

Avaliação de parâmetros do SCTP para transporte de tráfego VoIP em cenários com perdas

Rudolfo A. E. Rüncos¹, Eduardo P. Ribeiro¹

¹Departamento de Engenharia Elétrica – Universidade Federal do Paraná
UFPR – Curitiba – PR – Brasil

rrscar@gmail.com, edu@eletrica.ufpr.br

Abstract. *In multi-homed systems, an optimized use of the available paths can lead to resilience and quality gain for multimedia services. SCTP (Stream Control Transmission Protocol) is a transport protocol that allows multi-homed connections. Its default parameters are not best configured for real time traffic. This work focus on two parameters which strongly influence SCTP's behaviour: PMR (Path.Max.Retrans) and RTOMax (Retransmission TimeOut superior limit). Both are tested for multiple values through simulations of multi-homed hosts with background traffic and bursty losses. The impact of the delay-centric path selection algorithm is also evaluated on simulated scenarios. Results indicate that for VoIP calls transport, low values are recommended for PMR and for RTOMax.*

Resumo. *Em sistemas multi-abrigados, a otimização no uso das rotas disponíveis pode trazer ganhos de resiliência e qualidade a serviços multimídia. O SCTP (Stream Control Transmission Protocol) é um protocolo de transporte que suporta conexões multi-abrigadas. Seus parâmetros padrão, no entanto, não apresentam os valores mais adequados para transporte de tráfego em tempo real. Este trabalho estuda os parâmetros PMR (Path.Max.Retrans) e RTOMax (limite superior do Retransmission TimeOut), que apresentam forte influência sobre o comportamento do SCTP. São realizadas simulações envolvendo perdas em rajadas nas quais são testados diversos valores para PMR e para o RTOMax. Também é avaliado o impacto do uso do algoritmo de seleção de caminhos baseado em atraso (delay-centric) nos cenários simulados. Os resultados indicam que baixos valores para ambos são desejáveis para o transporte de chamadas VoIP.*

1. Introdução

Dispositivos multi-abrigados estão se tornando comuns devido à popularização de diversos meios de acesso, como WiFi, UMTS e WiMax. Um terminal ou sistema é dito multi-abrigado quando ele possui mais de uma interface de rede, física ou lógica, disponível para comunicação. Simultaneamente, há uma crescente demanda por serviços VoIP (*Voice over Internet Protocol*). Em sistemas multi-abrigados, mecanismos que realizem trocas de caminhos automaticamente podem beneficiar tais serviços.

A camada mais propícia para a implementação de mecanismos que envolvam trocas de rotas é a camada de transporte [Eddy 2004]. Implementar a troca de rotas na camada de transporte não exige mudanças no núcleo da rede, apenas nos pontos finais (terminais), o que torna essa solução mais viável economicamente. No entanto, o protocolo de transporte mais largamente utilizado por aplicações em tempo real, o UDP, não

dispõe de recursos adequados para se implementar esse tipo de solução. Nesse sentido, o SCTP surge como uma alternativa interessante, pois ele suporta conexões entre terminais multi-abrigados.

Chamadas VoIP são bastante sensíveis ao atraso na rede, sofrendo grande degradação se este for elevado. Mecanismos que realizam trocas de rotas automaticamente baseados no atraso que os caminhos apresentam já foram propostos [Kelly et al. 2004]. Eles atuam sobre o SCTP e os resultados provaram a eficácia de tais métodos. A esse tipo de mecanismo foi dado o nome de *delay-centric*, e assim também ele será referenciado a partir deste ponto neste artigo.

Além do atraso, outra questão relevante, especialmente quando se trata de enlaces sem-fio, é o impacto das perdas de pacotes em transmissões VoIP. O SCTP apresenta por padrão um mecanismo de detecção de falhas que é sensível a perdas. O funcionamento desse mecanismo é ajustado por parâmetros que podem ser configurados durante o estabelecimento de uma associação, que é o equivalente do SCTP a uma conexão TCP. Entretanto, a configuração padrão desses parâmetros não é adequada ao transporte de tráfego em tempo real.

O objetivo deste trabalho é avaliar através de simulações o impacto no desempenho do SCTP para o transporte de tráfego VoIP em terminais multi-abrigados causado por dois parâmetros: o *PMR* (*Path.Max.Retrans*) e o *RTOMax* (limite superior do parâmetro *Retransmission TimeOut*). Como esses parâmetros estão relacionados com o mecanismo sensível a perdas do SCTP, os cenários simulados envolvem perdas variáveis. Os cenários também apresentam tráfegos de fundo variáveis, que proporcionam diferentes condições de atraso. Avalia-se também o mecanismo *delay-centric* [Kelly et al. 2004] de forma a verificar como a sua presença interfere na configuração dos parâmetros sob estudo. A métrica de desempenho utilizada é o MOS (*Mean Opinion Score*) 3.4 estimado a partir do E-model 3.4.

Este artigo está estruturado da seguinte forma: A seção 2 consiste de uma breve revisão sobre trabalhos relacionados; a seção 3 aborda conceitos teóricos envolvidos neste trabalho; a descrição dos cenários simulados é apresentada na seção 4; os resultados aparecem na seção 5 e, por fim, as conclusões são apresentadas na seção 6.

2. Trabalhos Relacionados

Kelly et al. [Kelly et al. 2004] estudou o uso do SCTP para realização de trocas de rotas em redes sem fio. O autor propôs o algoritmo *delay-centric* para a tomada de decisão de troca de rotas baseada no atraso. Apesar de seus testes não envolverem cenários com perdas, o autor afirma que é viável a utilização do *delay-centric* no SCTP para aplicações em tempo real em terminais móveis. Gavriloff [Gavriloff 2009] ampliou os estudos sobre o *delay-centric*, implementando sua versão para o software de simulação de redes NS2 (*Network Simulator 2*).

Fracchia et al. [Fracchia et al. 2007] propôs um novo método denominado WiSE para estimação de congestionamento e falhas de transmissão em cenários sem fio com perdas em rajadas. Seu enfoque, diferentemente de Kelly et al. [Kelly et al. 2004], foi a transferência volumosa de dados. Através do uso de técnicas de estimação de largura de banda o autor afirma que o método permite um maior *throughput* com uma maior robustez a falhas, além de ser compatível com o TCP (*Transmission Control Protocol*).

