Análise de desempenho do protocolo SCTP (*Stream Control Protocol Transmission*)

Andrew Miranda da Silva[1], **Eduardo Maroñas Monks**[1]

[1]Curso Superior de Redes de Computadores – Faculdade de Tecnologia SENAC Pelotas (FATEC)
Rua Gonçalves Chaves 602 – 96015560 – Pelotas – RS – Brazil

andrew.mirandadasilva@hotmail.com.br, emmonks@gmail.com

Abstract. *This article presents a comparative analysis on the performance of transport layer protocols SCTP, TCP and UDP, by testing in scenarios with resource constraints. The goal is to compare the performance of protocols in network environments with delay, packet loss and bandwidth restrictions.*
Keywords: SCTP, TCP, UDP, analysis, performance.

Resumo. *Este artigo apresenta uma análise comparativa sobre o desempenho dos protocolos da camada de transporte SCTP, TCP e UDP, por meio de testes em cenários com restrições de recursos. O objetivo é comparar o desempenho dos protocolos em ambientes de rede com atraso, perdas de pacote e restrições de largura de banda.*
Palavras-Chave: SCTP, TCP, UDP, análise, desempenho.

1. Introdução

Os protocolos da camada de transporte são um dos critérios mais importantes para comunicação entre os computadores de uma rede, pois proporcionam serviços de entrega confiável ou não confiável de dados para as aplicações. Segundo [Farrel 2005], os protocolos mais importantes para o transporte de dados na Internet, são os mais conhecidos e utilizados, TCP (*Transmission Control Protocol*) [Postel 1981] e UDP (*User Datagram Protocol*) [Postel 1980]. Existem outras soluções que desempenham este papel, tal como o protocolo SCTP (*Stream Control Transmission Protocol*) [Stewart 2007], que propõe ser uma alternativa viável ao TCP e UDP na implementacao de sistemas distribuídos.

Na Internet, o protocolo TCP é bem estabelecido, porém houve uma necessidade da utilização do protocolo SCTP em conexões PSTN (*Packet Switched Telephone Network*), devido ao protocolo possuir características e recursos adicionais comparado aos protocolos TCP e UDP [Daniel 2005]. O SCTP possui algumas vantagens e diferenças em relação ao TCP e UDP, como a entrega sequencial de dados de usuário em múltiplos fluxos e tolerância a falhas de rede, através do suporte a múltiplos caminhos. Além de ser orientado a conexão, o protocolo é *Rate adaptative*, ou seja, adapta-se dinamicamente a variação do estado de rede, com isto, nota-se que o SCTP veio para ser uma alternativa mais robusta para transferência de fluxos de dados na rede [Daniel 2007].

Este trabalho tem como objetivo analisar o desempenho e o comportamento do protocolo SCTP em cenários de rede com restrições de recursos, utilizando ferramentas de análise de tráfego para coletar e verificar os resultados obtidos, comparando aspectos de desempenho em relação aos protocolos TCP e UDP.

2. Protocolos da Camada de Transporte

O TCP e UDP são os principais protocolos da camada de transporte nas apliações da internet. O TCP possui diversos serviços e algoritimos responsáveis pela garantia de envio dos dados durante a transmissão, como a retransmissão de segmentos perdidos e controle de congestionamento da conexão [Allman et al. 1999]. O TCP utiliza somente métodos de conexão fim a fim, onde não há possibilidade de utilização do protocolo em sistemas *multicast*. Enquanto a comunicação entre os dois *hosts* for mantida, o tratamento será da forma *full-duplex*, o que permite que ambos os envolvidos na troca de dados enviem e recebam informações ao mesmo tempo [Farrel 2005].

O protocolo UDP é utilizado para fornecer uma comunicação simples, porém rápida e eficiente, por padrão não possui um gerenciamento de controle preciso de erro ou fluxo [Daniel 2007]. O principal objetivo do protocolo é realizar multiplexação entre várias comunicações. O protocolo até possui uma verificação opcional de erros dos dados, porém esta verificação serve apenas para descartar dados corrompidos durante a transmissão na rede [Postel 1980]. Este protocolo é bastante utilizado em redes multimídia para serviços de rede em tempo real, ou até mesmo em comunicações de sistemas *multicast* [Daniel 2007].

