Investigação de alternativas para transmissão de
áudio e vídeo na Internet: avaliação dos algoritmos
de seleção de componentes na arquitetura S-
MOJOHON

Matheus Qualio Barbosa

Orientadora: Prof^a. Dr^a. Maria da Graça Campos Pimentel

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP – São Carlos

Junho de 2008

Investigação de alternativas para
transmissão de áudio e vídeo na Internet:
avaliação dos algoritmos de seleção de
componentes na arquitetura S-MOJOHON

Matheus Qualio Barbosa

A todos aqueles que de alguma forma contribuíram para a realização do trabalho, principalmente minha família, namorada e amigos.

Agradecimentos

Primeiramente gostaria de agradecer a Deus por ter permitido a realização dessa conquista e de muitas outras ocorridas durante este período. Em segundo lugar gostaria de agradecer a minha orientadora Maria da Graça Campos Pimentel e seus colaboradores Daniel Correa Lobato e Cesar Teixeira, essa galera me ajudou muito mesmo. Em terceiro lugar gostaria de agradecer muito a minha namorada Michele Bellini Perossi e a toda minha família (Magali, Camila, Babão, Tios, Tias, Avós, Primos, Primas, Cunhado, Cunhada, Rosinha, Rosa, Vô Ari, Vó Célia, Tio João, Mario, todos os Camargos, Barbosas, Qualios, Érica, Mauricio, Ari, Marilda e outros que nem lembro mais o nome), amigos, amigas e colegas, sendo alguns deles: Fabio, Eduardo, Felipe SS, Nina, Edson, Lobato (de novo), Patrícia, Cássio, André, Oswaldo Junior, Karin, Douglas, Marina, Miltão, Iara, Fernanda, Renata, João, Renato, Silvia, Ana Elisa, Juliana, Renato Bulcão, Taciana, Helder, Éder, Namba, Mario Avelino, Bella, galera das aulas, Patrícia Magna, Souza, Graça Nunes, Solange, Zao, Roseli, Neide, Elizete, Carlos, a escritora M. Cecília, Valéria, Rodrigo Mello, Claudia, meninos e meninas do Intermídia, a galera do Bar do Toco, Trem Bão, Redondo, Vila, A Vila, Pimenta, Beto Vaca, Silvio, outros bares, funcionários da USP e as meninas da pós, CNPq, FAPESP, galera do Intermídia antigo, Patrão, galera do LSC-COC, Tim, Henrique, Cássio, Pires e outros. E também meus amigos do COC, Barth, Berê, Felipe, Danilão, Érica, Alan, Roberta, Tucunaré, Julio, Maycon, Danilim, Ricardo, Barba, Moquenco, Lemão e Andrei. Tem também o pessoal aqui do emprego e todos que passaram nas repúblicas MilkBoys, Maiti, Abatedouro, obrigado. Valeu galera, sem vocês não teria conseguido nada. Agradeço a outros colegas que também participaram deste período de pós e que não vou lembrar, pois são muitos nomes. É isso aí galera, muito obrigado.

Resumo

No contexto da Internet convencional, são várias as soluções oferecidas para transferência de fluxos de áudio e vídeo. Este trabalho é realizado no contexto de uma rede de grande largura de banda e de baixa latência, como aquela que está sendo investigada no contexto do Programa FAPESP TIDIA.

O grupo de pesquisa no qual este trabalho foi realizado propôs a arquitetura S-MOJOHON (Lobato et al., 2007a), elaborada com o intuito de permitir transmissão de vários tipos de dados, principalmente áudio e vídeo, no ambiente TIDIA.

Os componentes intermediários de uma arquitetura voltada para transmissão de fluxos de áudio e vídeo na Internet, como é o caso da S-MOJOHON, precisam ser selecionados de maneira que possibilitem a entrega de informações entre componentes finais da melhor forma possível. Nesse sentido, o trabalho reportado nesta dissertação envolveu a avaliação dos algoritmos de seleção utilizados nas diferentes implementações da arquitetura S-MOJOHON.

Analisando os resultados obtidos, criou-se um novo algoritmo de seleção de componentes que explora as vantagens dos algoritmos avaliados anteriormente. Este texto apresenta a arquitetura S-MOJOHON, a avaliação dos algoritmos de seleção, o novo algoritmo proposto e sua avaliação.

Palavras-chave: Arquitetura, Áudio, Vídeo, Transmissão de fluxos, Ontologia, Algoritmos de seleção.

Abstract

Several solutions are available in the context of the Internet with respect to supporting the transfer of audio and video. The work reported in this text considers a gigabit Internet network environment, such as the one supported in the context of the FAPESP TIDIA Project.

The research group in which this work has been developed has proposed the S-MOJOHON architecture (Lobato et al., 2007), aimed at supporting the transfer of audio and video in the TIDIA environment.

The distributed and complementary nature of the components of an architecture such as the S-MOJOHON demands that they must be selected so as to allow the best transfer rate possible. In such a context, the work reported in this dissertation involved the evaluation of the algorithms deployed in two different S-MOJOHON implementations. As a result, a new algorithm has been designed and evaluated, as detailed in this text.

Keywords: Architecture, Video, Audio, Ontology, Streaming, Selection algorithms.

Sumário

1. Introdução	1
1.1 O Projeto TIDIA-Ae.....	6
1.2 Identificação do problema.....	8
1.3 Objetivos	8
1.4 Metodologia e Resultados Obtidos	9
1.5 Estrutura da Dissertação.....	9
2. Fundamentação e Trabalhos relacionados.....	11
2.1 Considerações Iniciais.....	11
2.2 Algoritmos de Seleção de Componentes.....	11
2.3 Web Semântica	13
2.4 Avaliações de Fluxos de Áudio e Vídeo.....	14
2.5 Áudio e Vídeo na Internet	15
2.6 Protocolos para Áudio e Vídeo na Internet.....	17
2.7 <i>Network Address Translation</i>	20
2.8 Arquiteturas para Transmissão de Áudio e Vídeo	23
2.9 Considerações Finais.....	29
3. Extensão da Arquitetura S-MOJOHON.....	30
3.1 Considerações Iniciais.....	30
3.2 Arquitetura S-MOJOHON	32
3.3 Etapas para estabelecimento de uma transmissão ...	36
3.4 Implementação do sistema de comunicação	40
3.5 Implementação de Captura e Apresentação	41
3.6 Implementação da arquitetura S-MOJOHON	42
3.7 Resultados da implementação.....	45
3.8 Considerações Finais.....	49
4. Algoritmos de seleção - avaliações	50
4.1 Considerações Iniciais.....	50
4.2 Algoritmo de Seleção Paralelo.....	50
4.3 Algoritmo de seleção ontológico	53
4.4 Algoritmo Híbrido.....	56
4.5 Discussão	59
4.6 Considerações Finais.....	61
5. Conclusões	62
Referências	67

Lista de Figuras

FIGURA 1: EXEMPLO DE UM ARQUIVO RDF QUE DESCREVE RECURSOS E RELAÇÕES DO S-MOJOHON (LOBATO ET AL., 2007A).....	14
FIGURA 2: CONFIGURAÇÃO DE REDE USANDO NAT (LOBATO ET AL., 2007).....	21
FIGURA 3: NARADA E SERVIÇO WEB DE ÁUDIO E VÍDEO. (BULUT ET AL., 2002).....	24
FIGURA 4: CAMADAS DA ARQUITETURA IDL. (KAMOLPHIWONG, 2002).....	25
FIGURA 5: SISTEMA <i>REAL TIME E-LEARNING</i> . (GUERRI ET AL., 2002).....	26
FIGURA 6: ARQUITETURA STARCAS (TSUCHIYA ET AL., 2006).....	28
FIGURA 7: EXEMPLO DE REPRESENTAÇÃO EM FORMA DE GRAFO DOS ASS E OS DIFERENTES ALGORITMOS DE ROTEAMENTO (LOBATO ET AL., 2007A).	31
FIGURA 8: REPRESENTAÇÃO GRÁFICA DA ONTOLOGIA QUE DESCREVE OS ASS (LOBATO ET AL., 2007A).....	32
FIGURA 9: MÓDULOS DA ARQUITETURA S-MOJOHON (LOBATO ET AL., 2007).	33
FIGURA 10: ALGORITMO DE CONFIGURAÇÃO E TRANSMISSÃO. FIGURA ADAPTADA DE LOBATO ET AL. (2007).	37
FIGURA 11: MÓDULOS DA ARQUITETURA S-MOJOHON (LOBATO ET AL., 2007A).	40
FIGURA 12: JMF API. (JMF PICTURE)	42
FIGURA 13: FOTO DO EXPERIMENTO REMOTO AUTOMATIZADO – GAIOLA DE SKINNER ..	46
FIGURA 14: AMBIENTE TIDIA-AE UTILIZANDO COMUNICADOR INSTANTÂNEO COM A ARQUITETURA S-MOJOHON	49
FIGURA 15: CLIENTES E REFLETORES – ALGORITMO PARALELO.....	52
FIGURA 16: ETAPAS DO PROCESSAMENTO DOS DADOS PARA OBTENÇÃO DO REFLETOR DESEJADO A PARTIR DE UMA BASE DE CONHECIMENTO.	54
FIGURA 17: CLIENTES E REFLETORES – ALGORITMO ONTOLÓGICO.....	55
FIGURA 18: CLIENTES E REFLETORES – ALGORITMO HÍBRIDO.	58

Lista de Tabelas

TABELA 1: REQUISITOS E COMPARATIVOS ENTRE AS ARQUITETURAS	28
TABELA 2: RESULTADO DA AVALIAÇÃO DA IMPLEMENTAÇÃO PARCIAL DO CONJUNTO DE COMPONENTES (LOBATO ET AL., 2007).	48
TABELA 3: MEDIDA DE TEMPO DE CONFIGURAÇÃO ENTRE O MÓDULO DE TRANSMISSÃO E MÓDULO DE MONITORAMENTO/CONFIGURAÇÃO EM SEGUNDOS.	51
TABELA 4: MEDIDA DE TEMPO DE CONFIGURAÇÃO ENTRE O MÓDULO DE TRANSMISSÃO E MÓDULO DE MONITORAMENTO/CONFIGURAÇÃO EM SEGUNDOS.	51
TABELA 5: AVALIAÇÃO DA UTILIZAÇÃO DE ARQUIVOS RDF COM TODOS OS SISTEMAS AUTÔNOMOS E REGRAS UTILIZANDO JENA EM SEGUNDOS	53
TABELA 6: RESULTADO DA AVALIAÇÃO UTILIZANDO ALGORITMO HÍBRIDO DE SELEÇÃO DE COMPONENTES. TEMPO DADO EM SEGUNDOS.	57

Lista de Siglas

A/V	Áudio e vídeo
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicativos)
CI	Comunicador Instantâneo – Tidia-Ae
HTTP	<i>Hypertext Transfer Protocol</i>
IP	<i>Internet Protocol</i> (Protocolo de Internet)
IPV6	<i>Internet Protocol</i> (Protocolo de Internet, Versão 6)
JMF API	<i>Java Media Framework API</i>
JMS	<i>Java Message Service</i>
JMStudio	<i>Java Media Studio</i>
JVM	<i>Java Virtual Machine</i> (Máquina Virtual Java)
MSN	<i>Messenger</i> (<i>Software</i> para Comunicação de Dados)
NAT	<i>Network Address Translation</i>
P2P	<i>Peer to Peer</i>
RTP	<i>Real Time Protocol</i>
RTSP	<i>Real Time Streaming Protocol</i>
SCTP	<i>Stream Control Transmission Protocol</i>
TCP	<i>Transmission Control Protocol</i>
TIDIA	Tecnologia da Informação no Desenvolvimento da Internet Avançada
TIDIA-Ae	Tecnologia da Informação no Desenvolvimento da Internet Avançada – Aprendizado Eletrônico
TMF	<i>Tidia Media Framework</i>
UDP	<i>User Datagram Protocol</i>
VLC	<i>Software</i> para manipulação e apresentação de fluxos de áudio e vídeo
VOD	<i>Video on Demand</i>
VoIp	Voz sobre IP

1. INTRODUÇÃO

A literatura reporta vários esforços na área de transmissão de áudio e vídeo pela Internet (Lobato et al., 2007) (Bulut et al., 2002) (Conceição et al., 2006) (Guha et al., 2006) (Ford et al., 2005) (Ribeiro & Leung, 2005) (Natarajan et al., 2006) (Wang et al., 2004) (Eppinger, 2005). Tal esforço ocorre também no projeto TIDIA-Ae¹, em particular para apoio ao desenvolvimento de ferramentas que permitem a comunicação síncrona entre usuários remotos como, por exemplo, o Comunicador Instantâneo (CI) publicado por (Lobato et al., 2005) (Lobato et al., 2006).

Esses esforços ocorrem também para a realização de experimentos remotos, como forma de “reduzir a distância física” entre os participantes (Capobianco et al., 2005) (Bischoff & Rohrig, 2004). O mercado está atento a essas necessidades, como evidencia o uso de ferramentas tais como Skype² e MSN³.

No contexto das ferramentas que fazem uso de áudio e vídeo (A/V), tarefas como, configuração inicial, início e finalização da transmissão de fluxos, são realizadas com frequência. Essas funcionalidades recorrentes indicam a possibilidade de criação de configurações padrão de arquiteturas para transmissão de dados. Com essas funcionalidades, podem-se elaborar *softwares* que fazem uso dos fatores comuns para integração entre diferentes arquiteturas. Algumas questões e decisões são comumente abordadas nestas soluções como, por exemplo, transparência do ambiente de rede, sincronismo, compatibilidade entre *softwares*, reusabilidade, direito de uso, escolha das linguagens de programação, tipos de protocolos e compatibilidade com sistemas operacionais.

¹ Portal do Projeto TIDIA-Ae (Tecnologia da Informação no Desenvolvimento da Internet Avançada – Aprendizado Eletrônico). Disponível em: <http://TIDIA-ae.incubadora.fapesp.br/portal>. Último acesso dia 27 de fevereiro de 2008.

² Skype. Disponível em: <http://www.skype.com/intl/pt/products/skypeout/>. Último acesso dia 27 de fevereiro de 2008.

³ Windows Live Messenger. Disponível em: http://promo.msn.com.br/discover_live/. Último acesso dia 27 de fevereiro de 2008.

Ainda considerando decisões relacionadas a boas práticas de Engenharia de *Software* incluem-se: o reuso de códigos e os métodos para desenvolvimento de novas soluções. Algumas soluções, como é o caso do Videolan (VLC)⁴ e da Java *Media Framework* (JMF) API⁵ fornecem bibliotecas para integração, desenvolvimento e elaboração de novas arquiteturas.