Para levar em conta tanto atraso como perdas em transmissões de tráfegos multimídia, Fitzpatrick et al. [Fitzpatrick et al. 2009] propôs o uso de um método de troca de

rotas que ele denominou ECHO (*Endpoint Centric Handover*). Esse método tem como parâmetro de decisão a qualidade do tráfego VoIP, estimada a partir do E-model. Através do ECHO, tanto o atraso como as perdas são levados em conta indiretamente na tomada de decisão pelo melhor caminho.

Estudos analíticos mostram que o mecanismo de detecção de falhas padrão do SCTP não realiza uma boa estimativa do tempo de falha [Budzisz et al. 2007]. No entanto, a escolha adequada dos parâmetros do SCTP pode melhorar sua performance em cenários com perdas. Qiao et al. [Qiao et al. 2008] sugere baixos valores de *PMR* (0 ou 1) para melhorar a velocidade de transferência de dados volumosa. Caro et al. [Caro et al. 2006] chega à mesma conclusão e ainda propõe uma política alternativa de retransmissão. Apesar do aumento de falsos-positivos para falhas causado por valores mais baixos de *PMR*, o aumento na velocidade de reação a falhas de caminho compensa e garante um ganho do *throughput*.

Outro parâmetro que tem grande influência sobre o comportamento do SCTP em relação a falhas de caminho é seu parâmetro *RTOMax*. Fallon et al. [Fallon et al. 2008] demonstra o bloqueio de transmissão causado em transmissões com perdas devido ao crescimento exagerado do *RTO*. Esse comportamento do SCTP, que a autora chama de contra-intuitivo, pode ser minimizado utilizando-se um valor menor de *RTOMax*, limitando assim o crescimento do *RTO* (*Retransmission Timeout*). Em outro trabalho [Fallon et al. 2009] a autora propõe um novo método (AORAN) para calcular o *RTO* e evitar os problemas causados por seu aumento excessivo em redes sem fio.

3. Revisão de Conceitos

3.1. SCTP

O SCTP, tal como o TCP, é um protocolo de transporte orientado a conexão padronizado em outubro de 2000 pela RFC2960 [Stewart et al. 2000]. Posteriormente outros documentos estenderam as funcionalidades deste protocolo [Stewart et al. 2006] [Stewart et al. 2004]. O SCTP apresenta controle de congestionamento e permite a entrega confiável de dados. Além disso, ele permite transmissão ordenada ou não ordenada de dados, fornece serviço de fluxos múltiplos (*multi-streaming*) e, devido à extensões (*Partial Reliability SCTP*, ou PR-SCTP) [Stewart et al. 2004], permite também entrega não confiável de dados, semelhante ao UDP, ou mesmo com confiabilidade parcial.

Uma conexão SCTP entre dois terminais é denominada de associação. Uma mesma associação SCTP permite o uso de mais de um endereço IP para comunicação, o que significa a possibilidade do uso de mais de uma rota para comunicação. Ou seja, o SCTP fornece suporte para associações multi-abrigadas. Essa capacidade do SCTP é o que torna este protocolo uma excelente alternativa para a implementação de sistemas automáticos de troca de rotas. Além disso, ele tem por padrão implementado um mecanismo que realiza a troca de rotas quando ocorrem falhas, de forma a aumentar a resiliência da associação.

Ao iniciar uma associação entre dois terminais multi-abrigados, um dos endereços IP de cada terminal envolvido na associação é escolhido como primário. Assim, uma rota se torna a primária enquanto as demais tornam-se secundárias. A menos que a rota primária se torne inativa, todo tráfego da associação irá fluir por essa rota. Caso o caminho primário se torne inativo, o SCTP irá enviar os pacotes através de uma rota alternativa (se disponível) até o restabelecimento do caminho primário. Um caminho se torna inativo se ocorrerem sucessivas ausências de recebimento de confirmação de envio de dados, por

exemplo durante uma rajada de perdas no caminho em questão.

3.1.1. *RTT*, *SRTT* e *RTO*

Para melhor entender o mecanismo de detecção de falhas do SCTP, é necessário antes entender a variável denominada *Retransmission Timeout (RTO)*. Cada caminho i apresenta o seu próprio RTO_i . O RTO determina o tempo máximo que o SCTP aguarda pela resposta a um pacote antes de retransmiti-lo. No início de uma associação o RTO é inicializado com o valor de RTO_{ini} , que por padrão é de $3s$.

Na sequencia, o valor do RTO é atualizado cada vez que uma medida de *Round Trip Time (RTT)* é realizada para o caminho em questão. O RTT é medido do instante em que um pacote é enviado até o instante em que seu ACK é recebido. O RTT é aferido no caminho primário a partir dos pacotes de dados e seus ACKs. Já nos caminhos secundários isso se dá através dos pacotes de pulsação e ACK de pulsação.

A partir da segunda medida de RTT , quando uma nova medida RTT é realizada, o RTO é atualizado como segue [Stewart and Xie 2001]:

$$RTTVAR = (1 - \beta).RTTVAR_{anterior} + \beta. |SRTT_{anterior} - RTT| \quad (1)$$

$$SRTT = (1 - \alpha).SRTT_{anterior} + \alpha.RTT \quad (2)$$

$$RTO = SRTT + 4.RTTVAR \quad (3)$$

onde o $RTTVAR$ indica a variação nas medidas do RTT e o $SRTT$ é a medida do RTT suavizada. β e α são constantes cujos valores recomendados são $1/4$ e $1/8$ respectivamente. Existem ainda limites para o valor de RTO . Este não pode ser inferior a RTO_{min} e nem superior a RTO_{max} . Por padrão, $RTO_{min} = 1s$ e $RTO_{max} = 60s$.

3.1.2. Pacotes de pulsação

Durante o tempo em que a associação existir, o SCTP irá avaliar periodicamente a disponibilidade de todas as rotas de forma a manter sempre atualizado o estado (ativa ou inativa) de cada uma delas. No caminho primário isso se dá através dos pacotes de ACK enviados pelo terminal receptor. Já nos caminhos alternativos, essa avaliação é realizada através de pacotes chamados de pulsação. A resposta a esses pacotes, chamada de ACK da pulsação, indica que o caminho se encontra ativo. O intervalo de tempo entre o envio de pacotes de pulsação, H_i , é calculado para cada caminho da seguinte forma [Stewart and Xie 2001]:

$$H_i = RTO_i + HBinterval.(1 + \delta) \quad (4)$$

Onde RTO_i é o valor atual do RTO para o caminho i , δ é um valor aleatório entre $-0,5$ e $0,5$ e $HBinterval$ é um parâmetro constante cujo valor padrão é $60s$. Toda vez que um pacote de dados ou de pulsação é enviado, um temporizador é inicializado com o ultimo valor calculado de H_i . Caso o temporizador expire, um novo pacote de pulsação é enviado. Isso ocorre também em caminhos inativos. Assim, se um ACK de pulsação é recebido em um caminho inativo, o estado desse caminho será modificado para ativo.