3. Protocolo SCTP (*Stream Control Transport Protocol*)

O protocolo SCTP (*Stream Control Transport Protocol*), foi desenvolvido para fins de troca de mensagens telefônicas na Internet, redes denominadas PSTN (*Packet Switched Telephone Network*). No ano 2000 foi lançada a primeira RFC 2960 [Stewart et al. 2000] por um grupo de engenheiros de rede do SIGTRAN (*Signaling Transport*) [Ong et al. 1999] que submeteram ao IETF *Internet Engineering Task Force*. Dois anos após a criação do protocolo, foi lançado uma nova RFC 3308 [IETF 2003] descrevendo as alterações no algoritimo de soma de verificação. Atualmente, o protocolo encontra-se na RFC 4960 [Stewart 2007].

O SCTP é considerado como um protocolo de transporte confiável e orientado a conexão, o mesmo utiliza recursos adicionais para melhorar o desempenho em rede, mas também possui funcionalidades do TCP e UDP embutidas [Stewart et al. 2000]. O SCTP opera em cima do protocolo IP, porém uma conexão só pode ser estabelecida se ambos os lados possuem o mesmo protocolo SCTP na camada de transporte. Em relação às portas de comunicação, o IANA definiu que as portas de comunicação TCP, estariam reservadas para SCTP, permitindo o uso concorrente de TCP e SCTP para o mesmo serviço no mesmo número de porta [Pfutzenreuter E 2004]. Segundo [Daniel 2007], o protocolo SCTP pode ser atrativo para aplicações comuns na Internet, principalmente, tratando-se de aplicações de multimídia.

3.1. Cabeçalho SCTP

Um pacote SCTP é composto por um *header* comum e *chunks*. Um *chunk* no cabeçalho SCTP é um campo que pode conter informações de controle ou dados de usuário [Stewart 2007]. Vários *chunks* podem ser multiplexados dentro de um pacote IP até que seja atingido o MTU. O *header* comum do SCTP é formado por 12 *bytes* contendo as portas de destinos e origem. A *Tag* de verificação indica o tamanho do *chunk* e o campo do *checksum* [Stewart 2007]. A Figura 1 mostra os campos que compõe o cabeçalho do protocolo.

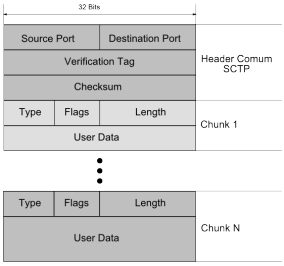

Figura 1. Estrutura do cabeçalho SCTP com vários chunks [UFRGS 2008]

3.2. Orientado a Mensagens

O protocolo trata os dados como blocos de mensagens independentes, ou seja, como sequência de dados fragmentados, dividindo os dados em partes e identificando cada parte para ser transmitida na rede, diferentemente do TCP, que trata os dados como sequência de octetos.

3.3. Associação

Uma comunicação estabelecida entre dois *hosts* que utilizam SCTP é feita através de uma associação, ou várias associações, negociada entre os participantes, de forma diferente do TCP, que realiza apenas uma sessão *full-duplex* [Stewart 2007]. Conforme a descrição da RFC [Stewart 2007], o canal de comunicação SCTP é unidirecional, podendo haver um número desigual de fluxos, em cada direção. Nas associações SCTP, não existem um estado meio fechado *half-closed*, isto serve tanto para associações e fluxos [Stewart 2007].

3.4. Multihoming

Uma das características mais interessantes do protocolo, é o suporte a multi-caminhos, denominado também como *multi-homed*. O objetivo deste mecanismo é criar mais de uma comunicação entre redes, com vários endereços IP, sem necessariamente ser um roteador [Stewart 2007]. Este recurso proporciona dois ou mais caminhos entre a origem e o destino. Segundo [Stewart 2007], este recurso tem uma grande importância, devido a tolerância à falhas de uma das conexões IP de um dos enlaces. O SCTP permite que cada *host* informe uma lista de IPs. O endereço IP utilizado durante a criação da associação é classificado como IP primário, isto significa que esta conexão é o principal canal de comunicação com o *host* de destino remoto. Os demais IPs informados são secundários, que podem ser utilizados caso falhe a comunicação com o IP primário [Stewart 2007].

4. Ferramentas

Nesta seção, são abordadas as ferramentas utilizadas nos testes comparativos entre os protocolo TCP, UDP e SCTP.