Existem também outras soluções, as quais podem ser consideradas arquiteturas, como, por exemplo, Narada (Bulut et al., 2002), IDL (Kamolphiwong et al., 2002) e *Real Time E-Learning* (Guerri et al., 2002). A utilização de algumas arquiteturas, ferramentas ou *frameworks* já existentes, podem apresentar dificuldades como:

- **Falta de Documentação:** algumas ferramentas e arquiteturas não fornecem documentação adequada sobre a tecnologia, as interfaces, os códigos e os exemplos. Isto dificulta a utilização, a correção de erros e a elaboração de novas soluções (arquiteturas, ferramentas ou *frameworks*).
- **Dificuldade de Utilização:** outras tecnologias, apesar de possuírem documentação vasta, tornam difícil a utilização dos métodos de modo a compor as tarefas necessárias.
- **Falta de Compatibilidade:** a falta de compatibilidade pode ocorrer em vários pontos, por exemplo, entre plataformas, como é o caso do JMF API e do VLC, entre sistemas operacionais e/ou entre linguagens de programação.
- **Protocolo de Transmissão de Dados:** algumas arquiteturas utilizam um único protocolo para transmissão de dados, mas nem sempre o protocolo utilizado é o mais adequado para determinado ambiente ou configuração de rede. Outras arquiteturas permitem a implementação de diferentes protocolos para a transmissão de dados, porém nem sempre é simples implementá-los, pois às vezes a

⁴ Videolan. Disponível em: <http://www.videolan.org/>. Último acesso dia 27 de fevereiro de 2008.

⁵ JAVA *Media Framework*. Disponível em: <http://java.sun.com/products/java-media/jmf/index.jsp>. Último acesso dia 27 de fevereiro de 2008.

linguagem escolhida não fornece recursos suficientes, ou o sistema operacional não permite tais implementações.

- **NAT:** Existem arquiteturas que não superam algumas barreiras de conectividade impostas pela presença de *NAT*. Estas arquiteturas normalmente utilizam protocolos UDP e não funcionam para todos os casos, como se observa nos trabalhos de *NewPort-NetWork*⁶ e Comer (2000), sendo uma delas a arquitetura JMStudio⁷.
- **Não Modularidade:** os sistemas necessitam ser construídos de forma modular, permitindo assim a criação e a adaptação de métodos e/ou recursos necessários para aplicações específicas. Um sistema modular permite melhor manutenção, entre outros benefícios.
- **Falta de Transparência:** algumas arquiteturas não possuem transparência em relação ao ambiente de rede a ser utilizado, ou seja, não funcionam com os mesmos métodos para diferentes ambientes de rede, por exemplo, redes que implementam, bloqueiam algum tipo de protocolo ou utilizam NAT. Se houvesse transparência, não haveria a necessidade de modificar métodos por parte do desenvolvedor.
- **Impossibilidade de Reutilizar Códigos:** algumas arquiteturas não possuem código fonte aberto para desenvolvimento ou reutilização. Possuem boas soluções, porém não permitem a utilização, por serem códigos comerciais.

Pode-se notar que alguns dos problemas citados não dependem somente das boas práticas de Engenharia de *Software*, mas também do ambiente de rede para o qual a solução é proposta. Nesse sentido, é relevante considerar os seguintes requisitos do ambiente TIDIA-Ae/Kyatera⁸:

⁶ Newport-Networks, White Papers: SIP, IMS, VoIP, NAT traversal. Disponível em: <http://www.newport-networks.com/whitepapers/nat-traversal1.html>. Último acesso dia 07 de Fevereiro de 2008.

⁷ JMStudio. Disponível em: <http://java.sun.com/products/java-media/jmf/2.1.1/jmstudio/jmstudio.html>. Último acesso dia 07 de Fevereiro de 2008.

⁸ Kyatera, Disponível em: <http://kyatera.fapesp.br>. Último acesso dia 07 de Fevereiro de 2008.

- **Redes de Alta Velocidade com Baixa Latência e Alta Largura de Banda:** a rede prevê ambientes que tenham a sua disposição redes Gigabits, onde os servidores e os clientes são interligados por meio de fibra ótica e cabos par-trançado. Isto permite a utilização de recursos de áudio & vídeo com a mínima compactação, transmitindo assim uma grande quantidade de dados com qualidade e baixa latência.
- **Presença de NAT e Firewall:** o ambiente de rede, apesar de controlado, pode possuir NAT e *firewall*, portanto a arquitetura deve trabalhar com NAT e *Firewall* de modo transparente quando do estabelecimento de conexões pelos usuários.
- **Necessidade de Realização de Multicast de Fluxos de Áudio & Vídeo:** o ambiente necessita de *multicast* dos fluxos de áudio & vídeo para reuniões, aulas, videoconferências, salas de bate-papo e outras aplicações.
- **Código Fonte Aberto:** com o código fonte aberto, é possível realizar modificações, como, adição, remoção de métodos, adaptação e manutenção.
- **Outras Características:** deve ser modular, possuir documentação, portabilidade, independência de plataforma e compatibilidade com as ferramentas desenvolvidas no projeto Tidia-Ae.

Com base nesses requisitos, foi proposta a arquitetura S-MOJOHON (Lobato et al., 2007). Esta arquitetura foi elaborada com o intuito de permitir transmissão de vários tipos de dados, principalmente áudio e vídeo. Em seu projeto foram consideradas características importantes que permitem a sua utilização no ambiente TIDIA-Ae.

A arquitetura S-MOJOHON foi elaborada com módulos que permitem a criação de canais com diferentes configurações de protocolos, sendo assim a arquitetura atende a requisitos necessários para interfaces de configuração, envio e finalização. No entanto, a arquitetura não necessariamente atende às necessidades das ferramentas que utilizam áudio & vídeo no ambiente TIDIA-Ae, tornando-se necessária a investigação de alternativas para a

implementação da arquitetura na forma de um conjunto de componentes que contemplem os requisitos identificados.

O trabalho reportado nesta dissertação atingiu o objetivo de investigar alternativas para a implementação da arquitetura S-MOJOHON. Para a tomada de decisões sobre as alternativas e as avaliações, realizaram-se pesquisas sobre arquiteturas e ferramentas de áudio & vídeo para Internet. As pesquisas realizadas incluem: os protocolos para transmissão de fluxos de áudio & vídeo por meio de *NAT*, as arquiteturas para transmissão de fluxos de áudio & vídeo por meio da Internet (realizando *unicast*), as avaliações de fluxos de áudio & vídeo em redes de alta velocidade, vídeo sob demanda, *players*, captura e digitalização de imagens e sons.

Ainda no tocante à pesquisa realizada, o trabalho incluiu a investigação de algoritmos de seleção de componentes de *softwares*, chamados também de servidores ou serviços. Arquiteturas com propósitos especiais e componentes distintos, como é o caso dos trabalhos de Qazzaz (2003) e Guha et al. (2006), necessitam realizar algum tipo de seleção de componentes, os quais fazem parte da arquitetura e, portanto, precisam ser selecionados da “melhor” maneira possível para buscar o melhor desempenho, baseados, por exemplo, na maximização e minização de métricas. A seleção destes componentes é feita com o objetivo de atender aos clientes da melhor maneira possível. Segundo Zegura et al. (2000), o melhor servidor (componente de *software*) vai depender do serviço e das preferências dos clientes. Já para Guo et al. (2002) dois tipos de informações são importantes para a escolha do melhor servidor: topologia da rede e o tamanho da fila do servidor.

Alguns autores como, por exemplo, Mendona & Silva (2005), Guo et al. (2002), Zegura et al. (2000), Sayal et al. (1998) e Gargolinski et al. (2005), apresentam políticas e algoritmos para seleção de componentes de *softwares* (ou servidores). Os autores avaliaram algoritmos como, por exemplo, *Paralelo*, *HTTP ping*, *Best Last*, *Best Median*, *Random*, *Closest-Server-First*, probabilístico e *Lowest-average-ping-time*, chegando à conclusão de que se pode selecionar um servidor com o algoritmo *Paralelo* atingindo um bom

desempenho quando a métrica desejada está diretamente relacionada com a distância.

No contexto de redes de baixa latência, como é o caso da rede KYATERA⁹ do projeto TIDIA¹⁰, faz-se necessária a seleção de componentes para a transmissão de fluxos de áudio e vídeo entre clientes. Com base nesse requisito, foram realizadas duas implementações distintas da arquitetura S-MOJOHON (Lobato et al., 2007a), as quais se diferenciam, principalmente, pelo algoritmo de seleção do componente de reflexão.

A arquitetura S-MOJOHON implementa um algoritmo paralelo, baseado na troca de mensagens de configuração, e um algoritmo de seleção baseado em regras, que utiliza informações provenientes de uma ontologia que descreve Sistemas Autônomos, refletores e suas propriedades. Partindo do princípio que, “o melhor” servidor para atender a uma determinada requisição de transmissão de fluxo de áudio & vídeo, seja o servidor mais próximo, faz-se necessária uma avaliação para a descoberta de qual dos algoritmos apresenta o melhor resultado para a seleção de servidores/refletores. Esta avaliação foi um dos focos do trabalho reportado nesta dissertação.

1.1 O Projeto TIDIA-Ae

O Projeto TIDIA, *Tecnologia da Informação no Desenvolvimento da Internet Avançada*, tem como objetivo estimular o desenvolvimento de novas tecnologias para a Internet de modo colaborativo, envolvendo comunidades de universidades e de centros de pesquisa. O Projeto TIDIA é composto por três sub-projetos: TIDIA-Kyatera que investiga problemas associados ao provimento e ao uso de uma rede de computadores de grande largura de banda e baixa latência; TIDIA-Ae (Aprendizado Eletrônico), que investiga a construção de um ambiente de ensino-aprendizagem com ferramentas

⁹ Projeto Kyatera disponível em: <http://kyatera.incubadora.fapesp.br/portal>. Último acesso dia 3 de Dezembro de 2007.

¹⁰ Projeto Tidia-Ae disponível em: <http://tidia-ae.incubadora.fapesp.br/portal> acessado em: 3 de Dezembro de 2007.

inovadoras que utilizem a capacidade da rede Kyatera; e TIDIA-*Incubadora*, que provê uma incubadora virtual de conteúdos e de projetos de *software*.

O Projeto TIDIA-Ae tem como principais objetivos: a pesquisa e o desenvolvimento na área de tecnologia da informação, voltados para a especificação, o projeto e a implementação de ferramentas aplicáveis à educação. O Projeto TIDIA-Ae visa desenvolver soluções flexíveis de grande impacto social a um baixo custo, gerando assim a disponibilização de seus resultados como *software* de código aberto e gratuito.

Uma das linhas de pesquisa do TIDIA-Ae é a de laboratórios e experimentos remotos, a qual consiste em disponibilizar experimentos automatizados para que alunos possam realizá-los de forma remota de qualquer computador.

Os WebLabs, laboratórios associados ao Projeto TIDIA-Kyatera, possuem recursos para automação dos equipamentos, como por exemplo o *software* Labview¹¹, e também devem explorar o uso das redes do Projeto TIDIA-Kyatera. Alguns dos WebLabs demandam transferência de áudio & vídeo para seus usuários, com intuito de mostrar o experimento sendo realizado com qualidade. O intuito é fornecer não só imagens do experimento, mas também outros dados medidos no experimento, por exemplo, dados de sensores, para que os usuários possam entender e analisar os dados disponibilizados.

Os usuários desses Weblabs podem estar em quaisquer computadores, utilizando ou não a tradução de endereços de rede (por exemplo, via NAT), ou até mesmo realizando um experimento em grupo que necessite de *multicast* dos dados e das imagens geradas pelo experimento.

A necessidade da transmissão de fluxos de áudio & vídeo, e da infraestrutura de rede gerada pelo projeto TIDIA-Ae em conjunto com os Weblabs, geram motivações e características de ambientes para os quais uma solução de transmissão de fluxos de áudio & vídeo deve ser disponibilizada.

¹¹ Software LabView disponível em: www.ni.com/labview/. Último acesso dia 3 de Dezembro de 2007.

1.2 Identificação do problema

O principal problema pesquisado no trabalho reportado nesta dissertação diz respeito à seleção de componentes intermediários para a transferência de fluxos de áudio e vídeo. Esta escolha deve ser feita de modo a atender aos participantes envolvidos na recepção dos fluxos de áudio e vídeo da melhor maneira possível. Isto significa que deve ser encontrado um componente que satisfaça às políticas estabelecidas pelos usuários, por um grupo de participantes ou pelas ferramentas que o utilizam. Para isso fazem-se necessárias pesquisas em relação aos algoritmos de seleção dos componentes, de modo a permitir a escolha do algoritmo que apresente melhor desempenho em relação ao tempo de seleção de servidores.

Considerando a necessidade de realizar comunicações por áudio e vídeo pela Internet entre participantes de uma determinada comunidade, o Projeto TIDIA-Ae deve permitir a comunicação em redes que possuam NAT ou *Firewall*.

Considerando estes dois problemas, o trabalho reportado nesta dissertação foi realizado no contexto de um serviço de comunicação de boa qualidade para transferência de fluxos de áudio e vídeo que supera problemas relacionados à existência de NAT e *Firewall*, que suporta *multicast*, é independente de plataforma e pode ser integrado às ferramentas do Projeto TIDIA-Ae -tais como o Comunicador Instantâneo (Lobato et al., 2005) e a *WhiteBoard* (Jardim et al., 2005).

1.3 Objetivos

O objetivo principal do trabalho é avaliar e analisar algoritmos de seleção de componentes para reflexão de fluxos. Pretende-se, com este trabalho, obter avaliações das implementações da arquitetura S-MOJOHON em relação ao desempenho de algoritmos de seleção de servidores de fluxos de mídias contínuas, discutindo as análises e as conclusões obtidas com implementações e com os estudos de casos realizados em laboratório.

Os objetivos específicos incluem a seleção de uma arquitetura para a realização do trabalho, buscando arquiteturas que satisfaçam as

especificações de integração do projeto Tidia-Ae. Ainda dentro dos objetivos específicos encontra-se a implementação de uma versão da arquitetura que supere problemas de conectividade na presença de NAT.

1.4 Metodologia e Resultados Obtidos

A pesquisa bibliográfica foi realizada com artigos, livros, revistas e Internet, com o intuito de obter opções para a solução para transmissão de fluxos de áudio e vídeo.

Houve inicialmente uma pesquisa sobre as arquiteturas, os algoritmos de seleção e as tecnologias existentes. A seguir foi realizada, em colaboração com outros membros do grupo, a criação e a extensão de uma arquitetura que atendesse aos requisitos levantados: a arquitetura S-MOJOHON.

Realizaram-se duas implementações da arquitetura S-MOJOHON, avaliando-as em relação a outras tecnologias e comparando os algoritmos selecionados com as pesquisas em artigos. Os resultados são apresentados com o objetivo de fornecer informações para futuras decisões de desenvolvimento de arquiteturas e de escolha de algoritmos de seleção de componentes.

