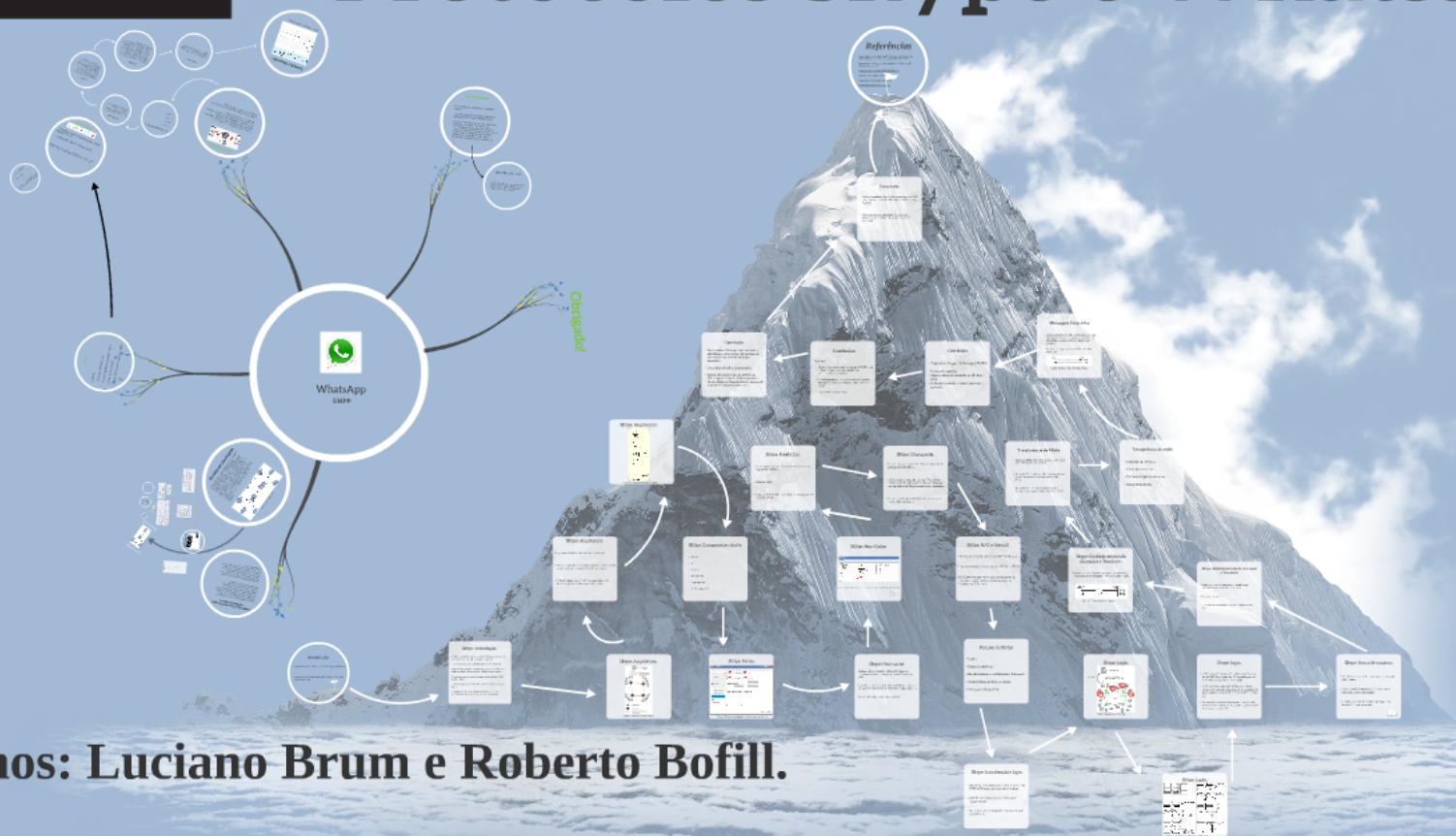
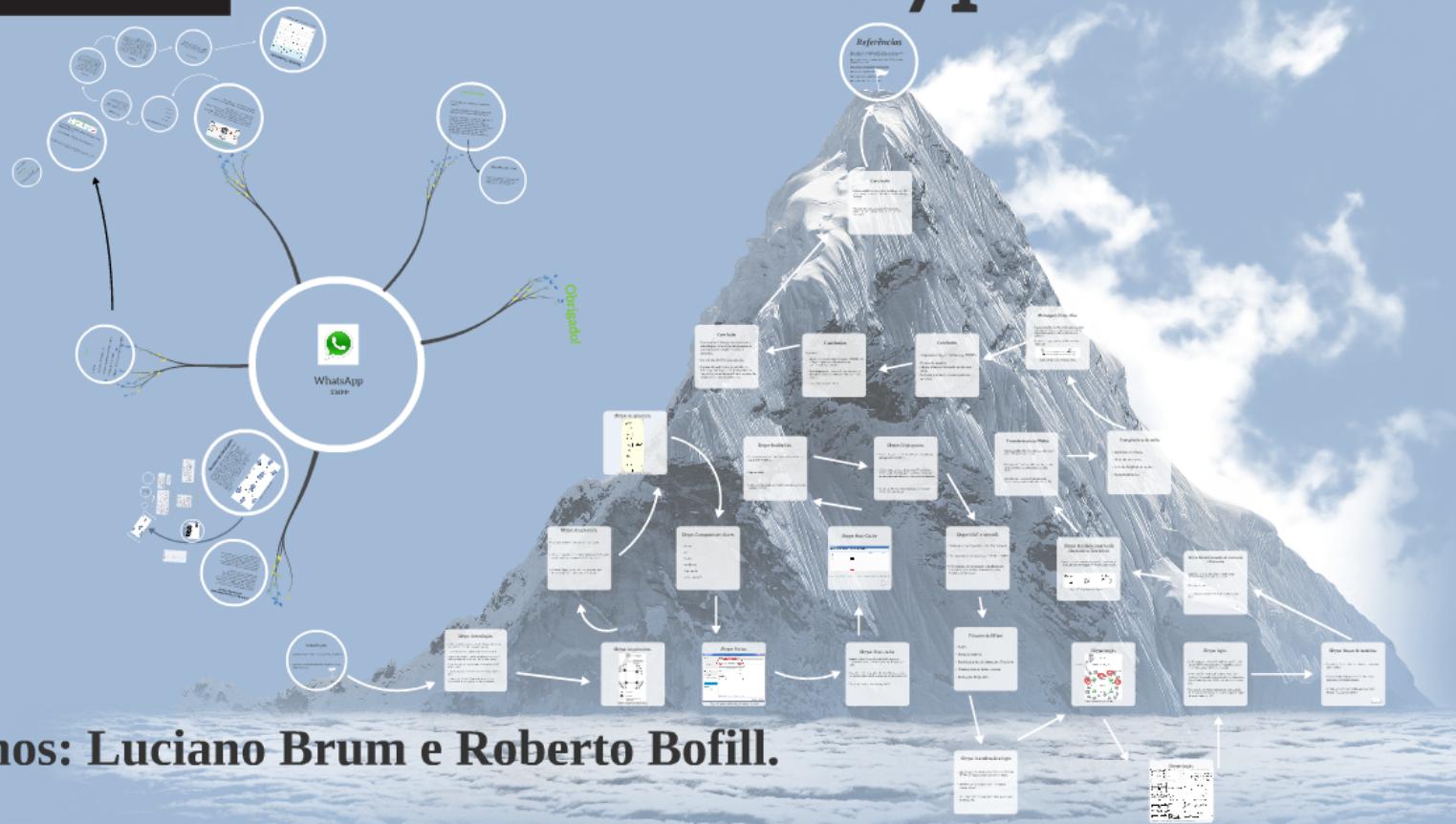


Uma Comparação entre os Protocolos Skype e Whatsapp



Alunos: Luciano Brum e Roberto Bofill.

Uma Comparaço entre os Protocolos Skype e Whatsapp



Alunos: Luciano Brum e Roberto Bofill.

Introdução

- Apresentação da arquitetura Skype e alguns conceitos.
- Apresentação do protocolo utilizado pelo Whatsapp e seus conceitos.



Skype: introdução.

- O Skype é um cliente de VoIP, desenvolvido pelo KaZaa, que atua sobre uma rede sobreposta ponto-a-ponto.
- No windows, foi desenvolvido em pascal usando Delphi
- Skype afirma que pode funcionar em NAT e firewalls e tem melhor qualidade de voz do que o MSN, Yahoo e outros.
- Possui dois tipos de nós: nós comuns (ordinary host) e SN (Super Nodes).
- Nomes de usuários e senhas são armazenadas no servidor de login.
- A autenticação de login é feita também neste servidor. (Centralizado, informações on e off descentralizadas!)

Skype: Arquitetura.

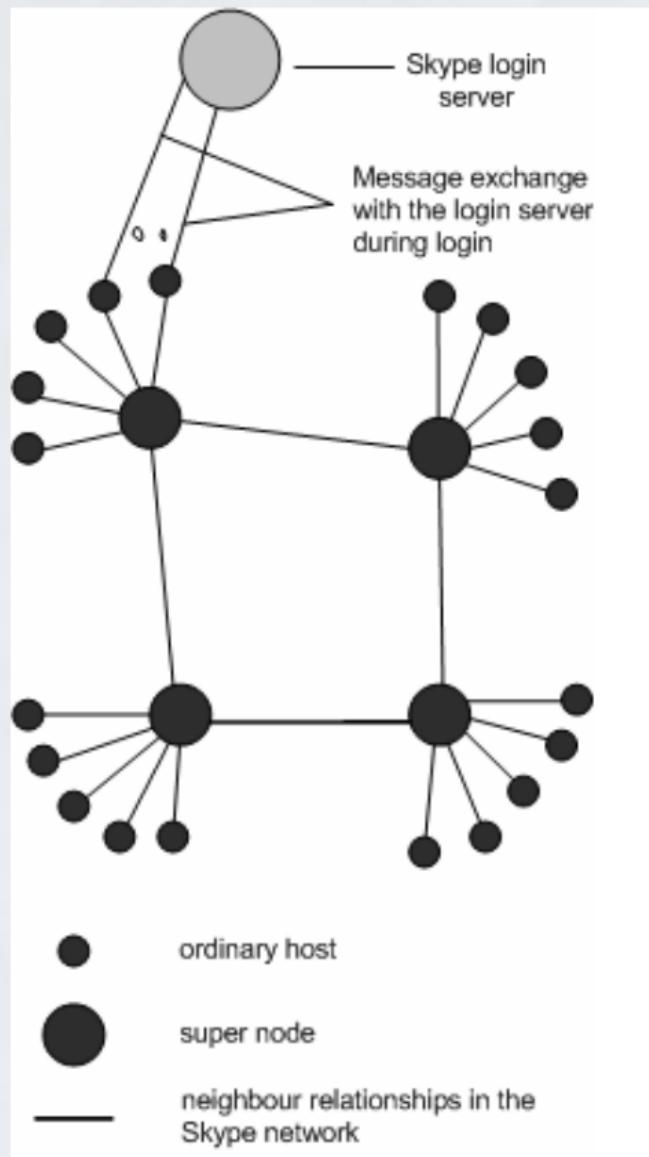


Figura 1: entidades do protocolo Skype.

Skype: Arquitetura.

- O protocolo TCP é utilizado para sinalização.
- Tanto o protocolo TCP quanto o protocolo UDP podem ser utilizados para transporte do fluxo de mídia.
- O cliente Skype, ao ser instalado, gera uma porta aleatória, que será utilizada para a conexão.

Skype: Arquitetura.

