

GESTÃO E MONITORAMENTO DE REDES DE COMPUTADORES GERENCIAMENTO DE SISTEMAS SENSÍVEIS

Autor: Me. Paulo Sérgio Pádua de Lacerda

Revisor: Rafael Rehm

INICIAR



introdução

Introdução

Caro(a) estudante,

Seja bem-vindo(a) à disciplina de Gestão e Monitoramento de Redes de Computadores! A proposta principal desta unidade é definir os conceitos, padrões e critérios fundamentais que compõem os sistemas sensíveis utilizados em redes de computadores. Isso inclui analisar os protocolos *Real-time Transfer Protocol* (RTP) e *Session Initiation Protocol* (SIP), mas também as ferramentas que os monitoram.

Com o cenário mundial convergindo para a Internet, as aplicações de comunicação têm seu crescimento aumentado nas empresas e entre os usuários, assim como os jogos online e aplicações de *stream* de vídeo. Essas aplicações são sensíveis a problemas na rede e precisam de uma rede com ótima qualidade de transferência de dados.

Em suma, você vai compreender os fundamentos de protocolos relacionados a sistema sensíveis à interferência na rede de computadores. Vamos lá?

Quality of Service

O universo da comunicação entre os usuários de Internet teve um crescimento exponencial nos últimos anos, aplicações de vídeo, como Skype, Duo, Zoom, foram adotadas como ferramentas para reuniões entre pessoas geograficamente distantes entre si; entretanto, reunidas por meio da aplicação em um mundo virtual.

Além disso, jogos online multiplayer e outras aplicações como *streaming* de vídeo (Netflix e Amazon Prime Video) também tiveram crescimento aumentado entre os usuários. Mas o que essas aplicações têm em comum? Quais os cuidados que o administrador de redes deve ter para determinar uma qualidade no fluxo de dados dessas aplicações?

Essas aplicações são sensíveis à interferência da rede e, por esse motivo, precisam de uma qualidade de rede para o fluxo de dados. E o que é qualidade de rede ou qualidade de serviço (QoS) em uma rede de computadores?

Desempenho da Rede

As redes são comumente medidas pelo termo *velocidade*, pois este determina seu melhor desempenho. Falamos, portanto, que uma rede é rápida ou lenta com base na velocidade. Entretanto, esse termo torna-se inadequado quando se trata de redes de computadores, pois uma rede considerada muito veloz hoje, passa a ser uma rede de velocidade média ou lenta em um período muito curto.

Com o objetivo de tornar a definição de velocidade associada a desempenho mais precisa, os engenheiros adotaram medidas mais quantitativas, que expressam melhor ou mais especificamente o desempenho da rede independentemente da época. São métricas como latência (*delay* ou atraso), vazão (*throughput*) e *jitter* (variação) (COMER, 2015).

Essas métricas determinam um melhor ou pior desempenho da rede e podem ser definidas como:

- Latência: o tempo requerido para transferir os dados ao longo da rede, desde a origem até o destino;
- Vazão: o montante de dados que pode ser transferido em um segundo; e
- *Jitter* : as mudanças no atraso e sua variação.

A latência em uma rede depende da distância entre a origem e o destino e sob quais condições de infraestrutura de Internet esses dados estão trafegando. Há diversos tipos de *delay*, mas podemos resumir os principais no Quadro 3.1.

Tipo de atraso	Descrição
Atraso de propagação	O tempo necessário para o sinal propagar através de um meio de transmissão
Atraso de acesso	Tempo necessário para um sinal ganhar o acesso a um meio de transmissão
Atraso de chaveamento (<i>switching</i>)	Tempo gasto no encaminhamento dos pacotes
Atraso de fila	O período de espera de um pacote na memória de um elemento de rede como roteador ou <i>switch</i> para ser encaminhado
Atraso do servidor	O tempo gasto pelo servidor no processamento de envio ou recebimento do pacote

Quadro 3.1 - Tipos de atraso (*delay*)

Fonte: Elaborado pelo autor.

Sistemas via cabo geralmente são mais rápidos que sistema *wireless* (sem fio), pois concedem menos atrasos. Geralmente, os computadores de jogos online são conectados via sistemas cabeados, para evitar os *delays* de sistemas *wireless* .

O *throughput* é a vazão, ou seja, a capacidade da rede de transmitir a maior quantidade de dados em unidade de segundos, especificado por *bits por seconds* (bps). Lembre-se de que, embora se trata de dados, pacotes a serem transmitidos, estamos falando na camada 1 tanto no modelo TCP/IP (*Transfer Control Protocol/Internet Protocol*) quanto OSI (*Open System Interconnection*), e

essa camada são os *links* e transferência dados em forma de sinal, e sinal é *bits*.

Então, as redes são medidas por meio de Kilo (1024) bps ou Kbps. Sendo assim, as redes de maior taxa de transmissão possuem *throughput* alto, em torno de Gbps (1024 * Mbps). Mas, como o *throughput* pode ser medido de diversas formas, há uma cautela ao se especificar o que realmente foi medido, pois diversas possibilidades existem, incluindo:

- A capacidade de um único canal ou vários canais;
- A capacidade teórica ou taxa de eficiência;
- A taxa de dados alcançados por uma aplicação chamada *goodput*.

Usuários de Internet comumente medem a velocidade da rede por dados transferidos por unidade de tempo ou pelo desempenho alcançado pela aplicação (um jogo online, por exemplo). Todavia, alguns *overheads* de rede são impostos por protocolos, e não medidos, desconhecidos ou ignorados, como:

- Envio de cabeçalhos de pacote, *trailers* e informações de controle (CRC - Ethernet);
- Imposição de um limite no tamanho da janela (receber *buffer* - Ethernet);
- Atraso ao resolver nomes e endereços (IP);
- Uso de um aperto de mão (TCP - *handshake*) para iniciar e encerrar a comunicação;
- Redução da taxa de transmissão quando o congestionamento for detectado;
- Retransmissão de pacotes perdidos.

Então, como a Internet é baseada no modelo de camadas TCP/IP, pode-se dizer que, no uso de uma aplicação como *File Transfer Protocol* (FTP), um *overhead* será aplicado na rede, devido aos cabeçalhos impostos em cada camada por cada protocolo (KUROSE; ROSS, 2013).

reflita

Reflita

Geralmente, as operadoras de telecomunicações vendem pacotes de acordo com a capacidade do *link* de dados. Mas, ao longo do uso, você poderá precisar de mais vazão de dados. Então, você poderá comprar mais *throughput* , porém não poderá comprar menos *delay* . Você sabe o porquê?

Por fim, a terceira métrica é o *jitter* . Essa medida está se tornando importante quando as redes são usadas para a transmissão de voz, vídeo em tempo real e jogos *on-line* , associada a outras duas métricas. Uma dica importante é que duas redes podem ter o mesmo atraso e não o mesmo *jitter* .

Mas como evitar o *jitter* ? Há duas abordagens gerais que são aplicadas para evitá-lo ou minimizá-lo: projetar uma rede isócrona sem *jitter* – redes de telefonia são projetadas com esse propósito –; ou usar protocolo que faz uma compensação do *jitter* . Com relação a protocolo, por exemplo, o protocolo de tempo real (RTP) faz compensação de problemas causados por *jitter* em um sistema de transmissão de voz ou vídeo.