1.5 Estrutura da Dissertação

Esta dissertação está organizada da seguinte maneira: No Capítulo 2 são apresentadas as pesquisas, artigos e técnicas estudados na área de comunicação na Internet, mostrando de que modo estas pesquisas podem colaborar com os objetivos do trabalho. O Capítulo 3 detalha a arquitetura S-MOJOHON, sua implementação e as extensões realizadas no contexto desta dissertação. O Capítulo 4 discute resultados de avaliações realizadas com as várias versões da arquitetura. O Capítulo 5 apresenta conclusões e trabalhos futuros.

2.FUNDAMENTAÇÃO E TRABALHOS RELACIONADOS

2.1 Considerações Iniciais

Neste capítulo são abordados alguns tópicos importantes para o entendimento do trabalho reportado nesta dissertação, como, por exemplo, fluxos de áudio e vídeo, tipos de protocolos de comunicação, plataformas de comunicação, métricas de avaliações e conceitos de NAT.

2.2 Algoritmos de Seleção de Componentes

Em arquiteturas compostas por componentes de *software* distribuídos, faz-se necessária a seleção entre componentes, os quais podem fazer parte de uma determinada transmissão de fluxo de áudio e vídeo. Neste sentido, utilizam-se algoritmos de seleção dos mais variados tipos com o objetivo selecionar o melhor componente de *software* dentre os disponíveis na arquitetura para realizar a transferência.

Vários autores (Mendona and Silva, 2005), (Guo et al., 2002), (Zegura et al., 2000) (Sayal et al., 1998) apresentam comparativos entre diversos tipos de algoritmos de seleção de componentes, dentre eles, destaca-se o uso do Algoritmo Paralelo como referência para testes com outros algoritmos. O Algoritmo Paralelo apresenta bons resultados para a seleção de componentes conforme apresentado nos trabalhos citados.

O Algoritmo Paralelo é um algoritmo que envia solicitação de serviço para todos os servidores possíveis e realiza o serviço no servidor que responder primeiro, ou seja, o algoritmo paralelo faz uma pré-solicitação de serviço para todos os servidores que estejam prontos a atendê-lo, após isso, executa o serviço no servidor que responder mais rápido. Em uma primeira análise, pode-se perceber que o Algoritmo Paralelo possui um bom desempenho para a seleção de servidores e de serviços, caso o “bom” ou “melhor” possa ser definido como mais próximo ou possua outras

características como: o servidor com a menor quantidade de trabalho, o servidor com poucos processos sendo executados, ou o servidor que possuir a conexão de menor latência com o cliente requisitante.

Gargolinski et al. (2005) apresentam um trabalho interessante para a seleção de *middleware* de jogos. Em seu trabalho, os autores realizam automaticamente uma avaliação entre a distância entre servidores e clientes utilizando o comando *ping*, selecionando os servidores que possuam uma menor média em relação a todos os clientes e descartando os outros. Deste modo é possível obter o servidor que está mais próximo em relação a vários clientes.

Os trabalhos de Maximilien e Singh (2004) e de Garcia e Toledo (2006) visam facilitar a seleção de serviços da maneira que os clientes desejam. Para isto utilizam informações sobre qualidade de serviço (QoS) e ontologias. Já os trabalhos de Davies et al. (2004) e Srinivasan et al. (2006) discutem as vantagens e a aplicabilidade de OWL-S¹², a qual pode ser utilizada para facilitar a descoberta e a seleção de serviços de modo adequado, ou seja, da maneira que mais se adequar entre o desejo do cliente e o serviço disponível. A diferença entre os trabalhos encontra-se basicamente nas técnicas utilizadas para realizar a seleção, porém ambos possuem os mesmos objetivos, obter o melhor serviço possível.

Um estudo detalhado dos trabalhos de Mendona e Silva (2005), Guo et al. (2002), Zegura et al. (2000), Sayal et al. (1998), Maximilien e Singh (2004), Garcia e Toledo (2006), Davies et al. (2004) e Srinivasan et al. (2006) levou à conclusão de mesclar os trabalhos e comparar algoritmos de seleção, principalmente o Algoritmo Paralelo como o algoritmo com informações semânticas para explorá-los e avaliá-los em relação ao desempenho. O Algoritmo Paralelo foi escolhido devido ao bom desempenho apresentado nos trabalhos citados anteriormente e devido à facilidade de implementação deste algoritmo já que o sistema de comunicação (JMS) escolhido utiliza esse algoritmo como padrão. Já o algoritmo baseado em web semântica, ontologia

¹² OWL-S: Semantic Markup for Web Services. <http://www.w3.org/Submission/OWL-S/>. David Martin et al. W3C Member Submission 22 November 2004. Último acesso dia 25 de Maio de 2008.

e regras foi utilizado no intuito de explorar e obter mais opções na seleção de componentes.

2.3 Web Semântica

A Web Semântica deve representar um avanço em relação à Web atual (Berners-Lee et al., 2001). Com o intuito de explorar a Web Semântica, faz-se necessária a utilização de linguagens que representam informações e o conhecimento de um dado domínio. Inicialmente foi proposto o padrão *Resource Description Framework* (RDF)¹³ (Klyne & Carroll, 2004), que provê um modelo de dados genéricos baseado em grafos, podendo representar recursos, propriedades e relações.

Outra linguagem importante para representação do conhecimento é a linguagem de ontologia *Web Ontology Language* (OWL)¹⁴, a qual é um meio de descrever informação de maneira explícita na Web Semântica (Bechhofer et al., 2004).

O conceito de Web Semântica foi explorado pelo grupo no qual este trabalho se realiza. Por exemplo, a Figura 1 ilustra como a arquitetura S-MOJOHON (Lobato et al., 2007a) faz uso de RDF para descrever os componentes da arquitetura, como o fato de que um dado sistema autônomo (por exemplo, o *AutonomousSystem_3* na linha 15) tem acesso a um refletor (linha 16), conforme especificado nas linhas 17 a 27. O arquivo contém informações que descrevem o conhecimento que pode ser processado para extração de informações.

¹³ <http://www.w3.org/RDF/>.

¹⁴ <http://www.w3.org/2004/OWL/>.

```

1 <?xml version="1.0"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns="http://www.owl-ontologies.com/2007/xpto.owl#"
5   xml:base="http://www.owl-ontologies.com/2007/xpto.owl">
6   <ASTarget rdf:ID="AutonomousSystem_2"/>
7   <ASSource rdf:ID="AutonomousSystem_1">
8     <Neighbourhood_Relation>
9       <Neighbourhood rdf:ID="Neighbourhood_4">
10        <neighbourGap rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</neighbourGap>
11        <ASneighbour rdf:resource="#AutonomousSystem_2"/>
12      </Neighbourhood>
13    </Neighbourhood_Relation>
14  </ASSource>
15  <ASSource rdf:ID="AutonomousSystem_3">
16    <hasReflector>
17      <Reflector rdf:ID="Reflector_1">
18        <reflectorName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Refletor1</reflectorName>
19        <reflectorUptime rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">P9900S</reflectorUptime>
20        <hasOperatingTime>
21          <OperationTime rdf:ID="OperationTime_1">
22            <endTime rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2007-05-19T02:00:00</endTime>
23            <beginTime rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2007-05-19T00:00:00</beginTime>
24          </OperationTime>
25        </hasOperatingTime>
26        <reflectorIP rdf:datatype="http://www.w3.org/2001/XMLSchema#string">10.6.3.9</reflectorIP>
27      </Reflector>
28    </hasReflector>
29    <ASequivalentTo rdf:resource="#AutonomousSystem_2"/>
30  </ASSource>
31 </rdf:RDF>

```

Figura 1: Exemplo de um arquivo RDF que descreve recursos e relações do S-MOJOHON (Lobato et al., 2007a).

2.4 Avaliações de Fluxos de Áudio e Vídeo

É importante a realização de avaliações para medir a diferença das implementações de arquiteturas em relação ao suporte à transmissão de fluxos de áudio e vídeo. Algumas alternativas para realizar tais avaliações são, por exemplo, simulação, monitoramento por *software*, monitoramento por *hardware*, *statecharts*, Redes de Petri e *benchmark*. As avaliações, que são elaboradas com intuito de comparar sistemas, são úteis para identificar pontos positivos e negativos e, eventualmente, melhorar as arquiteturas e suas implementações.

Nos trabalhos de Bulut et al. (2002) e Conceição et al. (2006), as avaliações reportadas foram realizadas por monitoramento de *software*. Uma característica importante dos monitores de *software* é que eles possuem um custo menor quando comparados aos monitores de *hardware*, apesar da interferência ser um pouco maior. Neste caso, a interferência pode ocorrer na captura dos pacotes, na rede a ser testada, no processamento, na memória e em outros pontos do sistema.

Tão importante quanto escolher que tipo de avaliação que será usada é a escolha das métricas para realização das medidas e para coleta dos dados

que possam indicar pontos fortes e fracos do projeto. Nos trabalhos de Bulut et al. (2002) e Conceição et al. (2006), por exemplo, foram selecionadas as métricas de *jitter* e de atraso total (*delay* ou latência).

Para a transmissão e recepção dos fluxos de áudio e vídeo, considera-se a métrica atraso total (latência) uma das mais importantes, sendo definida como o tempo entre o envio e o recebimento dos dados. Segundo Conceição et al. (2006), se a latência for maior que 200 milissegundos, a conversa em áudio e vídeo pode ser prejudicada, tornando-a não natural e deixando-a de baixa qualidade. Além desta métrica, pode-se utilizar outras para a realização da avaliação, elas podem ser quantitativas (*jitter*, perda de pacotes) e qualitativas.

As avaliações foram feitas utilizando monitores de *software*, medindo atrasos de configurações e atrasos totais dos dados, devido à facilidade de implementação e os bons resultados obtidos com avaliações prévias (Lobato et al., 2007).

2.5 Áudio e Vídeo na Internet

A utilização do termo *Áudio na Internet* remete ao protocolo *Voice over Internet Protocol (VoIP)*¹⁵, cujo objetivo principal é permitir a utilização da Internet convencional para a troca de voz. Um exemplo de aplicação que faz uso de *VoIp* é o Skype¹⁶.

Além de voz, imagens também são largamente utilizadas na Internet de várias outras formas como, por exemplo, vídeos gravados para serem reapresentados posteriormente¹⁷ e os fluxos de áudio e vídeo transmitidos em comunicadores instantâneos, como o MSN¹⁸ e o Skype. São vários os casos em que as transmissões de fluxos de áudio e vídeo se fazem com o intuito de permitir a comunicação voltada para o domínio educacional, como é o caso do Projeto TIDIA-Ae.

Além de serem vários os objetivos para a transmissão de fluxos de áudio e vídeo, são várias as configurações e inúmeros os ambientes de rede

¹⁵ <http://www.fcc.gov/voip/>. Último acesso dia 25 de Maio de 2008.

¹⁶ <http://www.skype.com>. Último acesso dia 25 de Maio de 2008.

¹⁷ Como os disponibilizados pelo YouTube em <http://www.youtube.com>. Último acesso dia 25 de Maio de 2008.

¹⁸ <http://www.msn.com>. Último acesso dia 25 de Maio de 2008.

nos quais a transmissão pode ocorrer. Assim, são utilizados protocolos, módulos e algoritmos de seleção variados. No ambiente do Projeto TIDIA-Ae, por exemplo, pode-se explorar recursos tais como baixa compactação e utilizar os protocolos RTP e UDP diretamente, uma vez que a rede permite uma maior vazão dos dados, imagens e sons de melhor qualidade e com atrasos menores, quando comparada com a Internet convencional.

Algumas pesquisas demandam uma alta definição de imagens, como por exemplo, a máxima qualidade possível (Tulu et al., 2003) exigida para imagens médicas. Isso porque o processo de compressão de imagens pode prejudicar a análise. Para estes casos, além da imagem possuir alta qualidade, o sistema deve oferecer também redes de alta velocidade.

Segundo Liu et al. (2008) nos últimos dois anos houve um aumento de esforços e muitas inovações técnicas para transmissão de fluxos de vídeos na Internet. Shankar and Schaar (2007) complementam a preocupação no sentido de transmitir fluxos através da Internet em tempo real. Já Ding et al. (2007) aplicam esforços em pesquisas de videoconferências entre funcionários e propõem melhorias e funcionalidades que podem ser criadas para reuniões. Já Hung et al. (2007) realizam pesquisas no sentido de descobrir pessoas dominantes em grupos de reuniões por vídeo-conferência. Todos os trabalhos citados anteriormente são importantes, pois buscam melhorias para comunicação pessoal, atuando no sentido de obter uma boa qualidade de comunicação e dados das mesmas. A motivação para isso deve-se ao baixo custo da comunicação, à necessidade de reduzir "a distância" entre os participantes, às possibilidades de se trabalhar, estudar em diferentes lugares, etc. Algumas motivações e objetivos são comuns com o trabalho do autor, o qual busca a transmissão de fluxos de A/V entre participantes em diferentes ambientes de rede, buscando qualidade e baixo *delay* na transmissão de dados entre os participantes de um determinado grupo.

A literatura reporta, ainda, pesquisas na área de *multicast* de *streams* de áudio e vídeo em redes de alta velocidade e em redes sem fio (Tomos &

Frossard, 2007) (Zamora et al., 2007) (Kostić et al., 2008) (Hlavacs & Buchinger, 2008) (Zamora et al., 2008).

Dependendo do ambiente de rede para o qual os sistemas são projetados, há maior ou menor preocupação com a compactação e a codificação dos dados, como é o caso dos projetos elaborados para a Internet, onde vários tipos de redes e conexões estão presentes. A compressão dos dados é necessária quando se possui baixa largura de banda, o que não é o caso do ambiente do Projeto TIDIA-Ae.

2.6 Protocolos para Áudio e Vídeo na Internet

Esta seção tem o objetivo de discutir alguns protocolos que podem ser utilizados para transmissão de fluxos de áudio e vídeo variando de acordo com a aplicação. São apresentadas as vantagens e desvantagens de cada protocolo.

SCTP¹⁹ - O *Stream Control Transmission Protocol* (SCTP – Protocolo de transmissão de controle de fluxo) é um protocolo do nível de transporte especificado pela RFCs 2960 e 3286. Segundo Ribeiro & Leung (2005) e Natarajan et al. (2006), este protocolo foi desenvolvido inicialmente para transporte de sinais em redes VoIp. O SCTP é um protocolo orientado à conexão e funciona sobre IP. O SCTP é similar ao TCP, no sentido de ser um protocolo confiável e orientado à conexão. Entretanto, é diferente do TCP em outros aspectos, como, por exemplo: uma conexão SCTP pode transferir vários tipos de fluxos para os clientes em diferentes associações, pode ter múltiplos modos de entrega e é menos sensível a ataques de negação de serviço, quando comparado ao TCP. As desvantagens do SCTP incluem os fatos de que não são todos os sistemas operacionais que estão preparados para utilização desse protocolo e nem todas as linguagens de programação fornecem apoio para a sua implementação.