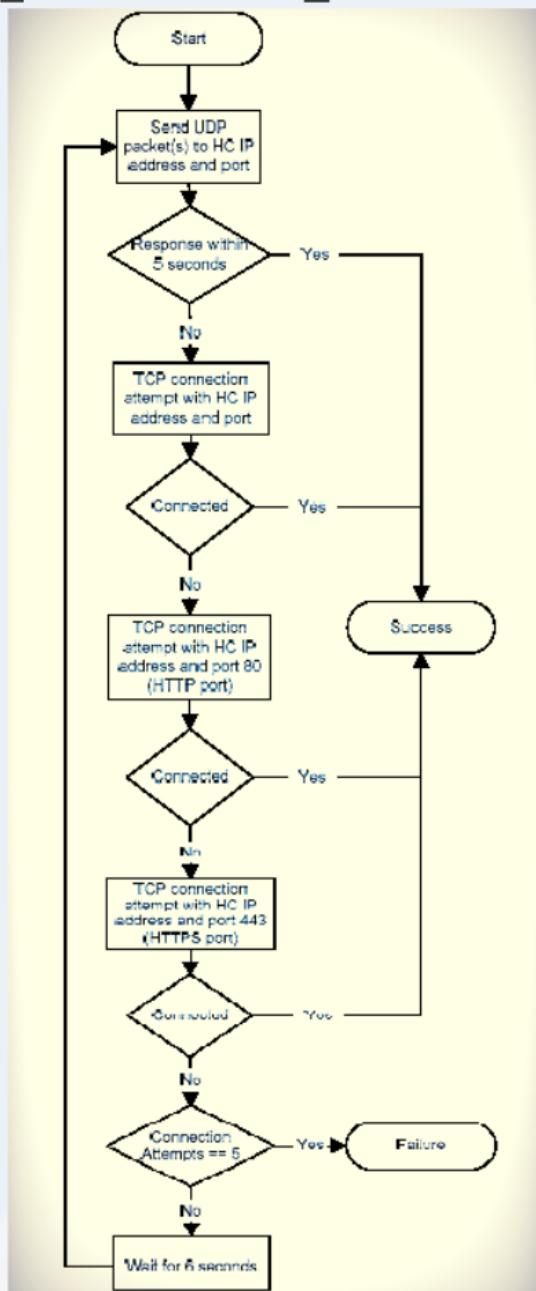


Figura 2: Algoritmo de login do Skype.

Skype: Componentes-chave.

- Portas
- HC
- Codecs
- Buddy List
- Criptografia
- NAT e Firewall

Skype: Portas.

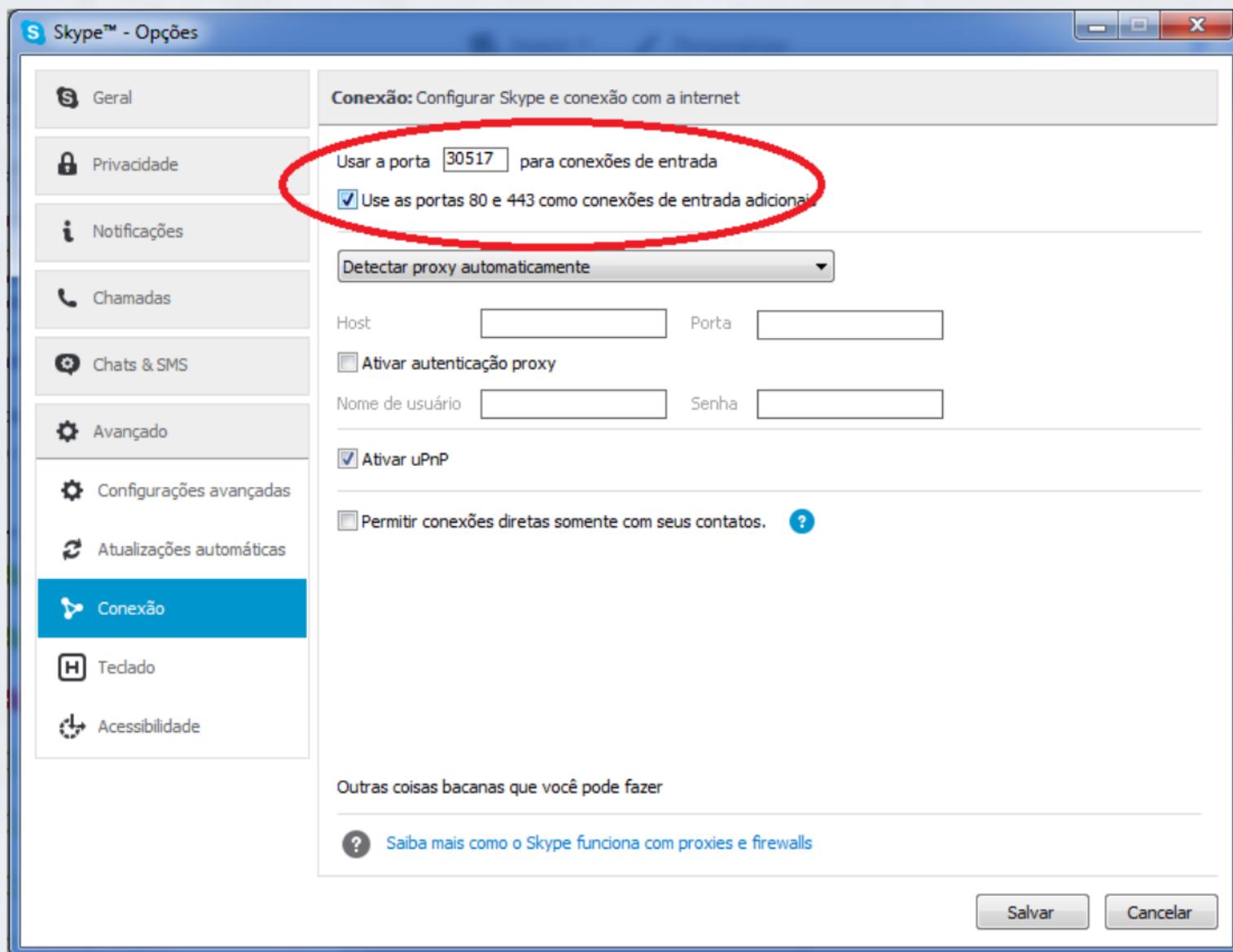


Figura 3: portas utilizadas pelo skype. (sockets)

Skype: Host-cache

- Quando o skype é iniciado, a tabela HC passa a ser incrementada com informações (endereço IP e porta) de SN's.
- Cada SC mantém uma tabela HC com SN's que estão ao seu alcance, relacionando os endereços IP e porta dos SN's.
- Tabela atualizada periodicamente pelo SC.

Skype: Host-Cache

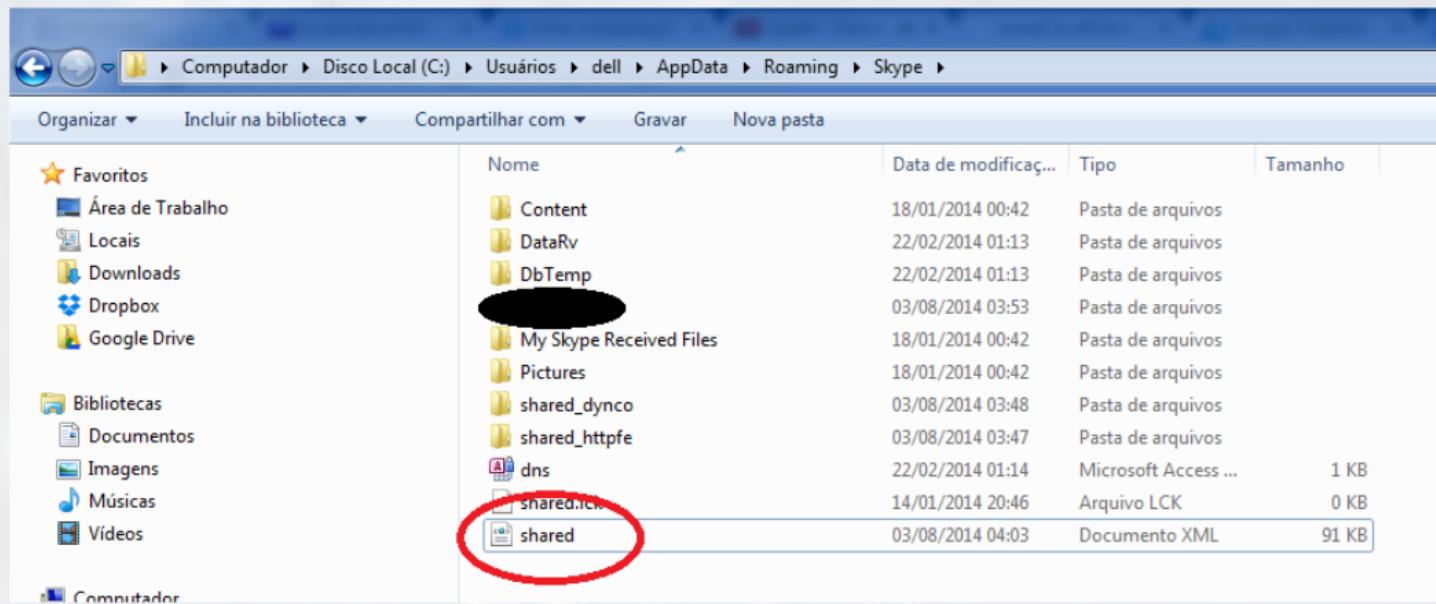


Figura 4: Localização do arquivo que contém informações de HC. (shared.xml).



Skype: Codecs

- Utiliza codecs de banda-larga.
- G.729: quadros de 10 ms, bit rate de 8kbits/s, amostragem fixa de 8kHz, muito utilizado em aplicações voIP.
- Silk: forma de compressão de áudio e codec de áudio desenvolvido pela Skype Limited (codificação de voz).
- ILBC: quadros de 20 ou 30 ms, amostragem fixa em 8kHz . (voIP)
- iSAC: quadros adaptativos de 30 a 60 ms, bit rate 10-32 kbps e amostragem fixa em 16 kHz. (voIP e aplicações de streaming de audio)
- TrueMotionVP: para chamadas de vídeo.

Skype: Buddy List.

- Skype armazena suas informações de contatos no registro do Windows.
- Criptografado.
- Local para uma máquina e não é armazenado em um servidor central.

Skype: Criptografia.

- Skype utiliza a criptografia AES.(Padrão utilizado pelo governo dos EUA).
- 256 bits com um total de 1.1×10^{77} possíveis chaves, a fim de criptografar os dados ativamente em cada chamada Skype ou mensagem instantânea.
- Skype usa RSA de 1536-2048 bits para negociar chaves AES simétricas.

Skype: NAT e firewall.

- Determina se está diante de um NAT ou firewall.
- Usa uma variação dos protocolos STUN e TURN.
- O SC atualiza esta informação periodicamente. Esta informação também é armazenado nos Registros do Windows.

Funções do Skype

- Login.
- Busca de usuários.
- Estabelecimento de chamadas e Teardown.
- Transferência de mídia e codecs.
- Mensagens Keep-alive.

Skype: Inicialização e login.

- Quando SC é executado pela 1º vez, é enviado um HTTP GET request para o servidor do skype.
- Inicializações subsequentes: HTTP request "getlastversion".
- SC Anuncia sua presença para outros peers e para sua buddy list.

Skype: Login.