3.1.3. Mecanismo de detecção de falhas

Cada vez que um pacote é enviado, um temporizador com o valor de RTO_i é iniciado. Se o temporizador expirar e não for recebido um ACK do pacote enviado, é dito que ocorreu um estouro de *timeout*. Quando ocorre um estouro de *timeout*, o contador de estouros de *timeout* do caminho em questão é incrementado, o RTO_i é duplicado, a janela de transmissão é reduzida e o pacote é retransmitido, caso o fluxo em questão esteja configurado para transmissão confiável. Se o SCTP estiver configurado para transmissão não confiável (PR-SCTP), ao invés de retransmitir o pacote que não teve ACK, o SCTP irá transmitir o pacote seguinte. Esse processo irá se repetir até que um ACK seja recebido ou até que o valor do contador de estouros de *timeout* seja maior do que o valor do parâmetro *Path.Max.Retrans* (PMR), que por padrão é de 5.

No primeiro caso, ao receber um ACK o contador de estouros de *timeout* é zerado e a transmissão segue normalmente. Já no caso de o valor do contador ser maior do que PMR , o estado do caminho é alterado para inativo e um caminho alternativo é escolhido para a transmissão dos dados. Dessa forma, o PMR determina a tolerância do mecanismo de detecção de falhas à ausência de confirmações de recebimento de dados. Em outras palavras, ele regula a sensibilidade desse mecanismo. Quanto menor for o valor do PMR , menor será a tolerância do SCTP a perdas e mais rápida será a troca de rotas. Seu valor padrão ($PMR = 5$) é um valor alto e pouco sensível a perdas. Para $PMR = 0$ não há tolerância, ou seja, qualquer ausência de recebimento de ACK inativa o caminho e causa a troca de rota.

3.2. Delay-centric

O algoritmo do *delay-centric* foi proposto originalmente por Kelly et al. [Kelly et al. 2004]. Esse mecanismo foi desenvolvido para aumentar a qualidade no transporte de dados sensíveis ao atraso da rede, como no caso do VoIP. Ele se baseia no monitoramento do atraso apresentado pelos caminhos envolvidos na associação SCTP e na troca automática para o caminho que apresente um menor atraso, de forma transparente às camadas superiores. Esse monitoramento do atraso é feito pela aferição do *SRTT* dos diferentes caminhos, através dos pacotes de pulsação nos caminhos alternativos ou do próprio tráfego útil no caso do caminho primário. Caminhos que apresentem um menor *SRTT* são redefinidos como primários. Nesse trabalho será utilizada a implementação feita por Gavriloff [Gavriloff 2009].

3.3. O modelo de perdas de Gilbert

O modelo de erros de Gilbert consiste de uma cadeia de Markov de 2 estados, um estado “bom” ou sem perdas, e outro estado “ruim” ou com perdas, conforme ilustrado pela figura 1. Três parâmetros definem esse modelo, que são $P1$, $P2$ e L . $P1$ é a probabilidade de mudança para o estado ruim, $P2$ é a probabilidade de mudança para o estado bom e L é a probabilidade de perda de pacote durante o estado “ruim”. Durante o estado “bom” não há perdas de pacotes.

Nesse tipo de modelo, o sorteio para troca de estado deve ser realizado periodicamente. O instante em que o sorteio é realizado pode ser cada vez que um bit é gerado, cada vez que um pacote é gerado, ou a intervalos de tempo constantes (independente da geração de pacotes). Para este trabalho escolheu-se o último caso, em que o sorteio para troca de estado ocorre a intervalos de tempo constantes $\Delta t = 5ms$, pois assim é possível extrair fórmulas simples para os fatores de interesse deste trabalho. Um deles é a perda

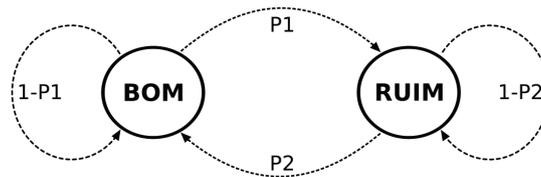


Figura 1. Modelo de Gilbert: uma cadeia de Markov de dois estados.

média (P_{med}) que, supondo que o modelo atue durante um tempo que tenda ao infinito, é dada por:

$$P_{med} = L \left(\frac{P1}{P1 + P2} \right) \quad (5)$$

3.4. MOS e o E-model

A métrica definitiva de desempenho para qualquer ligação telefônica é o MOS (*Mean Opinion Score*) [ITU-T 1996], aferido a partir de testes subjetivos realizados em condições adequadas e com um número significativo de indivíduos. Para se obter o MOS, cada indivíduo avalia a qualidade da chamada telefônica com uma nota de 1 a 5. O MOS é então calculado como a média aritmética de todos os indivíduos que avaliaram a chamada, resultando em um valor entre 1 e 5 que indica a qualidade da chamada.

Entretanto, essas avaliações subjetivas são custosas e, para os casos nos quais são necessárias muitas medições, tornam-se inviáveis. É para resolver esse tipo de problema que o E-model foi desenvolvido. O E-model é um padrão definido pela ITU-T como G.107 [ITU-T 2000] que utiliza várias informações e medições objetivas da rede para calcular um valor escalar de qualidade de chamada telefônica, o fator R , da seguinte forma:

$$R = R_0 - I_S - I_D - I_{e,eff} + A; \quad (6)$$

R_0 : Relação sinal-ruído básica;

I_S : Fator degenerativo simultâneo de codificação;

I_D : Fator degenerativo devido a transmissão, incluindo atraso;

$I_{e,eff}$: Fator degenerativo de equipamento, incluindo perdas de pacotes VoIP [ITU-T 2003];

A : Fator de vantagem.