Então, pode-se dizer que a medição dessas métricas é relativamente simples, principalmente de *throughput* e *jitter* , pois medir o *throughput* é somente medir a taxa de dados enviado por segundos, enquanto o *Jitter* é medir a chegada dos pacotes e comparar se seus intervalos são semelhantes. Mas, ao contrário desses dois, medir atraso depende da relação de sincronismo entre a origem e o destino por meio de um relógio sincronizado entre os dois.

No entanto, as aplicações de gerenciamento de rede monitoram por outras métricas como medir o tempo de ida/volta do pacote e dividir por dois. Por exemplo, o comando *ping* pode ser usado para medir essa métrica. Essa complexidade ou dificuldade de medir o atraso da rede pode ser resumida em quatro razões:

- Os roteadores podem ser assimétricos;
- As condições de rede são dinâmicas e podem mudar rapidamente;
- O gerenciamento pode afetar o desempenho;
- Tráfego volumoso.

Logo, os gestores de redes podem medir a rede por dois caminhos: de forma passiva ou ativa. A forma ativa ocorre inserindo-se tráfego na rede, e esse procedimento causa alteração em seu desempenho. O contrário acontece na forma passiva, pois monitora e analisa os pacotes, mas não interfere com inserção de dados na rede.

Essa complexidade de parâmetros a ser medidos tem uma única finalidade: manter a boa qualidade da rede, assunto do próximo tópico.

Prioridade de Serviços

No planejamento de uma rede de computadores, alguns requisitos devem ser levantados, por exemplo, se serviços sensíveis à interferência serão utilizados, como Voz sobre IP e *stream* de vídeo, entre outros. Contudo, os administradores de rede devem implantar mecanismos que garantam o bom desempenho desses serviços na rede.

Essa “garantia” é determinada pela prioridade de serviço implantada sobre a rede usada pelos usuários e conhecida como qualidade de serviço ou *Quality of Service* (QoS). Assim, os serviços na rede ganham prioridades, que são configuradas diretamente nos componentes de rede.

Vamos a um exemplo de aplicação de qualidade de serviço por meio de serviços oferecidos por um provedor, tal como a empresa que fornece *link* de

Internet. Quando um determinado usuário contrata um serviço de Internet na sua forma mais básica, ele geralmente contrata por taxa de banda, a exemplo, 5 Mbps, 10 Mbps etc. Por outro lado, as empresas oferecem diferentes pacotes com preços variados associados a um conjunto de garantias mais específicas, pois atrelam prioridades de serviços a esses pacotes.

Suponha que esse mesmo usuário altere o pacote dele para um pacote *Premium* e esse pacote subscreva um pacote básico. Então, todas as vezes que esse usuário usar a sua Internet para jogar ou ver filmes, o pacote *Premium* terá prioridade de execução sobre o pacote básicos dos clientes.

Porém, grandes companhias trabalham com contratos mais detalhados, com muitas impostas aos dois lados, para a provedora e para a empresa. Esses contratos são chamados de *Service Level Agreement* (SLA) e especificam o mínimo de garantias à empresa para que sua infraestrutura seja de boa qualidade e não indisponibilize serviços essenciais ao negócio.

Já o QoS implementado aos elementos de rede e roteadores determina prioridade no fluxo de dados ao longo da rede. Em vez de priorizar o tráfego individualmente, o sistema é implementado em duas classes, sendo que o roteador prioriza a classe à qual pertence o pacote. Logo, podemos exemplificar duas transmissões na rede: uma por voz ou vídeo e outra por e-mail. A transmissão de voz ou vídeo tem prioridade sobre o e-mail enviado pela rede.

Em suma, na implementação do QoS, um elemento de rede, um switch ou roteador, classifica e policia dados de entrada, encaminha e, posteriormente, insere cada pacote em uma fila de saída. Depois, através de um agendador de tráfego, seleciona os pacotes para enviar por uma porta lógica de saída quando esta estiver livre. Esse processo está ilustrado na Figura 3.1.

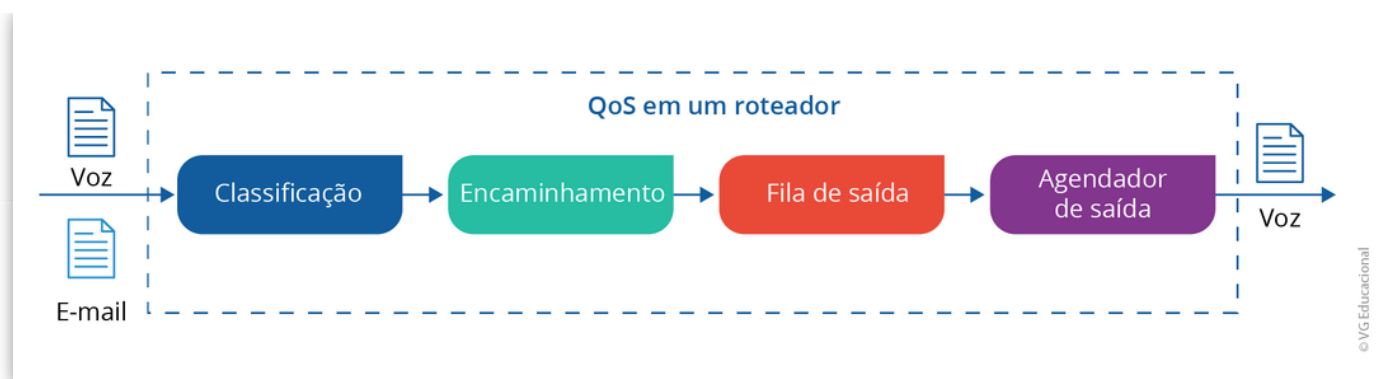


Figura 3.1 - Processo de QoS

Fonte: Elaborada pelo autor.

No processo apresentado na Figura 3.1, vários são utilizados no agendamento de saída que não são o foco desse estudo, mas, como ilustração, estão os algoritmos *leaky bucket* e *token bucket*. Porém, algumas ferramentas podem ser implantadas para monitorar a rede por meio dos protocolos como RTP. Esse será o assunto abordado no próximo tópico.

Vamos Praticar

Os jogos *on-line* estão cada vez mais presentes entre os usuários de Internet, seja jogos corporativos ou jogos de entretenimento, precisam de uma boa qualidade de infraestrutura de rede para que alcancem desempenho satisfatório.

Assinale a alternativa correta com relação a métricas de desempenho de rede e sua correta definição.

- ☐ **a) Jitter** : é o atraso ocorrido do pacote ao longo da rede.
- ☐ **b) Vazão**: capacidade de processamento dos equipamentos de rede.
- ☐ **c) Forward** : atraso provocado pelo acesso ao meio.

- **d)** Latência: atraso resultante da transferência de dados ao longo da rede.
 - **e)** Resposta do servidor: atraso provocado pelo acesso ao meio.
-

Protocolos de Sistemas Sensíveis

Agora, vamos abordar os protocolos que são utilizados por esses sistemas sensíveis como VoIP (Voz sobre IP), vídeos, entre outros, incluindo o protocolo RTP e o SIP.