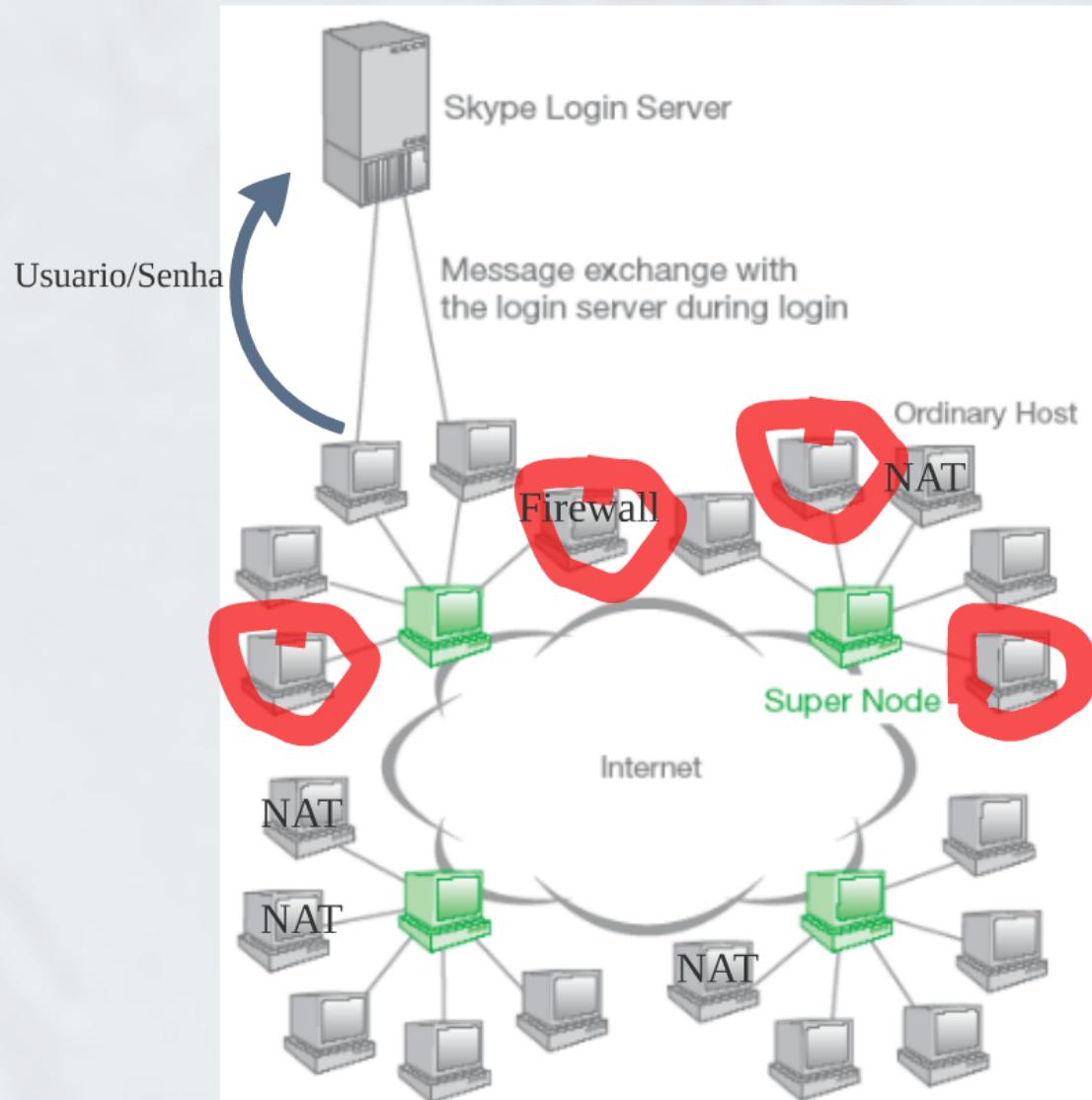
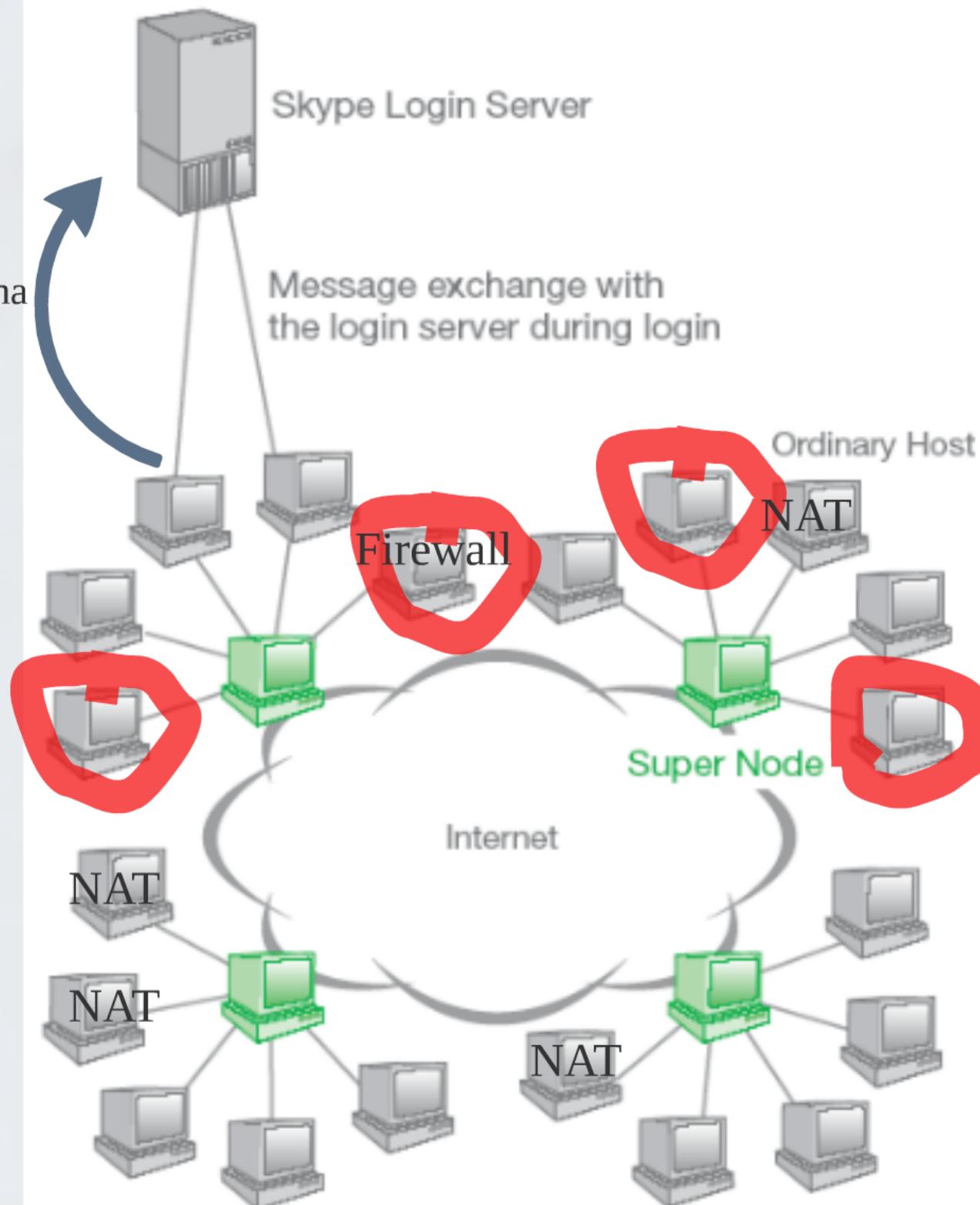
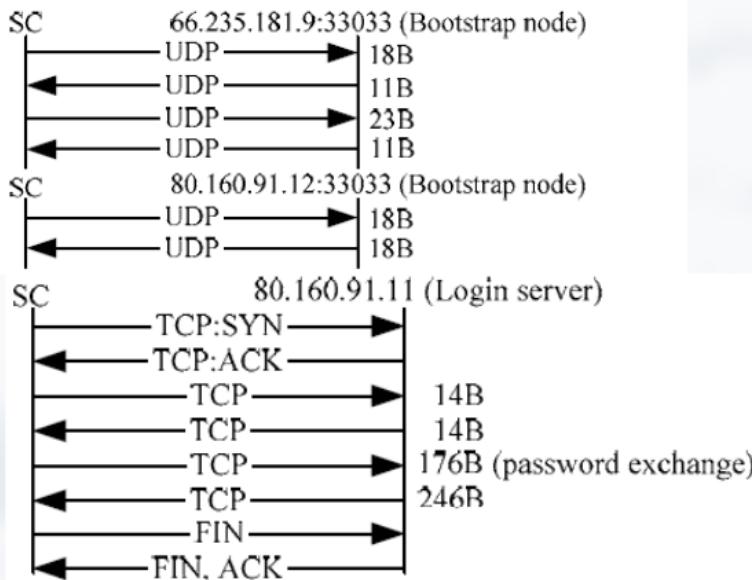


Figura 5: esquema de login do Skype.

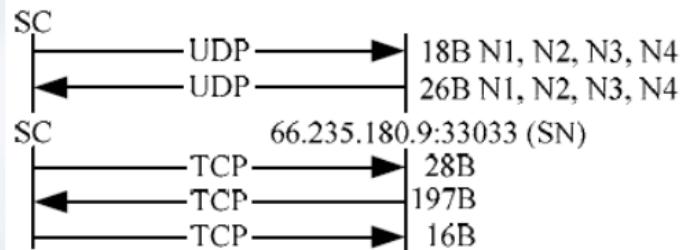
Usuario/Senha



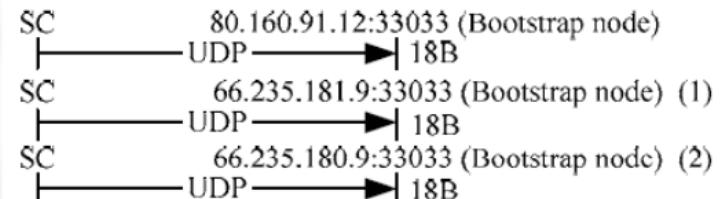
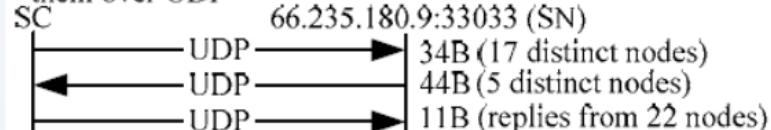
Skype: Login.



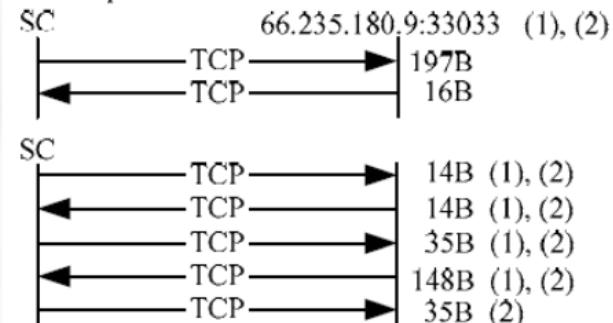
SC sends UDP packets to 4 distinct nodes and receives response over UDP. We believe that these nodes also run Skype.



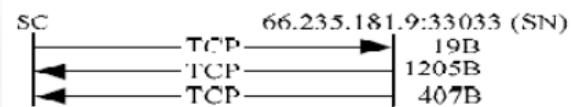
SC sends UDP packets to 22 distinct nodes and receives response from them over UDP.



Bootstrap nodes 66.235.180.9 and 66.235.181.9 are represented by labels (1) and (2) respectively in subsequent flows.



SC decides that it will retain TCP connection with 66.235.181.9. This node becomes a SN.



SC sends UDP packets to 4 distinct nodes. Since it is behind UDP restricted firewall, it cannot receive any responses over UDP.

