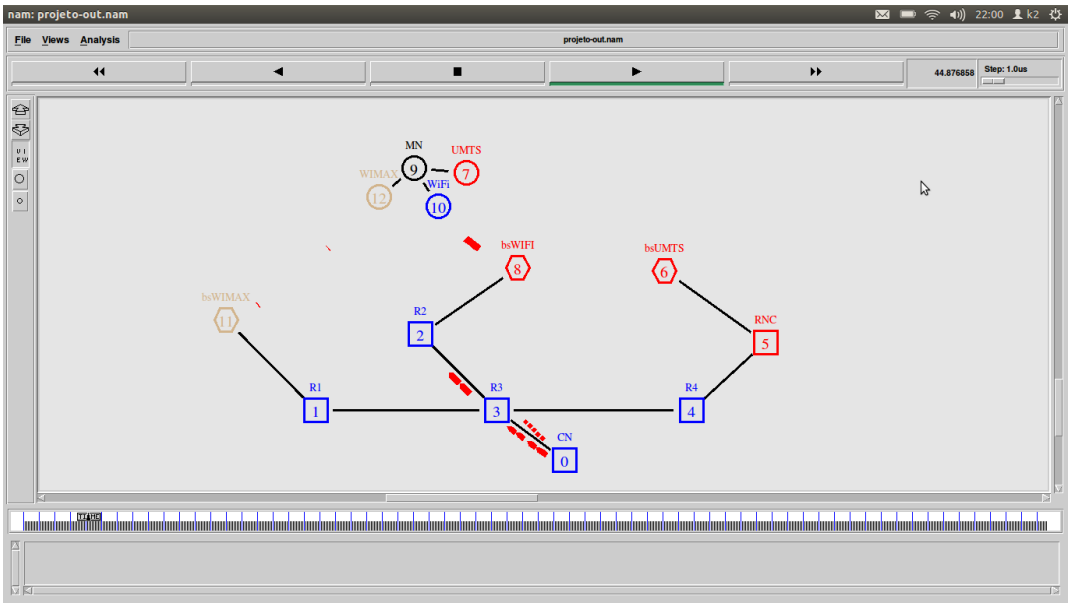


Figura 30. Captura de tela do NAM.

Este capítulo apresentou a metodologia utilizada para avaliar a proposta deste trabalho, além de detalhar o cenário utilizado nas simulações. A proposta foi avaliada no ns-2, utilizando-se o *framework* EvalVid-aCMT-UC para gerar tráfego de vídeo. Esse *framework* possibilitou a análise de QoE da proposta, sendo estes parâmetros apresentados na seção 6.2.

7. Resultados e Discussão

Foram realizadas noventa simulações, sendo trinta simulações para cada proposta (CMT, aCMT e aCMT-UC). Cada simulação possui a mesma semente correspondente nas outras propostas para comparação. O cenário utilizado nas simulações apresenta redes heterogêneas sobrepostas e é composto pelas tecnologias 802.11 (Wi-Fi), 802.16 (WiMAX) e 3G/UMTS. Este ambiente será bastante comum nas redes da nova geração ou NGN (*Next Generation Network*).

A avaliação é conduzida de duas formas, através de modelos de erros que variam de 1% a 10% de perdas nos caminhos e da inserção aleatória de tráfegos UDP na rede. O Sistema *Fuzzy* detecta os caminhos de melhor qualidade de enlace, com isso, pode desviar o fluxo para o caminho menos congestionado e com a menor probabilidade de erros na rede.

A Tabela 14 descreve a configuração das simulações. O tráfego *multihoming* é configurado no sentido *downlink* (do CN para o MN). Dez usuários são posicionados de forma aleatória, onde as coordenadas possuem uma distribuição gaussiana com média em torno da posição do AP 802.11 e variância limitada à área de cobertura do UMTS, para haver, em todos os casos, pelo menos duas redes na cobertura do usuário, caso contrário não haveria heterogeneidade.

Tabela 14. Configuração das Simulações.

Redes	802.11(*Wi-*	802.16	3G/*UMTS*
Taxa de Transmissão	54 Mbps	75 Mbps	2 Mbps
Raio de Cobertura	50m	1300m	1000m
Número de nós	10 usuários estáticos posicionados aleatoriamente. 1 Nó correspondente		
Tipo de Tráfego (para cada usuário)	FTP (limitado a 5 Mbps).		
Avaliação	Modelo de Erro (variando de 1% a 6% de		
Intervalo de monitoramento	*500 ms*		
Tamanho do pacote (*Data Chunk SCTP*)	1468		
Fragmentação máxima dos pacotes	1500		
Vídeo (*Elephants Dream*)	15.691 *frames*	Resolução: 352 x	
Warm-up	10% (60s) = 6s		
Tempo de cada simulação	654 segundos (10:54)		
Número de simulações	90		

A Figura 31 apresenta a média da vazão no receptor em relação à taxa de erro, que varia de 1% a 6%, inserida na rede. Neste gráfico, percebe-se que o CMT não se adapta em um cenário heterogêneo desvantajoso. Quando ocorrem erros na rede, o CMT perde pacotes e isso provoca a degradação da vazão total das associações heterogêneas. Por outro lado, o aCMT através do sistema fuzzy, detecta o cenário desvantajoso, desviando o fluxo para o enlace com melhor qualidade, mantendo a vazão quase constante. Utilizando-se uma taxa de erro de 6%, observa-se que a vazão média do aCMT foi quase 50% superior em relação ao CMT.

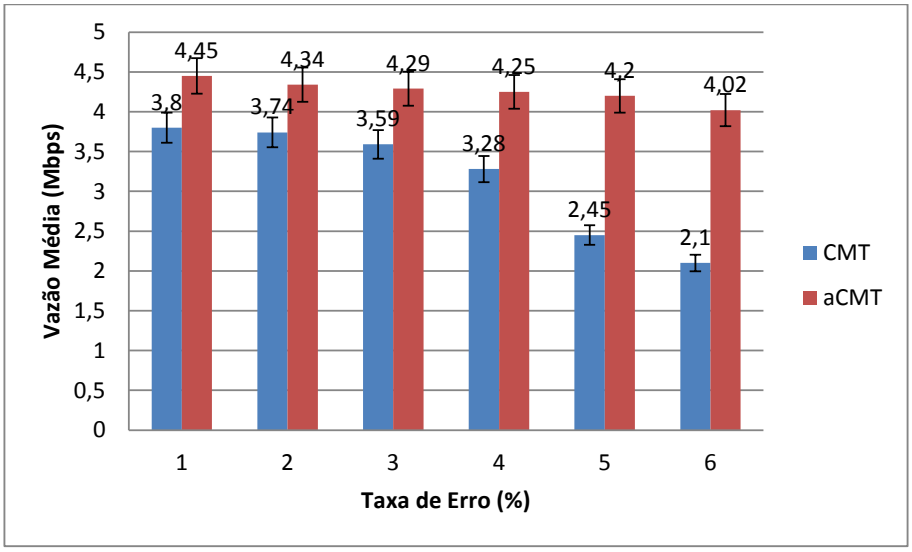


Figura 31. Comparação entre as médias das vazões do CMT e do aCMT que resultaram dos experimentos.

A Figura 32 exibe a média das perdas de pacotes em toda a rede durante as simulações. Nota-se que o aCMT novamente se mostra bastante robusto em comparação com o CMT, quando ocorrem erros na rede. Segundo o gráfico, com 6% de taxa de erro ocorrem 5,76% de perdas de pacotes utilizando o CMT e apenas 0,9% utilizando o aCMT. Isto acontece, pois o aCMT possui um Sistema *Fuzzy* que detecta falhas no caminho, deste modo, o fluxo é redirecionado para o melhor caminho, reduzindo, assim, as perdas de pacotes.

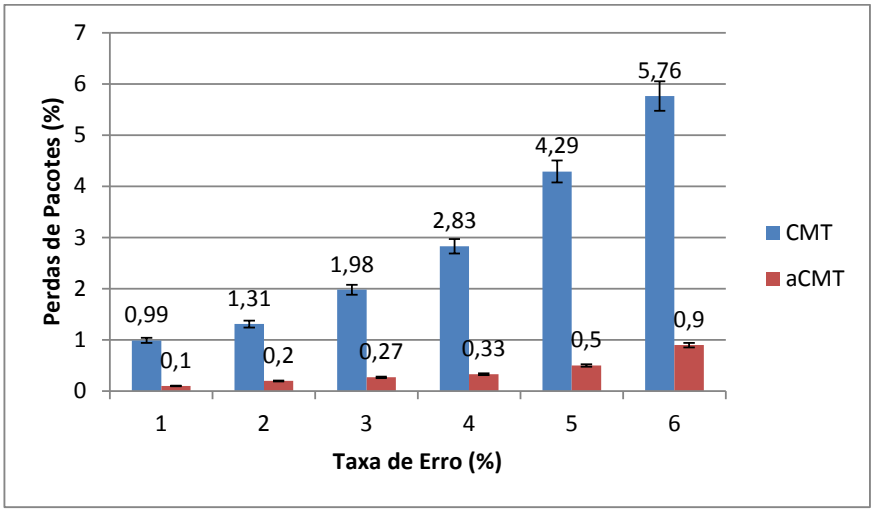
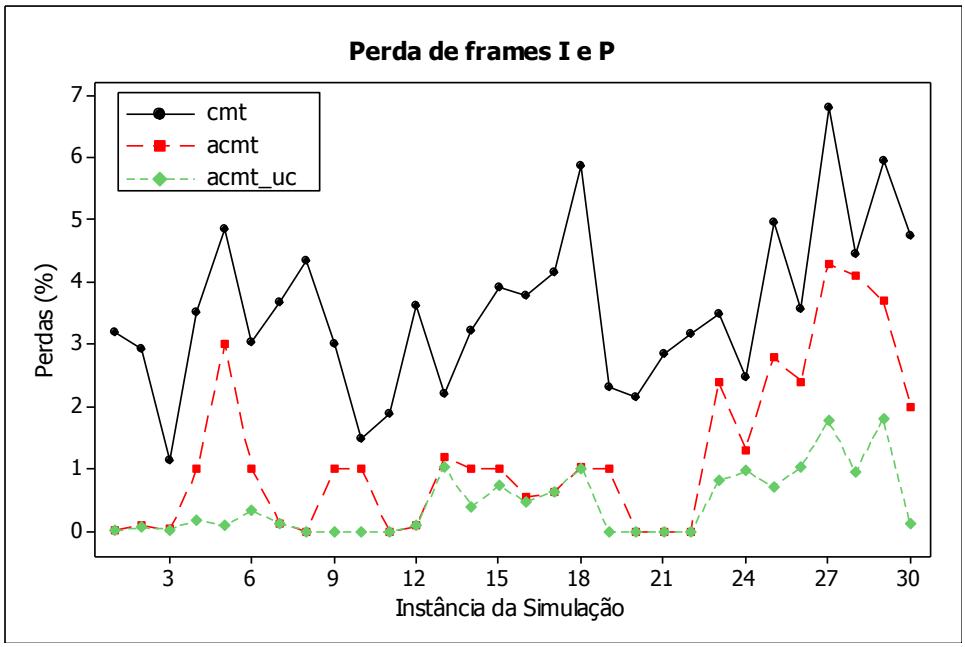


Figura 32. Perdas de pacotes no CMT e no aCMT.

A Figura 33 mostra as perdas apenas dos *frames* I e P, que são os *frames* mais importantes para a reconstrução da imagem. Nesta figura, observa-se que as perdas de *frames* utilizando o protocolo CMT são bem maiores em comparação com as do aCMT e do aCMT-UC. Isso se deve ao fato do CMT enviar pacotes para todas as interfaces sem tomar conhecimento da qualidade do enlace em questão. Apesar da proposta aCMT priorizar os melhores caminhos, ao ocorrer um descarte de pacote não há nada que impeça que os *frames* mais importantes sejam descartados.