Esses protocolos são usados para sistema de vídeo e áudio via Internet. Sistemas de vídeo e áudio via Internet são muito similares aos sistemas de áudio e vídeo por estação de rádio e TV. Porém, a única diferença é o meio de comunicação, pois o sistema via Internet usa a transmissão pelo modelo de camadas TCP/IP, ao contrário das estações de rádio e TV tradicionais que é o ar. Esses sistemas são sensíveis à interferência na rede, como atrasos, e não permitem retransmissões (FOROUZAN, 2008).

Todavia, há uma diferença muito significativa entre os sistemas de vídeo e áudio pela Internet, que é forma de comunicação. O sistema de transmissão de áudio via Internet usa o sistema de entrega de dados *unicast* e sob demanda, ao contrário do sistema de vídeo, que usa o sistema de entrega dos dados *multicast* e tempo real.

Esses sistemas permitem que as pessoas se comuniquem em tempo real, seja

por voz ou imagens (vídeos), mas possuem algumas características na transmissão em tempo real.

Características de Sistema em Tempo Real

Sistemas de tempo real com áudio e vídeo possuem algumas características importantes que serão detalhadas a seguir.

Relação de tempo: os dados de uma transmissão de áudio ou vídeo em uma rede comutada por pacotes necessitam que se preserve a relação de tempo entre os pacotes em uma mesma sessão. Essa relação tem que ser fixa e constante, havendo variação nesse tempo ocorre o chamado *jitter* para o usuário, um efeito de atraso no som e no vídeo. Uma solução para evitar o *jitter* é sincronizar os pacotes com *timestamp* (registro de horas). O registro é feito no envio e ajustado na recepção sem atraso nos tempos.

Buffer de reprodução: é uma memória que auxilia no desempenho de um produto de computação. Em um sistema de tempo real, armazena os dados até que seja possível a sua reprodução.

Ordem dos pacotes: quando um pacote é enviado da origem ao destino, ele recebe um identificador que indica a sua ordem no conjunto de pacotes, denominado número de sequência. Além do tempo e do *timestamp*, há necessidade desse número para ordenar o pacote na recepção e até detectar algum tipo de falha ou perda.

Multicasting : o sistema de transmissão *multicasting* é essencial para transmissão de vídeo e áudio via Internet, pois distribui os dados pelo método de *multicast*.

Tradução: um tradutor é um computador que diminui o formato de sinal de áudio ou vídeo para um tipo de menor largura de banda e qualidade.

Mixagem: em uma transmissão de vídeo, por exemplo, diversas fontes de

dados enviam o sinal. Um *mixer* transforma as diversas fontes de dados em um único fluxo de dados.

Suporte ao protocolo camada de transporte: o TCP não é adequado, pois solicita retransmissão. Nesse caso, os sistemas fazem uso do protocolo UDP (*User Datagram Protocol*).

Isto posto, essas características não são suficientes para uso de um sistema de tempo real, há necessidade do uso de recursos adicionais, como o protocolo RTP.

Protocolo RTP

Com o uso de aplicações em tempo real na Internet, foi desenvolvido um protocolo para esse fim, denominado RTP (*Real-Time Transfer Protocol*).

As características do protocolo são apresentadas no Quadro 3.2.

Protocolo RTP

Não suporta mecanismo de entrega *multicasting*

Opera em conjunto com o protocolo UDP

Seu posicionamento é entre o protocolo UDP e o aplicativo

Possui *timestamp* , mixagem e sequenciamento

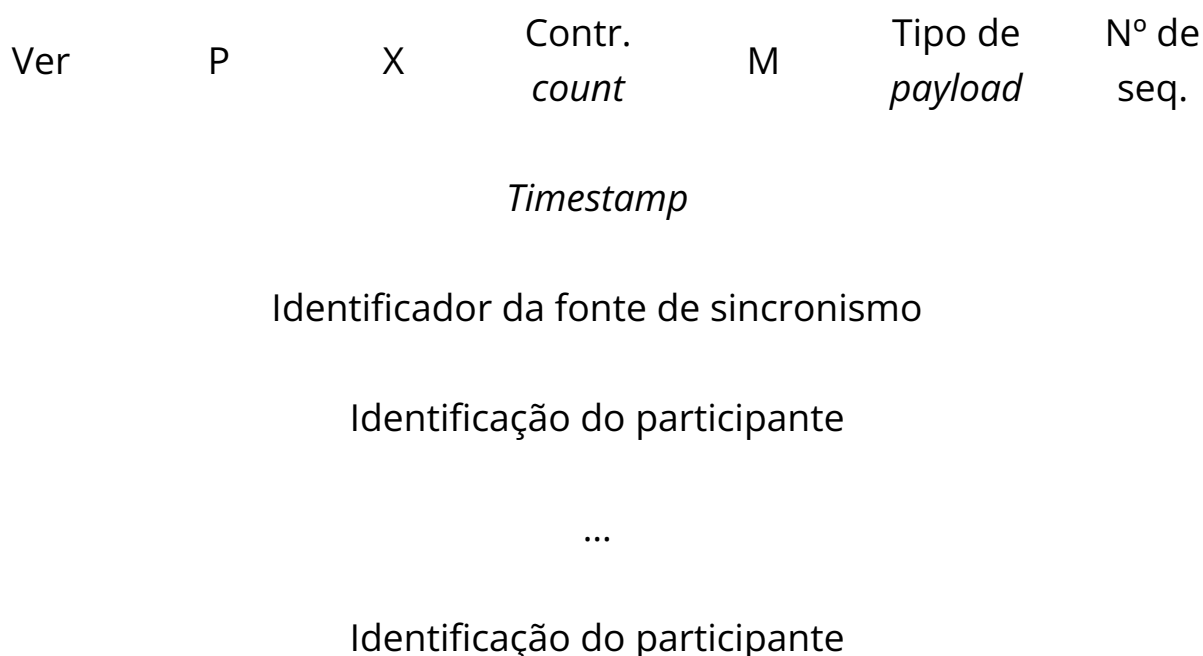
Situado na camada de transporte

Cabeçalho de pacotes simples

Quadro 3.2 - Características do RTP

Fonte: Elaborado pelo autor.

O protocolo RTP possui um formato simples de cabeçalho (Quadro 3.3). Mas, embora simples, atende a um conjunto de aplicações de tempo real. No processo de envio de dados da origem até o destino, o protocolo RTP faz uso de somente uma mensagem.



Quadro 3.3 - Cabeçalho RTP

Fonte: Elaborado pelo autor.

Os campos do cabeçalho apresentado no quadro são descritos por:

- Ver: número da versão
- P: determinação do preenchimento no final do pacote que define o comprimento total do pacote, caso seu valor seja 1.
- X: indica um cabeçalho de extensão extra.
- *Contributor count* : define o número de participantes, seu valor é de 4 bits.
- M: usado pela aplicação para determinar, por exemplo, o fim dos dados.
- *Payload type* : define o formato do *payload* (carga ou dados) que está sendo transmitido. Diversos formatos foram padronizados, incluindo o Áudio MPEG, valor 33.
- *Sequence Number* : usado para numerar os pacotes RTP.

- *Timestamp* : indicação da relação dos tempos de pacotes produzidos.
- *Synchronization source identification* : usado geralmente para determinar o mixer de diversas fontes.
- *Contributor Identifier* : identificador de cada fonte.