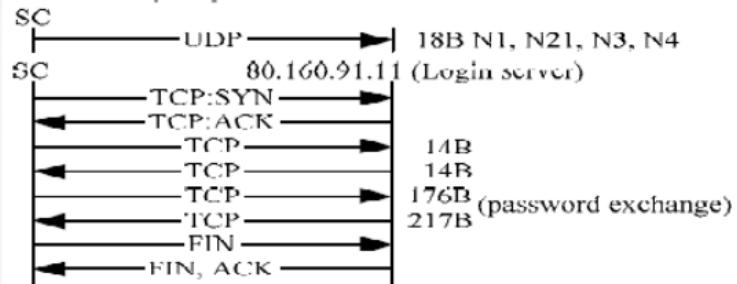
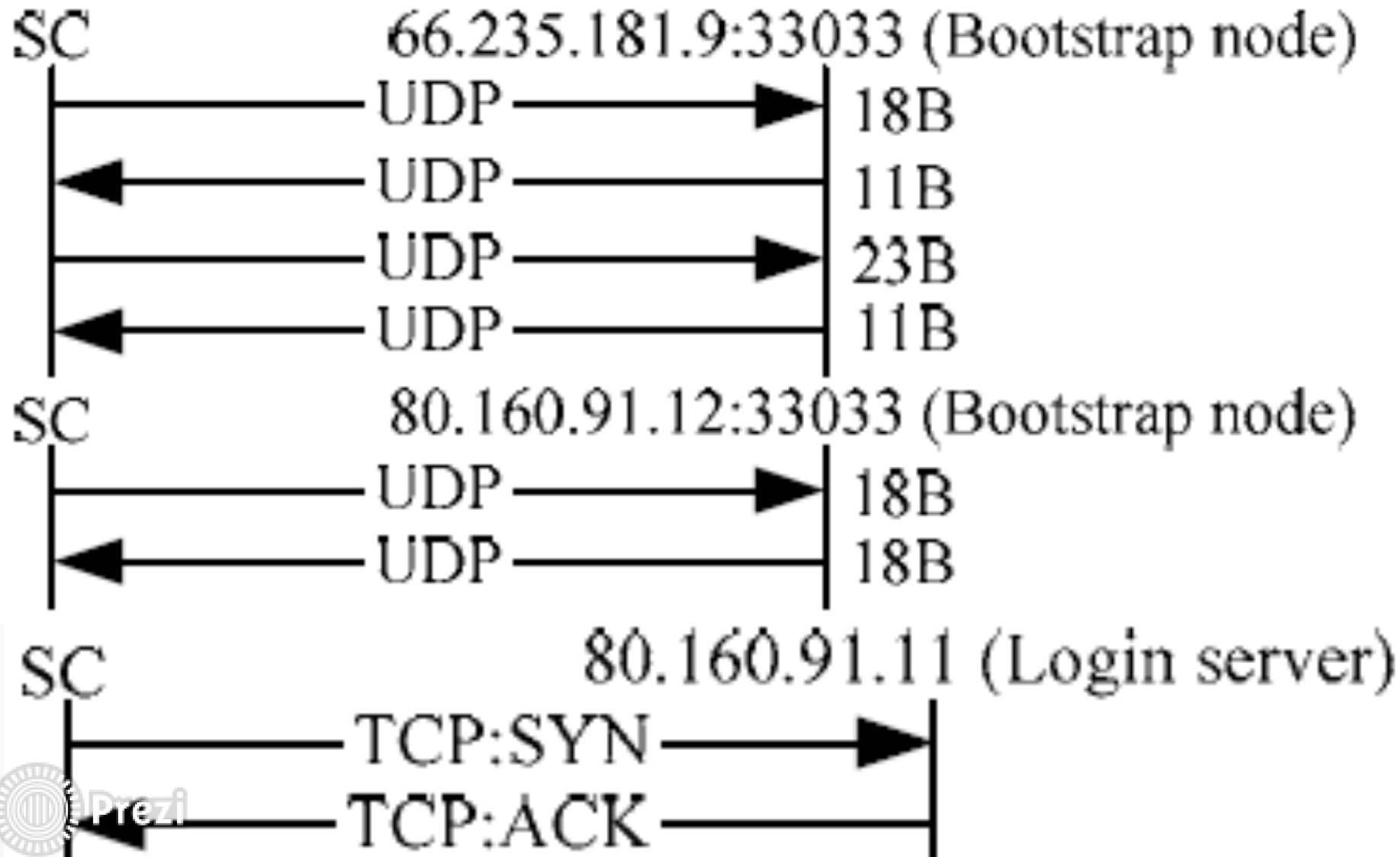
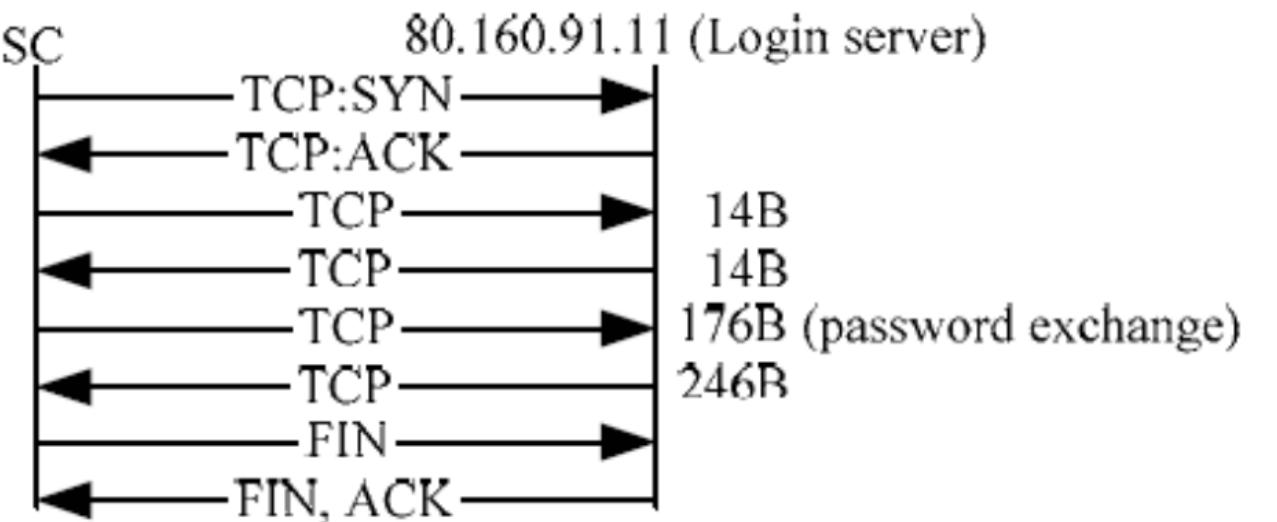


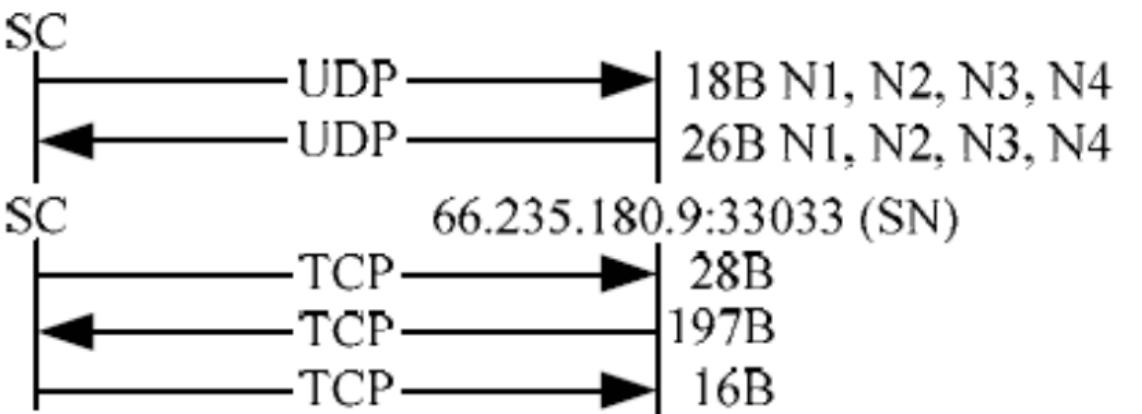
Figura 6: Troca de mensagens entre SN's, SC's e login servers.

Skype

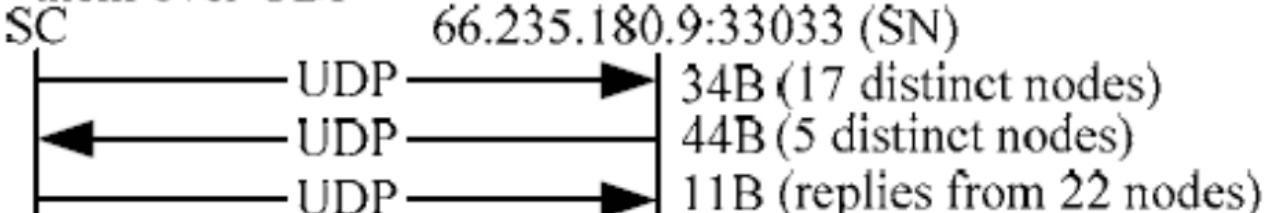


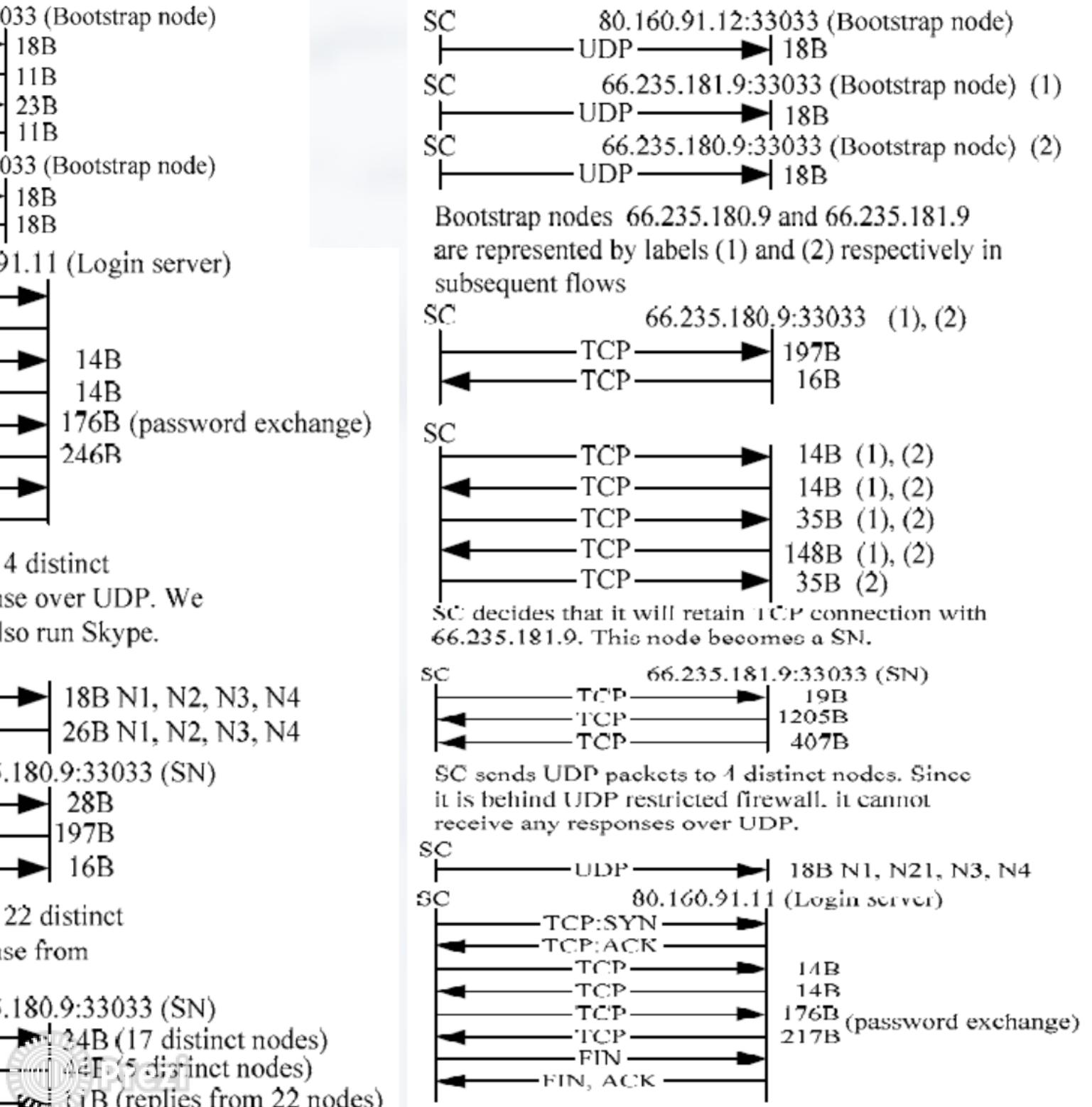


SC sends UDP packets to 4 distinct nodes and receives response over UDP. We believe that these nodes also run Skype.



SC sends UDP packets to 22 distinct nodes and receives response from them over UDP





Skype: login.

- O SC com um endereço IP público e um SC atrás de um NAT levou cerca de 3-7 segundos para ser concluído os procedimentos de login.
- O SC atrás de um firewall UDP-restrito levou cerca de 34 segundos para completar o processo de login. Enviou pacotes UDP para seus trinta nós na HC.
- Em seguida, ele tentou estabelecer uma conexão TCP com as entradas da HC e acabou por ser capaz de se conectar a um SN.

Skype: Busca de usuários.