Com o aCMT as perdas são menores, uma vez que ele prioriza o envio de pacotes pelo enlace que proporcionará melhor QoS para aplicação, porém, o aCMT não distingue o tipo de pacote enviado e desta forma alguns *frames* importantes, tipo I e P, são perdidos, causando queda considerável na qualidade do vídeo vista pelo usuário. Por outro lado, o aCMT-UC prioriza o envio dos pacotes I e P por caminhos menos congestionados e, portanto, com menor probabilidade de perdas.

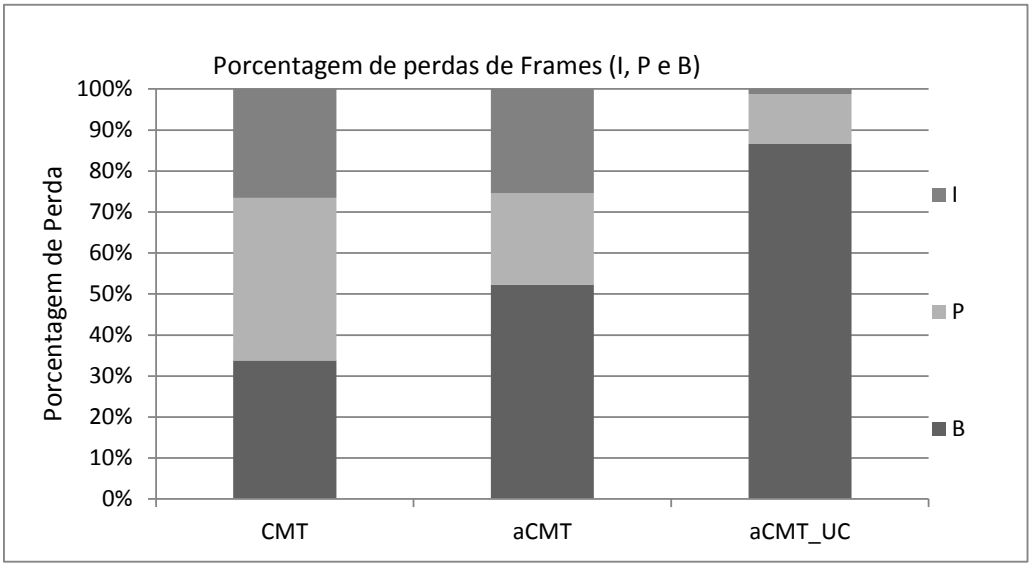
A Tabela 15 exibe algumas métricas de estatística descritiva da perda de *frames* I e P. Pode-se perceber que a perda com o aCMT-UC é bastante inferior a do CMT e do aCMT. A mediana é uma medida importante por mostrar o ponto de equilíbrio dos dados, ela equivale ao segundo quartil, ou seja, a mediana é o valor até o qual se encontra 50% da amostra em ordem, portanto ela não é afetada por valores discrepantes como na média. Nesta métrica, o aCMT-UC também é superior aos seus concorrentes.

Tabela 15. Estatística descritiva das perdas de *frames* I e P.

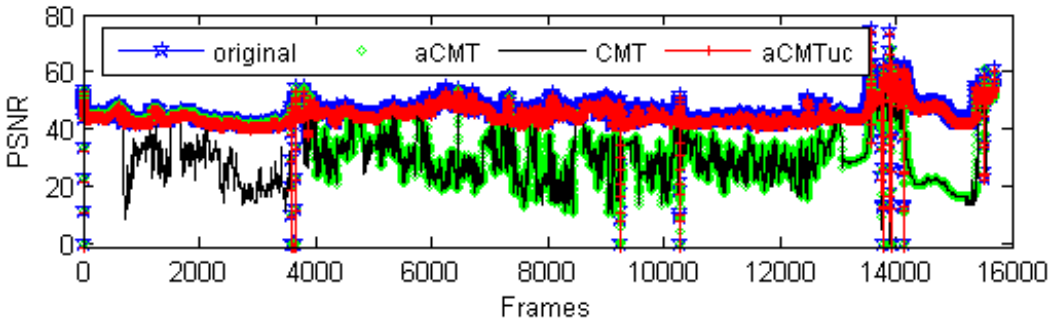
Protocolo	Média	Erro Médio	Desvio Padrão	Mediana
CMT	3,558	0,239	1,309	3,495
aCMT	1,226	0,235	1,289	1
aCMT-UC	0,4468	0,0968	0,53	0,1565

A Figura 34 mostra a porcentagem dos tipos diferentes de *frames* perdidos em relação ao total. É possível observar que o CMT perde *frames* de maneira proporcional, pois o envio de dados do CMT é realizado de maneira equivalente. Com o aCMT são perdidos mais *frames* do tipo B, pois esses *frames* são mais abundantes que os demais, no entanto é possível observar que *frames* do tipo I representam mais do que 20% do total de perdas, o que ainda é considerado alto.

O aCMT-UC por outro lado, prioriza o envio de *frames* I para que suas perdas, em cenários críticos, sejam minimizadas. É possível observar na Figura 34 que as porcentagens de *frames* I e P perdidas são menores em comparação com os outros protocolos em detrimento dos *frames* B que possuem menor importância. A perda de *frames* I geram maior distorção no vídeo recebido, sendo assim é melhor que um *frame* B seja perdido do que um *frame* I.



PSNR (*Peak Signal to Noise Ratio*) é uma das métricas objetivas mais bem conhecidas para avaliar QoS e por isso é utilizada neste trabalho para medir a qualidade do vídeo. PSNR mede o erro entre as imagens originais e reconstruídas. Deste modo, foi medido o erro entre o vídeo codificado no lado do remetente e vídeo decodificado no lado do receptor. Na Figura 35 é possível observar que tanto o aCMT quanto o CMT apresentam uma queda excessiva no PSNR, o que causará uma grande perda de qualidade do vídeo transmitido. No entanto, há uma melhoria na PSNR quando utiliza-se a proposta aCMT-UC, devido à priorização de *frames*.

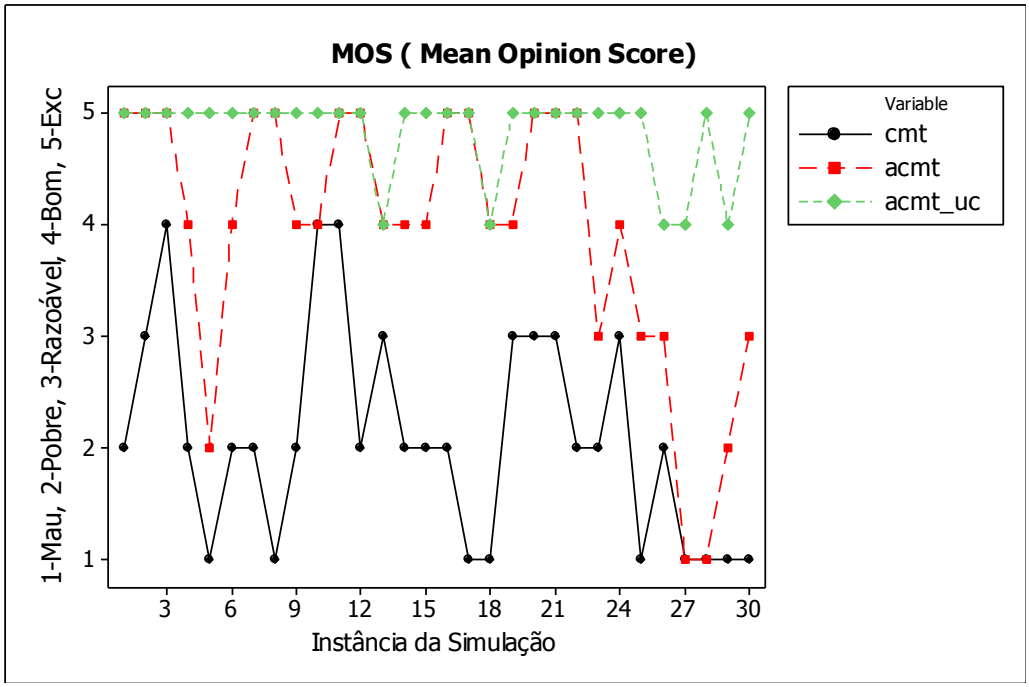


A Tabela 16 exibe algumas métricas de estatística descritiva em função do PSNR coletado (Figura 35). Pode-se perceber que a proposta aCMT-UC obteve um valor de PSNR bem próximo ao do vídeo original. No entanto, as propostas CMT e aCMT, como não possuem uma política que priorize os *frames* importantes, obtiveram valores mais afastados, cerca de 30% abaixo do valor esperado.

Tabela 16. Estatística descritiva do PSNR.

PSNR (dB)	Original	CMT	aCMT	aCMT-UC
Média	45,34	33,63	30,33	43,15
Desvio	5,33	10,37	10,01	5,37
Variância	28,43	107,59	100,31	28,87

A qualidade do serviço percebida por uma pessoa geralmente é dada em uma escala de 5 (melhor) a 1 (pior). Esta escala é chamada de MOS (*Mean Opinion Score*). A Figura 36 mostra o gráfico do MOS do vídeo recebido após cada simulação. Pode-se observar que as perdas de pacotes, ao utilizar os protocolos CMT e aCMT, afetam muito o MOS do vídeo recebido. Enquanto o aCMT-UC mantém valores de MOS entre "bom" e "excelente", o aCMT e o CMT atingem valores correspondentes a "pobre" e até "mau" em algumas simulações.



A Tabela 17 exibe a estatística descritiva, com 95% de intervalo de confiança, para a variável MOS. É possível observar que o aCMT-UC possui 4,8 de média de MOS, o que é mais do que o dobro da média do CMT. Na mediana, o aCMT-UC também se mostra superior tanto em relação ao CMT quanto ao aCMT.

Tabela 17. Estatística descritiva do MOS.

Protocolo	Média	Erro	Desvio	Mediana
CMT	2,1	0,175	0,96	2
aCMT	3,933	0,219	1,202	4
aCMT-UC	4,8333	0,0692	0,379	5

A Figura 37 exibe a comparação do frame 8211 do vídeo recebido para cada protocolo avaliado. É possível verificar que a degradação na qualidade do vídeo é claramente visível ao se utilizar o protocolo CMT, devido às perdas de pacotes. O frame do vídeo recebido com o aCMT melhora um pouco, mas a qualidade da imagem ainda é degradada, pois quando ocorreram perdas de pacotes não houve priorização de *frames*, ou seja, *frames* importantes para a reconstrução do vídeo foram descartados. No entanto, ao se analisar o vídeo recebido com o aCMT-UC nota-se, claramente, que não houve degradação na qualidade do vídeo. Isto possibilitou sua reconstrução no receptor, tornando-o muito semelhante ao frame original.

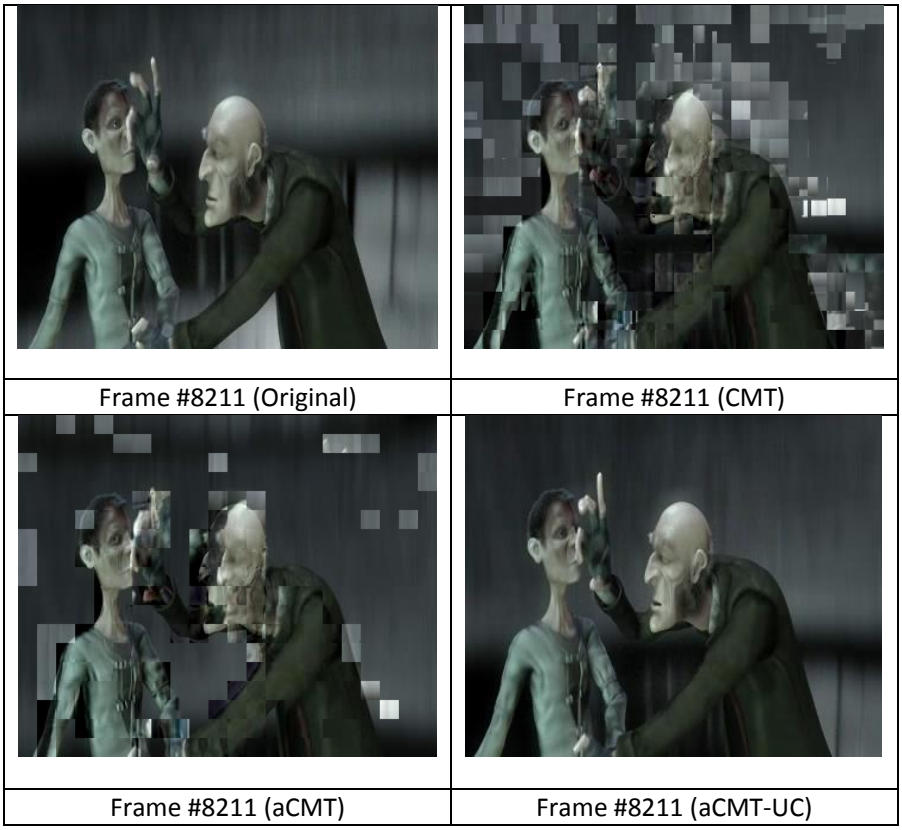


Figura 37. Comparação entre o frame número 8211 de cada proposta.