Um aspecto importante do protocolo RTP é que ele não faz uso de encapsulamento do protocolo IP, ao contrário, faz uso do protocolo UDP do usuário. Outra questão interessante é que a porta não é atribuída ao protocolo, mas solicitada sob demanda, sob a restrição única de ser par.

Após apresentarmos o protocolo RTP, agora precisamos entender como as sessões são criadas e mantidas em aplicações de áudio e vídeo interativo, por exemplo, jogos online. Esse é o tema do próximo tópico.

Protocolo RTP

A voz é transmitida via sistema de comutação por circuitos, ou seja, há necessidade de uma conexão estabelecida, para que possa haver uma comunicação entre os pares A e B, a exemplo da ligação por telefone discado.

A Figura 3.2 mostra uma imagem de uso da telefonia fixa, por linha, muito usada em décadas passadas, que utiliza comutação por circuitos e que vem sendo substituída, com o crescimento da Internet, por linhas de telefonia móvel e Voz sobre IP.

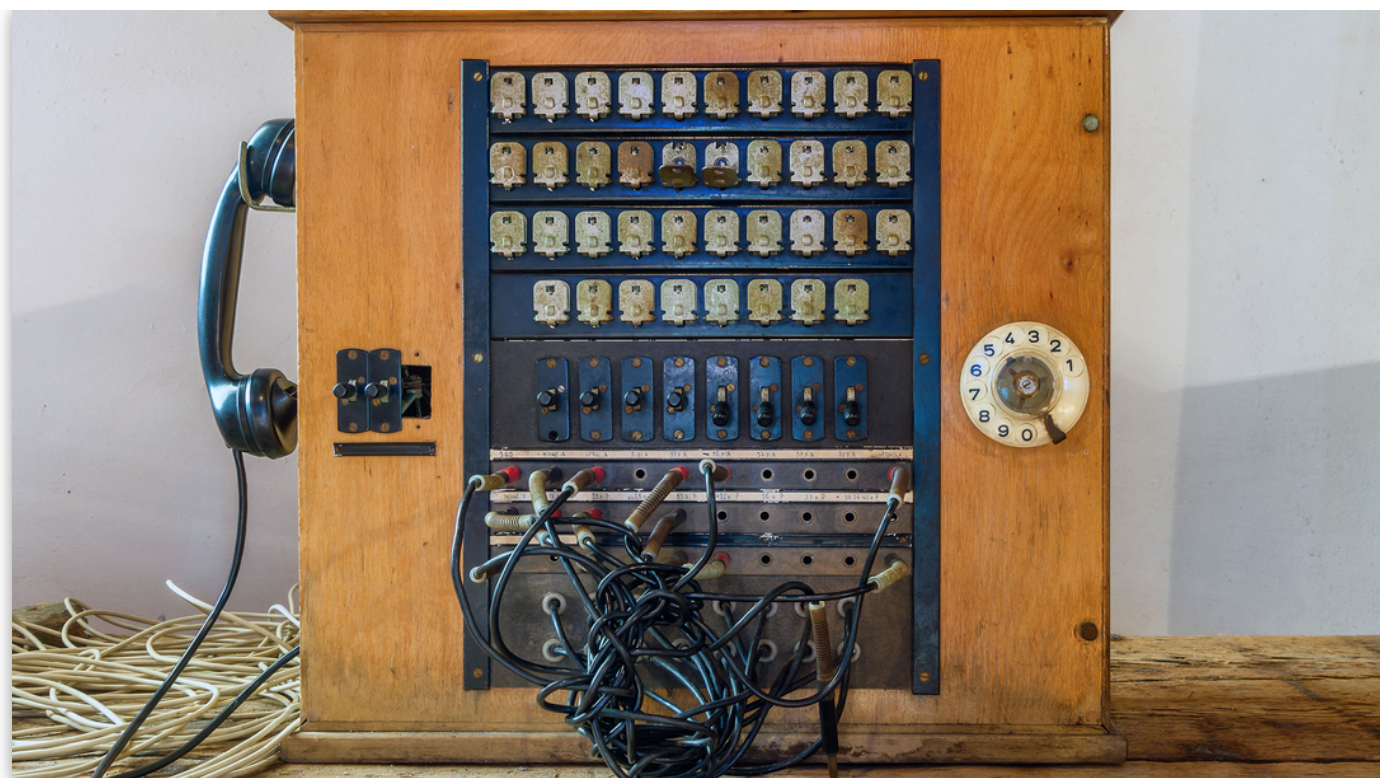


Figura 3.2 - Telefonia via cabo

Fonte: Josef / 123RF.

Similarmente, a telefonia tradicional, cabeada, vem sendo gradativamente sendo substituída pela telefonia VoIP, ou seja, está havendo uma substituição da comutação por circuitos por comutação por pacotes.

No intuito de promover a evolução da VoIP, o IETF (*Internet Engineering Task Force*) desenvolveu o protocolo SIP (*Session Information Protocol*). Esse protocolo de camada de aplicação tem a finalidade de estabelecer, gerenciar e encerrar uma sessão de comunicação multimídia. Seu desenvolvimento proveu uma independência do suporte da camada de transporte e, por essa razão, ele pode ser aplicado associado aos protocolos TCP e UDP.

Similar ao protocolo *Hypertext Transfer Protocol* (HTTP), pois também é baseado em troca de mensagens, o protocolo SIP define seis mensagens no seu processo de estabelecer a sessão, são elas: *invite*, *ack*, *bye*, *options*, *cancel* e *register* . Essas mensagens são ilustradas na Figura 3.3.

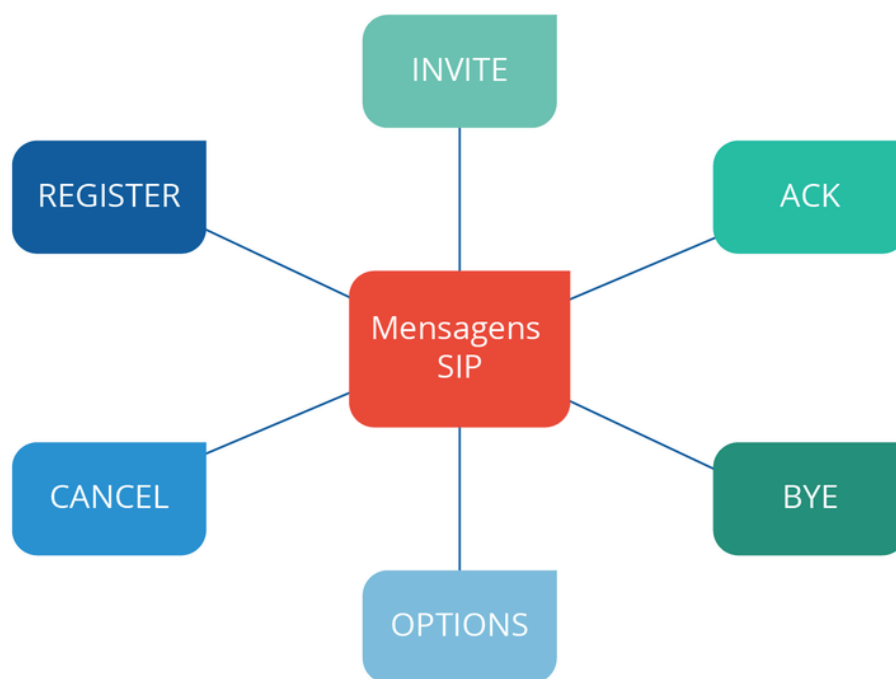


Figura 3.3 - Mensagens SIP

Fonte: Elaborada pelo autor.