Este fator R , por sua vez, pode ser mapeado em valores de MOS de acordo com a seguinte equação:

$$MOS = \begin{cases} 1 & R \leq 0 \\ 1 + 0.035R + R(R - 60)(100 - R) \times 7 \times 10^{-6} & 0 < R < 100 \\ 4.5 & R \geq 100 \end{cases} \quad (7)$$

Essa é uma avaliação objetiva que não substitui a avaliação subjetiva do MOS, mas serve como parâmetro de comparação para qualidade de voz na ausência de uma medida subjetiva.

4. Configuração das Simulações

Para executar as simulações foi utilizado o NS2 (*Network Simulator 2*), um *software* aberto de simulação de redes de computadores de eventos discretos largamente utilizado no meio acadêmico. A finalidade das simulações é avaliar a influência dos parâmetros *PMR* e, principalmente, *RTOMax* na qualidade das chamadas VoIP em diferentes condições de tráfego de fundo e perdas. Também são variados o estado do *delay-centric* de forma a estender a dimensão das análises.

A topologia simulada consta na figura 2. Nela, todos os enlaces são *full-duplex*, apresentam velocidades iguais a *1Mbps*, atraso de propagação de *10ms* e filas de saída com limite de 50 pacotes. Os agentes Sctp são conectados aos nós 8 e 9. Em todas as simulações o Sctp é configurado para tráfego não-confiável (PR-Sctp) e não-ordenado, e a opção de SACKs atrasados é desativada.

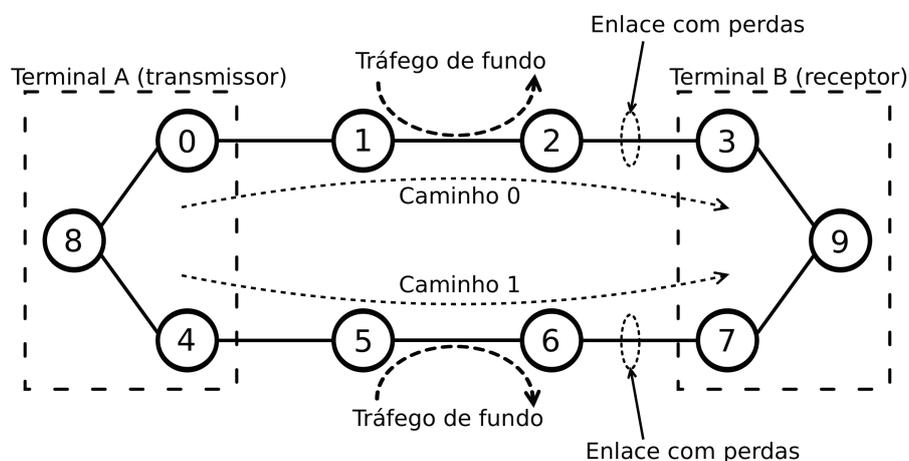


Figura 2. Topologia utilizada em todas as simulações. Cada link é *full-duplex*, com taxa de transmissão de *1Mbps* e *10ms* de atraso de propagação. O tráfego de fundo ocorre entre os nós 1 - 2 (caminho 0) e 5 - 6 (caminho 1). As perdas ocorrem nos enlaces entre os nós 2 - 3 e 6 - 7.

São mantidos constantes para todas as simulações, mas com valores diferentes do Sctp padrão os parâmetros $HBinterval = 0,1s$ e $RTOMin = 0,02s$. Para os demais parâmetros do Sctp são mantidos os valores padrão, com exceção do *PMR* e do *RTOMax* que são parâmetros ajustados no início da simulação e tem seus valores variados entre simulações distintas. No início de qualquer simulação, o caminho 0 é o escolhido como primário pelo Sctp.

O tráfego VoIP simulado representa o padrão G.711 [ITU.T 1988]. Dessa forma, se utiliza um gerador de tráfego CBR conectado ao agente Sctp do nó 8 que gera pacotes de *160bytes* a cada *20ms*. É feito uso de um *buffer-anti-jitter* fixo de *100ms* (5 pacotes). É também assumido o uso do algoritmo de PLC (*Packet Loss concealing*) conforme apêndice 1 do padrão G.711 [ITU-T 1999]. O *buffer-anti-jitter* assume o instante de chegada do primeiro pacote como sendo o tempo inicial. Qualquer pacote que apresentar um atraso superior a *100ms* em relação ao tempo previsto não será entregue à aplicação.

Há tráfego de fundo entre os nós 1 e 2, e 5 e 6. Os tráfegos utilizam UDP com pacotes de 500 bytes gerados a intervalos variáveis que seguem uma distribuição exponencial. Dessa forma, a fila formada é do tipo M/D/1. O atraso médio adicional que essa

fila imprime nos pacotes VoIP para cada caminho é fornecido como parâmetro de entrada da simulação. Assim, o tráfego de fundo tem seu intervalo médio entre pacotes calculado a partir deste atraso médio desejado. O atraso final experimentado por cada pacote é, portanto, o atraso de propagação e transmissão dos 3 enlaces somados (aproximadamente $35ms$ e constante para todas as simulações) mais o atraso sofrido na fila dos nós 1 ou 5, dependendo do caminho que o tráfego VoIP estiver seguindo, devido ao tráfego de fundo. O tráfego VoIP é levado em conta no cálculo da ocupação do enlace.

As perdas ocorrem entre os nós 2 e 3, e 6 e 7. Dessa forma os pacotes do tráfego de fundo não são perdidos, e os atrasos não são afetados pelas perdas. Apenas P_{med} é utilizado como parâmetro de entrada para as simulações, e a partir deste valor são determinados os valores dos parâmetros do modelo de Gilbert ($P1$, $P2$ e L) de forma que as durações médias dos estados estejam dentro das faixas desejadas. Procurou-se utilizar valores próximos aos utilizados por [Bhagwat et al. 1997]. Dessa forma, as durações médias dos estados estão entre $1s$ e $10s$ para o estado sem perdas, e entre $25ms$ e $250ms$ para o estado com perdas. No estado com perdas, $L = 0,8$, exceto para o caso de 10% de perdas, em que ele é $L = 0,85$. O intervalo de tempo entre

Em todas as simulações o tráfego VoIP é iniciado em um instante aleatório entre $10s$ e $20s$ de simulação e tem duração de $120s$. Essa duração é suficiente para que os mecanismos sob estudo possam atuar diversas vezes ao longo de uma simulação, além de ser uma duração razoável para uma ligação telefônica. Todos os casos simulados são executados 100 (cem) vezes, cada vez com uma sub-faixa diferente para o gerador de números aleatórios (RNG), de forma a gerar uma média dos valores do MOS para cada caso com intervalos de confiança estreitos.