8. Conclusão e Trabalhos Futuros

Este capítulo conclui o trabalho desenvolvido nesta dissertação. Na seção 8.1 são apresentadas as considerações finais e sumarização do trabalho desenvolvido. As contribuições são abordadas na seção 8.2. Por fim, a seção 8.3 apresenta e discute os trabalhos futuros.

8.1 Considerações Finais

Esta dissertação apresentou uma proposta de distribuição de carga via múltiplos caminhos utilizando uma extensão do protocolo SCTP chamada CMT. Verificou-se que o CMT não possui um bom desempenho em redes heterogêneas, por este motivo foi proposto, inicialmente, o aCMT que utiliza Lógica *Fuzzy* para a seleção e discriminação dos caminhos, atribuindo-lhes pesos dinamicamente. No entanto, notou-se em experimentos que o aCMT não era adequado quando se tratava de tráfego de vídeo, pois não discriminava o tipo de *frame* enviado. Deste modo, foi proposta uma melhoria do aCMT para ser utilizado como protocolo de transporte especificamente para aplicações de vídeo. Vale lembrar que o algoritmo aCMT-UC pode ser chaveado para que possa ser utilizado de forma intercambiável com o aCMT. Na presença de vídeo usa-se o aCMT-UC e nos demais casos pode-se utilizar o aCMT, pois a base do aCMT se encontra dentro da proposta aCMT-UC.

8.2 Contribuições

Esta seção apresenta as principais contribuições obtidas neste trabalho. A primeira contribuição foi caracterizar o problema do uso de diferentes tecnologias sem fio simultâneas, através de uma análise via Redes de Petri. Pode-se concluir com isso a necessidade de um mecanismo inteligente para gerenciar a distribuição de carga entre as diferentes redes utilizadas.

Partindo desse pressuposto, a segunda contribuição foi propor e avaliar uma solução adaptativa e centrada no usuário para o acesso heterogêneo sem fio com suporte a *multihoming*, QoS e QoE. Através da implementação de um Sistema Fuzzy foi possível obter um bom desempenho quando comparado ao protocolo CMT padrão.

A terceira contribuição foi a implementação de um *framework* EvalVid-aCMT-UC para avaliar as métricas de QoE da proposta. Com base nesse *framework* pode ser viável analisar tráfego de vídeo em ambientes que utilizam várias interfaces simultâneas, através da priorização de *frames* de vídeo mais significativos.

Parte das contribuições obtidas já foi publicada, submetida ou está em fase de preparação:

- Avelar, L. M. et al (2012) "Modelagem *Multihoming* em Redes Heterogêneas e Proposta de Distribuição de Carga utilizando o aCMT", XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2012.
- Avelar, L. M. et al (2013) "aCMT-UC: Um proposta de distribuição de carga centrada no usuário", XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2013. Submetido/Aguardando decisão.

8.3 Trabalhos Futuros

Este estudo e os resultados obtidos permitiram o surgimento de novas alternativas para pesquisa adicional, com a finalidade de melhorar e contribuir com o processo de distribuição de carga. A partir das opções para trabalhos futuros, pode-se destacar algumas possíveis pesquisas.

A possibilidade de realizar um estudo envolvendo mobilidade/*handover* vertical, já que neste trabalho focamos no balanceamento em cenário com usuário estático. A possibilidade do *handover* vertical também é uma meta tangível para pesquisas futuras e poderá ser considerada tanto para o balanceamento quanto para o aprovisionamento de QoS/QoE;

Outra alternativa seria uma solução *multihoming* energeticamente eficiente, pois o tráfego de aplicações de vídeo consome bastante banda e energia tanto dos dispositivos móveis quanto nos equipamentos da rede. Deste modo há a necessidade de estender o tempo de vida das baterias, bem como, reduzir o consumo de energia no núcleo da rede no contexto das então denominadas redes verdes (*Green Networking*) (GUPTA, 2009), uma tendência na pesquisa em TICs que visa, além do aspecto econômico, minimizar a emissão de CO_2 na atmosfera. A extensão do sistema fuzzy

para contemplar variáveis relacionadas ao consumo de energia tanto do dispositivo quanto da rede podem ser vislumbradas para a seleção do melhor caminho.

Para efeito de comparação é possível utilizar outras técnicas de inteligência computacional e otimização, tais como redes neurais, teoria dos jogos e algoritmos genéticos. Esta análise serviria para avaliar se a utilização da Lógica *Fuzzy*, adotada nesta dissertação, seria a melhor solução para tornar a proposta dinâmica.

Referências

Adeli, H. and Sarma, K. C. (2006). Cost Optimization of Structures: Fuzzy Logic, Genetic Algorithms, and Parallel Computing. Wiley.

ANSI (1996). Digital transport of video teleconferencing / video telephony signals. ANSI, 1996.

Avelar, L. M. et al (2012) "Modelagem Multihoming em Redes Heterogêneas e Proposta de Distribuição de Carga utilizando o aCMT", XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2012.

Aydin, I.; Chien-Chung Shen, "Performance Evaluation of Concurrent Multipath Transfer Using SCTP Multihoming in Multihop Wireless Networks," Network Computing and Applications, 2009. NCA 2009. Eighth IEEE International Symposium on , vol., no., pp.234,241, 9-11 July 2009.

Balbo, G; Bruell, S. C. and Sereno, M. Arrival theorems for product-form stochastic petri nets. In Proceedings of the 1994 conference on Measurement and modeling of computer systems, pages 87–97, Nashville, Tennessee, United States, 1994. ACM Press.

Bates, T. et al (1998) "Scalable Support for Multi-homed Multiprovider Connectivity", RFC 2260, January 1998. <http://www.ietf.org/rfc/rfc2260.txt>.

Budzisz, L. et al. (2009) "On Concurrent Multipath Transfer in SCTP-Based Handover Scenarios", IEEE International Conference on Communications, pp.1-6, 14-18 June 2009.

Caro, A. et al (2002) "ns-2 SCTP module", Version 3.5, http://www.armandocaro.net/ software/ns2sctp/.

Cisco (2012). Cisco visual networking index: Forecast and methodology, 2011 – 2016. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/. Accessed: 27/06/2012.

Cheng, L. and Marsic, I. Fuzzy Reasoning for Wireless Awareness, International Journal of Wireless Information Networks, Vol. 8, Issue 1, Jan. 2001, pp. 15-26.

Chung-Ming Huang; Ming-Sian Lin; , "Partially Reliable-Concurrent Multipath Transfer (PR-CMT) for Multihomed Networks," Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE, vol., no., pp.1-5, Nov. 30 2008-Dec. 4 2008.

CHUNG, Jae; CLAYPOOL, Mark. Ns by example [online] 2003. Disponível na Internet via URL: http://nile.wpi.edu/NS. Arquivo capturado em 27/11/2009.

Costa, Daniel Gouveia. SCTP – Uma Alternativa aos Tradicionais Protocolos de Transporte da Internet. Primeira Edição. Rio de Janeiro: Editora Ciência Moderna, 2005.

Dreibholz, T.; Rathgeb, .P.; R ngeler, I.; Seggelmann, R.; T xen, M.; Stewart, R.R.; , "Stream control transmission protocol: Past, current, and future standardization activities," Communications Magazine, IEEE , vol.49, no.4, pp.82-88, April 2011 doi: 10.1109/MCOM.2011.5741151.

Dreibholz, T.; Becke, M.; Rathgeb, E.P.; T xen, M.; , On the Use of Concurrent Multipath Transfer over Asymmetric Paths," GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference , vol., no., pp.1-6, 6-10 Dec. 2010 doi: 10.1109/GLOCOM.2010.5683579.

Espinosa, J., Vandewalle, J., Wertz, V.: Fuzzy logic, identification and predictive control. Springer, Heidelberg (2004).

Evalvid - A Video Quality Evaluation Tool-set home page http://www.tkn.tu-berlin.de/research/evalvid/, acessado em Agosto/2011.

Ford, A. et al (2011) "Architectural Guidelines for Multipath TCP Development", RFC 6182, March 2011. <http://www.ietf.org/rfc/rfc6182.txt>.

German, R.; Kelling, C.; Simmermann, A. and Hommel, G. "TimeNET: a toolkit for evaluating non-markovian stochastic petri nets". Performance Evaluation, v. 24, p. 69-87, 1995.

Greengrass, J.,Evans, J. and Begen, A. "Not all packets are equal, part 2: The impact of network packet loss on video quality," IEEE Internet Computing, vol. 13, pp. 74–82, March 2009.

R. Gupta, "Collaborative heterogeneity for energy efficient systems," in Proceedings of Workshop on Green Communications in conjunction with IEEE GLOBECOM, Hawaii, USA, December 2009.

Hanzo, L., Cherriman, P. J. and Streit, J. Wireless Video Communications. Digital & Mobile Communications. IEEE Press, 445 Hoes Lane, Piscataway, 2001.

Huang, C. et al. (2009) "An MIH-Assisted Handoff Mechanism for Concurrent Multipath Transfer in Wireless Multihomed Networks", Personal, Indoor and Mobile Radio Communications, p. 778 – 782.

ISO-IEC/JTC1/SC29/WG11. Evaluation methods and procedures for july mpeg-4 tests, 1996.

ITU-R Recommendation BT.500-10. Methodology for the subjective assessment of the quality of television pictures, March 2000.

Iyengar, J.R. et al (2006) "Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths", Networking, IEEE/ACM Transactions on, vol.14, no.5, pp.951-964.

Ke, Chih-Heng, et al. "An evaluation framework for more realistic simulations of MPEG video transmission." Journal of information science and engineering 24.2 (2008): 425-440.

Kim, T. et al (2010) "Concurrent Multipath Transfer using SCTP multihoming over heterogeneous network paths", 2010 International Conference on Control Automation and Systems (ICCAS), vol., no., pp.1598-1602, 27-30 Oct. 2010.

Mamdani, E.H. (1974) Application of Fuzzy Algorithms for Control of Simple Dynamic Plant. IEEE (Control and Science), v.121(12), p.1585-1588.

Marsan, M. A., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G. Modelling with Generalized Stochastic Petri Nets, ACM SIGMETRICS Performance Evaluation Review, v.26 n.2, p.2, August 1998.

Marsan, M. A.; Balbo, G. and Conte, G. Performance Models of Multiprocessor systems. The MIT Press, 1986.

Marsan, M. A.; Conte, G. and Balbo, G. A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems. ACM Transactions on Computer Systems, 2(2):93–122, 1984.

NAM - THE NETWORK ANIMATOR [online]. Disponível na Internet via URL: http://www.isi.edu/nsnam/nam/index.html. Arquivo capturado em 15/11/2011.

Natarajan, P.; Ekiz, N.; Yilmaz, E.; Amer, P.D.; Iyengar, J.; Stewart, R.; , "Non-Renegable Selective Acknowledgments (NR-SACKs) for SCTP," Network Protocols, 2008. ICNP 2008. IEEE International Conference on, pp.187-196, 19-22Oct.2008, doi:10.1109/ICNP.2008.4697037

Nguyen, S. et al. (2011) "Evaluation of multipath TCP load sharing with coupled congestion control option in heterogeneous networks," Global Information Infrastructure Symposium (GIIS), 2011, pp.1-5, 4-6 Aug. 2011.