¹⁹ SCTP for Beginners. Disponível em: http://tdrwww.exp-math.uni-essen.de/inhalt/forschung/sctp_fb/sctp_intro.html. Último acesso dia 07 de Fevereiro de 2007.

TCP – O TCP (*Transport Control Protocol*) é um protocolo antigo e comumente usado na Internet. O TCP está definido nos RFCs 793, 1122, 1323, 2018 e 2581, é orientado à conexão e tenta corrigir certas falhas da rede, como perda de pacotes e retransmissão indevida. O TCP permite garantia de entrega de pacotes o que para determinadas conexões é muito importante, mas que para o caso de mídias contínuas como áudio e vídeo pode ser menos importante. O TCP é um protocolo de transferência confiável dos dados, contando com recursos de detecção de erro, retransmissões, reconhecimentos cumulativos, temporizadores, campos de cabeçalho para números de seqüência e de reconhecimento. A ampla adoção do TCP ocorre pela facilidade de utilização e pelo bom funcionamento para qualquer tipo de fluxo de dados, quando não há congestionamento de rede e perda de pacotes.

UDP - O UDP (*User Datagram Protocol*) é um protocolo de transporte não orientado à conexão. É simples, básico, faz tão pouco quanto um protocolo de transporte pode fazer e é definido pela RFC 768. O UDP basicamente adiciona campos de portas no cabeçalho para o serviço de multiplexação/demultiplexação. Estas características podem ser de grande vantagem para determinadas aplicações pelos seguintes motivos: não há estabelecimento de conexões, não há estados de conexões e há uma pequena sobrecarga de pacotes. Isto pode ser visto como vantagem em alguns casos, como por exemplo: transferência de fluxos de áudio e vídeo, DNS e outras aplicações que não necessitam da garantia de entrega. Pode ser visto como desvantagem para casos onde a orientação à conexão é necessária ou casos onde esse tipo de protocolo é bloqueado por *firewalls*.

RTP – O RTP (*Real Time Protocol*) é um protocolo do nível de aplicação, definido pela RFC 1889 e é um padrão para transferência de áudio e vídeo via Internet. O RTP comumente roda sobre UDP e normalmente possui um cabeçalho de 12 bytes. É muito utilizado em conjunto com padrões de telefonia por Internet. Deve-se enfatizar que o RTP não fornece mecanismos de garantia de entrega. Os dados do cabeçalho RTP são transparentes para

os roteadores, são como os datagramas, este protocolo é utilizado em conjunto com o RTCP, um protocolo de controle que pode ser utilizado para as aplicações para possíveis melhorias nas transmissões e recepções dos fluxos.

Na maioria das vezes dados de áudio e vídeo são transmitidos usando protocolos RTP/RTCP, permitindo assim atingir baixo *delay* devido a não garantia de entrega. O RTP é bastante utilizado para fluxos de áudio e vídeo por não despende muito tempo na adição e correspondente retirada de cabeçalhos. Deste modo, permite a entrega de quadros ou pacotes mesmo que alguns deles não sejam entregues durante as transmissões. Um dos pontos negativos é que pode não ser possível utilizá-lo em algumas configurações de redes com NAT, como observado por alguns autores (Comer, 2000) (Wang et al., 2004) (Eppinger, 2005) (Guha et al., 2007) (Bulut et al., 2002).

Em alguns casos os dados de áudio e vídeo são transmitidos via TCP, principalmente para resolver problemas relativos ao uso de NAT e porque, hoje em dia, grande parte dos dados é transmitida via TCP (Wang et al., 2004) (Eppinger, 2005).

A transmissão de áudio e vídeo via TCP encontra problemas quando a rede enfrenta congestionamento ou perda de pacotes, porém em redes de alta velocidade este problema nem sempre existe (Lobato et al., 2007), (Muller 1998) e (De-jen et al. 1998). Nestes casos, pode-se considerar que os problemas de congestionamento serão de ordem menor, sendo assim o protocolo TCP pode ser utilizado para transmissão de fluxos de áudio e vídeo.

Segundo Wang et al. (2004) e Eppinger (2005), há conexões comerciais que utilizam protocolo TCP para *streams*. Segundo Bulut et al. (2002) e Ganjam & Zhang (2004), no pior caso, ou seja, diante de tais dificuldades para estabelecimento de conexão e presença de NAT/*Firewall*, os fluxos de áudio e vídeo devem utilizar o protocolo TCP.

Devido às vantagens e às desvantagens dos protocolos citados anteriormente, a arquitetura selecionada para a realização do trabalho contempla os dois protocolos, TCP e RTP.

2.7 Network Address Translation

Network Address Translation (NAT) (Tradução de Endereços de Internet) faz uma separação entre as máquinas de uma rede interna e máquinas da Internet com endereços IPs reais. Em uma rede com NAT, as máquinas possuem endereços IPs falsos que apenas são reconhecidos pela rede interna e não pela rede externa. As máquinas da rede interna tomam endereços e portas emprestadas do serviço de NAT por um período de tempo para fazer acesso às máquinas e serviços externos, porém as máquinas externas não possuem acesso direto às máquinas internas (Comer, 2000).

NAT pode ser uma solução para alguns problemas, mas pode gerar outros. É considerada solução do ponto de vista da economia de endereços IPs e também do ponto de vista da segurança. Por outro lado, a utilização de NAT pode ser um problema em relação à conectividade de diferentes computadores, pois pode ser impossível realizar uma conexão direta a máquinas externas sem necessitar de nenhum tipo de configuração na rede (Iana, 2002).

O uso de técnicas de tradução de endereço como uma alternativa à falta de endereços ou como uma comodidade nas mudanças de provedores de serviço, por exemplo, afeta o funcionamento correto daquelas aplicações que não seguem o modelo cliente/servidor, como é o caso de aplicações ponto-a-ponto. Este problema é ilustrado na Figura 2, que representa o que pode acontecer durante a transmissão de fluxos de dados entre duas instâncias de uma aplicação, quando cada instância é executada em uma máquina de uma rede local diferente, e as duas redes usam NAT. Neste cenário (Figura 2), não há uma forma simples de permitir que qualquer uma das duas instâncias, A ou B, inicie a comunicação com a outra instância, pois nem A nem B tem como endereçar pacotes diretamente uma à outra. A nuvem apresentada na Figura 2 ilustra a dificuldade desta localização quando os computadores estão localizados em redes diferentes, e ocorre a presença de NAT entre eles. Neste caso, o Computador A (localizado em uma das redes) não tem como endereçar pacotes diretamente ao Computador B

(localizado na outra rede), pois as duas redes implementam NAT e os computadores A e B não possuem endereços IPs reais.

Figura 2: Configuração de rede usando NAT (Lobato et al., 2007)

NAT atua como um divisor entre a Internet com IPs reais e a rede interna, sendo que todas as conexões da rede passam pela máquina que conecta e controla o acesso correspondente. Os endereços e as conexões solicitados por uma máquina de IP falso são traduzidos e encaminhados à Internet com uma máquina com IP real; a resposta é entregue à máquina que roda o serviço de tradução, a qual realiza a tradução de forma reversa e entrega os dados solicitados para a máquina que está dentro da rede com NAT.

2.7.1 Os tipos de NAT

Os diferentes tipos de NAT em relação à permissão de conexão são *Full Cone* NAT, *Restrict Cone*, *Port Restrict Cone*, *Symmetric*.

Full Cone NAT - Este é o tipo mais simples e por isso está sujeito a ataques de varredura de portas, ou seja, qualquer computador pode enviar dados para as portas que estão abertas que os dados serão entregues ao computador interno na rede.

Restrict Cone - Só entrega pacotes para a rede interna se os pacotes recebidos no NAT tiverem como destino a porta da máquina interna.

Port Restrict Cone - Parecido com o *Restrict Cone*, porém só transfere os pacotes de resposta se houver algum pacote enviado para o endereço e a porta da máquina que está enviando pacotes.

Symmetric - Toda requisição interna é mapeada em um endereço IP e porta, podendo receber dados apenas das máquinas na qual a conexão foi solicitada.

2.7.2 Alternativas de Comunicação Através de NAT

Abaixo serão apresentadas algumas técnicas existentes para estabelecimento de conexões através de redes que têm NAT/*Firewall*. Algumas das alternativas apresentadas foram utilizadas para a elaboração do trabalho e implementação da arquitetura.

Técnicas de Túnel - Consistem em utilizar um servidor antes e um servidor depois do NAT/*Firewall*. A técnica faz uso do canal de comunicação existente entre servidores externos e internos para trafegar os dados desejados dentro deste túnel. O túnel empacota os dados que entram no canal e desempacota no momento que saem deste canal, desta maneira os fluxos UDP são trafegados dentro de uma conexão TCP, permitindo assim a passagem de dados para algumas configurações de NAT/*Firewall*. Isto ocorre nos casos onde algumas conexões TCP são permitidas e as conexões UDPs não (Guha et al., 2007) e (Srisuresh et al., 2008).

Técnicas de Reflexão - Utilizam um servidor o qual todos têm acesso. Este servidor recebe fluxos de um lado e transfere para outro lado. Este tipo de refletor funciona porque os participantes necessitam se autenticar neste servidor-refletor para criar canais de comunicação que serão aproveitados para a transferência dos fluxos (Guha et al., 2007) e (Srisuresh et al., 2008).

Outras técnicas - Consistem na combinação de protocolos STUN, SIP, ICE, *Hole Punching* e modificações de protocolos como mostram os trabalhos de

(Guha, 2004) e (Ford et al., 2005). Estas técnicas tentam, através de outras conexões, identificar que tipo de ambiente e que tipo de NAT/*Firewall* está sendo utilizado em ambos os clientes. Também são utilizadas não só como meio de pesquisa do ambiente de rede, mas tentam fazer com que as conexões se estabeleçam entre os clientes dentro de NAT, com intuito de usar os protocolos RTP/UDP para os recursos de áudio e vídeo. A grande vantagem de usar algumas dessas técnicas ou combiná-las é que elas não adicionam cabeçalhos desnecessariamente, não exigem recursos computacionais de roteamento de pacotes e estão menos suscetíveis a ataques em seus servidores intermediários. Porém, a grande desvantagem é que não funcionam para todos os tipos de NAT/*Firewall* (Ford et al., 2005), fazendo com que, nos piores casos, todo o fluxo de dados seja trafegado utilizando protocolo TCP e/ou utilizando técnicas de túnel e reflexão (Bulut et al., 2002) (Ganjam & Zhang, 2004) e (Srisuresh et al., 2008).

2.8 Arquiteturas para Transmissão de Áudio e Vídeo

Nesta seção são apresentadas algumas arquiteturas e implementações similares à arquitetura e à implementação elaborada, assim como as linhas de pesquisas e tecnologias mais comuns. As arquiteturas e tecnologias serviram de base para análise, comparação e principalmente investigação de soluções para problemas comuns.

2.8.1 Arquitetura Narada

Algumas arquiteturas que propõem a solução de conectividade para redes que implementam NAT são do tipo *Publish/Subscribe*, normalmente estas arquiteturas utilizam um “*broker*” para fazer a transferência dos dados entre os clientes. No caso da arquitetura Narada, ilustrada na Figura 3 (Bulut et al., 2002), é utilizado um servidor JMS que faz o recebimento e entrega das mensagens entre os clientes (elemento *Narada Listener* na Figura 3). Sendo assim, o JMS transmite os dados para qualquer tipo de cliente, dentro ou fora de NAT.

Como um dos objetivos desta arquitetura também é a transmissão de fluxos de áudio e vídeo, o JMS é utilizado para transmitir os fluxos de áudio e vídeo do mesmo modo que transmite mensagem, portanto tudo passa pelo JMS. A Figura 3 representa a arquitetura Narada com todos os módulos necessários para realizar a comunicação de dados.

Figura 3: Narada e Serviço Web de áudio e vídeo. (Bulut et al., 2002)

Outro objetivo do projeto da arquitetura Narada foi realizar a integração do JMF API com o JMS, fazendo com que os dados e os fluxos de áudio e vídeo fossem transmitidos para realização de *unicast* e *multicast*. A idéia é realizar as transferências dos fluxos áudio e vídeo através do protocolo RTP/UDP, porém nos casos que não são possíveis devido à configuração da rede, utiliza-se TCP, sendo este o protocolo comumente utilizado pelo JMS para troca de mensagens entre os clientes.

A arquitetura Narada também tem o objetivo de ser escalável e possuir baixo *delay* (comparado ao JMF) de transmissão de áudio e vídeo, chegando assim no resultado de alguns milissegundos mais lento, porém mesmo assim é possível a utilização de recursos de áudio e vídeo integrados com o JMS.

2.8.2 Arquitetura IDL

A arquitetura IDL (Kamolphiwong et al., 2002) executa a transmissão de todos os dados via um *broker* particular, permitindo a transferência de vários

tipos de dados e a realização de *multicast* para áudio e vídeo, dados de *Whiteboard* e outras aplicações. Todos os dados utilizam protocolo TCP quando necessário, permitindo assim a resolução para os problemas de NAT/*Firewall*, porém o foco é transmitir via RTP/UDP sempre que possível. Esta tecnologia foi construída em camadas e permite a adição de novas ferramentas por este sistema de comunicação. A Figura 4 apresenta a modularidade da arquitetura, assim como a utilização do framework JMF e os protocolos de comunicação permitidos.

Figura 4: Camadas da Arquitetura IDL. (Kamolphiwong, 2002)

A IDL é genérica e sua implementação é em JAVA, voltada basicamente para o ensino à distância, construída sem a utilização de JMS, porém utilizando a tecnologia JINI²⁰, que permite criar ferramentas e arquiteturas com o objetivo de ser escalável e outras vantagens.

2.8.3 Arquitetura Real-Time E-learning

No trabalho de Guerri et al. (2002) é realizada a implementação de uma arquitetura que faz uso das tecnologias do JMF, *Windows Media Player* e sistemas de comunicação centralizada. Também é voltada para o ensino à

²⁰ Jini Network Technology, disponível em: <http://www.sun.com/software/jini/>. Último acesso dia 24 de Maio de 2008.

distância, juntamente com a integração de outras ferramentas e tem como meio de rede, um satélite, podendo assim explorar a rede de alta velocidade.

A arquitetura também utiliza a linguagem de programação Java e o *framework* JMF integrado com o *Windows Media Player*, porém alguns pontos da arquitetura e implementação não são em Java. A Figura 5 mostra uma representação geral da arquitetura, assim como o propósito e a conexão, mostrando a possibilidade de comunicação entre um professor e várias salas de aula remotas.