4.1. D-ITG

A ferramenta D-ITG (*Distributed Internet Traffic Generator*) [D-ITG 2013] é uma ferramenta capaz de produzir tráfego IPv4 e IPv6. Seu objetivo é medir as métricas de desempenho mais comuns, como taxa de transferência, atraso, *jitter* e perdas de pacotes. A versão da ferramenta utilizada nos testes foi a 2.8.

4.2. Iperf

A ferramenta Iperf [Iperf 2016] tem como funcionalidade principal o estresse da rede, realizando testes de performance entre *hosts*. O cliente gera um determinado fluxo de dados para o servidor Iperf em escuta, como em uma comunicação cliente e servidor, a ferramenta testa a capacidade máxima do meio de transmissão. A versão utilizada nos testes é a 3.1, possui o módulo de suporte ao protocolo SCTP, ainda na versão beta.

4.3. Tcpdump

O Tcpdump [Tcpdump 2016] é uma ferramenta de análise de tráfego em modo texto, também conhecida como *sniffers* de rede, onde seu objetivo é capturar e mostrar as conexões e transmissões de dados que trafegam pela interface de rede.

4.4. Wireshark

O Wireshark [Wireshark 2016] é uma aplicação de análise de tráfego de rede semelhante ao Tcpdump. Possui uma interface gráfica GUI (*Graphical User Interface*), onde é possível analisar as estatísticas de desempenho de uma determinada conexão. A versão utilizada para análise das capturas dos protocolos é a 2.0, que possui mecanismos de análise do protocolo SCTP.

4.5. WANem

O WANem (*Wide Area Network Emulator*) [Wanem 2014], é uma distribuição Linux que pode emular um roteador de rede com restrições de recursos, como atraso de rede, restrição da largura de banda, perdas de pacotes.

5. Cenários de Testes

Pare realização dos testes foram utilizadas duas máquinas reais com três servidores virtualizados no VMware Workstation 7.1. Em duas máquinas virtuais foi utilizado a distribuição Debian 8, com as ferramentas D-ITG e Iperf. A terceira máquina virtual instalada no hospedeiro, é o roteador WANem, que será responsável por realizar as restrições de recursos na rede. Os hospedeiros estão conectados em uma rede *ethernet*, em um *switch* de *100Mbit/s*.

5.1. Cenários sem Restrições

O primeiro cenário de testes é baseado no ambiente de rede local *100 Mbit/s*. Neste cenário, não foi necessário a utilização do roteador WANem apenas duas máquinas virtuais com o sistema operacional Debian 8. Uma das máquinas virtuais está em modo cliente e a outra como servidor.

5.2. Cenários com Restrições

Na criação dos cenários com restrições de recursos de rede acrescentou-se uma máquina virtual com a distribuição WANem versão 2.3, que efetuará o papel de um roteador para realizar as restrições de largura de banda, atraso e perdas de pacotes. Para estabelecer a comunicação do cliente com servidor na rede com restrições, foi necessário adicionar uma rota estática nos dois sentidos, passando pelo roteador WANem. A Figura 2 mostra a estrutura dos cenários com restrições de recursos.

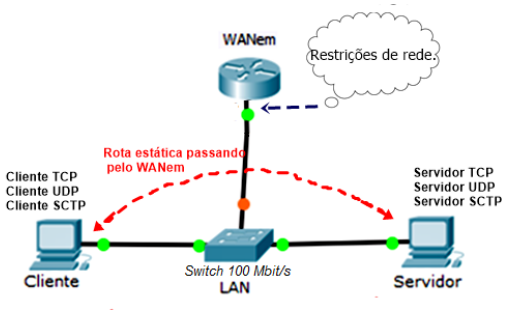

Figura 2. Estrutura dos cenários com restrições de recursos

5.3. Restrições de recursos

As restrições de recursos de rede foram baseadas em estatísticas de dois cenários reais da Internet e dois ambientes de simulação com grande quantidade de atraso e perdas. Um dos ambientes de rede, baseado em restrições reais foi baseado no estudo feito por [Chen et al. 2012] sobre o desempenho de redes WIFI, 3G e 4G em universidades americanas. O segundo cenário baseado em restrições de redes reais, foi um levantamento de planos de Internet oferecido pelos principais provedores de Internet da região Sul do Brasil, baseado na publicação do site G1 [g1.com 2015], com dados fornecidos pela Anatel. Este levantamento indica que a internet banda larga mais utilizada especificamente na cidade de Pelotas, possui a média de *6 MBit/s*.