Nightingale, J. et al (2012) "Removing path switching overhead in multipath mobile video delivery", IEEE International Conference on Consumer Electronics (ICCE), pp.275-276, 13-16 Jan. 2012.

NIST. (2011). [Online]. Disponível em: http://w3.antd.nist.gov/seamlessandsecure/pubtool.shtml#tools Acessado em 20 de agosto de 2011 às 15:00h

NS-2. (2011) “The network simulator ns-2.” [Online]. Disponível em: http://www.isi.edu/nsnam/ns/ Acessado em 16 de agosto de 2011 às 8:00h

Ortega, N. R. S. “Aplicação da teoria de conjuntos fuzzy a problemas da biomedicina”. 2001. 152p. Tese (Doutorado) – Universidade de São Paulo, São Paulo, SP.

Peterson, J. L. Petri nets. ACM Computing Surveys, 9(3):223–252, 1977.

Rüncos, R. et al. (2011) “Avaliação de parâmetros do SCTP para transporte de tráfego VoIP em cenários com perdas” XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos.

Schumacher, J.; Dobler, M.; Dillon, E.; Power, G.; Fiedler, M.; Erman, D.; de Vogeleer, K.; Ramos, M.O.; Argente, J.R., "Providing an User Centric Always Best Connection," Evolving Internet (INTERNET), 2010 Second International Conference on , vol., no., pp.80,85, 20-25 Sept. 2010

Song, W. et al. (2012) "Performance Analysis of Probabilistic Multipath Transmission of Video Streaming Traffic over Multi-Radio Wireless Devices", IEEE Transactions on Wireless Communications, vol.11, no.4, pp.1554-1564, April 2012.

Stewart, R.; XIE, Q. RFC 2960: Stream Control Transmission Protocol [online] 2000. Disponível na Internet via URL: http://www.ietf.org/rfc/rfc2960.txt. Arquivo capturado em 11/03/2009.

Stewart, R. e Xie, Q. 2004. "RFC 3758. Stream Control Transmission Protocol Partial Reliability Extension

Stewart, R. (2007) “Stream Control Transmission Protocol”, RFC 4960 September 2007. <http://www.ietf.org/rfc/rfc4960.txt>.

Tanenbaum, Andrew S. Redes de Computadores. Quarta Edição. Rio de Janeiro: Editora Campus, 2003.

TimeNET(2011). Disponível em < http://www.tu-ilmenau.de/fakia/8086.html>. Acesso Setembro/2011.

TRACEGRAPH [online]. Disponível na Internet via URL: http://www.tracegraph.com/. Arquivo capturado em 12/03/2009.

Tu, W. et al. (2012) "Efficient Resource Utilization for Multi-Flow Wireless Multicasting Transmissions", IEEE Journal on Selected Areas in Communications, vol.30, no.7, pp.1246-1258, August 2012.

Wallace, T. et al. (2012) "A Review of *Multihoming* Issues Using the Stream Control Transmission Protocol", IEEE Communications Surveys & Tutorials, vol.14, no.2, pp.565-578, Second Quarter 2012.

Wang, Z.; Bovik, A. C.; Sheikh, H. R. and Simoncelli, E. P. "Image Quality Assessment: From Error Visibility to Structural Similarity", IEEE Trans. on Image Proc., Vol. 13, No. 4, pp. 600-612, April 2004.

Wiegand, T.; Sullivan, G. J.; Bjøntegaard, G. and Luthra, A. “Overview of the H.264/AVC video coding standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576, Jul. 2003.

Wolf, S. and Pinson, M. Video quality measurement techniques. Technical Report 02-392, U.S. Department of Commerce, NTIA, June 2002.

Wu, D.; Hou, Y. T.; Zhu, Wenwu; Hung-Ju Lee; Tihao Chiang; Ya-Qin Zhang; Chao, H.J., "On end-to-end architecture for transporting MPEG-4 video over the Internet," Circuits and Systems for Video Technology, IEEE Transactions on , vol.10, no.6, pp.923,941, Sep 2000

Xiao, F. "DCT-based Video Quality Evaluation", Final Project for EE392J, winter 2000.

Yuan, Y. et al (2010) "Extension of SCTP for Concurrent Multi-Path Transfer with Parallel Subflows", IEEE Wireless Communications and Networking Conference (WCNC), pp.1-6, 18-21 April 2010.

Zhang, X. et al. (2009) "A Cross-layer approach to optimize the performance of Concurrent Multipath Transfer in wireless transmission", Wireless Days (WD), 2009 2nd IFIP, pp.1-5, 15-17 Dec. 2009.

Zhu, X. et al. (2009) "Distributed Rate Allocation Policies for Multihomed Video Streaming Over Heterogeneous Access Networks", IEEE Transactions on Multimedia, vol.11, no.4, pp.752-764, June 2009.

Apêndice A

Priorização dos *frames*

```
int nodeid;
        Node_S *spCurrNode_;
        int higher_priority_node;
        for (spCurrNode_ = sDestList.spHead;
             spCurrNode_ != NULL;
             spCurrNode_ = spCurrNode_->spNext) {

          spNewTxDest = (SctpDest_S *) spCurrNode_->vpData;
          n = Node::get_node_by_address(spNewTxDest->iNsAddr);

          higher_priority_node = GetHigherPriorityNode();
          if (frametype_ == I ) {
             if (higher_priority_node < 0) {
                printf("[ERROR] [%s] There's no priority node.", __FUNCTION__);
             }
             if (n->nodeid() == higher_priority_node) {
                break;
             } else
                continue;
          } else if(frametype_ == P || frametype_ == B) {

             if (higher_priority_node < 0) {
                printf("[ERROR] [%s] There's no priority node.", __FUNCTION__);
             }

             if (n->nodeid() != higher_priority_node) {

                if (firstTime == 1) {
                   ActiveJustOneInterface(spCurrNode, n->nodeid());
                   firstTime = 0;
                }
```

Apêndice B

Configuração do Sistema Fuzzy para a interface UMTS:

```
27
3     1
CWND       3       0     70000
          leftTriangle    4750     9000
          Triangle        9000     15000
          rightTriangle   15000    30000

VRTT         3       0     0.1
          leftTriangle    0.00001    0.002
          Triangle        0.002      0.008500
          rightTriangle   0.008      0.053

VAZAO              3       0       3
          leftTriangle    0       0.1
          Triangle        0.1     0.5
          rightTriangle   0.5     3

PESO     3           0       1
          leftTriangle    0.0     0.3
          Triangle        0.29    0.6
          rightTriangle   0.59    0.9


1         1         1         2
1         1         2         1
1         1         3         2
1         2         1         1
1         2         2         2
1         2         3         2
1         3         1         1
1         3         2         2
1         3         3         3
2         1         1         1
2         1         2         3
2         1         3         2
2         2         1         1
2         2         2         2
2         2         3         3
2         3         1         1
2         3         2         3
2         3         3         3
3         1         1         1
3         1         2         2
3         1         3         3
3         2         1         1
3         2         2         2
3         2         3         3
3         3         1         3
3         3         2         3
3         3         3         3
```

Configuração do Sistema Fuzzy para a interface WiFi:

```
27
3     1
CWND       3       0     70000
        leftTriangle    4750     9000
        Triangle        9000     15000
        rightTriangle   15000     30000

VRTT        3       0     0.1
        leftTriangle    0.00001    0.002
        Triangle        0.002     0.008500
        rightTriangle   0.008      0.053

VAZAO              3       0       6
        leftTriangle    0     0.3
        Triangle        0.3     0.8
        rightTriangle   0.8       1

PESO    3       0       1
        leftTriangle    0.0     0.3
        Triangle        0.29    0.6
        rightTriangle   0.59    0.9


1       1       1       2
1       1       2       1
1       1       3       2
1       2       1       1
1       2       2       2
1       2       3       2
1       3       1       1
1       3       2       2
1       3       3       3
2       1       1       1
2       1       2       3
2       1       3       2
2       2       1       1
2       2       2       2
2       2       3       3
2       3       1       1
2       3       2       3
2       3       3       3
3       1       1       1
3       1       2       2
3       1       3       3
3       2       1       1
3       2       2       2
3       2       3       3
3       3       1       3
3       3       2       3
3       3       3       3
```

Configuração do Sistema Fuzzy para a interface WiMAX:

```
27
3     1
CWND       3       0     70000
        leftTriangle    4750     9000
        Triangle        9000     15000
        rightTriangle   15000     30000

VRTT        3       0     0.1
        leftTriangle    0.00001    0.002
        Triangle        0.002     0.008500
        rightTriangle   0.008      0.053

VAZAO              3       0       6
        leftTriangle    0     0.3
        Triangle        0.3     0.8
        rightTriangle   0.8       3


PESO    3       0       1
        leftTriangle    0.0     0.3
        Triangle        0.29    0.6
        rightTriangle   0.59    0.9


1       1       1       2
1       1       2       1
1       1       3       3
1       2       1       1
1       2       2       2
1       2       3       3
1       3       1       1
1       3       2       2
1       3       3       3
2       1       1       1
2       1       2       3
2       1       3       3
2       2       1       1
2       2       2       2
2       2       3       3
2       3       1       1
2       3       2       3
2       3       3       3
3       1       1       1
3       1       2       2
3       1       3       3
3       2       1       1
3       2       2       2
3       2       3       3
3       3       1       3
3       3       2       3
3       3       3       3
```