- Tecnologia Global Index para buscar um usuário (distribuído).
- Skype não é um protocolo aberto e suas mensagens são criptografadas.
- Garante encontrar um usuário que logou nas últimas 72 horas. (se existir)



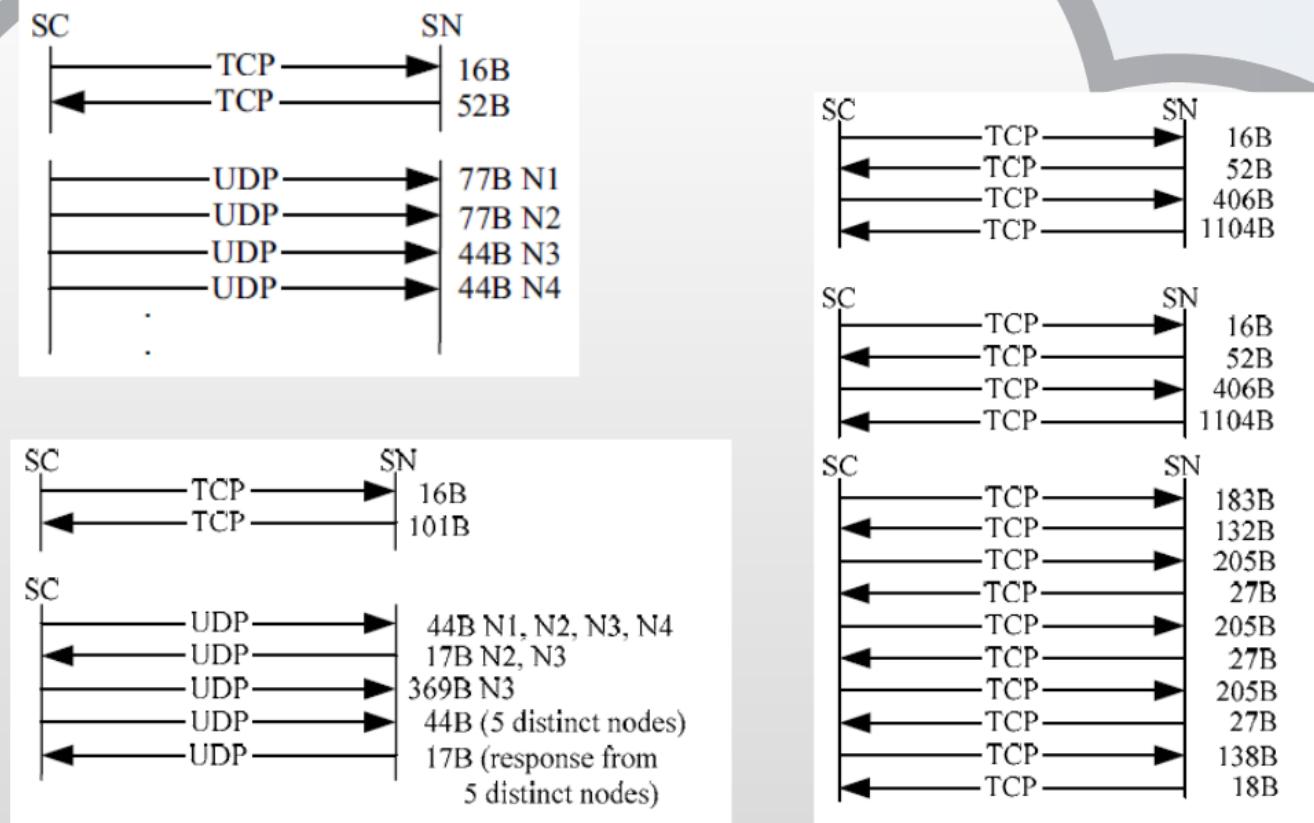


Figura 7: Troca de mensagens durante a busca de um usuário.

Caching nos resultados de busca ?

Skype: Estabelecimento de chamadas e Teardown

- Estabelecimento de chamadas em usuários que estão e não estão na buddy list do caller.
- Diferença: tempo.
- A sinalização da chamada é sempre realizada sobre TCP.



Skype: Estabelecimento de chamadas e Teardown.

Caller press dial

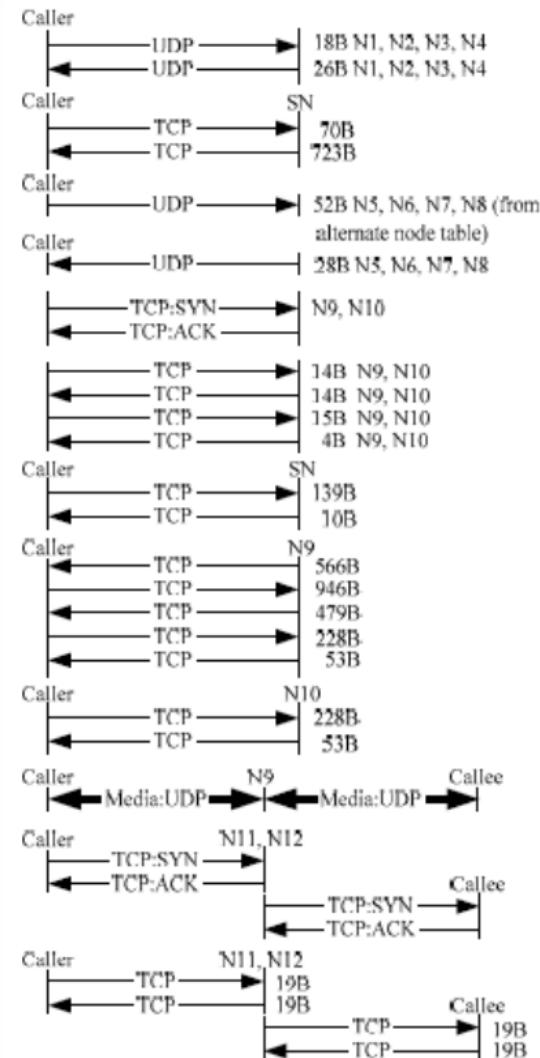
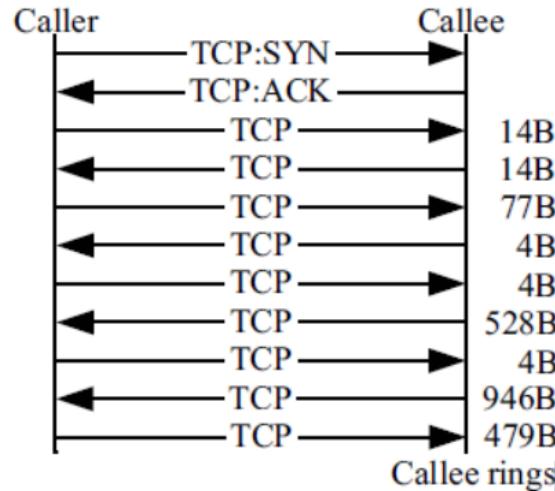
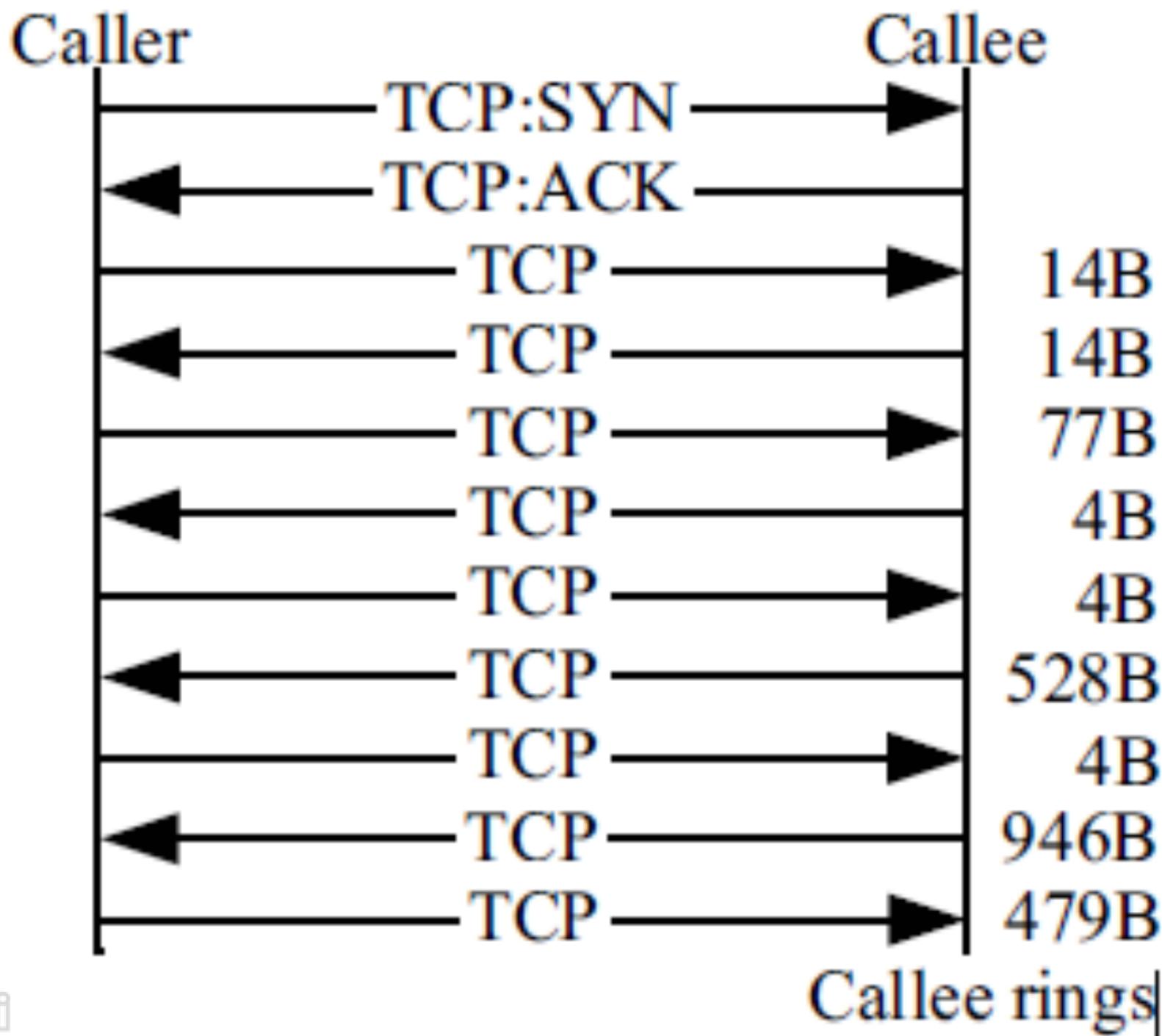
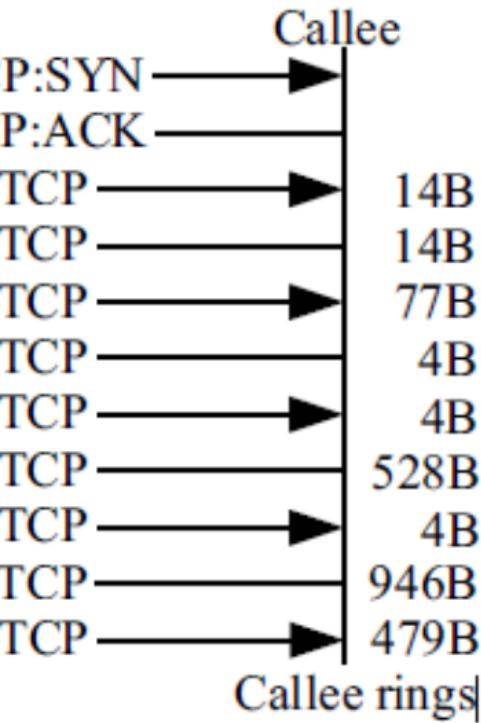


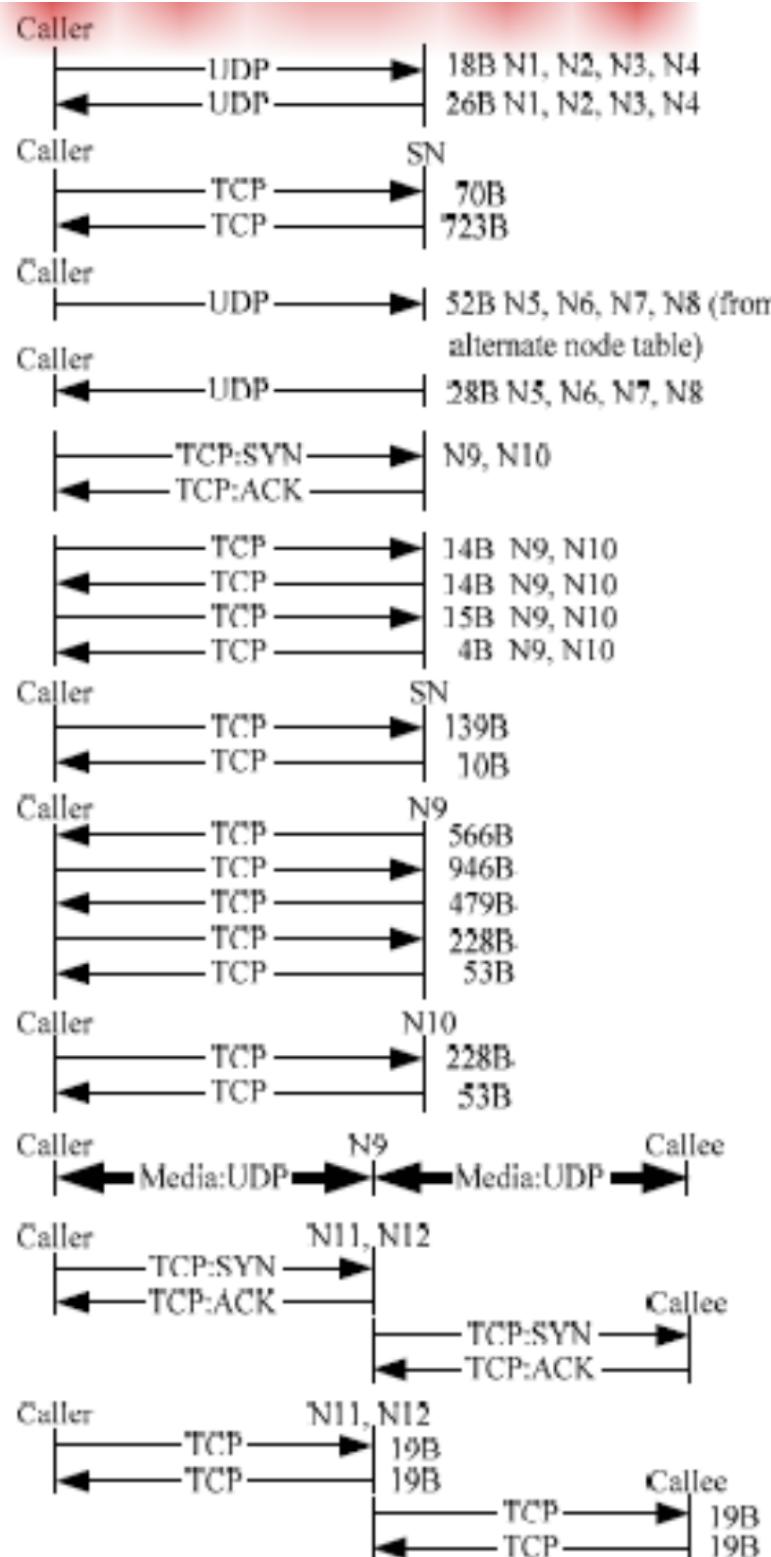
Figura 8: Estabelecendo chamadas em IP público e sobre firewall UDP restrito.