Figura 5: Sistema *Real Time E-Learning*. (Guerri et al., 2002)

2.8.4 TIDIA-Ae Media Framework

Para captura e apresentação dos fluxos de áudio e vídeo foi utilizada a ferramenta JMF por sua disponibilidade, ser em Java, ser multiplataforma, por permitir a mudança e seleção de diferentes níveis de compactação, qualidade e captura, além de permitir transferências com atrasos menores que 100 ms dependendo da rede.

Algumas justificativas para a escolha deste *framework* encontram-se no bom resultado obtido com pesquisas realizadas no projeto Tidia-Ae, relatadas nos trabalhos de Capobianco et al., (2005), Barbosa, (2005), Lobato et al, (2006) e Lobato et al., (2007). O resultado mais significativo é que o JMF não introduz atrasos maiores que 40 ms para captura e apresentação dos dados.

O TIDIA *Media Framework* (TMF) (Barbosa, 2005) e (Lobato et al, 2006) é uma adaptação do JMF com adição de controle para o funcionamento de transferência de dados entre clientes de forma direta. O TMF faz uso do JMF (captura e apresentação) e linguagem Java (transmissão e manipulação dos dados), utilizando-os como ferramenta na transferência de *streams* entre aplicações do projeto TIDIA-Ae, utilizado na primeira versão do Comunicador Instantâneo e na realização de experimentos remotos (Capobianco et al., 2005).

O funcionamento do TMF é simples se comparado às arquiteturas apresentadas no trabalho, porém foi o primeiro protótipo implementado pelo autor, o qual permite a transmissão de áudio e vídeo entre participantes dentro da mesma rede sem a presença de NAT.

2.8.5 Arquitetura STARCAST

O trabalho de Tsuchiya et al. (2006) utiliza servidores separados em camada de dados e controle para a transferência entre pontos distintos na Internet, também faz uso de sistemas de adaptação para transferência de dados em redes com presença de NAT/*Firewall* e para IPV6, além das tecnologias Java e JXTA para comunicação P2P.

A seleção dos componentes intermediários é realizada através de algumas métricas como, por exemplo, carga do processador, tempo de inicialização e número de conexões. Para a transmissão dos fluxos a arquitetura utiliza tanto TCP quanto UDP. A Figura 6 demonstra a localização dos clientes (Peer 1 e 2) que desejam participar de troca de informações, assim como os componentes de reflexão (JXTA *relay*) e os fluxos de dados e controle.

Figura 6: Arquitetura Starcast (Tsuchiya et al., 2006).

2.8.6 Comparação entre as Arquiteturas

A Tabela 1 sumariza uma comparação entre as arquiteturas apresentadas, no que tange requisitos importantes para transmissão de áudio e vídeo em redes de alta velocidade.

Tabela 1: Requisitos e comparativos entre as arquiteturas

	Narada	IDL	RealTime E-learning	StarCast
Compatível com JMF	√	√	√	√
Modular	√	√	√	√
Somente em JAVA	√	√	X	√
Suporta <i>NAT/Firewall</i>	√	√	√	√
Conectividade entre Ipv6 e Ipv4	X	X	X	√
Orientação a Canais	√	X	X	
Permite Video UDP(RTP)	√	√	√	√
<i>*Escalável</i>	√	√	X	√
Camadas	X	√	X	√
<i>Broker</i> (Mensagens de controle separadas dos dados multimídia)	X	X	X	√

**Escalável:* Depende do nível de escalabilidade, é importante realizar testes de escalabilidade com intuito de verificar o quanto cada servidor pode suportar sem comprometer a qualidade (Bulut et al., 2002) e (Kamolphiwong, 2002).

As implementações das arquiteturas Narada (Bulut et al., 2002), IDL (Kamolphiwong et al., 2002) e *RealTime E-learning* (Guerri et al., 2002), citadas neste capítulo, atendem a muitas das características e requisitos desejados no ambiente TIDIA-Ae, entretanto alguns requisitos necessários não são suportados. Dentre as características desejadas destacam-se compatibilidade com o JMF API, modularidade, habilidade para comunicação em ambientes com NAT e a utilização de fluxos de áudio e vídeo com protocolo UDP. Além destes, outro requisito de fundamental importância para o ambiente TIDIA-Ae é a separação do *broker* de comunicação da transmissão dos fluxos. A idéia geral da solução proposta no trabalho é atender os requisitos do ambiente TIDIA, permitindo integração com outras ferramentas na forma de componente.

Devido aos bons resultados obtidos durante o projeto Tidia-Ae com a arquitetura TMF, optou-se pela implementação de uma nova arquitetura aproveitando as boas características e objetivando novas funcionalidades no intuito de torná-la uma arquitetura que pudesse superar as arquiteturas citadas, surgindo assim a arquitetura S-MOJOHON.

2.9 Considerações Finais

Neste capítulo foram abordados alguns tópicos importantes para o entendimento do trabalho reportado nesta dissertação, como, por exemplo, fluxos de áudio e vídeo, tipos de protocolos de comunicação, plataformas de comunicação, métricas de avaliações e conceitos de NAT. Os tópicos foram abordados para permitir a avaliação de opções. O próximo detalha a arquitetura S-MOJOHON e suas extensões, realizadas no contexto desta dissertação.

3. EXTENSÃO DA ARQUITETURA S-MOJOHON

3.1 Considerações Iniciais

Este capítulo detalha a arquitetura S-MOJOHON, sua primeira implementação e os resultados obtidos a partir da mesma, sendo uma contribuição importante do autor, incluindo parte do projeto da arquitetura, implementações de todos os módulos e avaliação da arquitetura.

3.1.1 Trabalhos relacionados - S-MOJOHON

Sistemas Autônomos

Inicialmente é necessário o esclarecimento do significado de Sistemas Autônomos, sendo este a base de informações da ontologia do S-MOJOHON. A arquitetura de roteamento de pacotes IP de uma rede como a Internet geralmente está dividida em uma hierarquia de dois níveis (Huston, 2006), sendo que o ambiente de rede é dividido em domínios e cada domínio possui seu ambiente de roteamento interno.

Os protocolos de roteamento utilizados são de responsabilidade do administrador do domínio em questão. O segundo nível de roteamento é entre os domínios e descreve a maneira como estão conectados entre si. Cada um desses domínios é conhecido por Sistema Autônomo (SA ou AS do inglês *autonomous system*) e pode ser visualizado como uma coleção de redes IP e roteadores, todos sob controle de uma ou mais entidades que apresentam para a Internet uma determinada política de roteamento. Cada SA possui um identificador único e público, conhecido como ASN (AS number). A Figura 7 representa os ASs, os clientes que desejam se comunicar (H1 e H2), o fluxo de dados e os diferentes algoritmos de roteamento estabelecido entre eles.

Figura 7: Exemplo de representação em forma de grafo dos ASs e os diferentes algoritmos de roteamento (Lobato et al., 2007a).

Representação do conhecimento

Têm sido reportadas na literatura ações no sentido de utilizar informações semânticas para o gerenciamento de redes, como por exemplo, situação de roteadores e comutadores. Em Hama et al., (2006), usa-se RDF para criar a Network Description Language (NDL), capaz de descrever a topologia de uma rede, mas não há uma ontologia associada. Outros como Dabek et al., (2004), Gummadi et al. (2002) e Sharma et al. (2006) estão focados na identificação exata do valor de latência. Para implementação em questão obter latência exata não é extremamente importante, já que com o número de saltos entre os ASs consegue-se uma relação direta em relação à latência (Li & Sollins, 2004) e (Obrackza & Silva, 2000). Com os pontos necessários pode-se selecionar então o componente mais próximo do cliente utilizando informações e regras. A Figura 8 representa o modelo de dados que foi representado em formato RDF. Mostra também as relações entre os objetos e suas propriedades, sendo esta uma representação gráfica da ontologia que descreve os ASs, os componentes de reflexão, as relações de vizinhança entre eles e as suas propriedades.

Figura 8: Representação gráfica da ontologia que descreve os ASs (Lobato et al., 2007a).

3.2 Arquitetura S-MOJOHON

Os objetivos e funcionamento da arquitetura S-MOJOHON (Lobato et al., 2007) são: modularidade, uso de Java, compatibilidade com JMF e principalmente habilidade para superar NAT. Esses requisitos são justificados a seguir.

Abstração para o usuário: A arquitetura necessita ser abstrata em relação ao meio de rede em que está sendo usado, deve poder descobrir qual rede está sendo utilizada e se auto-adaptar para o funcionamento da mesma.

Modularidade: A arquitetura deve ser modular, com o intuito de facilitar não só o desenvolvimento, mas também facilitar a criação, modificação, adaptação e adição de novos módulos, métodos e funções.

Uso de JAVA: Deve ser toda desenvolvida em JAVA, pois segue o padrão dos projetos de pesquisa do TIDIA e contempla todas as vantagens e desvantagens da plataforma JAVA.

Compatível com JMF: Deve ser compatível com JMF, pois é um dos *frameworks* usados em projetos do TIDIA e outras arquiteturas. Assim é possível aproveitar ferramentas que já utilizam este *framework*, fazendo assim reuso dos códigos.

Habilidade para superar NAT: Deve possuir maneiras de superar NAT e *Firewalls*, fazendo com que os fluxos de áudio e vídeo sejam transmitidos e recebidos.

A Figura 9 apresenta os módulos que compõem a arquitetura S-MOJOHON (Lobato et al., 2007): módulo de transmissão, módulo de reflexão, módulo de adaptação, módulo de monitoramento/configuração e módulo de recepção. Esta seção detalha cada um desses módulos.

Figura 9: Módulos da arquitetura S-MOJOHON (Lobato et al., 2007).

Módulo de transmissão: Este módulo é responsável pela transmissão dos fluxos de áudio e vídeo, podendo fazer a transmissão sobre protocolos UDP/TCP e transmitir diretamente para um cliente ou para um refletor qualquer. O módulo de transmissão recebe informações sobre o local onde os fluxos devem ser enviados e também informações sobre o protocolo a ser usado. Este módulo é baseado no JMF (*Java Media Framework*) com devidas adaptações quando necessária a transmissão de pacotes através do protocolo TCP.

Módulo de reflexão: Além de gerenciar fluxos entre os clientes, o módulo de reflexão deve entregar os fluxos da maneira exata na qual os clientes necessitam, fazendo com que o problema de conectividade NAT/*Firewall* seja resolvido caso necessário. Este módulo possui a configuração pré-definida de vários tipos de canais, combinando entre eles UDP, TCP e *multicast*, sendo estes os canais responsáveis pelo recebimento e entrega dos fluxos entre os participantes. Os canais de comunicação são a base do módulo de reflexão. Os canais são criados de acordo com a necessidade e a solicitação do usuário e/ou sistema. Os canais são constituídos basicamente de entrada e saída de dados. As entradas são responsáveis por receber os fluxos de um cliente e as saídas são responsáveis por entregar o fluxo a um ou mais clientes. Em relação à configuração, tem-se a combinação entre os protocolos UDP e TCP na entrada e saída, obtendo assim canais UDP/TCP, onde a entrada é UDP e a saída TCP respectivamente. Pode-se chegar então a 4 combinações, além das combinações para realização de *multicast*. Os canais podem ser construídos e utilizados de diferentes formas, desde de que sejam associados com o módulo de adaptação quando necessário. Podemos descrevê-los de uma forma mais simples como sendo um roteador de pacotes de áudio e vídeo entre diferentes portas e IPs. Os canais e combinações também permitem transferências de conjunto de bits (áudio, vídeo, mensagens, imagens, etc) de um transmissor para um receptor.

Módulo de Monitoramento e Configuração (*broker*): Este módulo tem a importância de monitorar o refletor através de métricas escolhidas pelo usuário. Estas métricas de monitoração são importantes para verificar não somente o número máximo de fluxos de áudio e vídeo suportado, mas também permitindo uma política de escolha de qual refletor vai ser escalonado para atender os clientes. Esta monitoração será feita a partir de monitores de *softwares*, pois são mais fáceis de implementar do que monitores de *hardware*. As métricas serão escolhidas futuramente, mas pretende-se avaliar *jitter*, atraso, consumo da largura de banda, entre outros. Além da monitoração este módulo é responsável pelo gerenciamento dos fluxos, ou seja, por abrir e fechar canais entre os clientes. O módulo de configuração identifica os participantes que desejam transmitir fluxos, o tipo de protocolo que será usado para a transmissão e recepção e deve gerenciar todos os fluxos. Além disso, o módulo deve ser capaz de fechar conexões que não estão sendo usadas, atualizar as conexões que estão ativas, e abrir canais de fluxos para os clientes que solicitarem.

Módulo de Adaptação: Módulo que pode ou não ser utilizado, dependendo da maneira como a transmissão do *broker* para o receptor será feita. Quando a transmissão do *broker* para o módulo de recepção for através do protocolo TCP, faz-se necessária a adaptação do fluxo, ou seja, receber o fluxo TCP e entregar para o módulo de recepção o fluxo no formato UDP da mesma maneira em que foi transmitido, dessa forma obtem-se o desacoplamento dos módulos.

Módulo de recepção: O módulo de recepção opera em conjunto com o de adaptação, se necessário, sendo responsável por receber os fluxos e apresentá-los ao usuário, podendo ser áudio ou vídeo. Este módulo é também baseado no JMF, o que permite a reutilização de código e principalmente a captura e apresentação dos fluxos de dados sem acrescentar muito tempo no processamento de sinais, ou seja, a captura e apresentação é realizada no tempo máximo de 35 ms.

3.3 Etapas para estabelecimento de uma transmissão

Configuração

Nesta fase são feitas as configurações necessárias para a transmissão do fluxo de áudio e vídeo. Esta configuração envolve a escolha do tipo de canal desejado, podendo usar as várias combinações disponíveis na implementação. Esta fase também tem o intuito de identificar que tipo de ambiente de rede os clientes estão, definindo assim um melhor ajuste nos canais para uma transmissão de melhor qualidade.

Ainda na fase de configuração, os clientes se comunicam entre si e com o refletor para escolher e informar o tipo de canal que será utilizado na transmissão e onde os dados serão entregues para o cliente. Esta configuração é necessária para informar o cliente que vai receber o fluxo de dados e informar onde encontrá-lo.

Após a configuração dos canais desejados e a informação para os clientes sobre a localização dos fluxos, executa-se a transmissão dos dados.