Apêndice C

Script que roda todo o programa

```
#!/bin/bash

set -e

################################################
# User Configutations
################################################

#Configurations Variables
NUM_ARG=$#
EVALVID_PATH=/home/k2/Documents/CMT/evalvid/evalvid-2.7

#Give a reference name for your videos and converts
REF_NAME="a01"
#About movie
FPS=24
WIDTH=352
HEIGHT=288
QSCALE="" #"-qscale .1"

SLEEP_TIME=1

NS_CORE=ns-2.29-cmt/ns
NS_PATH=/usr/local/ns-allinone-2.29/$NS_CORE


################################################
#System Names - You shouldn't change this names
################################################
SEND_DUMP="sd_$REF_NAME"
RECV_DUMP="rd_$REF_NAME"
SEND_TRACE="st_$REF_NAME"
TRANSMITED_VIDEO="$REF_NAME"'e'
RECV_WITH_EG="$RECV_DUMP"'eg'
```

```
function clean() {
        rm -f -v *~
        rm -f -v *.txt
        rm -f -v *.tr
        rm -f -v *.nam
        rm -f -v $SEND_DUMP
        rm -f -v $RECV_DUMP
        rm -f -v video1.dat
        rm -f -v *$TRANSMITED_VIDEO*

}

function cleanall() {
        clean
        rm -f -v *"$REF_NAME"*
        rm -f -v *"SctpAgent"*
}

function usage() {
        echo "Evalvid/NS-2 Simulation Script. V 1.0"
        echo "Created by Lorena Marques"
        echo -e "usage: $0 [OPTIONS] \n"

        echo " -b, --before <FILE NAME>                 : Run steps before simulation. name
without (.yuv)"
        echo " -a,--after  <FILE NAME>                  : Run steps after simulation. name
without (.yuv)"
        echo " -e, --eval   <FILE NAME>                 : Execute the video evaluation after
simulation"
        echo " -eg, --error <BER VALUE>                 : Generating Error model in recepted
file."

        echo -e "\nMiscellaneous:"
```

```
	echo " -p, --plot <FILE NAME>                    : Plot 1 curve in a independent
chart"
	echo " -p, --plot <FILE NAME1> <FILE NAME2>      : Plot 2 curves in the same
chart"
	echo " --extract, -ex <FILE NAME>     : Extract informations to plot"
	echo "--ns-all, -all                        : Run all commands at once."
	echo "--plot-args                           : Plot a chart using arguments
provided"


	echo " -c, --clean                        : Clean simulations file"
	echo " -cl, --cleanall                : Clean all files"
	echo " -h, --help                         : Display this help and exit"
	echo " -v, --version                      : Display version info and exit"

}

function about() {

	echo "****************************************"
	echo "$0 v1.0"
	echo "It's a program used to run CMT/Evalvid related command"
	echo "try $0 --help to more info"
	echo "created by Lorena Marques(llm@cin.ufpe.br)"
	echo "****************************************"

}


case $NUM_ARG in
0)
	usage
	;;
1)
	if [ "$1" == "--clean" ] || [ "$1" == "-c" ]; then
		clean
```

```
        elif [ "$1" == "--cleanall" ] || [ "$1" == "-cl" ]; then
                cleanall
        elif [ "$1" == "--help" ] || [ "$1" == "-h" ]; then
                usage
        elif [ "$1" == "--version" ] || [ "$1" == "-v" ]; then
                about
        else

                usage
        fi

        ;;
2)

        if [ "$1" == "--before" ] || [ "$1" == "-b" ]; then

                if [ -f "$2.yuv" ]; then

                        echo "[FFMPEG]: ffmpeg -s cif `$QSCALE` -r 25 -b 64000 -bt 3200 -g
30 -i "$2".yuv -vcodec mpeg4 "$REF_NAME".m4v";sleep $SLEEP_TIME

                        #For more informations: man ffmpeg
                        #-s cif   : Frame size
                        #-qscale        : Quality Scale of video, higher values less quality
                        #-r             : Rate 24 mps
                        #-b                     : bitrate bits/s -- can be omitted
                        #-bt            : bitrate tolerance -- can be omitted
                        #-g                     : set Group of Picture
                        #-i             : File name
                        #-vcodec        : set codec
                        #ffmpeg -s cif -qscale .1  -r 24 -b 64000 -bt 3200 -g 30 -i "$2".yuv -
vcodec mpeg4 "$REF_NAME".m4v
                        $EVALVID_PATH/ffmpeg/ffmpeg -s cif -r 24 -g 20 -bf 3 -qscale 0.1  -
i "$2".yuv -vcodec mpeg4 "$REF_NAME".m4v

                        echo "[MP4BOX]: MP4Box -hint -mtu 1072 -fps "$FPS" -add
"$REF_NAME".m4v "$REF_NAME".mp4"; sleep $SLEEP_TIME
```

```
                        MP4Box -hint -mtu 1056 -fps $FPS -add "$REF_NAME".m4v
"$REF_NAME".mp4

                        echo "[FFMPEG]: ffmpeg -i "$REF_NAME".mp4
"$REF_NAME"_ref.yuv";sleep $SLEEP_TIME
                        $EVALVID_PATH/ffmpeg/ffmpeg -i "$REF_NAME".mp4
"$REF_NAME"_ref.yuv
                        ffmpeg -i "$REF_NAME".mp4 "$REF_NAME"_ref.yuv

                        echo "[MP4TRACE]: mp4trace -f -s 192.168.0.2 12346
"$REF_NAME".mp4 > $SEND_TRACE";sleep $SLEEP_TIME
                        $EVALVID_PATH/mp4trace -f -s 192.168.0.2 12346
"$REF_NAME".mp4 > $SEND_TRACE

                else
                        echo "Error: The file '$2' is not a valid file or doesn't exists"
                fi

        elif [ "$1" == "--after" ] || [ "$1" == "-a" ]; then

                if [ -f "$2.yuv" ]; then
                        #To avoid double free error
                        export MALLOC_CHECK_=0

                        if [ "$1" != "-a" ]; then
                                echo "[PSNR]: psnr $WIDTH $HEIGHT 420 "$2".yuv
"$REF_NAME"_ref.yuv > ref_psnr.txt ";sleep $SLEEP_TIME
                                $EVALVID_PATH/psnr $WIDTH $HEIGHT 420 "$2".yuv
"$REF_NAME"_ref.yuv > ref_psnr.txt

                                #echo "[SSIM] psnr $WIDTH $HEIGHT 422 "$2".yuv
"$REF_NAME"_ref.yuv [ssim]";sleep $SLEEP_TIME
                                #$EVALVID_PATH/psnr $WIDTH $HEIGHT 422 "$2".yuv
"$REF_NAME"_ref.yuv [ssim] > ssim_"$REF_NAME".txt
                        fi
```

```
                echo "[ETMP4]: etmp4 -F -x $SEND_DUMP $RECV_DUMP
$SEND_TRACE $REF_NAME.mp4 $TRANSMITED_VIDEO ";sleep $SLEEP_TIME
                $EVALVID_PATH/etmp4 -F -x $SEND_DUMP $RECV_DUMP
$SEND_TRACE $REF_NAME.mp4 $TRANSMITED_VIDEO

                echo "[FFMPEG]: ffmpeg -i "$TRANSMITED_VIDEO".mp4
"$TRANSMITED_VIDEO".yuv   ";sleep $SLEEP_TIME
                $EVALVID_PATH/ffmpeg/ffmpeg -i "$TRANSMITED_VIDEO".mp4
"$TRANSMITED_VIDEO".yuv


        else
                echo "Error: The file '$2' is not a valid file or doesn't exists"
        fi
    elif [ "$1" == "--eval" ] || [ "$1" == "-e" ]; then

        if [ -f "$2.yuv" ]; then

                echo "[PSNR] psnr $WIDTH $HEIGHT 420 "$2".yuv
"$TRANSMITED_VIDEO".yuv";sleep $SLEEP_TIME
                $EVALVID_PATH/psnr $WIDTH $HEIGHT 420 "$2".yuv
"$TRANSMITED_VIDEO".yuv > psnr_"$TRANSMITED_VIDEO".txt

                #echo "[SSIM] psnr $WIDTH $HEIGHT 422 "$2".yuv
"$TRANSMITED_VIDEO".yuv [ssim]";sleep $SLEEP_TIME
                #$EVALVID_PATH/psnr $WIDTH $HEIGHT 422 "$2".yuv
"$TRANSMITED_VIDEO".yuv [ssim] > ssim_"$TRANSMITED_VIDEO".txt

                echo "[AWK] Delay and Jitter"; sleep $SLEEP_TIME
                awk '{print $3}' delay_"$TRANSMITED_VIDEO".txt |
$EVALVID_PATH/hist - 0 .05 50 > hist_"$REF_NAME".txt

                echo "[MOS] Calculating MOS.";sleep $SLEEP_TIME
                $EVALVID_PATH/mos . ref_psnr.txt 25 > mos.txt

                echo "[MIV] Calculation MIV.";sleep $SLEEP_TIME
                $EVALVID_PATH/miv . > miv.txt
```

```
        else
                echo "Error: The file '$2' doesn't exists"
        fi


elif [ "$1" == "--ns" ] || [ "$1" == "-ns" ]; then

        if [ -f "$2" ]; then
                echo "Executing: $NS_PATH $2"; sleep $SLEEP_TIME
                $NS_PATH $2
        else
                echo "Error: The file '$2' is not a valid file or doesn't exists"
        fi
elif [ "$1" == "--plot" ] || [ "$1" == "-p" ] ; then

        if [ -f "$2" ]; then


                echo "Ploting $2 with gnuplot"

                if [ "$1" == "-p" ]; then
                        #gnuplot -persist << EOF
                        #plot "f0.tr" with linespoints
                        #EOF
                        echo "plot '$2' with points" | gnuplot -persist
                else
                        echo "plot '$2' with lines" | gnuplot -persist

                fi

        else
                echo "Error: The file '$2' is not a valid file or doesn't exists"
        fi

elif [ "$1" == "--extract" ] || [ "$1" == "-ex" ] ; then
```

```
METRIC='metrics'

rm -fr $METRIC
mkdir -p $METRIC
echo "##################################################"
echo "The files will be laid at \"$METRIC\" directory"
echo "##################################################"

        if [ -f "$2" ]; then


                nomes=( "vrtt_wifi" "srtt_wifi" "cwnd_wifi" "ithr_wifi"
                "vrtt_wimax" "srtt_wimax" "cwnd_wimax" "ithr_wimax"
                 "vrtt_umts" "srtt_umts" "cwnd_umts" "ithr_umts" "fuzzy_wifi"
                 "fuzzy_wimax" "fuzzy_umts" "tsn_graph" "priority_umts"
                 "priority_wifi" "priority_wimax")

                x=0;
                #echo "########################################"
                while [ $x != ${#nomes[@]} ]
                do

                  #echo "[checking] "${nomes[$x]}""
                  echo -ne `cat $2 | grep "${nomes[$x]}" >
$METRIC/"${nomes[$x]}".txt`
                  #echo "[ok] File "${nomes[$x]}".txt created."

                  FILE="$METRIC/"${nomes[$x]}".txt"

                  if [[ -s $FILE ]] ; then
                                echo "[Success] $FILE has been created."
                        else
                                #echo "[Fail] Fail to created $FILE"
                                rm -f $FILE
                        fi ;

                  #if [ $x == 0 ]
```

```
                #then
                #       echo "A menor \"unidade\" de dados binÃ¡rios "
                #       echo "tem o nome de "${nomes[$x]}"."
                #       echo ""
                #else
                #       echo "1 "${nomes[$x]}" Ã© o conjunto de bits."
                #fi
                let "x = x +1"
              done
                #echo "("${nomes[@]:2}")"
                #echo "sÃ£o os conjuntos de bits"
                #echo "("${nomes[@]:1:3}")"
                #echo "sÃ£o os conjuntos menores que 32 bits"
                #echo "########################################"




        else
                echo "Error: The file '$2' is not a valid file or doesn't exist"
        fi
else
        usage
fi

;;

3)
        if [ "$1" == "--plot" ] || [ "$1" == "-p" ]; then

                if [ -f "$2" ] && [ -f "$3" ] ; then


                        echo "plot '$2' with lines, '$3' with lines" | gnuplot -persist

                else
```

```
                echo "Error: The files '$2' or '$3' are not valid files or don't exist"
        fi
elif [ "$1" == "--plot-args" ]; then

        if [ -f "$2" ];then
                echo "plot '$2' $3" | gnuplot -persist
        else
                echo "The file $2 doesn't exist."
        fi

elif [ "$1" == "--sum" ] || [ "$1" == "-s" ] ; then

        if [ -f "$2" ]; then

                #awk "{a+=$"$3"}END{print a}" $2
                awk "BEGIN{line=$"$3";max=0;min=1000000} { line=$"$3"; if (max
< line){ max = line }; n+=1; sum+=line; if(min > line){ min = line; } } END{ print
\"min:\",min,\"average:\",sum/n,\"max:\",max}" $2

        else
                echo "Error: The file '$2' is not a valid file or doesn't exists"
        fi

elif [ "$1" == "--ns-all" ] || [ "$1" == "-all" ] ; then

        if [ -f "$2" ]; then

                if [ -f "$3.yuv" ]; then
                        echo "`./$0 --clean`";
                        echo "`./$0 --ns $2`";
                        echo "`./$0 --after $3`";
                        echo "`./$0 --eval $3`";
                        echo "`./$0 --extract monitoring_file.txt`";

                else
                        echo "Error: The file '$3' is not a valid file or doesn't exists"
                fi
```

```
                else
                        echo "Error: The file '$2' is not a valid file or doesn't exists"
                fi

        else
                usage
        fi

        ;;

esac



#Problems faced
#[PSNR]
#The psnr tool doesn't not deals with large files due to 32bits limitation architecture,
#so we needed to make a little hack in Makefile to change the file offset to 64bits
#-D_FILE_OFFSET_BITS=64
#http://learn-from-the-guru.blogspot.com.br/2008/02/large-file-support-in-linux-for-cc.html
```