Caller press dial





Callee rings



Skype: Estabelecimento de chamadas e Teardown.

- Durante o encerramento da conexão(teardown), é sinalizada uma mensagem TCP entre caller e calle.

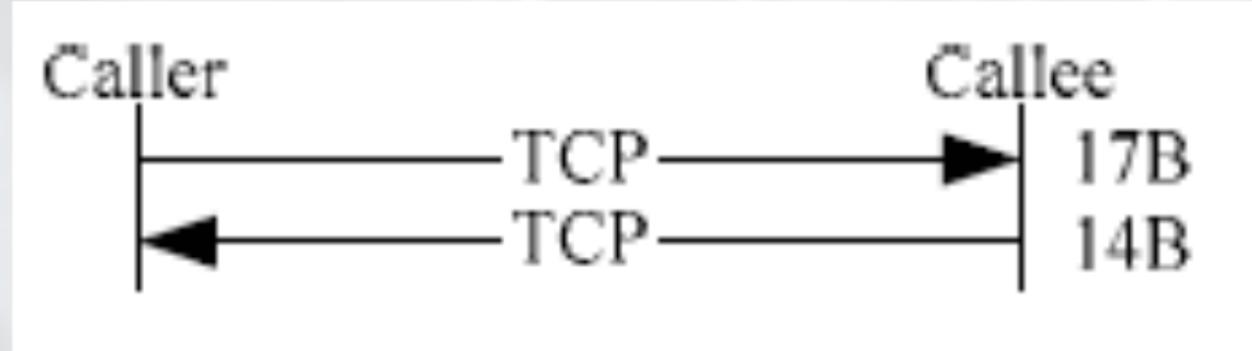


Figura 9: Teardown no skype.

Transferência de Mídia

- Tráfego de mídia entre SC's sem NAT e firewalls é sobre UDP (preferencialmente).
- SC's sobre NAT ou firewall UDP restrito, utilizam um nó alternativo para transmissão de mídia. (TCP)
- SC's sobre NAT utilizam nó alternativo que funciona como proxy de mídia entre eles. (UDP)

Transferência de mídia

- Supressão de silêncio.
- Chamada em espera.
- Faixa de frequência do codec.
- Congestionamento.

Mensagens Keep-Alive

- É uma mensagem enviada por um dispositivo para outro para verificar se a ligação entre os dois está funcionando, ou para evitar esta ligação seja quebrada.
- SC envia mensagens para o seu SN a cada 60 s sobre TCP.

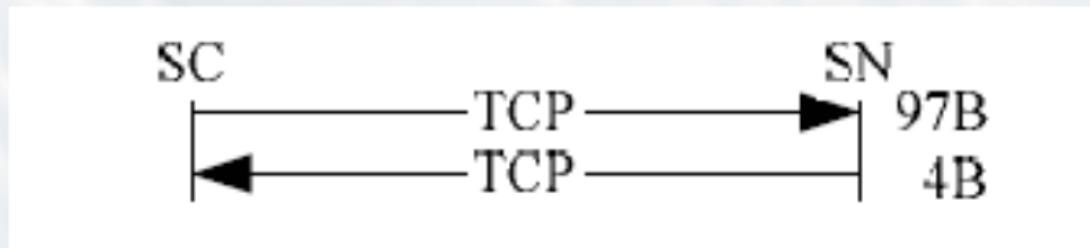


Figura 10: Keep-alive message skype.



WhatsApp

XMPP

Índice

1. Sobre.
2. Segurança X Portabilidade.
3. Projeto OpenWhatsapp (Open_Source).
4. Protocolo Base XMPP.
5. Sistema P2P baseado no protocolo XMPP.

Plataformas Base



O WhatsApp também é compatível com versões antigas do Nokia:

Asha 201, Asha 305, Asha 310 e Asha 311.

Symbian (Sistema Operacional da Nokia Lançado em 1997, possuiu três versões: Symbian¹, Symbian² e Symbian³). O último smartphone com Symbian foi o Nokia 808 PureView.

O Aplicativo

WhatsApp Messenger é um aplicativo de mensagens multiplataforma que permite trocar mensagens pelo celular sem pagar por SMS

http://www.whatsapp.com/?l=pt_br

Jan Koum - Criador do WhatsApp

Compra do Facebook por R
\$ 45 Bilhões.

Em termos de segurança



Como o WhatsApp usa um código fechado e o seu protocolo é personalizado a partir do XMPP, a segurança da comunicação e conexão de aplicativos fica inacessível, então não temos idéia de como eles utilizam as permissões que você deu a eles.

Outras alternativas de códigos abertos para usar que se comparam com o protocolo WhatsApp.

Aplicativos XMPP Open Source

- TextSecure
- SureSpot
- Kontalk
- ChatSecure



TextSecure

Existente a 4 anos, suporta chat-group e pode enviar anexos tanto quanto qualquer outro aplicativo de transferencia de arquivos. (Versão somente disponivel para Android).

Brevemente para MAC e Desktop.

Observação: O aplicativo criptografa tudo, desde os arquivos enviados até as mensagens trocadas.

SureSpot

SureSpot possui muitas consultas de pesquisa desde que o WhatsApp foi comprado pelo Facebook, o aplicativo não é tão leve quanto aparenta, para adicionar um contato deve-se mandar um convite, parecido com o procedimento do MSN, SKYPE, diferente do TextSecura que importa todos contatos salvos em sua memória pessoal para o aplicativo.

Isso traz segurança devido ao problema de que o código não pode filtrar todos seus dados indesejadamente, mas por outro lado fica mais difícil procurar todos contatos, pois você deve fazê-lo a mão.

Fora isso, SureSpot também tem um armazenamento local criptografado e suporta mensagens de áudio.

Kontalk

Kontalk é um muito bom e pequeno. É open source e de fácil utilização. Agora eles trabalham com um protocolo proprietário, mas logo eles vão mudar para o XMPP, que vai ajudar a conectar este aplicativo para outros programas, de modo que será possível conversar a partir do telefone para um ambiente de desktop. Eles também estão tentando implementar uma versão iOS. Os servidores usados são descentralizadas e todos podem criar a sua / seu próprio servidor. Eles são administrados através de uma rede de sistema de confiança. Todas as mensagens são criptografadas e anexos são transmitidos muito bem (como é WhatsApp).



Chat Secure

Utiliza Open Source, o que deixa a desejar é sua interface e o chat em grupo que o mesmo não possui, é um aplicativo muito bom em questões de segurança, também tem a característica de criptografar tudo.



Tabela de Comparação

APP NAME	SIZE OF USER- BASE	MOBILE PLAT- TFORMS	GROUP CHAT	AT- TACH- MENTS	USER FRIENDLY	TOP PROGRAM	EN- CRYPTED	OPEN SOURCE
What-sApp	~450 million							
TextSecure	0.1 - 0.5 million							
SureSpot	> 50.000							
Kontalk	10.000 - 50.000							
ChatSecure	0.1 - 0.5 million							

Figura x: comparação entre os app's.

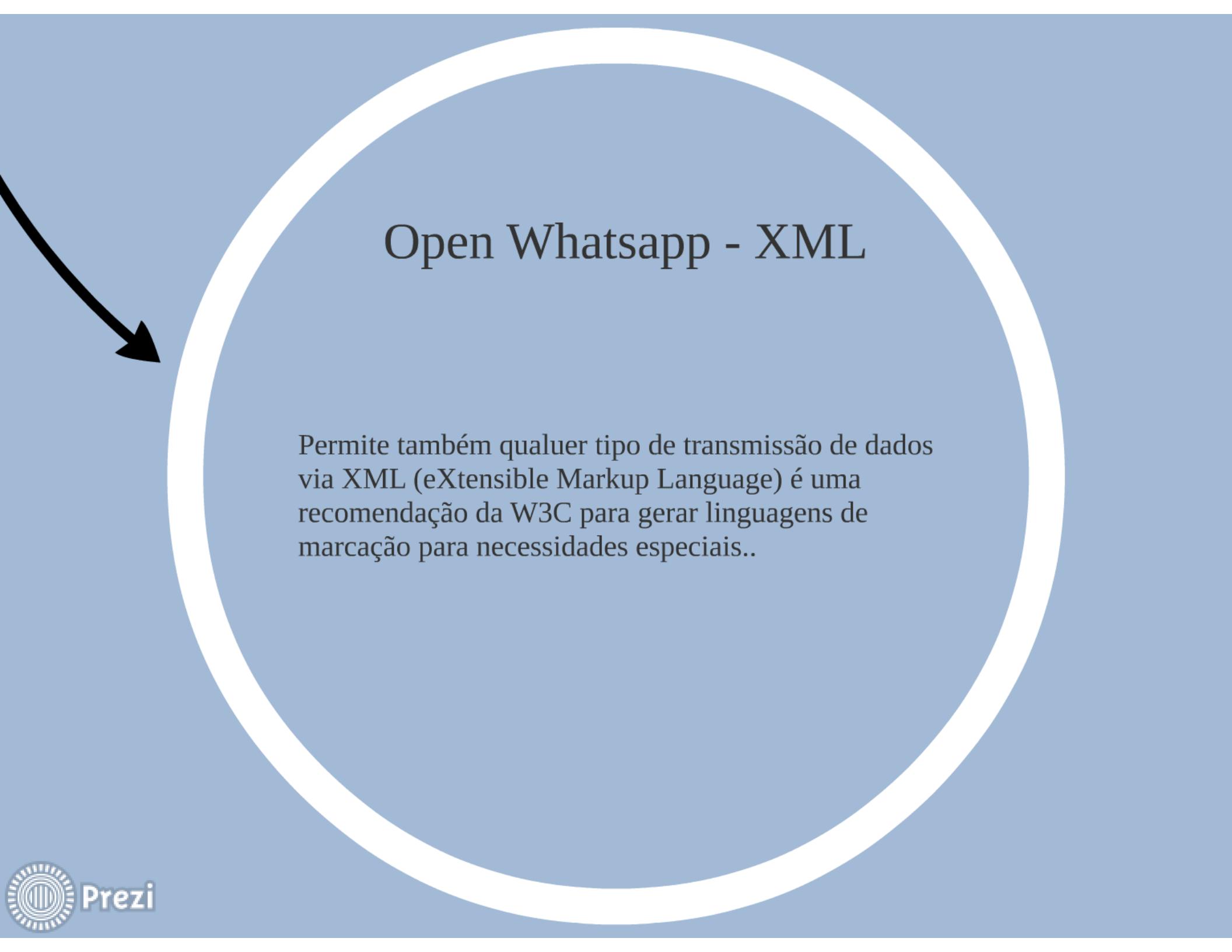
APP NAME	SIZE OF USER-BASE	MOBILE PLAT-FORMS	GROUP CHAT	AT-TACH-MENTS	USER FRIENDLY	TOP PROGRAM	EN-CRYPTED	OPEN SOURCE
What-sApp	~450 million		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TextSecure	0.1 - 0.5 million		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SureSpot	> 50.000		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kontalk	10.000 - 50.000		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ChatSecure	0.1 - 0.5 million		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

OpenWhatsApp

O WhatsApp não usa padrões de protocolos abertos.