São 8 os parâmetros que devem ser especificados para cada simulação. A tabela 1 lista todos esses parâmetros com os respectivos valores utilizados nas simulações. Todas as combinações possíveis foram simuladas.

Parâmetro de simulação	Valores simulados
Estado do <i>delay-centric</i>	ligado e desligado
Atraso médio na fila do caminho 0	5ms, 25ms e 75ms
Atraso médio na fila do caminho 1	5ms, 25ms e 75ms
PMR	0, 1, 2 e 5
Média de perdas do caminho 0	0.5%, 2% e 10%
Média de perdas do caminho 1	0.5%, 2% e 10%
RTO_{max}	10s, 5s, 2s, 1s, 0.5s, 0.2s, 0.12s, 0.1s, 0.08s, 0.05s e 0.02s
Sub-faixa de RNG	0 a 99

Tabela 1. Parâmetros de entrada das simulações. Cada combinação resulta em uma simulação diferente.

5. Resultados

Um dos problemas que resultam numa queda de qualidade da chamada VoIP diz respeito à interrupção da transmissão. Esse problema pode se dar devido a um aumento excessivo do RTO em casos de $PMR = 5$ ou devido ao estabelecimento do estado dormente da associação em casos de PMR muito baixo (especialmente $PMR = 0$). A figura 3 mostra um exemplo do primeiro caso. Nesse gráfico, cada ponto corresponde a um pacote, com

o instante no qual ele foi gerado representado pelo eixo X e o atraso pelo eixo Y, ambos em segundos. A linha tracejada indica o valor máximo de atraso tolerado pelo *buffer-anti-jitter*. Os pacotes descartados na rede são representados sobre o eixo X (atraso de 0s). Nesse gráfico, os pacotes que estão acima da linha tracejada foram descartados pelo *buffer-anti-jitter*, e portanto contaram como perdas adicionais para fins de cálculo do MOS.

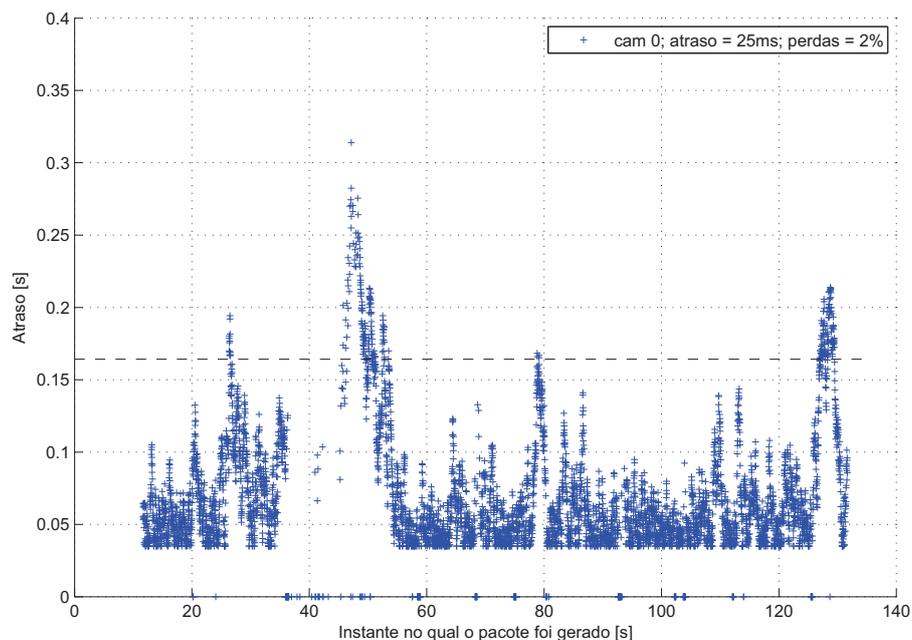


Figura 3. Atrasos dos pacotes. Simulação com *delay-centric* desligado, $PMR = 5$ e $RTOMax = 10s$. Caminho 0: atraso = 25ms, perdas = 2%; caminho 1: atraso = 25ms, perdas = 2%. A média do atraso dos pacotes é 70ms. MOS = 2,58.

Essa figura traz um caso no qual $PMR = 5$ e ocorre uma interrupção na transmissão de pacotes devido a uma rajada de perdas em torno dos 40s da simulação (*delay-centric* desativado). Essa interrupção acarreta também num grande aumento do atraso nos pacotes enviados logo após esse intervalo devido a um acúmulo de dados no *buffer* de saída do SCTP. Nesse exemplo, o valor do $RTOMax$ é de 10s.

Na figura 4 consta o mesmo caso da figura anterior, com a única diferença sendo o valor do $RTOMax$, que neste caso é de 0,1s. Dessa forma é possível verificar que não ocorre a interrupção observada na figura 3. Com o RTO limitado a um valor pequeno, o SCTP não espera longos períodos para enviar um pacote mesmo após a ausência de um SACK e detecção de uma perda. Dessa forma, pequenas rajadas de perdas não prejudicam o fluxo VoIP além das próprias perdas, primeiramente porque assim que a rajada de perdas cessar o receptor voltará a receber os dados, o que significa um restabelecimento mais rápido da chamada. Além disso, não havendo interrupção poucos dados serão mantidos no *buffer* de saída do SCTP, o que não causa o aumento no atraso observado no caso anterior.