Cada uma dessas mensagens designa processos diferentes:

- *Invite* : mensagem enviada para iniciar uma sessão.
- *Ack* : mensagem de confirmação enviada pelo receptor após o Invite.
- *Bye* : encerra uma sessão.
- *Options* : faz consulta à máquina a recursos disponíveis.
- *Cancel* : cancela um processo já iniciado.
- *Register* : estabelece uma conexão quando há impossibilidade no receptor da chamada.

A telefonia por circuitos contém um endereço, os números telefônicos. Todavia, quando se trata de telefonia IP, o conceito é bem semelhante. Quando é Voz sobre Ip (VoIP), cada ramal possui um endereço. Entretanto, devido ao protocolo SIP, esse endereço é mais flexível, pois pode endereçar ao ramal por meio de um e-mail, endereços lógicos, etc., um número de telefone convencional e outros tipos de endereços na identificação tanto do transmissor quanto do receptor. Uma única restrição tem que ser atendida: todos os endereços devem estar no formato do protocolo SIP, também conhecido como *schema* SIP, como, por exemplo,

sip:estudante@201.25.60.112 (formato IP), sip:estudante@faculdade.br (formato e-mail) e sip:estudante@9-9999-8888 (formato telefone) (FOROUZAN, 2008).

O protocolo SIP estabelece uma sessão simples em três fases distintas: fase 1 – estabelecimento de sessão; fase 2 – comunicação; e fase 3 – encerramento da sessão. Um ponto muito interessante no uso do protocolo SIP é o processo de descoberta do ramal de destino, ou seja, descobrir o destinatário quando este não está à frente da aplicação, e sim em outro sistema ou em outra estação (STALLINGS, 1999).



Você ficou curioso(a) para entender mais sobre o protocolo SIP. Então, leia o RFC 3261, que o apresenta detalhadamente. Uma excelente fonte de aprofundamento de seus conhecimentos. Boa leitura!

Para ler o conteúdo, acesse o *link* a seguir.

ACESSAR

O protocolo SIP possui um mecanismo de descoberta de endereço do destinatário que usa servidores de registro. Cada usuário está registrado em pelo menos um servidor de registro. Então o protocolo faz a consulta a esse servidor para descobrir o destinatário.

A Figura 3.4 ilustra o processo de descoberta do destinatário por meio de servidor de registro.

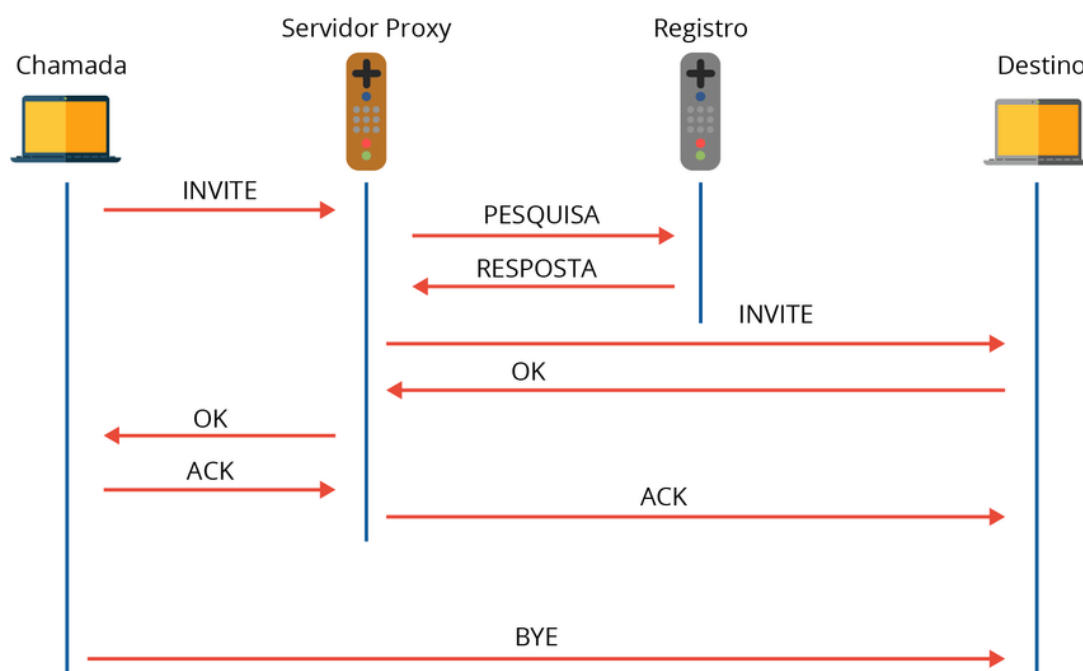


Figura 3.4 - Descoberta de destinatário pelo SIP

Fonte: Elaborada pelo autor.

O processo de descoberta do protocolo SIP é similar ao processo de descoberta do DNS (*Domain Name Server*) na Internet. Porém, nem tudo são flores!

No cotidiano dos usuários, diversas redes são usadas. No entanto, esses usuários não sabem quais equipamentos são seguros e confiáveis, pois somente fazem uso devido à transparência da rede, ou seja, essas redes são chamadas de redes não confiáveis. Como resultado do uso de redes não confiáveis, surgem diversas reclamações sobre o serviço de rede, como travamento de sinalização, queda, não conexão etc.

Então, nesse cenário de insatisfação, criaram-se dois mecanismos de medição da qualidade da rede, além das métricas tradicionais, como vazão, tempo de retorno, *jitter* , taxa de erros etc. Esses mecanismos são o fator R e a pontuação MOS (*Mean Opinion Score*). Ambos os fatores somam a experiência do usuário às métricas de medição da qualidade da rede (LIMA, 2006).

Em redes IP com VoIP, o fator R é usado para medir a qualidade da

comunicação entre os usuários. É um fator escalar, ou seja, recebe valores entre 0 (menor satisfação) e 100 (satisfação plena), de acordo com variáveis como o sinal do ruído (R_o), quantização da voz (I_s), fator de degradação (qualidade da voz), atraso I_d , I_e relacionado à distorção de sinal (codecs, atraso, perda de pacotes, *buffers*, compensação de *jitter*) e, por último, o valor aceitável pelo usuário quanto ao uso da aplicação simbolizado por (A). Vamos a um exemplo demonstrativo. Suponha que o valor padrão do sinal do ruído seja 94,77, o valor padrão da degradação do sinal 1,44, qualidade da voz 20 (esses valores vão de 0 a 60), percepção do usuário 5 (valores de 0 a 20) e, por último, a relação da distorção do sinal, que varia entre 0, 1 (0 sem perda, igual a 1 perda aleatória, e maior que 1, perda de pacotes).

A especificação G.107 da ITU-T de 2015 usa a fórmula $R = (R_o - I_s - I_d - I_e - A)$ para calcular o valor do fator R e associa ao quadro de referência o resultado para medir a satisfação do usuário, ilustrado no Quadro 3.4.