No cenário três, foi utilizado uma porcentagem razoável de perdas de pacotes, para analisar a performance e o comportamento dos protocolos em ambientes com perdas maiores. A Tabela 1 mostra as restrições de rede aplicadas em cada cenário.

Tabela 1. Tabela de restrições

Restrições	Cenário 1 (Wifi 3G/4G)	Cenário 2 (Pelotas)	Cenário 3 (Perdas)
Largura de Banda *(Mbit/s)*	*4,95 Mbit/s*	*6 Mbit/s*	*5,9 Mbit/s*
Atraso *(ms)*	*30 ms*	–	*20 ms*
Perdas pacotes (%)	*1 %*	–	*10 %*

6. Testes

Os testes foram realizados utilizandos os três protocolos da camada de transporte, TCP, UDP e SCTP. No tráfego de rede gerado pelo cliente, com TCP e SCTP utilizou-se a geração de fluxo de dados padrão. Nos testes efetuados com UDP, foi necessário apontar a largura de banda utilizada no cenário, pois o padrão da ferramenta é utilizar fluxos de *1 Mbit/s.*

Com a ferramenta D-ITG 2.8 aplicaram-se as mesmas metodologias de testes realizadas com o Iperf 3.1, porém alguns parâmetros para gerar o tráfego de rede foram acrescentados no cliente. O cliente Iperf 3.1, por padrão gera o tráfego com o tamanho dos pacotes de aproximadamente *1500 Bytes*, para não haver influência nos resultados de vazão e tempo das conexões, foi adicionado no cliente D-ITG o parâmetro para utilizar pacotes de *1500 Bytes*. A quantidade de dados utilizada na transmissão foram valores

fixos de 100 *Megabytes*, 50 *Megabytes* e 10 *Megabytes* nos testes realizados para ambas ferramentas.

7. Análise dos Resultados

Os cenários de rede, utilizados nos testes com a ferramenta Iperf 3.1, foram mantidos nos testes realizados com o D-ITG. Os resultados de vazão e tempo, tiveram pequenas diferenças entre as ferramentas, porém os protocolos apresentaram o mesmo comportamento em ambos os testes.

O gráfico na Figura 3 mostra a média de vazão dos testes realizados em rede local. Na figura 4, o gráfico mostra os resultados de vazão dos cenários com restrições de recursos.

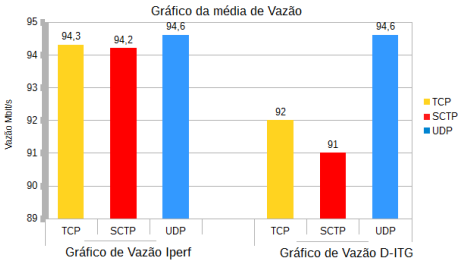

Figura 3. Gráfico da média de vazão, nos testes realizados em rede Local

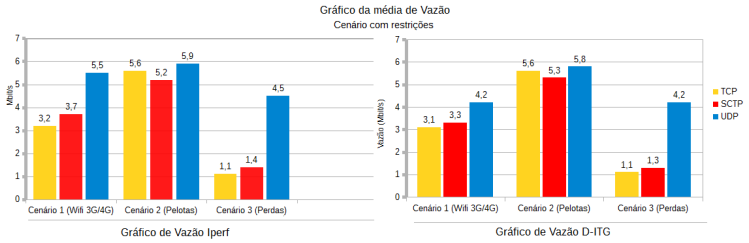

Figura 4. Gráfico da média de vazão, nos testes realizados em cenários com restrições

O protocolo TCP, teve o desempenho em rede igual ou superior ao protocolo SCTP em cenários sem restrições e com restrições de largura de banda. A similaridade entre o protocolo SCTP e TCP é justificada por ambos serem confiáveis, orientado a conexão e possuirem recursos de controle de congestionamento de fluxo [Stewart 2007].

Na avaliação do protocolo UDP, entre os três, foi o que melhor obteve vazão e menor tempo nas transmissões de dados, porém nos testes realizados em cenários com restrições de recursos, especificamente os que possuem características de atraso e perdas de pacotes, o protocolo teve a média de *18 %* de perdas nos dados transferidos e aproximadamente *46 %* de perda nos dados no cenário 3. Os resultados com o protocolo foram esperados, pois a característica do UDP é realizar transmissões de dados sem verificar o recebimento no destino.