O segundo caso em que ocorre uma interrupção na transmissão de pacotes VoIP é devido ao estabelecimento do estado dormente da associação, que é quando todos os caminhos disponíveis se tornam inativos. Como esperado, essa situação foi detectada em

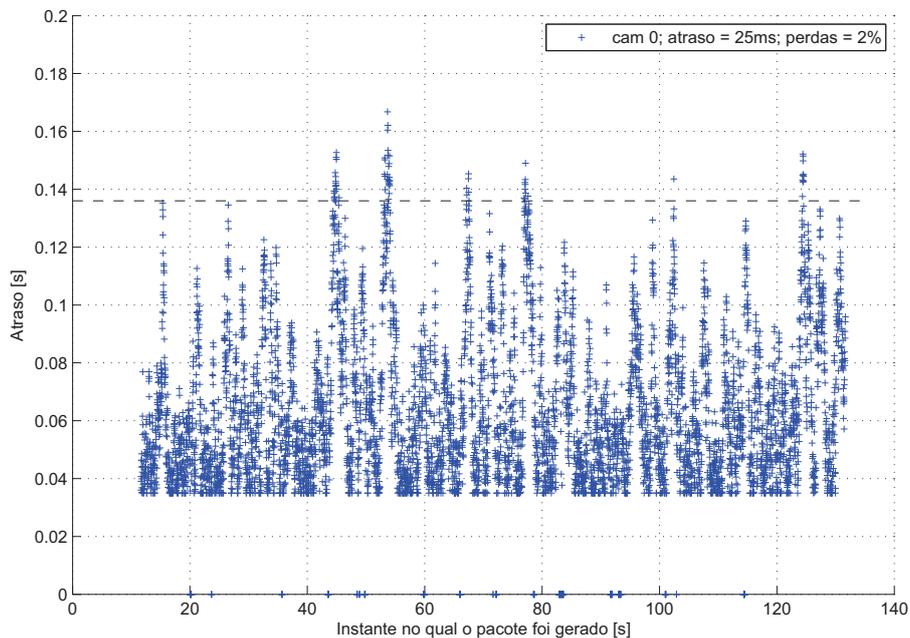


Figura 4. Atrasos dos pacotes. Simulação com *delay-centric* desligado, $PMR = 5$ e $RTOMax = 0,1s$. Caminho 0: atraso = 25ms, perdas = 2%; caminho 1: atraso = 25ms, perdas = 2%. A média do atraso dos pacotes é 63ms. MOS = 3,94.

casos de valores baixos de PMR (0 e 1). Para que a associação saia do estado dormente, é necessário que algum caminho volte a se tornar ativo, através da resposta de um pacote de pulsação. Como o intervalo entre o envio de pacotes de pulsação é dependente do RTO (vide equação 4), valores mais baixos de $RTOMax$ garantem intervalos menores entre o envio de pacotes de pulsação. Dessa forma, o estado dormente pode ser encerrado com mais antecedência, garantindo uma retomada mais rápida da transmissão e, por consequência, valores maiores de MOS.

O gráfico da figura 5 mostra o quanto o valor do $RTOMax$ influencia, em média, uma chamada VoIP. Nesse gráfico cada ponto é calculado como a média do valor do MOS obtido com 100 simulações que usam sub-faixas diferentes do RNG. Os intervalos de confiança são de 95% e supõem uma distribuição normal para o MOS. As simulações que compõe esse gráfico apresentavam todas a mesma configuração de atrasos e perdas nos caminhos. O eixo Y representa valores de MOS e o eixo X representa valores simulados para $RTOMax$ (em escala logarítmica).

Na figura 5 consta um caso de atraso médio moderado (25ms) e perdas altas (10%) em ambos os caminhos. Para esses casos de grandes perdas fica evidente a grande contribuição que baixos valores de $RTOMax$ trazem à qualidade das chamadas para todos os valores de PMR e estados do *delay-centric*. Para valores de $RTOMax$ entre 1s e 0,1s, a medida em que este diminui ocorre um aumento no MOS. Para valores maiores de $RTOMax$ há pouca variação no MOS, assim como para valores menores do que 0,1s.

O gráfico da figura 5 resume as principais características encontradas em relação à influência do parâmetro $RTOMax$. O ganho no MOS é observado diminuindo-se o valor do $RTOMax$ até cerca de 0,1s. Abaixo disso não se observa mais ganho no MOS.

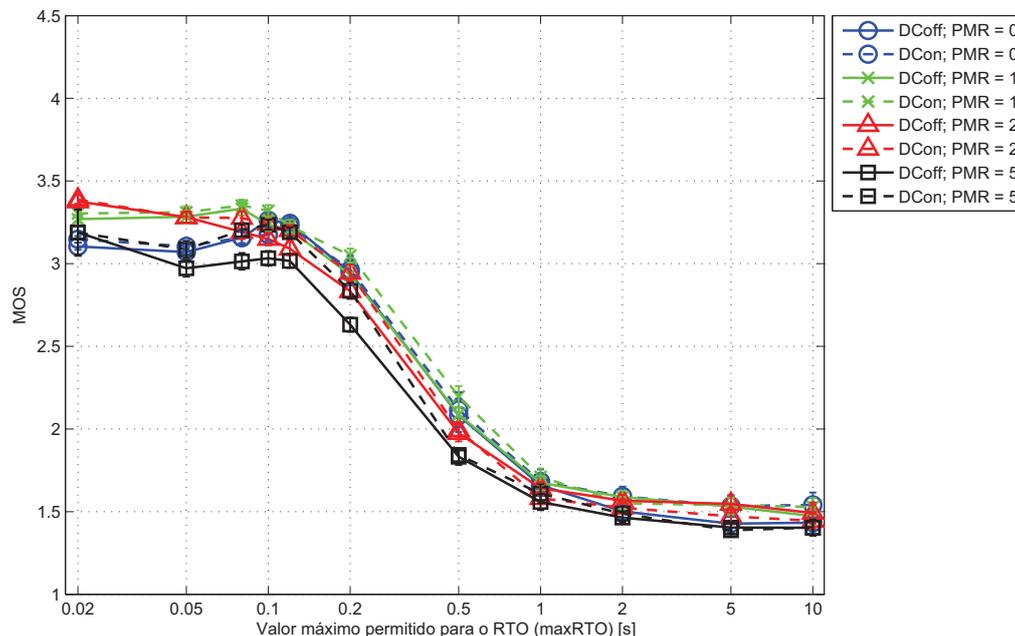


Figura 5. Média do MOS com IC de 95% para casos de: atraso = 25ms e perdas = 10% no caminho 0; atraso = 25ms e perdas = 10% no caminho 1.

Em alguns casos ocorre até uma leve queda no MOS abaixo de 0,1s, o que sugere este valor como um ponto ótimo para o $RTOMax$ para este cenário especificamente. De acordo com o gráfico, diminuir o $RTOMax$ de 10s para 0,1s trouxe o MOS médio das chamadas de um valor impraticável de cerca de 1,5 para um valor aceitável de 3,3. Isso sugere que mesmo em casos de grandes perdas de pacotes na rede o uso do SCTP pode ser viabilizado para transporte de tráfego VoIP se o $RTOMax$ for drasticamente reduzido.