Fator R	Satisfação do usuário
90 -100	Muito satisfeito
80 - 90	Satisfeito
70 - 80	Alguns usuários insatisfeitos
60 - 70	Muitos usuários insatisfeitos
50 a 60	Praticamente todos insatisfeitos
0 - 50	Não recomendável

Quadro 3.4 - Relação fator R x Satisfação do usuário

Fonte: Lima (2006, p. 35).

Como resultado do exemplo, temos: $R = 94,77 - 1,44 - 20 - 5 - 0$, então, $R = 68,33$. Fazendo referência ao Quadro 3.1, temos que muitos usuários estão

satisfeitos com relação à qualidade do serviço. Em média, fatores acima de 70 têm boa qualidade de comunicação.

Ao contrário do fator R, o método MOS, ITU-T P.800 utiliza valores bem mais baixos: 1 para comunicação ruim e 5 para excelente comunicação, em média acima de 3.6 é considerado satisfatório. Outra diferença entre essas métricas é que o fator R tem como referência o *jitter*, atraso e perda de pacotes. Em contrapartida, a métrica MOS também faz uso da experiência de usuário. O MOS pode ser calculado por meio do fator R pela seguinte fórmula: $MOS = 1 + 0.035R + 7 \times 10^{-6} \times R(R - 60)(100 - R)$ (ITI-T, G.107, 2015). Então, com o fator R sendo 68,33, temos $MOS = (1 + 0,035 \times 68,33 + 7 \times 10^{-6} \times 68,33 \times (68,33 - 60) \times (100 - 68,33))$. Como resultado o fator 3,517. Nesse caso, faz-se o uso do quadro de referência, quadro 3.5, para medir a satisfação do usuário segundo MOS.

MOS	Satisfação do usuário
4,3 a 5	Muito satisfeito
4,0 a 4,3	Satisfeito
3,6 a 4,0	Alguns usuários insatisfeitos
3,1 a 3,6	Muitos usuários insatisfeitos
2,8 a 3,1	Praticamente todos insatisfeitos
1,8 a 2,8	Não recomendável

Quadro 3.5 - Relação MOS x Satisfação do usuário

Fonte: Lima (2006, p. 35).

A referência do quadro com o exemplo demonstrativo indica que muitos usuários estão insatisfeitos. No próximo bloco, será apresentada uma ferramenta para uso de protocolos como o SIP e o RTP.

Ferramenta *StarTrinity*

A *StarTrinity* é uma ferramenta que analisa protocolos SIP e RTP. Um *software* desenvolvido em base de *feedback*, que disponibiliza no próprio site diversos tutoriais para uso e teste de rede com aplicações sensíveis à interferência como atraso e *jitter*.

A aplicação possui uma arquitetura subdividida em módulos, mas também permite o uso de protocolos, contém conceitos técnicos e tecnologias. A Figura 3.5 mostra essa arquitetura.

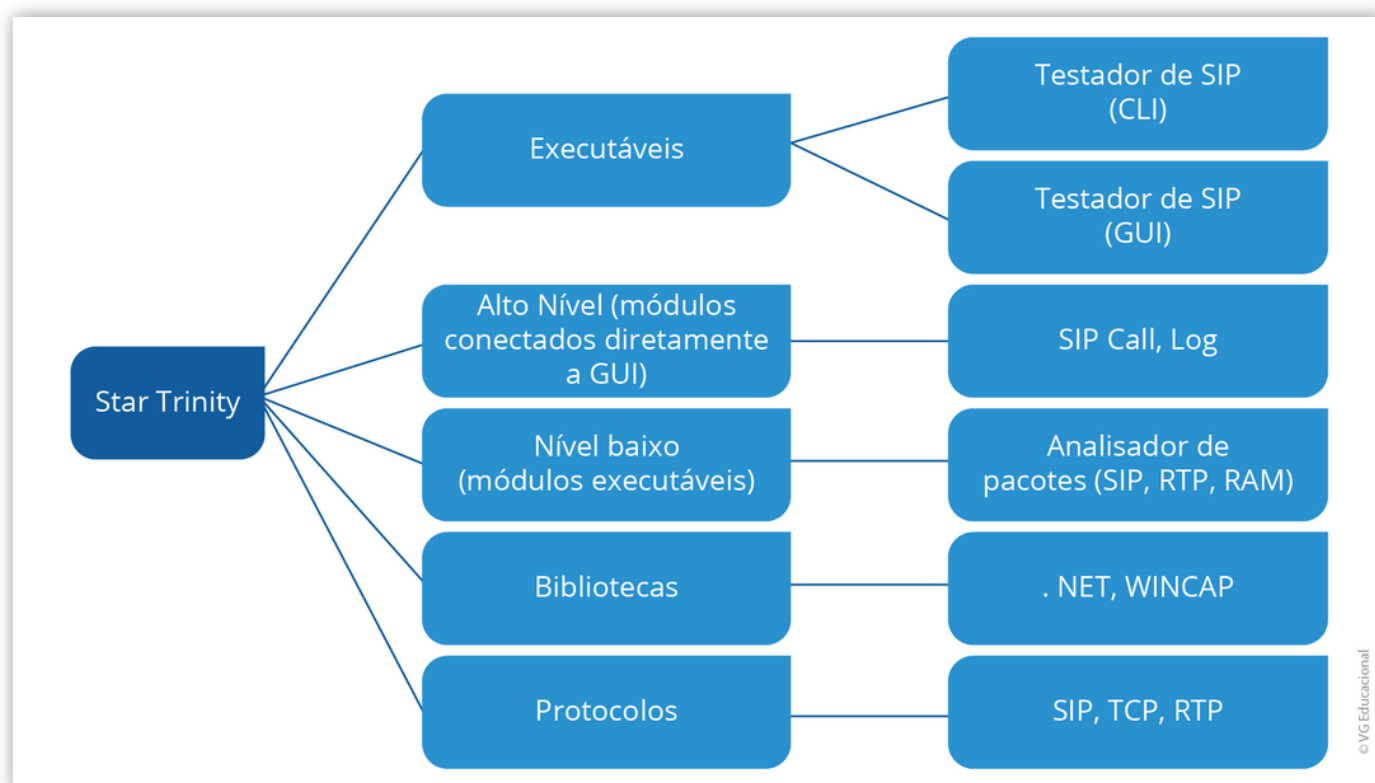


Figura 3.5 - Arquitetura do software *StarTrinity*

Fonte: Elaborada pelo autor.

Pela Figura 3.5, nota-se diversos blocos que dividem a aplicação em sub-blocos de programas executáveis, módulos de apresentação por interfaces gráficas, modelo de execução (*runtime*), bibliotecas usadas no sistema e protocolos. Essa arquitetura permite uma visão global das opções que podem ser usadas por um administrador de rede.

O uso da ferramenta é dependente da implantação de um sistema VoIP para teste, mas esse não é foco da unidade, e sim apresentar como a ferramenta é

aplicada e os seus resultados. Para tal demonstração, o próprio site oferece testes online com *feedbacks* para análise de comportamento da ferramenta com relação aos protocolos.

Vamos a um exemplo de teste utilizando a ferramenta disponível no site da *StarTrinity* . Após acessar o site, clique na opção VoIP Status para termos acesso à *StarTrinity* VoIP Status. Uma instância de um servidor SIP que envia e recebe chamadas SIPs faz a análise e envia um relatório. Muito útil para o entendimento inicial de como funciona o protocolo. Na própria página da ferramenta, uma ilustração mostra toda a arquitetura do sistema usado para teste.