Apêndice D

Script de configuração da simulação no ns-2.

```
#!/usr/local/ns-allinone-2.29/ns-2.29-cmt/ns


#############################################################################
###
#               SIMULATION OF HANDOVER VERTICAL (UMTS/WIFI)
#                   USING MULTIHOMING FEATURE OF SCTP
#
#
#   Author : Lorena Marques Avelar
#         E-mail: eng_lorena@yahoo.com.br
#
#   Copyrigthy(c) 2009
#
*****************************************************************************
#   *                                                 *
#   * This source is free; redistribution and use in source and binary forms, *
#   * with or without modification, are permitted provided that the following *
#   * conditions describe at the terms of the GNU General Public License as   *
#   * published by the Free Software Foundation; either version 2 of the      *
#   * License, or (at your option) any later version.                         *
#   *                                                 *
#
*****************************************************************************
#############################################################################
###


#
=============================================================
==
# Define options
```

```
#
======================================================================
==
set opt(chan)    Channel/WirelessChannel      ;# channel type
set opt(prop)    Propagation/TwoRayGround     ;# radio-propagation model
set opt(netif)     Phy/WirelessPhy               ;# network interface type
set opt(mac)     Mac/802_11                  ;# MAC type
set opt(ifq)     Queue/DropTail/PriQueue      ;# interface queue type
set opt(ll)      LL                          ;# link layer type
set opt(ant)     Antenna/OmniAntenna          ;# antenna model
set opt(ifqlen)          5000                ;# max packet in ifq
set opt(adhocRouting)    DSDV                ;# routing protocol

set opt(x)       1000                        ;# x coordinate of topology
set opt(y)       1000                        ;# y coordinate of topology
set opt(stop)    900.0                       ;# time to stop simulation
set opt(seed)    0.0
set opt(app-start)       0                   ;# time to start FTP aplication
#
======================================================================
==
set num_wired_nodes      5                   ;# number of wired nodes
set num_bs_nodes         2                   ;# number of base station
set opt(nn)              3                   ;# number of mobilenodes

set lifetime 900

set debug_var 0


#Queue
#Puts this values to high because it have been dropping
Queue/DropTail           set mean_pktsize_ 1024
Queue/DropTail           set limit_ 1000
```

```
#
# ==========================================================================
==
# Main Program
#
# ==========================================================================
==
# create simulator instance
set ns   [new Simulator]

Trace set show_sctphdr_ 1

set output_dir .

set quiet 0

# seed the default RNG
global defaultRNG

#if {$argc == 1} {
#    set seed [lindex $argv 1]
#    if { $seed == "random"} {
#        $defaultRNG seed 0
#    } else {
#        $defaultRNG seed [lindex $argv 1]
#    }
#}

if {$argc != 5} {
        puts "\[ERROR\] We need 5 arguments"
        puts "Usage:"
        puts "\tFirst: wifi inicial priority "
        puts "\tSecond: umts inicial priority"
        puts "\tThird: wimax inicial priority"
        puts "\tFouth: seed for random number generation"
        puts "\tFifth: Protocol type. 0-CMT,1-ACMT,2-ACMT_UC"
        puts "\n\tExemple: ./cmt-evalvid_v6 30 30 30 1 0\n"
```

```
        exit 1
}

#Definindo a Ã¡rea de cobertura(AC) para as estaÃ§Ãµes 802.11: 200m coverage
# Para calcular a AC utilizou-se o programa definido em:
#/usr/local/ns-allinone-2.29/ns-2.29/indep-utils/propagation
Phy/WirelessPhy set Pt_ 0.281838
Phy/WirelessPhy set freq_ 2412e+6
Phy/WirelessPhy set RXThresh_ 5.25089e-10

Phy/WirelessPhy set CSThresh_ [expr 0.9* [Phy/WirelessPhy set RXThresh_]]

#define frequency of RA at base station
Agent/ND set maxRtrAdvInterval_ 1.0
Agent/ND set minRtrAdvInterval_ 0.5
Agent/ND set minDelayBetweenRA_ 0.1
Agent/ND set maxRADelay_ 0.2

#define MAC 802_11 parameters
#Mac/802_11 set bss_timeout_ 1
#Mac/802_11 set pr_limit_ 1.1 ;#for link going down
Mac/802_11 set client_lifetime_ $lifetime ;# since we don't have traffic, the AP would
disconnect the MN automatically after client_lifetime_
Mac/802_11 set basicRate_ 6Mb
Mac/802_11 set dataRate_ 5Mb
Mac/802_11 set bandwidth_ 5Mb


#wireless routing algorithm update frequency (in seconds)
Agent/DSDV set perup_ 1

# Define global simulation parameters

#define DEBUG parameters
Agent/ND set debug_ $debug_var
Agent/MIH set debug_ $debug_var
Agent/MIHUser/IFMNGMT/MIPV6 set debug_ $debug_var
```

```
Agent/MIHUser/IFMNGMT/MIPV6/Handover/Handover1 set debug_ $debug_var

#Quem decide o handover eh o chaveamento da interface
#Agent/MIHUser/IFMNGMT/MIPV6/Handover/Handover1 set case_ 2

Mac/802_11 set debug_ $debug_var


# set up for hierarchical routing
$ns node-config -addressType hierarchical
AddrParams set domain_num_ 4               ;# number of domains
lappend cluster_num 5 1 1 1                ;# number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 1 1 1 1 3 2 2        ;# number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel     ;# of each domain
$ns use-newtrace
set tracefd [open projeto-out.tr w]
set namtrace [open projeto-out.nam w]
$ns trace-all $tracefd                     ;# All traces are written in the tracefd

;# all wireless nam traces are written into the namtrace
$ns namtrace-all-wireless $namtrace $opt(x) $opt(y)

# Create topography object
set topo   [new Topography]

# define the topology's boundaries
$topo load_flatgrid $opt(x) $opt(y)

# create God
# God needs to know the number of all wireless interfaces
create-god [expr $opt(nn) + $num_bs_nodes]

#create wired nodes
set temp {0.0.0 0.1.0 0.2.0 0.3.0 0.4.0}           ;# hierarchical addresses
for {set i 0} {$i < ($num_wired_nodes)} {incr i} {
   set WD($i) [$ns node [lindex $temp $i]]
```

```
}

$WD(0) install-default-ifmanager



############
# UMTS
############

#Configurando UMTS.
$ns set hsdschEnabled_ 1addr
$ns set hsdsch_rlc_set_ 0
$ns set hsdsch_rlc_nif_ 0

# Cofigurando nÃ³ RNC
$ns node-config -UmtsNodeType rnc
set rnc [$ns create-Umtsnode 2.0.0] ;# node id is 0.

#COnfigurando BS do UMTS
$ns node-config -UmtsNodeType bs \
                -downlinkBW 1.5Mbs \
                -downlinkTTI 10ms \
                -uplinkBW 1.5Mbs \
                -uplinkTTI 10ms \
                -hs_downlinkTTI 2ms \
                -hs_downlinkBW 1.5Mbs z



set bsUMTS [$ns create-Umtsnode 2.0.1] ;# node id is 1
if {$quiet == 0} {
   puts "bsUMTS: tcl=$bsUMTS; id=[$bsUMTS id]; addr=[$bsUMTS node-addr]"
}
```

```
#Conectando RNC na BS UMTS
$ns setup-Iub $bsUMTS $rnc 622Mbit 622Mbit 15ms 15ms DummyDropTail 2000

$ns node-config -UmtsNodeType ue \
                -baseStation $bsUMTS \
                -radioNetworkController $rnc



set MH_if1 [$ns create-Umtsnode 2.0.2]
#IFACE UMTS
$MH_if1 set X_ 410.000000000000
$MH_if1 set Y_ 200.000000000000
$MH_if1 set Z_ 0.000000000000


#############

$ns duplex-link $rnc $WD(4) 622Mbit 0.4ms DropTail 1000
$rnc add-gateway $WD(4)


# Create channel
set chan_ [new $opt(chan)]

# Configure for ForeignAgent and HomeAgent nodes
$ns node-config  -mobileIP OFF \
                                -adhocRouting $opt(adhocRouting) \
            -llType $opt(ll) \
            -macType $opt(mac) \
            -ifqType $opt(ifq) \
            -ifqLen $opt(ifqlen) \
            -antType $opt(ant) \
            -propType $opt(prop) \
            -phyType $opt(netif) \
            -channel $chan_ \
            -topoInstance $topo \
```

```
        -wiredRouting ON \
        -agentTrace ON \
        -routerTrace ON \
        -macTrace OF \
        -movementTrace OFF

# Create bsWIFI and bsUMTS
set bsWIFI [$ns node 1.0.0]
#Configurando a BS (bsWIFI)
set bstationMacbsWIFI [$bsWIFI getMac 0]
set AP_ADDR_0 [$bstationMacbsWIFI id]
$bstationMacbsWIFI bss_id $AP_ADDR_0
$bstationMacbsWIFI enable-beacon
$bstationMacbsWIFI set-channel 1



#set FA [$ns node 2.0.0]
#Configurando a BS (FA)
#set bstationMacFA [$FA getMac 0]
#set AP_ADDR_1 [$bstationMacFA id]
#$bstationMacFA bss_id $AP_ADDR_1
#$bstationMacFA enable-beacon
#$bstationMacFA set-channel 1

$bsWIFI random-motion 0

# Position (fixed) for base-station nodes (bsWIFI & bsUMTS) and wired node
$bsWIFI set X_ 70.000000000000
$bsWIFI set Y_ 200.000000000000
$bsWIFI set Z_ 0.000000000000

$bsUMTS set X_ 430.000000000000
$bsUMTS set Y_ 230.000000000000
$bsUMTS set Z_ 0.000000000000

$WD(3) set X_ 250.000000000000
$WD(3) set Y_ 100.000000000000
```

```
$WD(3) set Z_ 0.000000000000

$rnc set X_ 420.000000000000
$rnc set Y_ 220.000000000000
$rnc set Z_ 0.000000000000
# create a mobilenode that would be moving between bsWIFI and bsUMTS.

# note address of MH indicates its in the same domain as bsWIFI.
#$ns node-config -wiredRouting ON
set MH [$ns node 1.0.1]
set MH_if0 [$ns node 1.0.2]

#set MH_if1 [$ns node 2.0.1]
#[$MH_if1 getMac 0] set-channel 1

$MH_if0 color Blue
$MH_if1 color Red


#XXX: WIMAX
#####################################################################
#XXX: WIMAX
#####################################################################
#XXX: WIMAX
#####################################################################



WimaxScheduler/BS set dlratio_ 0.56
Mac/802_16 set debug_ $debug_var
Mac/802_16 set frame_duration_ 0.010
Mac/802_16 set fbandwidth_          7e+6
Mac/802_16 set queue_length_ 5000
Mac/802_16 set client_timeout_ $lifetime ;#to avoid BS disconnecting the SS
```

```
#Diferentes tipos de modulaÃ§Ã£o
#set default_modulation            OFDM_BPSK_1_2
#set default_modulation            OFDM_QPSK_1_2
#set default_modulation            OFDM_QPSK_3_4
#set default_modulation            OFDM_16QAM_1_2