Protocolo padrão do WhatsApp é o Extensible Messaging and Presence Protocol (XMPP).

Este protocolo dispõe de uma tecnologia aberta, extensível, baseada em XML que realiza comunicação em tempo real e que permite a criação de vários tipos de aplicações, tais como as de mensagens instantâneas, presença, chats multiusuários, chamadas de voz e vídeo, colaboração, middleware, content syndication.



Open Whatsapp - XML

Permite também qualquer tipo de transmissão de dados via XML (eXtensible Markup Language) é uma recomendação da W3C para gerar linguagens de marcação para necessidades especiais..

XMPP: Basic Extensible Messaging and Presence Protocol

O protocolo XMPP foi desenvolvido para trabalhar em um ambiente com transmissão garantida, não tendo a necessidade de confirmar o recebimento das mensagens, ele foi pensado para que a maior parte do processamento e armazenamento de informações fique a cargo do servidor onde em toda a troca de mensagens os servidores responsáveis pelos usuários envolvidos devem aparecer nas trocas de mensagens.

O Protocolo XMPP é uma alternativa aos protocolos proprietários vigentes, possuindo uma certa segurança por ser um software livre devido sua participação colaborativa da comunidade de desenvolvedores e uma instalação e configuração simples, objetiva, levada a prova todos os dias por meio de diversos aplicativos de mensagens instantâneas.



Uma abordagem alternativa é o uso de STREAM depois que por ali passou um desses elementos, isto é, no contexto da STREAM, e a conexão permanece persistente).

Resumo de abordagem

A Extensible Messaging and Presence Protocol (XMPP) é um perfil de aplicação da Extensible Markup Language (XML) que permite a troca quase em tempo real de dados estruturados ainda extensível entre duas ou mais entidades de rede. Este documento define os principais métodos de protocolo de XMPP: instalação e desmontagem de fluxos XML, criptografia de canal, autenticação, tratamento de erros e de comunicação primitiva para troca de mensagens, a disponibilidade da rede ("presença"), e as interações de solicitação-resposta.

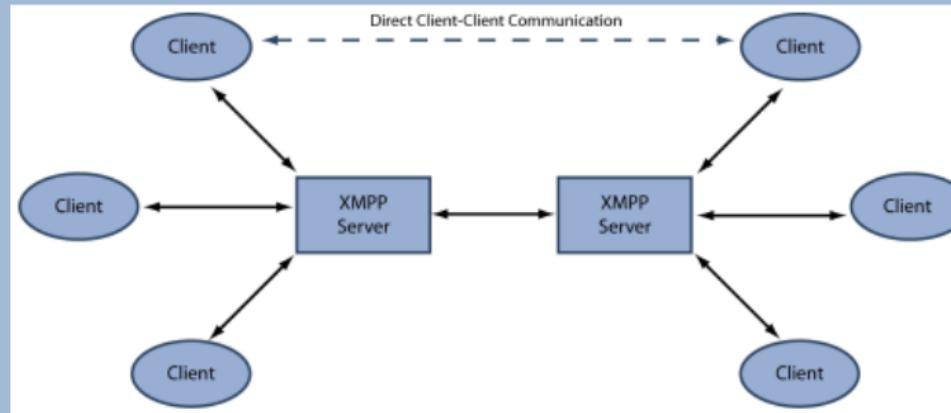


Figura x: Arquitetura XMPP.

rede ("presença"), e as interações de solicitação-resposta.

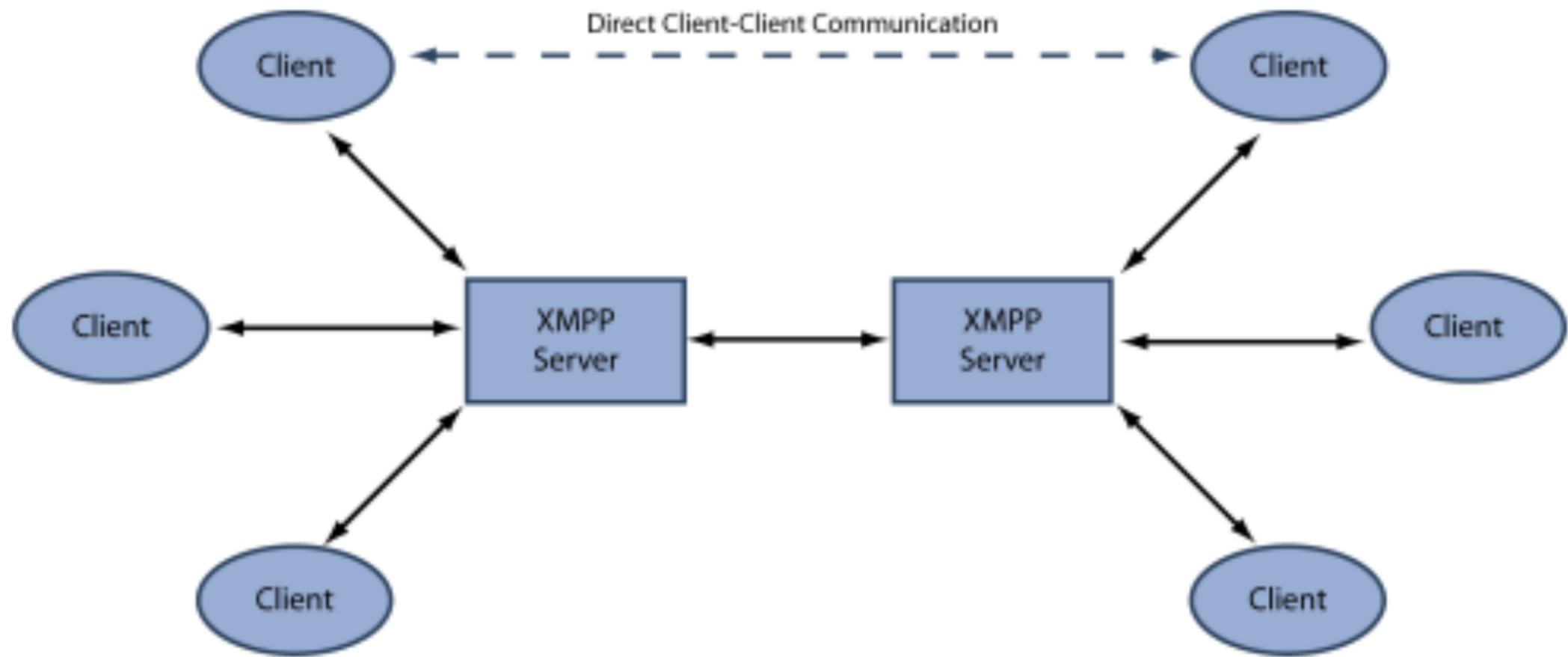


Figura x: Arquitetura XMPP.

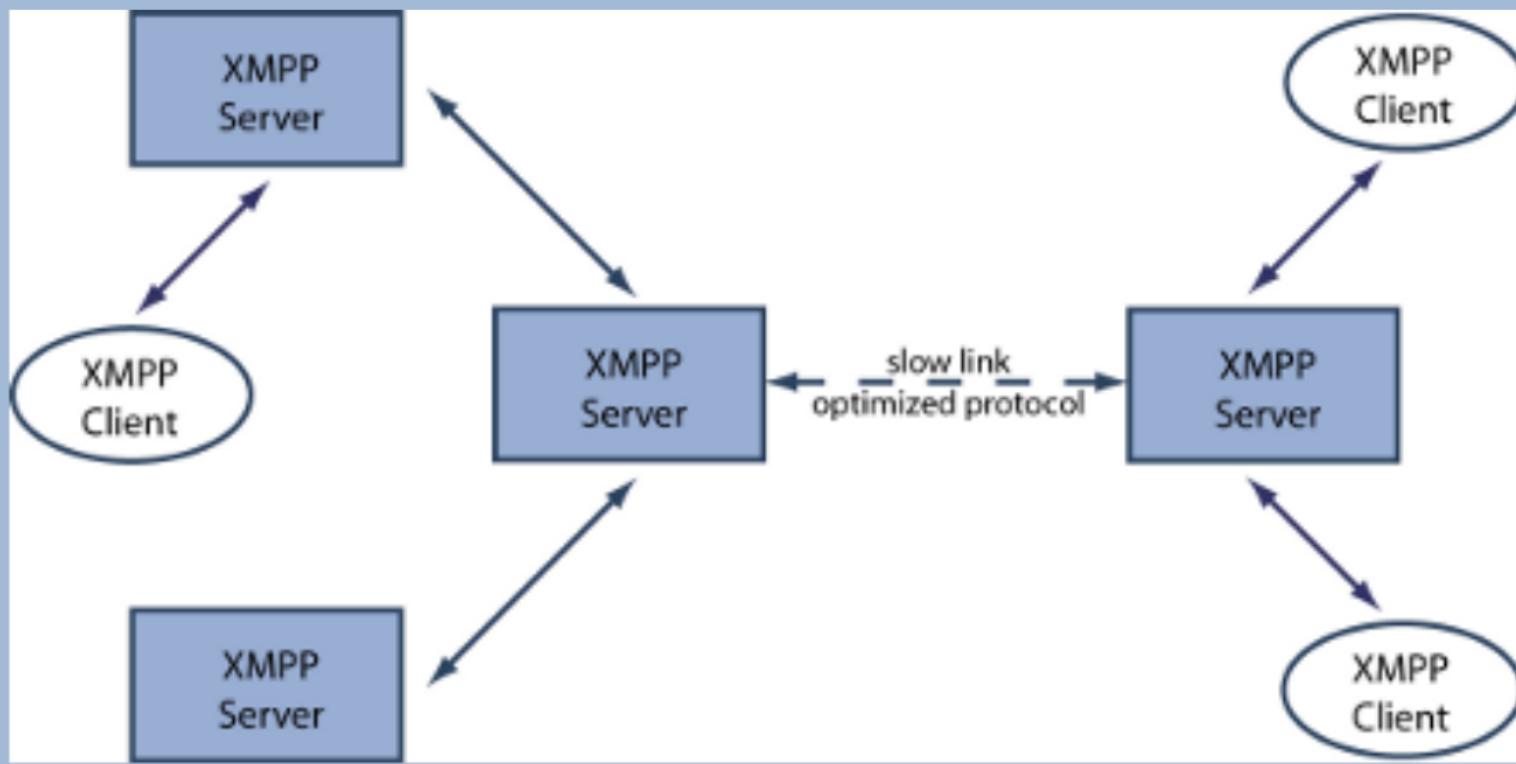


Figura x: outro modo de arquitetura XMPP.

Protocolo XMPP

O protocolo XMPP fornece uma tecnologia para a troca assíncrona, fim-a-fim de dados estruturados por meio de fluxos XML persistentes diretamente entre a rede distribuída de endereços globais, os clientes e servidores.

A conexão XMPP pode ser considerada persistente, e utiliza conexões TCP de longa duração. E para dar os saltos de um cliente a um servidor, ou servidor para servidor, um sistema "always-on" é ajustado para que a transferência de pacotes possa se dar a qualquer momento.

Mas então, como se da a estrutura em questão de cliente e servidor, em outras palavras, como eles se comunicam?