Na seção *Free trial* , clique no *link free trial* para a página da criação de um novo VoIP test. Um formulário para configuração surgirá na tela com o ID do teste, o servidor, algoritmo de codec usado em sistema VoIP G729, G711U e G711A, número de canais concorrentes, duração da conexão, e pronto.

Saiba mais

Se você ficou interessado(a) na ferramenta *StarTrinity* , o site da aplicação contém um livro em formato HTML muito completo e rico em informações. Navegando pelos *links* , você terá a oportunidade de enriquecer os seus conhecimentos a respeito da aplicação. Boa leitura!

Para ler o conteúdo, acesse o *link* a seguir.

ACESSAR

Na verdade, você tem de escolher somente o tipo de código e deixar o resto

com configuração padrão. Após a caixa de configuração ser preenchida, clique no botão de criar. Um *link* para baixar o executável surgirá na tela. Baixe o executável e execute-o. Um teste começará a ser executado. A ferramenta também disponibiliza um *trial* para teste. A tela inicial da ferramenta é mostrada na Figura 3.6.

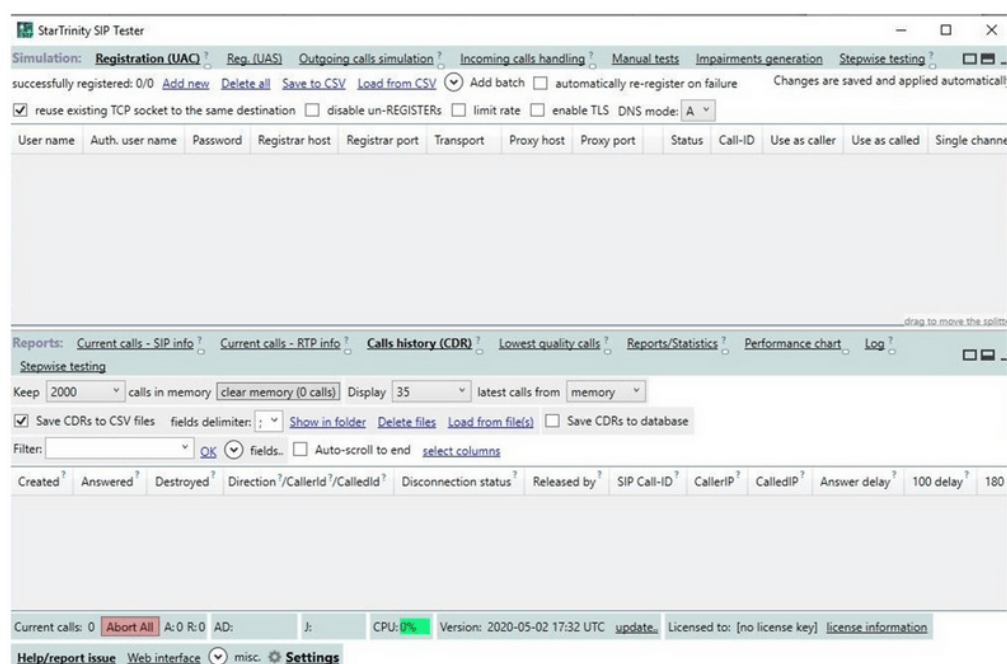


Figura 3.6 - Ferramenta StarTrinity

Fonte: Arquitetura... (2020, on-line).

Após a conclusão do teste, clique no *link view results* para verificar o resultado. A conclusão do teste dependerá da infraestrutura que estiver aplicando o *software*, podendo levar mais ou menos tempo, porém o interessante da ferramenta é que ela mostra o teste sendo realizado, as portas de conexão e outras informações.

Mas a telefonia possui outros mecanismos que também são usados como serviços pela rede, esse é o caso do *Asterisk*, tema abordado do próximo tópico.

Fundamentos da Telefonia IP

A convergência dos modelos de negócio para um modelo baseado na Internet é inquestionável, tanto que a telefonia, antes comutada por circuitos, agora passa à comutação por pacotes, e como meio faz uso também da Internet (TANENBAUM; WETHERALL, 2011).

O protocolo de Internet (IP), segundo Kurose e Ross (2013, p. 452),

provê um serviço de melhor esforço. Em outras palavras, a Internet faz seu melhor esforço para transportar cada datagrama do remetente ao receptor o mais rapidamente possível, mas não faz nenhuma promessa sequer sobre o atraso fim a fim para um pacote individual, ou sobre o limite de perdas de pacotes.

Então, o termo VoIP representa a comunicação por voz sobre o protocolo IP ou telefonia IP.

Telefonia IP

O Skype é uma das mais variadas aplicações usadas pelos usuários de Internet com base na telefonia IP ou VoIP. Em termos de infraestrutura, as operadoras substituem os tradicionais comutadores de circuitos telefônicos por roteadores. A razão é simples: os roteadores trazem um custo menor e o sistema VoIP agrega mais valores, por exemplo, fazer reuniões com pessoas geograficamente distantes e virtualmente conectadas, fora o envio de dados e o uso de uma rede compartilhada.

O conceito básico da telefonia IP é a amostra contínua de áudio, ou seja, converta cada amostra em forma digital, envie o fluxo digitalizado resultante por uma rede IP em pacotes e converta o fluxo de volta para analógico para reprodução (COMER, 2015).

A telefonia IP faz uso de diversos protocolos para sinalização de chamadas VoIP, o H.323 e o SIP. A Figura 3.7 mostra uma macrovisão do sistema VoIP.



Figura 3.7 - Sistema VoIP

Fonte: alexlmx / 123RF.

Observa-se pela imagem que uma organização para adotar o sistema VoIP somente precisa da aquisição de licenciamento de um servidor VoIP (qualquer empresa que opera na Nuvem), pois o sistema será implantado respeitando o

legado de rede da empresa contratante do serviço, além dos equipamentos de telefone VoIP físicos. Muitas empresas preferem o uso de seus próprios serviços e, por sua vez, fazem uso de sistema de telefonia de código aberto como o Asterisk.

Fundamentos de Gerência sobre Asterisk

O Asterisk é um sistema de código aberto, ou seja, um *software*, não um *hardware*, que pode ser baixado da Internet, configurado e usado como um servidor de telefonia IP, inclusive pode ser associado a *softphones* para aplicação pessoal, mas, principalmente, empresarial (WINTERMEYER; BOSCH, 2009).

Uma característica do Asterisk é que ele trabalha com uma gama de tecnologias, incluindo a analógica, ISDN e VoIP. A implantação do Asterisk permite ilimitadas possibilidades, desde uma diminuição dos custos até a ampliação de funcionalidades do sistema.

Saiba mais

Gostou do sistema Asterisk? Se deseja compreender mais sobre o sistema Asterisk, no site FreePBX há diversas imagens prontas (.iso) e com a instalação Asterisk já realizada. É um excelente estudo, pois você precisa somente baixar, instalar e começar a praticar e consolidar seus conhecimentos.

Para conferir o conteúdo, acesse o *link* a seguir.