A Figura 10 mostra um exemplo do processo de transmissão e recepção de dados entre dois participantes. Os passos 1, 2 e 3 da Figura 10 indicam a solicitação do módulo de transmissão para o módulo de monitoramento um refletor. Os passos 4, 5, 6 e 7 indicam a configuração de um módulo de reflexão pelo monitoramento e a disponibilização do módulo de reflexão para o módulo de transmissão. Nestas etapas ocorrem a seleção do refletor utilizando algum algoritmo. Os passos 8, 9 e 10 fazem parte da configuração do módulo de recepção, neste momento o módulo de reflexão recebe o endereço do refletor. Os passos 11, 12 e 13 indicam a transmissão dos dados do módulo de transmissão para o refletor e do refletor para a recepção, fechando assim a transferência de dados entre um cliente e outro, podendo ser um fluxo de áudio, vídeo ou outros dados. A partir da etapa 14 inicia-se o processo de fechamento das conexões e parada de transmissão dos fluxos entre a transmissão e a recepção, onde são finalizados os módulos de adaptação, reflexão, transmissão e recepção.

Transmissão

Nesta fase ocorre a transmissão dos fluxos de dados entre um cliente e outro, isto é realizado logo após as configurações, sendo assim é necessária a escolha do tipo de transmissão que será realizada, áudio ou vídeo.

Os dados capturados pelas câmeras de vídeo e microfones são adaptados, se necessário, para serem transmitidos até o servidor intermediário (refletor). Comumente poderá ser utilizada a transmissão dos dados via protocolo UDP sem nenhuma adaptação até o refletor, sendo que do refletor até o receptor dos dados, pode-se utilizar os recursos de adaptação se necessário.

Figura 10: Algoritmo de configuração e transmissão. Figura adaptada de Lobato et al. (2007).

Finalização

Após o uso dos canais são necessárias as finalizações e os fechamentos dos mesmos. Para isto o sistema de troca de mensagens é utilizado para informar ao *broker* e ao refletor que a comunicação será interrompida, fazendo assim com que o refletor feche os canais e portas. Esta comunicação

também é feita para o cliente que estava recebendo os dados, para que ele feche a comunicação com o refletor.

Desta forma, as portas e *sockets* não utilizados são fechados, liberando recursos da máquina para que possam ser utilizados novamente pelo mesmo ou outros processos.

Persistência

Para a persistência de documentos RDF e para as inferências adotamos o arcabouço de Web Semântica Jena²¹. Populou-se a base de conhecimento com uma topologia de sistemas autônomos similar a Internet. Cada módulo de reflexão, ao se registrar junto a um módulo de monitoramento e configuração, é incluído também na base de conhecimento. Para selecionar o módulo de reflexão, o módulo de transmissão publica uma mensagem para um módulo de configuração no tópico de sinalização compartilhado. Estas mensagens contêm um objeto que inclui, entre outras informações, o endereço IP do módulo de transmissão e o tipo de canal que deve ser configurado.

Assim que a mensagem chega ao módulo de configuração, o último identifica o Sistema Autônomo correspondente ao endereço IP do primeiro e, então, recupera a topologia, dessa forma, a topologia e as regras são, então, encaminhadas ao motor de inferência.

Funcionamento e troca de informações entre os módulos

Adicionando Web Semântica através de uma ontologia sobre ASs, surge então o S-MOJOHON.

A ontologia criada no S-MOJOHON representa os sistemas autônomos, as localizações e tempos de funcionamento dos módulos de reflexão da arquitetura, além de estabelecer a distância entre ASs vizinhos. Desta maneira é possível localizar na forma de um grafo, os sistemas autônomos que possuem refletores e que são vizinhos dos sistemas autônomos nos quais os módulos de transmissão estão localizados.

²¹ Jena. Disponível em: <http://jena.sourceforge.net/>. Último acesso dia 27 de fevereiro de 2008.

A representação da implementação S-MOJOHON que envolve ASs descritos na linguagem RDF e OWL nos leva às Figuras 7 e 8, onde são apresentados os nós e arcos, em formas de grafo. Os nós representam recursos e os arcos representam propriedades ou relações descrevendo recursos.

A representação do S-MOJOHON é apresentada pela Figura 11, onde existe um módulo de configuração responsável por acessar a base de conhecimento, obter a topologia da rede, inferir na base de sistemas autônomos utilizando regras e fornecer ao transmissor o melhor refletor, baseado nas regras.

Periodicamente o módulo de monitoramento testa cada refletor sob sua responsabilidade para obter suas localizações, verificar se está operacional, e coletar as métricas (passo 1 na Figura 11).

Toda vez que o módulo de monitoramento detecta uma mudança de estado de um refletor, ele atualiza a informação armazenada no repositório de conhecimento, criando um novo documento RDF (passo 3 na Figura 11) baseado na ontologia adequada (passo 2 na Figura 11). Este documento RDF é armazenado no repositório de conhecimento por um módulo de persistência de RDF (passo 4 Figura 11).

Sempre que um módulo de transmissão necessita configurar um ponto de reflexão, ele encaminha os requisitos desse refletor para o módulo de configuração (passo 5 Figura 11). O módulo de configuração obtém a informação armazenada no repositório de conhecimento através do módulo de persistência de RDF (passo 6 Figura 11). No exemplo da Figura 11, esse conhecimento é a topologia de sistemas autônomos e a localização dos refletores nesses sistemas autônomos (passo 7 Figura 11).

Com base no documento RDF obtido no passo 7, um motor de inferência produz outro documento RDF (passo 9) de acordo com as regras (passo 8) impostas pelos requisitos do passo 5. Novamente no exemplo da Figura 11, o documento resultante no passo 9 contém os refletores que satisfazem às restrições impostas pela aplicação. O documento RDF do passo 9 é processado pelo módulo de configuração, que obtém o endereço do refletor, o configura e encaminha o mapeamento dos canais para o módulo

A implementação do barramento *publisher/subscriber* é provido pelo JBossMQ, uma implementação do Java *Message Service* (JMS²²) que faz parte do servidor de aplicações JBoss 4.0. A adoção do JBossMQ pode ser justificada pelo fato de haver disponível um arcabouço de aplicação que especializa as classes de JBoss e facilita o desenvolvimento de aplicações que precisam de comunicação *publisher/subscriber* tornando JBossMQ uma ferramenta ideal para a prototipação rápida de aplicativos que precisam desse modelo de comunicação.

É importante deixar claro que o sistema de comunicação é independente do monitor e dos clientes que desejam transmitir fluxos. Este sistema está sendo utilizado por ser livre, por ser do projeto TIDIA e por ter um bom funcionamento no ambiente da Internet, mas nada impede que possa ser trocado por qualquer outro sistema de comunicação.

As mensagens trocadas são através de objetos, onde tais objetos possuem características como IP do refletor, IP do cliente, porta de recebimento do fluxo do refletor (entrada), porta de transmissão do fluxo (saída) para outro cliente, identificador do fluxo e outros campos.

3.5 Implementação de Captura e Apresentação

JMF API (Java *Media FrameWork*) é um *framework* Java que permite captura, manipulação e apresentação de recursos de áudio e vídeo. É independente de plataforma, pois funciona sobre a máquina virtual Java e permite a compactação de dados também. Essa tecnologia está sendo usada pelas arquiteturas de Tulu et al. (2003), Guerri et al. (2002), Bulut et al. (2002), Lobato et al. (2007), Lobato et al. (2006), Capobianco et al. (2005) e Barbosa, (2005).

JMF possui uma vasta API na qual se pode implementar e modificar exemplos e códigos de acordo com a necessidade do usuário. Este *framework* possui também uma ferramenta (JMStudio) para mostrar o que o JMF pode

²² Java *Message Service*. Disponível em: <http://java.sun.com/products/jms/>. Último acesso dia 27 de fevereiro de 2008.

fazer. Como padrão transmite os fluxos de áudio e vídeo via RTP/UDP para os clientes. Possui também vários tipos de exemplos, para *player*, gravação, transmissão, recebimento e clonagem de recursos.

O *framework* JMF está sendo usado em transmissões de vídeo, áudio, vídeo-conferências, programas P2P e em muitas outras aplicações. É uma tecnologia robusta, com código fonte aberto, possui boa documentação e permite várias modificações. A Figura 12 mostra o JMF, suas características, etapas de processamento e as suas funcionalidades gerais.

Figura 12: JMF API. (JMF picture²³)

O interessante desse *framework* é que pode facilitar o desenvolvimento e/ou a utilização de recursos audiovisuais. Isto ocorre, pois existem alguns exemplos de *softwares* disponíveis no próprio *site* que podem ser utilizados para o desenvolvimento rápido de aplicações e protótipos.

3.6 Implementação da arquitetura S-MOJOHON

O trabalho reportado nesta dissertação inclui a investigação de alternativas para implementação de uma arquitetura que permita trafegar fluxos de áudio e vídeo via Internet, em Internet de alta velocidade. A implementação da arquitetura possui vantagens do Skype e do MSN, em termos de

²³ JMF Picture, Disponível em: http://sdhc-esd.sun.com/ESD4/JSCDL/jmf/2.0/jmf2_0-guide.pdf?AuthParam=1169642674_bf3a77d2351e011be7560462e78459c3&TUrl=an1npDpbKod7kSYrROhENTonleQ4W0D1Lc4nXz+pGFFranixdCdgxDTPbW4=&TicketId=dlh/PAJGM+A8+w==&GroupName=SDLC&BHost=sdhc5g.sun.com&FilePath=/ESD4/JSCDL/jmf/2.0/jmf2_0-guide.pdf&File=jmf2_0-guide.pdf Último acesso em: 28/02/2007

atravessar e receber fluxos através de NAT, além de possuir baixa latência (comparável à latência do JMF API) na transmissão de áudio e vídeo.

O conjunto de componentes foi implementado em Java para futura utilização dos laboratórios remotos WebLabs e clientes TIDIA-Ae.

A implementação contempla itens descritos na Tabela 1, está de acordo com os requisitos do ambiente TIDIA-Ae e com as características da arquitetura S-MOJOHON. Vantagens e desvantagens são:

- **Vantagens:** além de velocidade e qualidade compatíveis (próximas) ao JMF, está em camadas, em módulos, 100% desenvolvida em Java e compatível com TIDIA-Ae.
- **Desvantagens:** dependência de servidores (*brokers*) entre os participantes e a não comunicação com telefones e celulares.

3.6.1 Algoritmos de Seleção de Componentes (Refletores)

Algoritmo Paralelo

O Algoritmo Paralelo é uma implementação simples, porém eficiente em relação à seleção de componentes intermediários, pode-se dizer eficiente quando se trata da métrica de obtenção do servidor mais disponível para atender ao serviço. O algoritmo de seleção de servidores paralelo funciona da seguinte maneira, primeiro envia-se mensagem de requisição de servidores a todos os servidores registrados nos canais do serviço de comunicação, após isso os servidores irão responder à mensagem com intuito de atender à requisição, para finalizar, o requisitante executa o serviço no servidor que responder mais rápido.

Para simplificar pode-se fazer uma analogia com um patrão requisitando ajuda de funcionários para realizar uma determinada tarefa, por exemplo, imagine que o patrão necessita de um funcionário pra entregar um conjunto de informações para dois setores distintos dentro da empresa no menor tempo possível. Para o patrão realizar a seleção do melhor funcionário pra realizar a tarefa descrita anteriormente, ele pode utilizar qualquer tipo de métrica ou usar o Algoritmo Paralelo. Usando o algoritmo paralelo o patrão se dirige até a sala onde estão todos os funcionários

trabalhando e a disposição ao mesmo tempo e espalha a mensagem de requisição na sala, realizando isso em difusão, ou seja, solicita a todos os funcionários que realizem determinada tarefa. Após este momento no qual todos os funcionários receberam a mensagem, alguns deles irão se prontificar a atender o patrão. Os funcionários que se prontificaram é porque escutaram a requisição do patrão mais rápido do que os outros funcionários, ou porque são os funcionários que estão menos ocupados em relação aos outros. Sendo assim, o funcionário que responder à requisição do patrão primeiro é o funcionário que está mais perto ou menos ocupado, ou seja, é o melhor funcionário pra executar a tarefa no menor tempo possível.

Algoritmo baseado em regras

No sentido de investigar alternativas para implementação da arquitetura, foram explorados conceitos da Web Semântica. As principais alterações realizadas na arquitetura encontram-se nos módulos de monitoramento e configuração, além da adição de novos módulos, o de persistência, repositório e de inferência. As transferências de dados também sofreram alterações, pois há a necessidade de consultas sobre topologia, informações em RDF e ontologia. Para esse tipo de algoritmo faz-se necessário o uso de ontologia e regras. As regras devem garantir em nossa prova de conceito que o refletor selecionado para um determinado fluxo seja (1) o mais próximo do módulo de transmissão, e; (2) no caso de haver mais de um refletor na mesma distância, o que esteja a mais tempo operando sem falhas.

A regra a seguir, por exemplo, seleciona o refletor no caso mais simples, quando dois ASs são equivalentes. Se sistemas autônomos, A e C, possuem *gap* zero, então eles são equivalentes e, portanto, o refletor mais próximo do sistema autônomo A é o que está instanciado no sistema autônomo C.

```
[rule1: (Neighbourhood_Relation(?A, ?B), ASneighbour(?B, ?C), neighbourGap(?B,
?X),
equal(?X, 0) -> ASEquivalentTo(?A, ?C) ), hasReflector(?A, ?Y) ->
SuitableReflector(?Y)]
```


Depois da inferência, um documento RDF com o refletor a ser usado estará disponível e será processado pelo módulo de configuração. O endereço do refletor e o mapeamento dos canais é, então, publicado no tópico de sinalização. Por fim, o refletor e o módulo de transmissão podem se configurar de acordo com o mapeamento publicado.

3.7 Resultados da implementação

3.7.1 Captura Apresentação e Integração

Em 2005/2006, com o desenvolvimento do TMF, obteve-se resultado em uma aplicação do projeto TIDIA-Ae, sendo que com o TMF foi possível realizar um experimento de forma remota, através da gaiola de Skinner (Capobianco et al. e Barbosa, 2005), onde há a necessidade de fluxos de áudio e vídeo com tempos de transmissão menores que 200 ms.

O experimento foi realizado com sucesso em duas ocasiões, uma delas na rede local da Computação da UFSCar e outra realizada da UFSCar e USP utilizando a rede óptica de alta velocidade do TIDIA-Ae. O experimento foi reportado em Barbosa (2005). No entanto, a grande desvantagem do TMF foi o problema de transmissão para redes com presença de NAT/*Firewall*.

O resultado do experimento também foi uma motivação para utilização do JMF como padrão para futuros desenvolvimentos de arquiteturas relacionadas à transmissão de áudio e vídeo pela Internet. As Figuras 13 e 14 mostram o *hardware*/experimento que foram construídos para o projeto Tidia-Ae, o qual utilizou o TMF como recurso para a realização do experimento.