Estrutura cliente-servidor-servidor-cliente XMPP

Por exemplo:

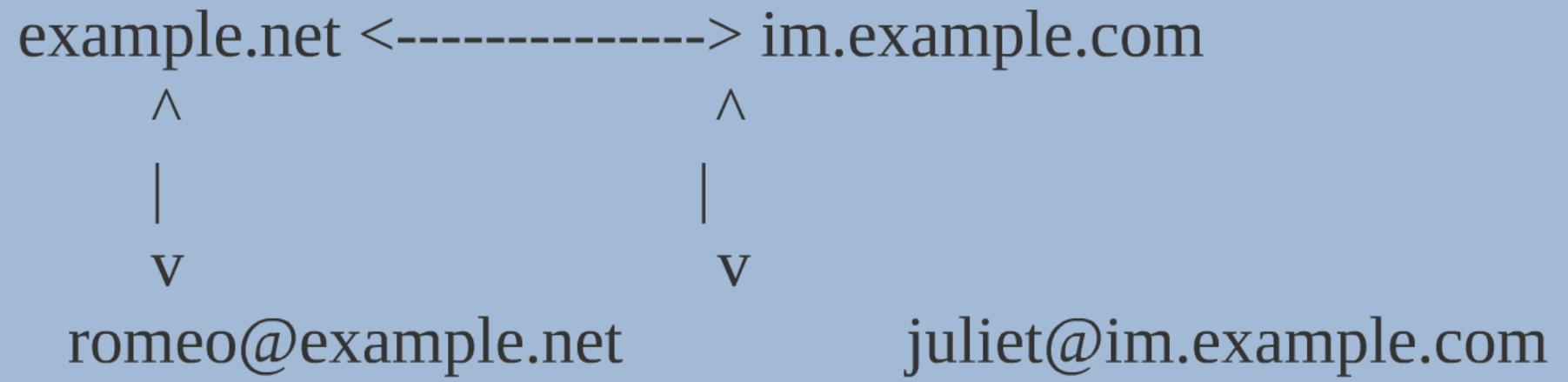
Utilizador 1 -> <juliet@im.example.com>
Servidor -> im.example.com

Utilizador 2 -> <romeo@example.net>
Servidor -> example.net

O Cliente acima é associado a um servidor <im.example.com>, tendo esse princípio ele pode ser capaz de trocar mensagens, status e outros tipos de dados estruturados com <romeo@example.net>.

Esse protocolo é padrão para protocolos que fazem uso de endereçamento mundial, uma conexão fim-a-fim em XMPP é lógicamente P2P, mas fisicamente o padrão se da cliente-servidor-servidor-cliente.







Autenticação e registro do cliente a um servidor.

Um cliente é uma entidade que estabelece um fluxo de XML com um servidor por autenticação com as credenciais de uma conta já registrada, e que, então, completa um vínculo a fim de permitir a entrega de estrofes (stanzas)XML entre o servidor e o cliente sobre a stream(passagem/córrego) negociada.

O cliente então usa XMPP para a comunicação com o seu servidor, outros clientes, e quaisquer outras entidades na rede, onde o servidor é responsável pela entrega de stanzas para outros clientes conectados no

mesmo servidor ou roteia-los para servidores remotos. Vários clientes podem se conectar simultaneamente a um servidor em nome do mesmo registro de conta, onde cada cliente é diferenciada pelo "resourcepart" de um endereço XMPP (por exemplo, <juliet@im.example.com/balcony> vs. <juliet@im.example.com/chamber>



Veja que basicamente, a XML é usada para tipos de dados e o XMPP para protocolo de comunicação; o servidor registra o cliente com um "resourcepart" e



Veja que basicamente, a XML é usada para tipos de dados e o XMPP para protocolo de comunicação; o servidor registra o cliente com um "resourcepart" e realiza uma comunicação entre dois clientes de mesmo servidor, além de estabelecer a comunicação, é ele quem faz o roteamento para outros servidores remotos.

Mais cedo, como se dá a comunicação entre cliente e servidor, em outras palavras, como eles se comunicam?

Características do Servidor que usa XMPP

Um servidor é uma entidade cujas responsabilidades principais são:

1º -> Gerir fluxos XML com os clientes conectados e entregar estrofes XML para os clientes sobre a Stream já alocada; o que inclui a responsabilidade de garantir que um cliente autentique com o servidor antes de ser concedido o acesso a rede XMPP.

2º -> Políticas locais de serviços na comunicação servidor-servidor, gerenciar fluxos XML com servidores remotos e stanzas via XML para os servidores sobre a stream negociada.

Dependendo da aplicação, as responsabilidades secundárias de um Servidor XMPP pode incluir:

1º -> Armazenamento de dados que são usados pelos clientes (por exemplo, listas de contato para usuários de aplicativos de mensagens instantâneas e presença baseados em XMPP); neste caso, a stanza XML relevante é tratada diretamente pelo próprio servidor em nome do cliente.

2º -> Serviços de alojamento de add-on que também usam XMPP como base para comunicação, mas que fornecem funcionalidades adicionais. Exemplos incluem serviços de conferência multi-usuário, e serviços de publicação-assinatura.



STREAM, como funciona a segurança.

Como um servidor ou qualquer tipo de entidade na rede abre uma STREAM de conexão, seu protocolo funciona como um gatekeeper para o domínio dos serviços de TI, que impõe certas condições para a conexão ser bem sucedida.

No mínimo a entidade que quer se conectar tem de ao menos se apresentar a entidade que esta abrindo a STREAM antes de enviar stanzas XML.

Para Cliente-Servidor utiliza-se SASL (Simple Authentication and Security Layer), um autenticador simples para garantir o minimo de segurança e confiabilidade.



Se for uma conexão que a entidade que esta abrindo a STREAM considerar não regular ou duvidosa (geralmente quando a chamada for de um roteamento de outro servidor), além do protocolo de autenticação SASL, é obrigatório o uso da criptografia TLS (Transport Layer Security/Segurança da Camada de Transporte), que se dá basicamente por criptografia com chaves de segurança, é baseado no protocolo TCP.

Flush da STREAM

Uma abertura de STREAM pode ser considerada uma vulnerabilidade caso dados fiquem "estagnados" durante a comunicação, logo, uma STREAM depois de encerrada, deve descartar o conhecimento de dados que por ali passou antes de ser encerrada e após o sucesso de transição desses elementos, isso é feito fazendo um flush(lavagem) no antigo contexto da STREAM, e a transação de novos cabeçalhos nos contextos da STREAM através da conexão TCP. (Não esquecendo que a conexão é persistente).

Servidor completo XMPP

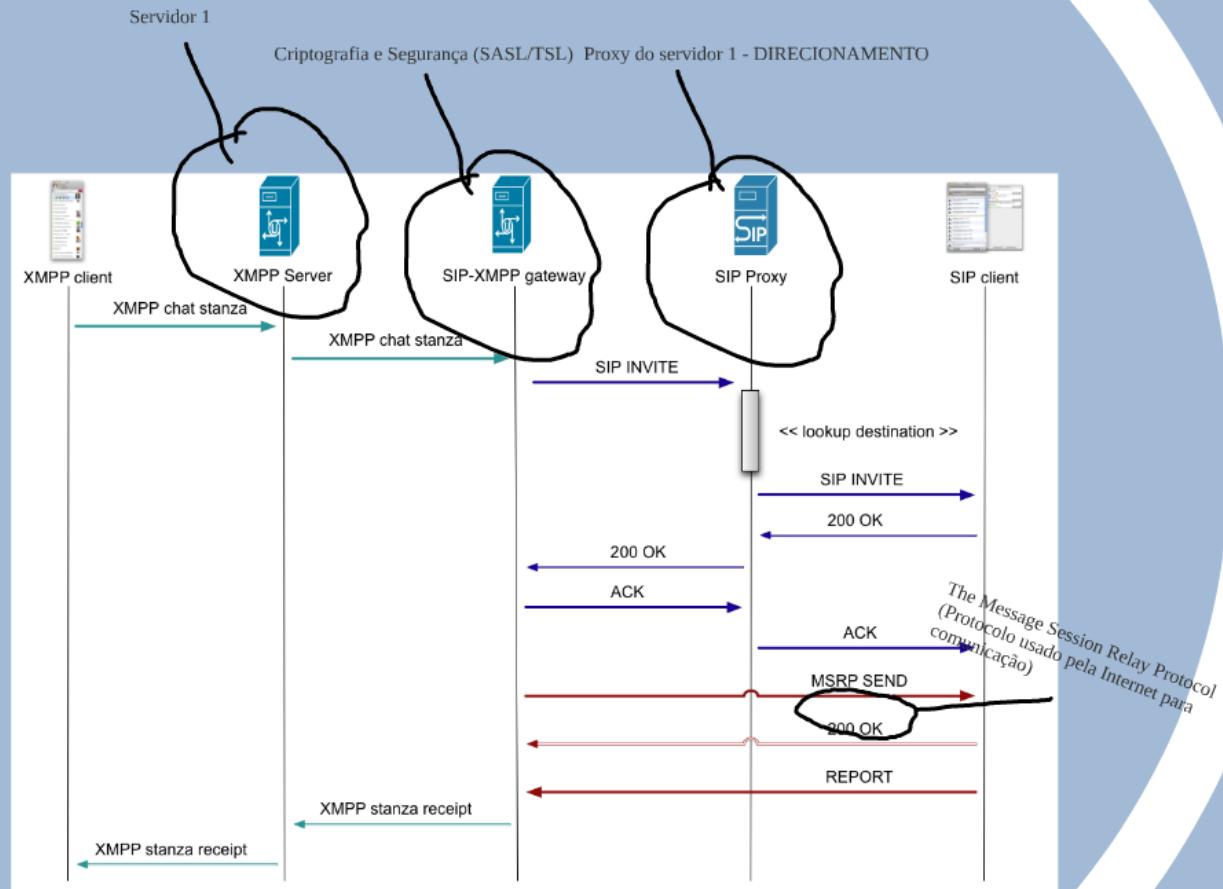
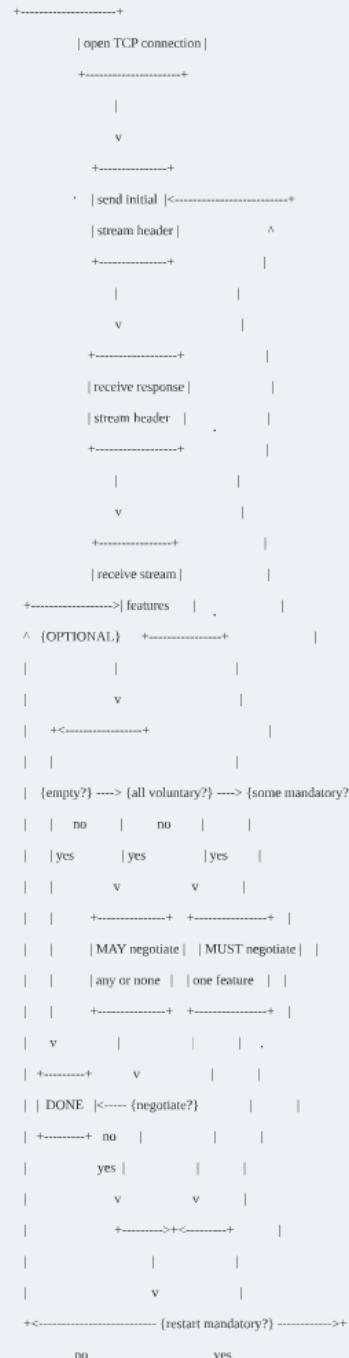


Figura x: funcionamento suposto do Whatsapp.

Negociação de uma STREAM



Conclusões

- Arquitetura Skype x Whatsapp (XMPP).
- Buscas de usuário:
- Skype a busca é efetuada na HC dos SN's.
- Whatsapp a busca é feita a partir do servidor.

Conclusões

Segurança:

- Skype é mais seguro que o whatsapp (XMPP), pois o Skype Login Server não interfere na comunicação entre clientes.
- O whatsapp possui um servidor que está sempre interagindo com a comunicação entre os clientes (flush).
- AES e RSA x SASL e TLS.

Conclusão

- Desempenho: Whatsapp não é necessária autenticação. Já no skype, um processo de autenticação no servidor do skype é necessário.
- Ponto único de falha dos protocolos.
- Excesso de usuários em um servidor do Whatsapp causa queda de desempenho. O fato do Skype ser descentralizado, excesso de usuários não congestionam a rede.

Conclusão

- Tráfego de mídia no Skype é feito geralmente por UDP e possui uma qualidade de áudio superior ao do whatsapp. (codecs)
- Whatsapp não possui chamadas de voz, apenas arquivos de áudio. Skype é possível chamadas de voz. (voIP)

Referências

<https://support.skype.com/pt/faq/FA148/quais-sao-as-portas-que-devem-estar-abertas-para-poder-usar-o-skype-no-windows>

http://www2.ic.uff.br/~eoliveira/Disciplinas_D.Sc/Redes_MM/Skype04_Resumo.pdf

<http://arxiv.org/ftp/cs/papers/0412/0412017.pdf>

<http://tools.ietf.org/html/rfc6120>

<http://www.whatsapp.comopensource/>

<http://www.wired.com/2014/05/briar/>

Obrigado!