#set default_modulation            OFDM_16QAM_3_4
#set default_modulation            OFDM_64QAM_2_3
set default_modulation            OFDM_64QAM_3_4


#ParÃ¢metros globais para WIMAX
Mac/802_16 set dcd_interval_       5 ;#max 10s
Mac/802_16 set ucd_interval_       5 ;#max 10s
set contention_size               5 ;#for initial ranging and bw
Mac/802_16 set t21_timeout_        0.02 ;#max 10s, to replace the timer for looking at preamble
Mac/802_16 set client_timeout_     $lifetime

# add Wimax nodes
set opt(netif)        Phy/WirelessPhy/OFDM      ;# network interface type 802.16
set opt(mac)          Mac/802_16                ;# MAC type 802.16

# radius =
Phy/WirelessPhy set Pt_ 0.025
Phy/WirelessPhy set RXThresh_ 1.26562e-13 ;#1000m radius
Phy/WirelessPhy set CSThresh_ [expr 0.9*[Phy/WirelessPhy set RXThresh_]]

# configure Access Points
$ns node-config  -mobileIP OFF \
                                 -adhocRouting $opt(adhocRouting) \
            -llType $opt(ll) \
            -macType $opt(mac) \
            -channel [new $opt(chan)] \
            -ifqType $opt(ifq) \
            -ifqLen $opt(ifqlen) \
            -antType $opt(ant) \
            -propType $opt(prop)    \
            -phyType $opt(netif) \
```

```
            -topoInstance $topo \
            -wiredRouting ON \
            -agentTrace ON \
            -routerTrace ON \
            -macTrace OFF  \
            -movementTrace OFF

# configure Base station 802.16
set bstation802_16 [$ns node 3.0.0] ;
$bstation802_16 set X_ 250.0
$bstation802_16 set Y_ 200.0
$bstation802_16 set Z_ 0.0
#$bstation802_16 namattach $namtrace

set clas [new SDUClassifier/Dest]
[$bstation802_16 set mac_(0)] add-classifier $clas
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set bs_sched [new WimaxScheduler/BS]
$bs_sched set-default-modulation $default_modulation
[$bstation802_16 set mac_(0)] set-scheduler $bs_sched
[$bstation802_16 set mac_(0)] set-channel 0

# creation of the wireless interface 802.11
#$ns node-config -wiredRouting OFF ;#\
#               -macTrace ON
set MH_if2 [$ns node 3.0.1]

$MH_if2 random-motion 0                                         ;# disable random motion
$MH_if2 base-station [AddrParams addr2id [$bstation802_16 node-addr]] ;#attach mn to
basestation
$MH_if2 set X_ 10.0
$MH_if2 set Y_ 230.0
$MH_if2 set Z_ 0.0
$MH_if2 random-motion 0
#$MH_if2 namattach $namtrace
set clas [new SDUClassifier/Dest]
[$MH_if2 set mac_(0)] add-classifier $clas
```

```
#set the scheduler for the node. Must be changed to -shed [new $opt(sched)]
set ss_sched [new WimaxScheduler/SS]
[$MH_if2 set mac_(0)] set-scheduler $ss_sched
[$MH_if2 set mac_(0)] set-channel 0


#ND MODULES
# now WIMAX
set nd_bs2 [$bstation802_16 install-nd]
#$nd_bs2 set-router TRUE
$nd_bs2 router-lifetime $lifetime
#$ns at 1 "$nd_bs2 start-ra"

#XXX: WIMAX
#####################################################################
#XXX: WIMAX
#####################################################################
#XXX: WIMAX
#####################################################################


#FIXME mobile IP OFF
#set HAaddress [AddrParams addr2id [$HA node-addr]]
#[$MH set regagent_] set home_agent_ $HAaddress

#set HAaddress [AddrParams addr2id [$bsUMTS node-addr]]
#[$MH_if0 set regagent_] set home_agent_ $HAaddress


#FIXME mobile IP OFF

#$MH base-station [AddrParams addr2id [$HA node-addr]]
#$MH_if0 base-station [AddrParams addr2id [$HA node-addr]]
#$MH_if1 base-station [AddrParams addr2id [$bsUMTS node-addr]]

#Multiplas interbsUMTSces
$ns multihome-add-interface $MH $MH_if0 ;# Interface WIFI
```

```
$ns multihome-add-interface $MH $MH_if1 ;# Interface UMTS
$ns multihome-add-interface $MH $MH_if2 ;# Interface WIMAX


$MH random-motion 0
$MH_if0 random-motion 0



$MH_if0 set X_ 0.000000000000
$MH_if0 set Y_ 200.000000000000
$MH_if0 set Z_ 0.000000000000

# movement of the MH
$MH set X_ 10.000000000000
$MH set Y_ 215.000000000000
$MH set Z_ 0.000000000000

# MH starts to move towards bsUMTS
#$ns at 4.000000000000 "$MH setdest 400.000000000000 215.000000000000
10.000000000000"
#$ns at 4.000000000000 "$MH_if0 setdest 390.000000000000 200.000000000000
10.000000000000"
#$ns at 4.000000000000 "$MH_if2 setdest 400.000000000000 220.000000000000
10000000000000"

#$ns at 4.000000000000 "$MH_if1 setdest 410.000000000000 200.000000000000
50.000000000000"



#give label, shape and color for nodes
$bsWIFI shape "hexagon"
$bsUMTS shape "hexagon"
$bsWIFI color "red"
$bsUMTS color "red"
$WD(0) shape "square"
```

```
$WD(0) color "blue"
$WD(1) shape "square"
$WD(1) color "blue"
$WD(2) shape "square"
$WD(2) color "blue"
$WD(3) shape "square"
$WD(3) color "blue"

$WD(4) shape "square"
$WD(4) color "blue"
$bsWIFI    label bsWIFI
$bsUMTS    label bsUMTS
$rnc label RNC
$rnc color "red"
$rnc shape "square"
$WD(0) label CN
$WD(1) label R1
$WD(2) label R2
$WD(3) label R3
$WD(4) label R4
$MH    label MN
$MH_if0 label WiFi
$MH_if1 label UMTS
#wimax
$bstation802_16 color tan
$bstation802_16 label bsWIMAX
$bstation802_16 shape "hexagon"
$MH_if2 label WIMAX
$MH_if2 color "tan"


# create links between wired and BaseStation nodes
$ns duplex-link  $WD(3) $WD(4)  20Mb  2ms DropTail 2000
$ns duplex-link  $WD(0) $WD(3)  20Mb  2ms DropTail 2000
$ns duplex-link  $WD(1) $bstation802_16  20Mb  2ms DropTail 2000
$ns duplex-link  $WD(2) $WD(3)  20Mb  2ms DropTail 2000
```

```
$ns duplex-link  $WD(2) $WD(3)     20Mb  2ms DropTail 2000
$ns duplex-link  $WD(2) $bsWIFI     20Mb  2ms DropTail 2000

$ns duplex-link  $WD(3) $WD(1) 20Mb  2ms DropTail 2000
#$ns duplex-link-op  $WD(3)  $bstation802_16  orient up

$ns duplex-link-op  $WD(1) $bstation802_16  orient up
$ns duplex-link-op  $WD(2)  $WD(3)  orient left-up
$ns duplex-link-op  $WD(0)  $WD(3)  orient right-up
#$ns duplex-link-op  $bsWIFI     $WD(3)  orient right-down
$ns duplex-link-op  $rnc $WD(4)  orient left-down
$ns duplex-link-op  $rnc $bsUMTS orient right-up


$ns duplex-link-op $WD(3) $WD(4) queuePos 0.5
$ns duplex-link-op $WD(0) $WD(3) queuePos 0.5
$ns duplex-link-op $WD(3) $WD(1) queuePos 0.5
$ns duplex-link-op $WD(2) $WD(3) queuePos 0.5

$ns duplex-link-op $WD(1) $bstation802_16 queuePos 0.5



# define color index
$ns color 0 tan

#*************************************************************


# configure WLAN interface of each Base Satation
#HA
#    set nd_HA [$HA install-nd]
 #   $nd_HA set-router TRUE
  #  $nd_HA router-lifetime 18
   # $ns at 1 "$nd_HA start-ra"
    #set mih_HA [$HA install-mih]
   # set tmp(0) [$HA set mac_(0)] ;#in 802.11 one interface is created
```

```
   #$tmp(0) mih $mih_HA
   #$mih_HA add-mac $tmp(0)


 #UMTS
set nd_rncUMTS [$rnc install-nd]
$nd_rncUMTS set-router TRUE
$nd_rncUMTS router-lifetime $lifetime
$nd_rncUMTS enable-broadcast FALSE

$nd_rncUMTS add-ra-target 2.0.1
set nd_ue [$MH_if1 install-nd]

# now WLAN
set nd_bs [$bsWIFI install-nd]
$nd_bs set-router TRUE
$nd_bs router-lifetime $lifetime
$ns at 0.0 "$nd_bs start-ra"

#set ifmgmt_bs [$bsWIFI install-default-ifmanager]
#set mih_bs [$bsWIFI install-mih]
#$ifmgmt_bs connect-mih $mih_bs
#set tmp2 [$bsWIFI set mac_(0)]
#$tmp2 mih $mih_bs
#$mih_bs add-mac $tmp2

#set nd_mn [$MH_if0 install-nd]

set handover [new Agent/MIHUser/IFMNGMT/MIPV6/Handover/Handover1]
$MH install-ifmanager $handover


#$nd_mn set-ifmanager $handover
#$nd_ue set-ifmanager $handover


set mih [$MH install-mih]
```

```
$handover connect-mih $mih
#$handover nd_mac $nd_mn [$MH_if0 set mac_(0)]



#$handover add-flow $sink $tcp $iface0 1 ;#2000.

$ns node-config -llType UMTS/RLC/AM \
   -downlinkBW 1.5Mbs \
   -uplinkBW 1.5Mbs \
   -downlinkTTI 20ms \
   -uplinkTTI 20ms \
   -hs_downlinkTTI 2ms \
   -hs_downlinkBW 1.5Mbs



#*************************************************************
#set downlink 1

#Criando agente Tx
#set sender [new Agent/SCTP]
#if { $downlink == 1 } {
#$ns attach-agent $WD(0) $sender
#} else {
#$ns multihome-attach-agent $MH $sender
#}


#*************************************************************
#**********************    CMT    ***************************
#**********************    CMT    ***************************
#*************************************************************
set max_fragmented_size   1008

#add SCTP header(12 bytes) and IP header (20 bytes)
```

```
#1024 + 32 = 1054
set packetSize 1024
set mtu 1056


#Criando agente Tx
set sender [new Agent/SCTP/aCMT-PR/Evalvid]
$ns attach-agent $WD(0) $sender
$sender set fid_ 0
$sender set debugMask_ -1
$sender set debugFileIndex_ 0
$sender set numOutStreams_ 1            #SCTP/CMT-PR
$sender set numUnrelStreams_ 1                  #SCTP/CMT-PR
$sender set unordered 0                         #SCTP/CMT-PR

$sender set mtu_ $mtu
$sender set dataChunkSize_ $packetSize
$sender set useCmtReordering_ 0   # turn on Reordering algo.
$sender set useCmtCwnd_ 0         # turn on CUC algo.
$sender set useCmtDelAck_ 0       # turn on DAC algo.
$sender set eCmtRtxPolicy_ 4      # rtx. policy : RTX_CWND
$sender set_tx_filename sd_a01

#Criando agente Rx
set sink [new Agent/SCTP/aCMT-PR/Evalvid]
$ns multihome-attach-agent $MH $sink
$sink set debugMask_ -1
$sink set debugFileIndex_ 1
$sink set mtu_ $mtu
$sink set dataChunkSize_ $packetSize
$sink set initialRwnd_ 65536
$sink set useDelayedSacks_ 1
$sink set useCmtDelAck_ 1
$sink set_rx_filename rd_a01

$ns color 0 Red
$ns color 1 Blue
```

```
if {$argc == 5} {
   $sink protocol_type  [lindex $argv 4]

 }
##$ns attach-agent $MH $sender
#
##Criando agente Rx
#set sink [new Agent/SCTP]
#$sink set class_ 2
#if { $downlink == 1 } {
#$ns multihome-attach-agent $MH $sink
#} else {
#$ns attach-agent $WD(0) $sink
#}

#Conectando o Tx ao Rx
$ns connect $sender $sink

#Application #1
#Camada de aplicaÃ§Ã£o sobre camada de transporte
#No CBR da pra ver melhor os pactoes, pois o FTP utiliza janelas de envio
#set app [new Application/FTP]

#Application #2
#set app [new Application/Traffic/CBR]
#$app set packetSize_ 1100
#$app set interval_ 0.001

#Application #3
#set app [new Application/Telnet]