Figura 13: Foto do Experimento Remoto Automatizado – Gaiola de Skinner

Figura 14: Gaiola de Skinner, experimento remoto USP/UFSCAR

3.7.2 Transmissão de A/V e Integração

Realizou-se uma implementação da arquitetura S-MOJOHON e que foi utilizada por uma das ferramentas do ambiente TIDIA-Ae, Comunicador Instantâneo (CI) (Lobato et al., 2005) e (Lobato, et al., 2006). Já em 2006/2007 realizou-se a integração do Comunicador Instantâneo do TIDIA-Ae com a implementação do S-MOJOHON. Nesta integração removeu-se a utilização da TMF e adicionou-se o S-MOJOHON. A partir desse momento

pôde-se realizar comunicação entre clientes através de diferentes redes, com ou sem a presença de NAT/*Firewall*.

O CI então foi utilizado para testes e avaliações, sendo que um dos testes permitiu à ferramenta CI a transferência de fluxos através de clientes NAT, validando assim um dos objetivos da implementação de um conjunto de componentes.

Os testes foram realizados utilizando um monitor de *software* para capturar o tempo (*delay*) entre os *frames* capturados e os recebidos após a transmissão.

Um dos testes, realizado com transmissão de áudio e vídeo em rede local de 100 Mb, permitiu a transmissão e recepção dos fluxos com um atraso máximo de 0,172 segundo, sendo um tempo um pouco maior sem a utilização do sistema S-MOJOHON, a qual foi de 0,142 segundo, validando assim um outro objetivo da implementação do conjunto de componentes.

Com estes testes, conseguiu-se validar a arquitetura e perceber a qualidade e velocidade dos fluxos transmitidos, concluindo que a arquitetura atinge alguns dos objetivos desejados, ou seja, velocidade compatível com o JMF e habilidade de realizar conexões entre máquinas dentro de NAT.

Em uma outra avaliação, mais confiável e mais controlada, com máquinas iguais e numa rede local de 100 Mbps, como mostra a Tabela 3 os resultados são distintos em 2 configurações diferentes.

Na primeira configuração sem componentes S-MOJOHON, utilizando apenas o JMF API onde os dados são entregues de forma direta, ou seja, algo muito próximo ao ideal em relação à transmissão dos dados.

Na segunda configuração houve a utilização de apenas um módulo da arquitetura S-MOJOHON, o módulo refletor, sendo assim os dados foram transmitidos entre os clientes, ainda utilizando protocolo UDP.

Na terceira configuração, além do refletor, houve a adição do módulo de adaptação. Nesta configuração, os protocolos utilizados foram UDP e TCP. Os resultados foram:

Tabela 2: Resultado da Avaliação da implementação parcial do conjunto de componentes (Lobato et al., 2007).

Configuração	Média (ms)	Variância(ms)	1.º Quartil(ms)	Mediana(ms)	3.º Quartil(ms)
1 (JMF)	76, 00	15, 06	67, 50	75, 00	90, 00
2 (JMF + Refletor)	75, 20	14, 47	65, 00	80, 00	80, 00
3 (JMF + Refletor + Adaptador)	76, 40	11, 14	70, 00	80, 00	80, 00

Observando os resultados da Tabela 2 e após a execução do teste de *one-way ANOVA* ²⁴, o qual obteve o valor $p=0,949$, considerando $\alpha=0,05$, pôde-se constatar que os resultados não são diferentes, ou seja, usar a arquitetura S-MOJOHON (Lobato, et al., 2007) não implica em atrasos significativos para o cliente que está recebendo o fluxo de vídeo, comparada com a não utilização do sistema. Isto prova que o maior atraso não ocorre devido ao módulo refletor e nem aos canais de comunicação adicionados a ele, mas sim à captura da imagem, a qual é realizada pela JMF API.

Ainda na fase de integração, levantaram-se algumas sugestões para a melhoria da implementação do S-MOJOHON, como, por exemplo, a utilização de outras ferramentas para captura e apresentação de áudio e vídeo que não fosse o JMF, além de uma mudança do serviço de comunicação, para algo P2P e não centralizado como é realizado no momento. As sugestões surgiram devido à dificuldade de escalabilidade do serviço de comunicação centralizado. A Figura 15 mostra o CI em funcionamento com outro usuário realizando a transmissão de A/V através da arquitetura S-MOJOHON dentro da plataforma do Tidia-AE utilizando a Internet convencional com um dos clientes dentro de uma rede com a presença de NAT.

²⁴ *Analysis of Variance – One Way*. Disponível em: <http://www.uwsp.edu/psych/stat/12/anova-1w.htm> Último acesso dia 27 de fevereiro de 2008.

Figura 14: Ambiente TIDIA-Ae utilizando comunicador instantâneo com a arquitetura S-MOJOHON

3.7.3 Whiteboard com S-MOJOHON e TIDIA-Ae

Em 2007 houve uma integração da arquitetura S-MOJOHON com a ferramenta Whiteboard, realizada com o intuito de transmitir áudio para vários alunos, ou seja, realizar *multicast* para um grupo de alunos. Diferente da integração do CI, a integração com a ferramenta Whiteboard envolveu uma adaptação do código da implementação do S-MOJOHON.

Essa integração ainda está em desenvolvimento e em testes para melhorias, como, por exemplo, armazenamento e recuperação das informações passadas por áudio numa sessão de aula. Toda a integração entre a ferramenta Whiteboard e arquitetura S-MOJOHON foi realizada dentro do contexto do projeto TIDIA-Ae pelo grupo USP-Ribeirão Preto, trabalho realizado por Tatiana Bolão e Hugo Peste.

3.8 Considerações Finais

Este capítulo apresentou detalhes sobre a arquitetura S-MOJOHON, sua primeira implementação e os resultados obtidos a partir da mesma, sendo esta a primeira principal contribuição reportada nesta dissertação. O Capítulo 5 apresenta a outra principal contribuição do autor, em termos de investigação e algoritmos de seleção, e os resultados obtidos.

4. ALGORITMOS DE SELEÇÃO - AVALIAÇÕES

4.1 Considerações Iniciais

Este capítulo fornece dados sobre algumas avaliações realizadas sobre as implementações dos algoritmos de seleção com o objetivo de obter dados e quantificar a qualidade das implementações, até mesmo os problemas para futuras tomadas de decisões na elaboração de novas arquiteturas e/ou implementações, sendo este capítulo a grande maior contribuição do autor.

4.2 Algoritmo de Seleção Paralelo

Nesta seção são discutos os resultados obtidos a partir das avaliações do algoritmo de seleção paralelo. Os resultados são apresentados em tabelas comparativas, mostrando as diferenças, vantagens e desvantagens observadas através dos experimentos.

Avaliação em relação à seleção de refletores utilizando o algoritmo paralelo e tempo de resposta das mensagens

A Tabela 3 apresenta a relação de tempo de configuração entre a MaqU e os dois refletores, R181 e R211. A Tabela 4 apresenta a relação de tempo de configuração entre a MaqC e os dois refletores, R211 e R181.

- Os valores de Média apresentados nas tabelas indicam o tempo que um módulo de transmissão leva para enviar e receber a mensagem de configuração para o módulo de monitoramento/configuração. Após receber a mensagem de volta o refletor já está configurado.
- R211 e R181 são módulos de monitoramento/configuração e reflexão sem nenhuma carga, esperando mensagens de configurações.

- MaqU: Módulo de transmissão, simulando um cliente solicitando refletores para transferência de fluxos, máquina localizada próxima fisicamente e topologicamente a servidor R211, as duas máquinas pertencem à mesma rede de 100Mbps.
- MaqC: Módulo de transmissão, simulando um cliente solicitando refletores para transferência de fluxos, máquina localizada próxima fisicamente e topologicamente a servidor R211, máquinas em redes diferentes com velocidades de 100 Mbps ou superior, sujeita a tráfegos de informações aleatórios.

Tabela 3: Medida de tempo de configuração entre o módulo de transmissão e módulo de monitoramento/configuração em segundos.

MaqU	Média (s)	Desvio Padrão(s)	Variância(s)
R181 (12 saltos)	9,80	6,5	42,79
R211 (1 saltos)	0,69	0,04	0,002

Tabela 4: Medida de tempo de configuração entre o módulo de transmissão e módulo de monitoramento/configuração em segundos.

MaqC	Média(s)	Desvio Padrão(s)	Variância(s)
R181 (1 salto)	0,62	0,28	0,08
R211 (12 saltos)	13,03	7,55	57,05

A Figura 16 apresenta as possibilidades dos vários componentes de reflexão, que podem ser selecionados pelos clientes, e destaca a seleção elaborada pelo algoritmo paralelo no caso do refletor D responder à solicitação mais rapidamente que os outros.

Paralelo: Refletor D responde mais rápido (está mais perto ou menos sobrecarregado) ao PC-Y. Exemplo do tempo de seleção: 1 segundo

Figura 15: Clientes e refletor – Algoritmo Paralelo

Pode-se perceber na Tabela 3, que a relação entre a MaqU e o R211 possui uma média de configuração (0,69), desvio padrão e variância com valores baixos, podendo esta ser considerada a configuração ideal em termos de uma rede de alta velocidade. Isto ocorre, pois as máquinas encontram-se na mesma rede, sem interferências de informações externas.

Na Tabela 4, a relação entre MaqC e R181 possui valores baixos (0, 62), próximos aos considerados ideais. Isto ocorre devido à proximidade das máquinas e devido à baixa interferência de informações externas entre os módulos.

Já as relações entre a Tabela 3, MaqU, R181(9, 8), e a Tabela 4 MaqC, R211(13) apresentam média, desvio padrão e variância altos, pois as máquinas estão distantes de forma topológica (12 a 14 saltos respectivamente), conectadas via Internet, sujeitas a vários fluxos externos e de forma totalmente aleatória, causando assim o aumento do tempo de configuração entre clientes (módulos de transmissão) e *midlleware* (módulos de monitoramento/configuração e reflexão). Estas relações evidenciam que os tempos de configurações dos canais de transmissão de fluxos variam em

função da rede e da distância topológica, sendo assim estas métricas podem ser utilizadas para selecionar o refletor mais próximo ao cliente.

Ainda neste sentido, se imaginarmos uma situação onde o algoritmo paralelo é utilizado para seleção de servidores (refletores), as respostas mais rápidas seriam dos refletores que estão mais próximos aos clientes ou menos ocupados, provando assim o bom funcionamento do algoritmo paralelo para a seleção de refletores que possam atender ao cliente. A relação de proximidade pode ser o maior destaque, porém existem outras variáveis como, por exemplo, configuração da máquina na qual o serviço está disponibilizado, sistema operacional, processos em funcionamento no servidor, etc.

4.3 Algoritmo de seleção ontológico

O algoritmo de seleção ontológico é todo baseado em regras, motor de inferência e possui como base de dados uma ontologia, a qual foi apresentada na descrição da arquitetura S-MOJOHON. Este algoritmo infere uma base de dados através de regras, o qual gera resultados de relações, objetos e atributos de acordo com a descrição das regras.

Nesta seção são discutidos os resultados obtidos a partir das avaliações do algoritmo de seleção ontológico, os resultados são apresentados em tabelas comparativas, mostrando as diferenças, vantagens e desvantagens observadas através dos experimentos.

A Tabela 5 mostra a relação de tempo de configuração entre a máquina que solicita o serviço e a resposta das possíveis máquinas que podem prestar o serviço.

Tabela 5: Avaliação da utilização de arquivos RDF com todos os Sistemas Autônomos e regras utilizando JENA em segundos

	TC (s)	TAR (s)	TPBC (s)	TPC (s)	TRC (s)
Média	5,36	0,10	286,25	0,054	0,016
Variância	0,008	0,0001	115,63	0,008	0
Desvio Padrão	0,083	0,007	10,20	0,085	0,004

Onde:

- TC: Tempo de carregamento da base de conhecimento na memória em segundos.
- TAR: Tempo de adição de regras à base de conhecimento em segundos.
- TPBC: Tempo de preparo da base de conhecimento juntamente com as regras (pré-inferência) em segundos
- TPC: Tempo de consulta à base de conhecimento em segundos.
- TRC: Tempo de re-consulta à base de conhecimento após adição de novos refletores em segundos.

A avaliação foi realizada utilizando a seguinte regra:

rules="[r1: (?a eg:hasViz ?b), (?b eg:vizID ?c), (?c eg:hasRefletor ?d)->(?a eg:hasRefletor ?d)]"

e o motor de inferência *GenericRuleReasoner* da API JENA. As inferências foram realizadas numa base conhecimento de aproximadamente 5MBytes a qual possui toda a descrição de sistemas autônomos, vizinhança e refletores.

Figura 16: Etapas do processamento dos dados para obter o refletor desejado a partir de uma base de conhecimento.

Para seleção de um refletor utilizando o algoritmo ontológico serão necessários os seguintes tempos: troca de mensagens entre os módulos de transmissão e monitoramento e os tempos das tabelas acima, podendo ser das seguintes maneiras:

- Tempo da primeira seleção composto por: TC+TAR+TPBC+TPC. Este tempo é o tempo total que se leva entre carregar o modelo, realizar a inferência e obter os refletores disponíveis para o AS

em questão. Tempo da primeira seleção é aproximadamente 5 minutos.

- Tempo de reconsulta: TRC, considerando que todas as etapas anteriores já foram realizadas é de aproximadamente 20 milissegundos.

Outra forma de realizar a consulta sem a preparação da base de conhecimento também pode ser feita, dessa forma a consulta seria realizada da seguinte maneira: TC+TAR+TPC. Isso pode gerar um tempo maior que o tempo de primeira seleção, pois dessa forma não ocorre a preparação na base de conhecimento. O tempo total deste tipo de seleção é de aproximadamente 9 minutos. A Figura 18 mostra o Refletor C selecionado devido à política estabelecida.

Figura 17: Clientes e Refletores – Algoritmo Ontológico

O maior tempo reportado na Tabela 5 é o tempo de preparo da base de conhecimento. Esse tempo de preparo é um tipo de pré-inferência, após isso os tempos de consulta (54ms) e re-consulta (16ms) são baixos e aceitáveis. Os tempos variam em função da quantidade de parâmetros nas regras, em

função da quantidade de inferência necessária na base de conhecimento. Quanto maior as inferências a serem feitas, maior o tempo de consulta e re-consulta. A adição ou remoção dos dados após a primeira consulta no modelo inferido fica em média 2 segundos, dependendo da informação adicionada.

A regra apresentada como *r1* é uma regra que adiciona refletores a todos os vizinhos de Sistemas autônomos que possuem refletores. Caso exista um grafo conexo de sistemas autônomos, todos os SAs possuirão um ou mais refletores. Este tipo de seleção de refletores é lento (5 minutos) em relação à seleção por algoritmo paralelo (1 a 13 segundos), porém pode-se adicionar mais refinamento ao tipo de pesquisa. A seleção de refletores com regras diferentes pode criar novos tipos de seleção e possuir até mesmo variação nos tempos apresentados, porém se ganha em qualidade da seleção.