Na análise do protocolo SCTP, observa-se que o desempenho em rede foi muito similar ao TCP, principalmente em cenários sem restrições de recurso. Nos cenários com restrições de largura de banda e atraso, não apresentou nenhuma grande superioridade em relação aos demais protocolos, tanto nos testes realizados com D-ITG e Iperf. O grande diferencial do SCTP, foi especificamente em cenários com perdas de pacotes, pois transferiu o fluxo íntegro de dados, com menor tempo de conexão comparado ao TCP, e com mais integridade que o UDP. Este aproveitamento do protocolo SCTP, pode ser justificado pelo controle de congestionamento e de perdas de pacotes tratados pelas mensagens SACKs, o tratamento dos dados retransmitidos obteve maior eficiência e foi mais rápido que o do TCP. Desta forma o SCTP obteve melhor aproveitamento do que o TCP e UDP.

8. Considerações Finais

Em uma análise geral dos resultados, o SCTP manteve-se com médias proporcionais entre os protocolos TCP e UDP, não apresentando nenhuma superioridade , a não ser em cenários com perdas de pacotes, onde obteve uma vantagem considerável. Nos testes realizados nota-se uma grande similaridade e características do SCTP com TCP. O protocolo SCTP possui atributos interessantes, como o uso de *Multihoming* e *Multistreaming*, porém não estavam disponíveis por completo nas ferramentas e aplicações de teste. Conclui-se que o protocolo SCTP é tão bom quanto o TCP, e devido aos recursos adicionais, poderia obter melhor desempenho. Os protocolos TCP e UDP são muito bem estabelecidos, pois a maioria dos sistemas operacionais e aplicações não possuem suporte ao SCTP, as vantagens de utilização sobre o TCP, serão bem maiores quando houver maior disponibilidade deste protocolo em sistemas operacionais e aplicações. O protocolo SCTP, mostrou-se eficiente, porém não foi possível analisar seus recursos adicionais, como o *Multihoming*, devido algumas limitações das ferramentas de teste, seria de suma importância a investigação de disponibilidade do recurso *Multihoming* em cenários de rede com múltiplos caminhos. Outro segmento de pesquisa bastante interessante, seria analisar métodos de segurança do protocolo SCTP.

Referências

Allman, M., Paxson, V., and Stevens, W. (1999). Tcp congestion control. RFC 2581, RFC Editor.

Chen, Y.-C., Towsley, D., Nahum, E. M., Gibbens, R. J., and Lim, Y.-s. (2012). Characterizing 4g and 3g networks: Supporting mobility with multi-path tcp. *University of Massachusetts Amherst, Tech. Rep*.

D-ITG (2013). Distributed internet traffic generator. `http://traffic.comics.unina.it/software/IT/`. Acessado em: 2016-09-30.

Daniel, G. (2005). SCTP uma alternativa aos tradicionais protocolos de transporte da internet. Editora Ciência Moderna, Rio de Janeiro.

Daniel, G. (2007). Comunicações multimidia na internet. pages 219–252. Editora Cincia Moderna, Rio de Janeiro.

Farrel, A. (2005). A internet e seus protocolos. pages 219–252. Editora Campus, Rio de Janeiro.

Iperf (2016). Iperf. `https://iperf.fr/`. Acessado em: 2016-03-30.

Ong, L., Rytina, I., Garcia, M., Schwarzbauer, H., Coene, L., Lin, H., Juhasz, I., Holdrege, M., and Sharp, C. (1999). Framework architecture for signaling transport. RFC 2719, RFC Editor.

Postel, J. (1980). User datagram protocol. STD 6, RFC Editor. http://www.rfc-editor.org/rfc/rfc768.txt.

Postel, J. (1981). Transmission control protocol. STD 7, RFC Editor. http://www.rfc-editor.org/rfc/rfc793.txt.

Stewart, R. (2007). Stream control transmission protocol. RFC 4960, RFC Editor. http://www.rfc-editor.org/rfc/rfc4960.txt.

Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and Paxson, V. (2000). Stream control transmission protocol. RFC 2960, RFC Editor.

Tcpdump (2016). Tcpdump. https://www.tcpdump.org.

UFRGS (2008). Protocolo sctp stream control transmission protocol. http://www.inf.ufrgs.br/ cechin/Net/sctp/sctp.html.

Wanem (2014). Wanem. http://wanem.sourceforge.net/. Acessado em: 2016-09-30.

Wireshark (2016). Wireshark. https://www.wireshark.org/. Acessado em: 2016-09-30.