Outra observação em relação ao gráfico é que as curvas referentes ao $PMR = 1$ estão levemente acima das curvas de $PMR = 0$. Esse também foi um resultado padrão deste cenário. Uma explicação para esse fato é que a rápida reação a perdas propiciada pelo $PMR = 0$ também ocorre com $PMR = 1$ quando o RTO está sob limites estreitos. Como o uso do $PMR = 1$ resulta em uma menor chance de ocorrência de estado dormente, a combinação dessas duas propriedades é uma explicação para a leve superioridade apresentada pelo $PMR = 1$ neste cenário. Entretanto, a diferença é muito pequena, e dada a pequena abrangência de casos simulados neste cenário, não se pode afirmar qual dos dois valores é a melhor opção.

Quando as perdas não são intensas o ganho alcançado pela redução do $RTOMax$ é muito menor, conforme mostra o gráfico da figura 6. Nesse caso o atraso é o mesmo do caso da figura 5 em ambos os caminhos (25ms) mas as perdas são moderadas (2% em ambos). Apesar do ganho ser menos evidente, ele está presente para valores menores de $RTOMax$. Novamente percebe-se que em 0,1s está o ponto ótimo para $RTOMax$. Não só pelo valor médio de MOS maior, mas também pela diminuição dos intervalos de confiança em torno desse ponto, o que indica um comportamento mais regular da qualidade das chamadas.

Ainda na figura 6 percebe-se a leve vantagem que o $PMR = 1$ leva em relação

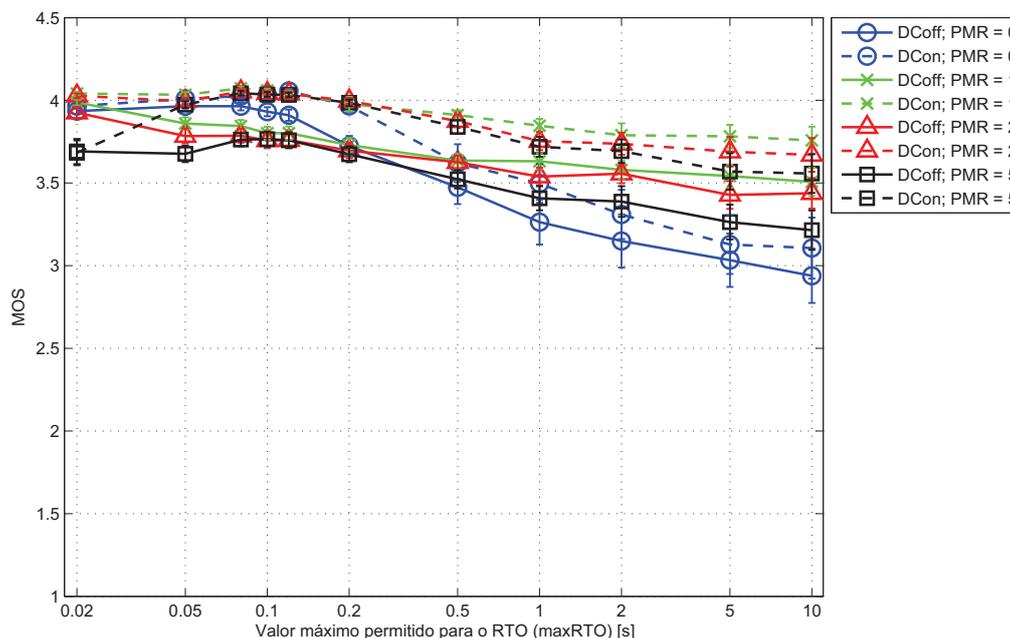


Figura 6. Média do MOS com IC de 95% para casos de: atraso = 25ms e perdas = 2% no caminho 0; atraso = 25ms e perdas = 2% no caminho 1.

aos demais valores de PMR . Mesmo assim a diferença, especialmente para $RTO_{max} = 0, 1$, é muito pequena e não permite assegurar uma tendência. Outra observação é que o ganho de MOS com a diminuição do RTO_{max} para os casos de $PMR = 0$ é o mais acentuado dentre todos os casos simulados.

A tabela 2 contém as médias do MOS das simulações englobando todos os casos em que as perdas nos dois caminhos são de 10%. Cada valor de MOS nessa tabela corresponde a um valor diferente de RTO_{max} (sentido das colunas) e de PMR (sentido das linhas) com ou sem *delay-centric*. Todos os casos de atraso em todas as combinações estão presentes nessa tabela.

Essa tabela mostra que, em média, para casos de perdas elevadas, baixos valores de RTO_{max} (em torno de 0,1s ou 0,08s) proporcionam uma melhor qualidade para o tráfego VoIP em relação a valores maiores de RTO_{max} . O emprego do *delay-centric* ajudou também nesse sentido. Além disso, percebe-se que os casos de $PMR = 1$ obtiveram uma média superior a dos demais valores de PMR . A maior média nessa tabela ocorre para os casos de $RTO_{max} = 0,08s$, $PMR = 1$ e com *delay-centric* ativado.

O RTO_{max} também apresentam uma forte influência sobre o mecanismo de controle de congestionamento do SCTP. Ao diminuir os valores desses parâmetros o SCTP se torna menos “tolerante” ao congestionamento, pois seus tempos de espera para envio de dados se tornam menores. Dessa forma utilizar valores muito pequenos de RTO_{max} (e, por consequência, de RTO_{min}) torna o tráfego gerado pelo SCTP parecido com aquele produzido pelo UDP, conforme já havia demonstrado Park et al. [Park et al. 2007] em seus estudos.

$RTOMax$ (s)	<i>Delay-centric</i> DESLIGADO				<i>Delay-centric</i> LIGADO			
	Valor de PMR				Valor de PMR			
	0	1	2	5	0	1	2	5
10	1,391	1,517	1,497	1,418	1,547	1,603	1,520	1,460
5	1,402	1,521	1,499	1,400	1,536	1,578	1,512	1,471
2	1,463	1,547	1,525	1,462	1,577	1,612	1,554	1,527
1	1,561	1,646	1,607	1,538	1,734	1,778	1,654	1,602
0,5	1,924	2,003	1,899	1,783	2,249	2,280	2,041	1,900
0,2	2,613	2,683	2,610	2,502	3,029	3,066	2,985	2,843
0,12	3,006	2,880	2,776	2,739	3,222	3,255	3,215	3,190
0,1	3,019	2,934	2,826	2,765	3,177	3,282	3,236	3,219
0,08	3,037	3,001	2,863	2,767	3,177	3,329	3,264	3,211
0,05	2,909	3,020	2,966	2,679	3,069	3,311	3,276	3,100
0,02	2,895	3,139	3,094	3,059	2,951	3,240	3,375	3,150

Tabela 2. Médias do MOS para casos de 10% de perdas e todos os valores de atraso. As médias estão classificadas por $RTOMax$, PMR e estado do *delay-centric*.