ACESSAR

O servidor Asterisk pode ser gerenciado via Zabbix, bastando, para isso, adicionar como host o servidor Asterisk. Mas, antes disso, você precisa configurar o Asterisk para que suporte ao protocolo SNMP e MIB. Logo depois da instalação e das configurações do servidor Asterisk, você precisa resetar o *software* para que as configurações sejam ativadas. Em seguida, pode-se fazer um pequeno teste de funcionamento. Para tal processo, faça uso do comando `snmpwalk` para verificar do OID para MIB (`snmpwalk -v2c -c 127.0.0.1`). Ambos os comandos foram especificados em tópicos anteriores.

Após os itens configurados no servidor Asterisk e depois de ter criado o *host* na aplicação Zabbix, é preciso criar os itens a serem gerenciados no servidor Asterisk. A seguir, um exemplo de configuração do item para resgate da versão do Asterisk.

- Nome: versão do asterisco
- Tipo: Agente SNMPv2
- Chave: asterisk.version

- OID: 1.3.6.1.4.1.22736.1.1.1.0
- Comunidade SNMP: {\$ SNMP_COMMUNITY}
- Tipo de informação: texto
- Intervalo de atualização: 30s
- Aplicação: Asterisk

Basicamente, os itens são alterados com relação à chave e ao respectivo OID, o tipo de informação texto, numérico, e o tempo. Caso queira, é possível adicionar *triggers* (gatilhos) que sejam disparados e possam executar ações. Lembrando que a função básica de um *trigger* {<server>:<key>.<function> (<parameter>)}<operator><constant>, em que *server* é o servidor, *key* a chave, *functions* são as funções próprias do Zabbix, *parameter* são parâmetros passados, por exemplo, tempo em segundos, *operator* determina operadores como (ou) operador lógico OU, (+) soma aritmética, *constant* um valor constante de referência conforme a expressão do *trigger*.

A seguir, algumas expressões *triggers* :

Análise de carga de processamento na CPU:
{www.teste.com:system.cpu.load[all,avg1].last()}>5

Troca de senha: {www.teste.com:vfs.file.cksum[/etc/passwd].diff()}=1

Servidor indisponível: {asterisk.asterisk.com:icmpping.count(30m,0)}>5

Verifica se o horário local do *host* monitorado e o do servidor do Zabbix estão sincronizados: {asterisk:system.localtime.fuzzytime(10)}=0

Entretanto, o uso das ferramentas tanto de gerenciamento Zabbix quanto de implantação de VoIP, Asterisk, depende de um bom planejamento e levantamento correto dos requisitos de rede pelo profissional de TI responsável. O resultado da boa qualidade de rede está diretamente associado a esse planejamento.

praticar

Vamos Praticar

Usando a ferramenta de simulação *packet tracer*, crie uma pequena rede VoIP e utilize para essa rede: um roteador, um switch e dois telefones IP. Faça a conexão entre esses dispositivos, depois um teste de ligação entre os aparelhos e verifique os protocolos no modo simulação da ferramenta. Vamos à prática?

indicações

Material Complementar



FILME

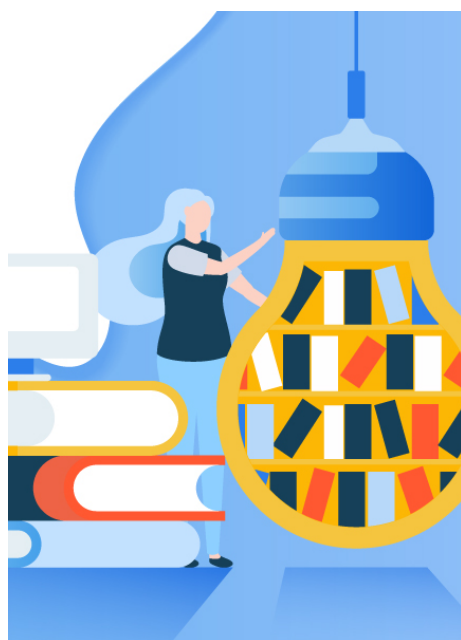
O quinto poder

Ano: 2013

Comentário: esse filme mostra a história do compartilhamento de dados e o uso de redes sociais. Na Internet, por meio das redes sociais, os usuários não têm limites. Excelente filme para o estudante compreender e analisar requisitos que devem ser monitorados e gerenciados em uma rede empresarial, com foco nas redes sociais e no fluxo de informação.

Para conhecer mais sobre o filme, acesse o *trailer* a seguir:

TRAILER



LIVRO

Asterisk: the definitive guide

Editora: Novatec

Autores: Jim Van Meggelen, Russell Bryant e Leif Madsen

ISBN: 978-85-7522-821-0

Comentário: esse livro descreve os conceitos sobre a aplicação Asterisk. O capítulo 3, em especial, apresenta os requisitos necessários para a instalação do produto, com dicas de instalação, apresentação de erros,

comandos shell e muito mais. Um excelente complemento de estudos para a unidade.

conclusão

Conclusão

Ao longo desta unidade, estudamos os conceitos iniciais, aprendemos sobre o conceito de QoS, vimos que a qualidade de uma rede está associada ao conceito, bem como aos serviços ofertados pela empresa de *links* de Internet.

Observamos também ferramentas usadas para monitoramento de protocolos como o SIP e o RTP. Esses protocolos são usados por ferramentas na comunicação entre usuários de Internet, *stream* de vídeo e jogos online e compensam problemas de transmissão de dados entre origem e destino.

Logo em seguida, apresentamos o conceito sobre telefonia IP, tecnologia de vasta implantação nas organizações, bem como os fundamentos de uso da ferramenta de gerenciamento Zabbix sobre um servidor Asterisk.

Então, nesta unidade foi possível definir os conceitos de padrões e critérios sobre QoS, assim como entender uma ferramenta para medição, a *StarTrinity*. Também foi possível identificar o uso dos protocolos SIP e RTP, analisar expressões de *trigger* e reconhecer o uso do servidor Asterisk.

referências

Referências Bibliográficas

ARQUITETURA do software **StarTrinity** . StarTrinity. Disponível em: <https://startrinity.com/VoIP/Architecture.aspx>. Acesso em: 20 abr. 2020.

COMER, D. E. **Computer networking and internet** . 6. ed. New Jersey: Pearson Education, 2015. 668 p.

FOROUZAN, B. A. **Comunicação de dados e redes de computadores** . Porto Alegre: AMGH Editora, 2009.

KUROSE, J.; ROSS, K. W. **Redes de computadores e a internet** : uma abordagem top-down. Tradução por Daniel Vieira. Revisão técnica por Wagner Luiz Zucchi. 6. ed. São Paulo: Pearson Education do Brasil, 2013.

LIMA, A. F. M. **Monitoramento SNMP para avaliar a qualidade das chamadas em um ambiente VoIP** . 109 f. Dissertação (Mestrado em Engenharia Teleinformática). Centro de Tecnologia – Universidade Federal do Ceará, Ceará, 2006.

STALLINGS, W. **SNMP, SNMPv2, SNMPv3, and RMON1 and 2** . 3. ed. Reading, Massachusetts: Addison-Wesley, 1999.

TANENBAUM, A. S.; WETHERALL, D. **Redes de computadores** . 5. ed. Pearson, 2011.

WINTERMEYER, S.; BOSCH, S. **Practical asterisk 1.4 and 1.6** : from beginner to expert. New Jersey: Pearson Education, 2009.