#$app set interval_ 0.001


#Application #4
#set app [new Application/SctpApp1]
```

```
#$app set interval_ 0.001

#Application #5 - Evalvid

set original_file_name st_a01
set trace_file_name video1.dat
set original_file_id [open $original_file_name r]
set trace_file_id [open $trace_file_name w]

set pre_time 0


while {[eof $original_file_id] == 0} {

   gets $original_file_id current_line

   scan $current_line "%d%s%d%d%f" no_ frametype_ length_ tmp1_ tmp2_
   set time [expr int(($tmp2_ - $pre_time)*1000000.0)]


   if { $frametype_ == "I" } {
      set type_v 1
      set prio_p 0
   }

   if { $frametype_ == "P" } {
      set type_v 2
      set prio_p 0
   }


   if { $frametype_ == "B" } {
      set type_v 3
      set prio_p 0
   }

   if { $frametype_ == "H" } {
```

```
        set type_v 1
        set prio_p 0
    }

    puts  $trace_file_id "$time $length_ $type_v $prio_p $max_fragmented_size"
    set pre_time $tmp2_

}

close $original_file_id
close $trace_file_id
set end_sim_time $tmp2_
puts "End Video Time: $end_sim_time"


set trace_file [new Tracefile]
$trace_file filename $trace_file_name
set app [new Application/Traffic/myCMTEvalvid]
$app attach-agent $sender
$app attach-tracefile $trace_file

set start_time 10.0
set end_time 40.0
$ns at $start_time "$app start"

set total_end_time [expr $end_sim_time + $start_time + $end_time]

$ns at $total_end_time "$app stop"
$ns at $total_end_time "$sink close_tx_file"
$ns at $total_end_time "$sink close_rx_file"
$ns at $total_end_time "stop"

#$ns at 700 "$app stop"
#$ns at 700 "$sink close_rx_file"
#$ns at 700 "$sink close_tx_file"
#$ns at 700 "stop"
```

```
$app attach-agent $sender

#$ns at 0.0 "$app start"
#$ns at $opt(stop).0001 "stop"


#*************************************************************


set dch0 [$ns create-dch $MH_if1 $sink]
#$ns attach-dch $MH_if1 $handover $dch0
#$ns attach-dch $MH_if1 $nd_ue $dch0

#set tmp2 [$MH_if1 set mac_(2)] ;
#$tmp2 mih $mih
#$mih add-mac $tmp2
#set tmp2 [$MH_if0 set mac_(0)]
#$tmp2 mih $mih
#$mih add-mac $tmp2

#Multiface node is receiver
        #$MH attach-agent $sink $MH_if0
#       $handover add-flow $sink $sender $MH_if0 1 ;#2000.

        #CN is transmitter

#*************************************************************

#$ns at 1.0000 "$sender set-primary-destination $MH_if0"
#$ns at 10.0 "$sender set-primary-destination $MH_if2"
#$ns at 1.0000 "$sender force-source $MH_if1"

global wifi_p
global umts_p
global wimax_p
```

```
if {$argc == 5} {
   set wifi_p  [lindex $argv 0]
   set umts_p  [lindex $argv 1]
   set wimax_p [lindex $argv 2]

        puts "########################"
        puts "Setting priority:"
        puts "Setting wifi_p to: $wifi_p"
        puts "Setting wifi_p to: $umts_p"
        puts "Setting wimax to: $wimax_p"

} else {

puts "\[WARNNING\] Wrong number of arguments, you must inform the wifi, umts and wimax
Priority;"
set wifi_p  100
set umts_p  100
set wimax_p 100
puts "Using default values. wifi: $wifi_p, umts: $umts_p and wimax $wimax_p;"

}



#Configura a prioridade das interfaces CMT
# "3" Ã© o Numero MÃnimo de enviados
$sender set-priority $MH_if0 $wifi_p    ;#WIFI
$sender set-priority $MH_if1 $umts_p    ;#UMTS
$sender set-priority $MH_if2 $wimax_p           ;#WIMAX

global random_number

if {$argc == 5} {
        set seed [lindex $argv 3]
        if { $seed == "random"} {
        $defaultRNG seed 0
        } else {
```

```
        $defaultRNG seed [lindex $argv 3]
        }
}

set max 0.50
set min 0.01
set n [new RandomVariable/Uniform]
$n set max_ $max
$n set min_ $min

set error_rate [$n value]



set offset 2.0

set max [expr $total_end_time/2 ]
set min [expr $offset + $start_time ]
set n2 [new RandomVariable/Uniform]
$n2 set max_ $max
$n2 set min_ $min

set begin [$n2 value]

set max [expr $total_end_time ]
set min $begin
set duration_var [new RandomVariable/Uniform]
$duration_var set max_ $max
$duration_var set min_ $min
set duration [$duration_var value]



#Create the error model
set em [new ErrorModel]
$em unit pkt
$ns at $begin "$em set rate_ $error_rate"   ;#  PER = 3%
$ns at $duration "$em set rate_ 0 "
```

```
set mark "(ERROR_MODEL)"

puts "$mark ****************************************"
puts "$mark Error Model:"
puts "$mark Begin_time: [format \"%.3f\" $begin] with rate [format \"%.5f\" $error_rate] "
puts "$mark end_time [format \"%.3f\" $duration] "

$em ranvar [new RandomVariable/Uniform]
$em drop-target [new Agent/Null]


set max 3
set min 0
set n3 [new RandomVariable/Uniform]
$n3 set max_ $max
$n3 set min_ $min

set which_one [$n3 value]



if { $which_one < 1} {

puts "$mark Genarating error at '$WD(2) $bsWIFI"'
$ns link-lossmodel $em $WD(2) $bsWIFI

} elseif {$which_one < 2} {

puts "$mark Genarating error at 'WD(1) bstation802_16"'
$ns link-lossmodel $em $WD(1) $bstation802_16

} else {

puts "$mark enarating error at 'rnc WD(4)"'
$ns link-lossmodel $em $rnc $WD(4)
}
```

```
puts "$mark ****************************************"

#attach the model to the link
#$ns link-lossmodel $em $WD(2) $bsWIFI
#$ns link-lossmodel $em $WD(1) $bstation802_16
#$ns link-lossmodel $em $rnc $WD(4)



############################
#Fuuuuuzyyyyyyyyyy
#XXX: WIFI
set fuzzy_wifi [new FuzzySystem]
$fuzzy_wifi set fuzzifyFlag 0
$fuzzy_wifi set outputCombinationFlag 1
$fuzzy_wifi set fuzzifyFlag 1
$fuzzy_wifi result-file "result-wifi.txt"
$fuzzy_wifi read-input-file "cmt-rules-wifi.rules"
$fuzzy_wifi print-fuzzy-rule-set "fuzzy-rules-wifi.txt"
$sink install-fuzzy-system $fuzzy_wifi $MH_if0

#XXX: WIMAX
set fuzzy_wimax [new FuzzySystem]
$fuzzy_wimax set fuzzifyFlag 0
$fuzzy_wimax set outputCombinationFlag 1
$fuzzy_wimax set fuzzifyFlag 1
$fuzzy_wimax result-file "result-wimax.txt"
$fuzzy_wimax read-input-file "cmt-rules-wimax.rules"
$fuzzy_wimax print-fuzzy-rule-set "fuzzy-rules-wimax.txt"
$sink install-fuzzy-system $fuzzy_wimax $MH_if2

#XXX: UMTS
set fuzzy_umts [new FuzzySystem]
$fuzzy_umts set fuzzifyFlag 0
$fuzzy_umts set outputCombinationFlag 1
$fuzzy_umts set fuzzifyFlag 1
$fuzzy_umts result-file "result-umts.txt"
```

```
$fuzzy_umts read-input-file "cmt-rules-umts.rules"
$fuzzy_umts print-fuzzy-rule-set "fuzzy-rules-umts.txt"
$sink install-fuzzy-system $fuzzy_umts $MH_if1


#$ns at 40.0000 "$sink set-priority $MH_if0 0" ;#WIFI
#$ns at 40.0000 "$sink set-priority $MH_if2 100" ;#WIMAX

#$ns at 40.0000 "$sink set-priority $MH_if2 1 ;#WIMAX"
#$ns at 40.0000 "$sink set-priority $MH_if0 0 ;#WIFI"

# Define initial node position in nam
# 12 defines the node size in nam
#$ns initial_node_pos $MH 12

#######################################
#XXX Criando InterferÃªncia na camada de transporte
#######################################
############TODO: TRÃ • FEGO WIMAX
set udp0 [new Agent/UDP]
$ns attach-agent $WD(0) $udp0
$udp0 set class_ 2
$ns color 2 Red

#traffic
set cbr [new Application/Traffic/CBR]
#$cbr set packetSize_ 1100
#$cbr set interval_ 0.005
$cbr attach-agent $udp0

#sink Rx
set null [new Agent/LossMonitor]
$ns attach-agent $MH_if2 $null

#conect
$ns connect $udp0 $null
#$ns at 10.0 "$cbr start"
```

```
#$ns at 60.0 "$cbr stop"


############TODO: TRÃ • FEGO WIFI
set udp1 [new Agent/UDP]
$ns attach-agent $WD(2) $udp1
$udp1 set class_ 3
$ns color 3 Blue

#traffic
#set cbr1 [new Application/Traffic/CBR]
#$cbr1 set packetSize_ 1400
#$cbr1 set interval_ 0.00250
#$cbr1 attach-agent $udp1
#

set traffic [new Application/Traffic/Exponential]
$traffic set packetSize_ 800
$traffic set interval_ 0.003
$traffic set burst_time_ 0.005
$traffic set idle_time_ 0.005
$traffic set rate_ 9M
$traffic attach-agent $udp1

#sink Rx
set null1 [new Agent/LossMonitor]
$ns attach-agent $MH_if0 $null1

#conect
$ns connect $udp1 $null1
#$ns at 20.0 "$traffic start"
#$ns at 40.0 "$traffic stop"

#$ns rtmodel-at 80.0 down $WD(3) $bstation802_16
#$ns rtmodel-at 85.0 up $WD(3) $bstation802_16
```

```
############TODO: TRÃ • FEGO UMTS
#Criando agente Tx
set udp2 [new Agent/UDP]
$ns attach-agent $WD(0) $udp2
$udp2 set class_ 4
$ns color 4 Yellow

#traffic
set cbr2 [new Application/Traffic/CBR]
$cbr2 set packetSize_ 900
$cbr2 set interval_ 0.005 ;#900 (* 9 bits) / 0.001s = 1.6Mbps ; 1B = 9b
$cbr2 attach-agent $udp2

#sink Rx
set null2 [new Agent/LossMonitor]
$ns attach-agent $MH_if1 $null2

#conect
$ns connect $udp2 $null2
#$ns at 50.0 "$cbr2 start"
#$ns at 50.0 "$cbr2 stop"


########################################
#XXX
########################################
#VAZÃO NO RECEPTOR
set f0 [open f0.tr w]
set f1 [open f1.tr w]
set f2 [open f2.tr w]

proc record {} {
        global sink f0 f1 f2 null1
        #Peganod um instancia do simulador
        set ns [Simulator instance]
        #tempo para que a funÃ§Ã£o seja chamada novamente
        set time 0.5
```

```
        #Pegando o numero de bytes que chegou no receptor
        set bw0 [$sink set bytes_]
                set bw1 [$null1 set bytes_]

        #puts "BW0 = $bw0"
        #Pegando o tempo corrente
        set now [$ns now]
        #Calculando a largura de banda
        puts $f0 "$now [expr $bw0/$time*8/1000000]"
        #puts $f1 "$now [expr $bw1/$time*8/1000000]"

        #Resetando os bytes recebidos
        $sink set bytes_ 0
                $null1 set bytes_ 0

        #Re-escalonando o processo
        $ns at [expr $now+$time] "record"
}
$ns at 0.0 "record"


# Tell all nodes when the simulation ends
$ns at $opt(stop).0 "$bsWIFI reset";
$ns at $opt(stop).0 "$MH reset";
$ns at $opt(stop).0 "$MH_if0 reset";
$ns at $opt(stop).0 "$MH_if1 reset";
$ns at $opt(stop).0 "$MH_if2 reset";

$ns at 0 "[eval $MH_if1 set mac_(2)] disconnect-link" ;#UMTS UE
$ns at 0.1 "[eval $MH_if1 set mac_(2)] connect-link"  ;#umts link

$ns at $opt(stop).0 "$bsUMTS reset";
#$ns at $opt(stop).0002 "$ns halt"
#$ns at $opt(stop).0001 "stop"

proc stop {} {
```

```
    global ns tracefd namtrace

    close $tracefd
    close $namtrace

        exec nam projeto-out.nam &
        exit 0
}

$ns run                 ;# starts the simulator
```