6. Conclusões

Neste trabalho foi estudado o desempenho do SCTP juntamente com o algoritmo de seleção automática de rotas baseado no menor atraso (*delay-centric*) para o tráfego VoIP (G.711 com PLC). Os cenários simulados contemplavam casos de usuários multi-abrigados cujas rotas continham tráfego de fundo aleatório (atraso) e perdas em rajada variáveis. Os principais parâmetros avaliados quanto à influência na qualidade de chamadas VoIP foram o PMR e o $RTOMax$. A qualidade da chamada de cada simulação era medida usando o MOS estimado a partir do E-model.

Em relação aos valores de PMR , os menores valores apresentaram os melhores resultados. Em média, $PMR = 1$ se mostrou um pouco mais vantajoso do que $PMR = 0$ para os casos simulados. Mas haviam casos em que o PMR nulo obteve valores maiores de MOS. Como este trabalho simulou apenas algumas das várias possibilidades de cenários, pode-se afirmar apenas que menores valores de PMR (0 ou 1) são os mais recomendados para transporte tráfego VoIP.

Igualmente para o $RTOMax$, valores menores garantiram um melhor desempenho ao SCTP nas simulações realizadas, independentemente do estado do *delay-centric* ou valor do PMR . Encontrou-se um valor ótimo em torno de 0,1s para o $RTOMax$. Quando as perdas são altas, este valor proporciona os maiores valores de MOS e uma diferença expressiva em relação ao MOS obtido com valores mais altos de $RTOMax$, tais como de 1s ou maiores. Apenas para perdas muito baixas é que não foi possível detectar o efeito benéfico da diminuição $RTOMax$.

Dos resultados obtidos pode-se concluir que o uso de valores menores tanto de PMR quanto de $RTOMax$ propiciam uma melhor qualidade para o transporte de dados sensíveis ao atraso, tais como chamadas VoIP. O uso do *delay-centric* também traz vantagens nesse sentido na grande maioria dos casos, que também são os mais comumente encontrados na prática. A combinação de $PMR = 0$ ou $PMR = 1$, juntamente com um $RTOMax = 0,1s$ e com o *delay-centric* ativado resultou, nos cenários simulados, nos maiores valores de MOS.

Referências

- Bhagwat, P., Bhattacharya, P., Krishma, A., and Tripathi, S. K. (1997). Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs. *Wirel. Netw.*, 3(1):91–102.
- Budzisz, L., Ferrus, R., Grinnemo, K.-J., Brunstrom, A., and Casadevall, F. (2007). An analytical estimation of the failover time in SCTP multihoming scenarios. *Wireless Communications and Networking Conference (WCNC)*, pages 3929–3934.
- Caro, A. L., Amer, P. D., and Stewart, R. R. (2006). Rethinking end-to-end failover with transport layer multihoming. *Annals of Telecommunications*.
- Eddy, W. (2004). At what layer does mobility belong? *Communications Magazine, IEEE*, 42(10):155–159.
- Fallon, S., Jacob, P., Qiao, Y., and Murphy, L. (2009). An adaptive optimized RTO algorithm for multi-homed wireless environments. *Wired/Wireless Internet Communications - Lecture Notes in Computer Science*, 5546:133–145.
- Fallon, S., Jacob, P., Qiao, Y., Murphy, L., Fallon, E., and Hanley, A. (2008). SCTP switchover performance issues in WLAN environments. *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 564–568.
- Fitzpatrick, J., Murphy, S., Atiauzzaman, M., and Murphy, J. (2009). Using cross-layer metrics to improve the performance of end-to-end handover mechanisms. *Computer Communication Preprint Online*, 10.1016:13 Pages.
- Fracchia, R., Casetti, C., Chiasserini, C.-F., and Meo, M. (2007). Wise: Best-path selection in wireless multihoming environments. *IEEE Transactions on Mobile Computing*, 6(10):1130–1141.
- Gavriloff, I. (2009). Análise de aspectos envolvidos no mecanismo de seleção de caminho baseado em atraso para sistemas multiabrigados utilizando sctp. Master's thesis, UFPR, Curitiba, Brazil.
- ITU-T (1996). Methods for subjective determination of transmission quality. *ITU-T Recommendation P.800*.
- ITU-T (1999). A high quality low-complexity algorithm for packet loss concealment with g.711. *G.711 appendix I*.
- ITU-T (2000). The e-model, a computational model for use in transmission planning. *G.107*.
- ITU-T (2003). Estimates of ie and bpl parameters for a range of codec types. *SG12 D106*.
- ITU.T (1988). Pulse code modulation (pcm) of voice frequencies. *G.711*.
- Kelly, A., Muntean, G.-M., Perry, P., and Murphy, J. (2004). Delay-centric handover in SCTP over WLAN. *Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE*.
- Park, H. L., Kim, M., and Kim, J.-S. (2007). Evaluation of stream control transmission protocol as a transport for VoIP over WLAN. *Advanced Communication Technology, The 9th International Conference on*, 3:1613–1617.
- Qiao, Y., Fallon, E., Murphy, J., Murphy, L., Zhu, X., Hayes, G., Matthews, A., and Hanley, A. (2008). Performance analysis of multi-homed transport protocols with network failure tolerance. *IET Communications*, 2:336–345.
- Stewart, R., Arias-Rodriguez, I., Poon, K., Caro, A., and Tuexen, M. (2006). Stream control transmission protocol (SCTP) specification errata and issues. *IETF RFC 4460*, (4460).
- Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and Conrad, P. (2004). Stream control transmission protocol (SCTP) partial reliability extension. *RFC 3758*.
- Stewart, R. R. and Xie, Q. (2001). *Stream Control Transmission Protocol (SCTP) - A reference Guide*. Addison-Wesley.
- Stewart, R. R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H. J., Taylor, T., Rytina, I., Kalla, M., and Zhang, L. (2000). Stream control transmission protocol. *IETF RFC 2960*.