4.4 Algoritmo Híbrido

O algoritmo híbrido foi desenvolvido pelo autor com a finalidade de juntar as vantagens do algoritmo semântico com a velocidade do algoritmo paralelo.

O algoritmo híbrido ainda utiliza regras, Jena, sistemas autônomos e arquivos RDF para a base de dados, porém a grande diferença encontra-se na regra utilizada e na busca dos refletores mais próximos ao cliente.

Com a realização dos experimentos e avaliações notou-se que a busca em grafo de todos os elementos pode ser um limitante da solução, ou seja, um “gargalo” no algoritmo ontológico. Sendo assim, decidiu-se criar um algoritmo híbrido, o qual realiza busca em grafo dos melhores refletores para o melhor AS realizada através de algoritmos conhecidos de varredura em grafo, como por exemplo, algoritmos recursivos e o algoritmo de Dijkstra (1959). Dessa forma, percorrer o grafo torna-se rápido e eficiente, pois além do grafo ser percorrido e encontrar os possíveis refletores para um AS, o algoritmo utilizado não espalha os efeitos para os outros ASs, sendo assim necessitam-se de novas consultas para novos sistemas, diferentemente do algoritmo apresentado anteriormente.

O algoritmo híbrido é mais adequado para ambientes dinâmicos, já o baseado puramente em regras é mais adequado para ambientes mais estáticos. Realizando então a varredura em grafo por um algoritmo eficiente que retorna apenas os refletores que são possíveis para determinado AS, faz-se necessária a seleção do melhor refletor, ou seja, utiliza-se as regras para selecionar atributos dos refletores em questão, sendo que o caminho já foi fornecido pelo algoritmo anterior. Desta forma, obtém-se os refletores de menores caminhos e os “melhores” refletores selecionados pelas regras em JENA.

A Tabela 6 detalha que, com este algoritmo, obteve-se o tempo total de seleção em no máximo 12 segundos, mesmo variando a quantidade de refletores e suas localizações. Este tempo de 12 segundos não leva em consideração a troca de mensagens entre os módulos, adicionando o tempo total de troca de mensagens, pode-se obter o tempo de aproximadamente 25 segundos.

Tabela 6: Resultado da avaliação utilizando algoritmo híbrido de seleção de componentes. Tempo dado em segundos.

	TC(s)	TI(s)	TAR(s)	TPC(s)	TT(s)
MEDIA	4, 08	8, 60	0, 30	0, 12	13, 13
Variância	0, 00	0, 05	0, 04	0, 01	0, 02
Desvio Padrão	0, 06	0, 21	0, 20	0, 08	0, 12

Onde:

TC: Tempo de carregamento da base de conhecimento na memória em segundos.

TI: Tempo de inferência do algoritmo de busca em grafo.

TAR: Tempo de adição de regras à base de conhecimento em segundos.

TPC: Tempo de consulta à base de conhecimento em segundos.

TT: Tempo total de consulta à base de conhecimentos menos troca de mensagens

A Figura 19 mostra a seleção do componente refletor G devido à proximidade, o fator de carga baixo e também a política.

Figura 18: Clientes e Refletores – Algoritmo Híbrido.

Com base nos dados da Tabela 6, pode-se perceber que o tempo de carregamento da base de conhecimentos (TC) para memória ainda é alto, contribuindo dessa forma para o aumento do tempo total de inferência e consulta. Outro valor alto é o tempo de inferência do algoritmo de varredura em grafo, que varia em função da distância percorrida e quantidade de refletores presentes, pode-se em média adotar este valor de 12 segundos.

Existem tentativas de melhorias na implementação do algoritmo no intuito de diminuir o tempo para consulta. Contudo, pode-se perceber que o tempo total é em média 25 segundos, ainda alto, porém bem menor que o tempo de 5 minutos, e esse tempo nos fornece uma seletividade baseada em distância e outras características e propriedades encontradas na descrição dos refletores.

4.5 Discussão

A utilização do algoritmo paralelo para a seleção de refletores tem as seguintes vantagens:

- Fácil implementação, devido à utilização do serviço de comunicação na arquitetura S-MOJOHON.
- Resposta em função da distância, onde o tempo de resposta dos módulos de monitoramento/configuração para o módulo de transmissão varia em função da distância entre eles, sendo assim o módulo que primeiro responder é o módulo mais próximo dos clientes, possuindo o tempo de transmissão mais baixo possível para o ambiente.
- Resposta em função da rede, sistema operacional, *hardware*, processos em andamento, entre outros. Desta forma, o *broker* selecionado será o menos carregado, o que possui mais processamento livre, memória, menor latência entre o cliente, etc.

O algoritmo paralelo possui também desvantagens tais como:

- Seleção de refletores de forma limitada, ou seja, com este algoritmo é possível selecionar apenas refletores mais próximos, mas pode ser que este não seja o mais adequado para o cliente. Com algoritmo paralelo fica impossível selecionar refletores com outras características como, por exemplo, tempo em que o servidor está no ar, ou último servidor selecionado, ou servidor com o maior tempo em funcionamento e outras características desejadas, ou seja, o melhor servidor depende das métricas escolhidas.

A utilização do S-MOJOHON e regras Jena para a seleção de refletores têm as seguintes vantagens:

- Poder de seleção com diferentes métricas, pois a seleção de refletores baseados em regras e propriedades dos refletores descritas na ontologia permite a escolha do refletor baseada em diferentes métricas, como, por exemplo, seleção do refletor com mais tempo em

funcionamento, do refletor mais próximo, do refletor com outras qualidades que podem estar descritas na ontologia.

- Outra vantagem da utilização de ontologias e regras é a possibilidade de mudança de regras e a utilização dos dados e metadados para novas maneiras de seleção dos refletores.

Quanto às desvantagens:

- A implementação do sistema, da ontologia, das regras, armazenamento e consulta são mais trabalhosas e complicadas em relação ao algoritmo paralelo.
- Os tempos de consulta dos possíveis refletores para o sistema autônomo desejado variam em função do tamanho da base de dados, das regras e do motor de inferência.

Devido às vantagens citadas anteriormente, o algoritmo paralelo é fácil de programar e realiza a seleção de refletores com boas qualidades.

As vantagens da utilização de Jena, regras e ontologia são as de liberdade de escolha de métricas e parâmetros para diferentes tipos de seleção de refletores, porém existe um custo de implementação que pode não compensar dependendo da quantidade de dados e refletores que serão utilizados. Ganha-se em liberdade de escolha e perde-se no custo da implementação. Ambos podem selecionar bons refletores buscando baixo *delay*, porém nenhum deles vai garantir que isso ocorra, pois as variáveis estão em função da rede na qual os clientes e os refletores estão localizados.

A utilização do algoritmo híbrido, constituído de seleção baseada em grafos e baseado em regras, tem como vantagens:

- O tempo de seleção de refletores, ou seja, os possíveis refletores com as distâncias são fornecidos num tempo menor comparado ao algoritmo puramente baseado em regras.

- Facilidade na implementação, pois o algoritmo de busca em grafo é de conhecimento de todos e permite atuação do conjunto de regras num conjunto menor de ASs e de propriedades.
- Poder de seleção com diferentes métricas, pois a seleção de refletores baseados em regras e propriedades dos refletores descritas na ontologia permite a escolha do refletor baseada em diferentes métricas, como, por exemplo, seleção do refletor com mais tempo em funcionamento, do refletor mais próximo, do refletor com outras qualidades que podem estar descritas na ontologia.
- Outra vantagem da utilização de ontologias e regras é a possibilidade de mudança de regras e a utilização dos dados e metadados para novas maneiras de seleção dos refletores.

Em termos de desvantagens, deve-se considerar que, caso haja necessidade de realizar a seleção com dados e propriedades pertencentes à ligação/relação entre ASs, pois hoje o algoritmo realiza a seleção observando somente a propriedade distância, caso haja outras propriedades na relação *hasViz* haverá a necessidade de mudanças no algoritmo.

4.6 Considerações Finais

Este capítulo forneceu dados de avaliação sobre o serviço de comunicação utilizado nas implementações, mostrando vantagens e desvantagens entre eles, sendo esta a maior contribuição do autor. O próximo capítulo sumariza as conclusões e discute trabalhos futuros.

5. CONCLUSÕES

O projeto e a implementação da arquitetura S-MOJOHON, bem como modificações necessárias para os diferentes algoritmos de seleção, exploram vários tópicos de pesquisa, como, por exemplo, técnicas de realizar conexões em redes que implementam NAT, protocolos de áudio e vídeo, comunicação, avaliação e algoritmos de seleção. Isto ilustra o fato de que, para implementação e avaliação de uma arquitetura, é necessário conhecer esses tópicos e explorar detalhes de cada um e de todos ao mesmo tempo.

Além de explorar as diferentes áreas e tópicos de pesquisa, é necessária coordenação para que o projeto final atenda necessidades e objetivos definidos inicialmente. Uma das maneiras de analisar o funcionamento desse tipo de arquitetura é avaliá-la e testá-la em funcionamento. Neste sentido, o trabalho reportado atende aos requisitos de avaliação e teste de funcionamento.

Em relação aos objetivos e conclusões de pontos específicos, o trabalho, o qual envolve implementações e avaliações da arquitetura e seus componentes, apresenta pontos fortes e fracos. Entre os pontos fortes pode-se destacar a modularidade da arquitetura, a exploração de várias áreas de conhecimento, as decisões tomadas sobre os problemas a serem superados, a comparação e equivalência com outras arquiteturas, a exploração e os resultados obtidos em relação aos algoritmos de seleção de componentes.

Ainda em termos dos algoritmos investigados, o trabalho mostra que existe uma grande área de avaliação e pesquisa sobre algoritmos de seleção de componentes, assim como a exploração de web semântica, utilização de regras e políticas para tomada de decisões, e que nem sempre um único algoritmo traz grandes vantagens para solução de problemas. Pode ser necessária a mescla entre pontos fortes e fracos de determinados algoritmos para se obter bons resultados. Cada caso tem sua característica especial, e nem sempre determinados algoritmos ou tecnologias obterão resultados significativos.

Considerando os trabalhos de extensão e de avaliação da arquitetura S-MOJOHON, obteve-se um bom resultado em algumas decisões realizadas durante a pesquisa. Por outro lado, decisões de utilização de algumas tecnologias como JMF, serviço de comunicação centralizado e a possibilidade de incluir pesquisas sobre IPV6 tornam o sistema não tão robusto, difícil de escalar e não excelente para todos os casos, como se vê em produtos comerciais.

De modo geral, os objetivos propostos inicialmente para o trabalho foram atingidos. Os sistemas projetados e implementados serviram de base para investigações importantes que foram reportadas em publicações após validadas em ferramentas do projeto TIDIA-Ae.

O trabalho realizado abre caminho para trabalhos futuros que incluem melhorar as implementações atuais e também novas pesquisas. Destacam-se pesquisas futuras para diminuir os tempos de consulta da implementação S-MOJOHON e a avaliação de novas implementações com diferentes máquinas de inferência.

Outra linha de pesquisa resulta da substituição do serviço de comunicação centralizado (Pimentel. et al., 2005) por um sistema de comunicação P2P. Com um novo sistema de comunicação P2P, pode-se obter escalabilidade, fazendo com que vários refletores e clientes possam se comunicar. Essa pesquisa já está sendo realizada no contexto do projeto TIDIA-Ae fase II, por um aluno de iniciação científica e outro de doutorado.

Faz-se oportuna também a avaliação de uma nova implementação da arquitetura S-MOJOHON que utilize conceitos de OWL-S para seleção de *middleware*.

Visando ainda os contextos de serviços dos Projetos TIDIA-Ae e Kyatera, em particular em termos de ferramentas que explorem redes de baixa latência, pode-se explorar outros sistemas para captura de fluxos de áudio e vídeo, sistemas que impliquem em atrasos menores que os obtidos por JMF. O objetivo seria obter latências ainda mais baixas com alta qualidade de fluxos de áudio e vídeo.

Uma nova linha de pesquisa pode ser investigada no sentido de oferecer o módulo de reflexão como um serviço Web, ou até mesmo utilizar

OWL-S. OWL-S é uma ontologia para descrição de serviços, suas utilidades e funcionamento, objetivando descoberta e composição automática de serviços. Neste contexto seria necessário especificar propriedades *Profile*, *Model* e *Grounding*. Alguns trabalhos que podem servir de base para a pesquisa são: (Maximilien and Singh, 2004), (Garcia and Toledo, 2006), (Davies et al., 2004), (Srinivasan et al., 2006). Tal abordagem permitiria encontrar o componente refletor que melhor se encaixa com os parâmetros passados pelos clientes.

Expansão e modificações na ontologia proposta pelo S-MOJOHON, adicionando possibilidades de novas regras, com isso, cria-se um sistema com mais informações de refletores e clientes, com o objetivo de obter o servidor mais adequado no menor tempo e com a maior qualidade possível.

Publicações realizadas no contexto deste trabalho

LOBATO, D. C.; **BARBOSA, M. Q.**; PIMENTEL, M. G.; TEIXEIRA, C. A. C. - ACM SAC 2007) MOJOHON: A Channel-driven Communication Architecture for Applications Deployed on the Internet. In: THE 22ND ANNUAL ACM SYMPOSIUM ON APPLIED COMPUTING, 2007, Seoul, Korea. Proceedings of the 22nd ACM Symposium of Applied Computing. EUA : ACM Press, 2007. v. 2. p. 1061-1066

LOBATO, D. C.; BULCÃO NETO, R.; **BARBOSA, M. Q.**; PIMENTEL, M. G. C.; TEIXEIRA, C. A. C. S-MOJOHON: Towards a semantic architecture for message exchanging. In: IEEE International Conference on Semantic Computing, 2007, Irvine, CA. Proceedings of the 1st IEEE International Conference on Semantic Computing. Los Alamitos. CA : IEEE CS Press, 2007. P. 759-766.

LOBATO, D. C.; BULCÃO NETO, R.; **BARBOSA, M. Q.**; PIMENTEL, M. G. C.; TEIXEIRA, C. A. C. Exploiting Semantic Information on a Message Exchanging Middleware. In: In: THE 23RD ANNUAL ACM SYMPOSIUM ON APPLIED COMPUTING, 2008, Fortaleza, Brasil. Proceedings of the 23rd ACM Symposium of Applied Computing. EUA : ACM Press, 2008 (poster)

Outras publicações

LOBATO, D. C.; **BARBOSA, M. Q.**; PIMENTEL, M. G.; TEIXEIRA, C. A. C. A Channel-driven Communication Architecture for Applications Deployed on the Internet. In: WORKSHOP TIDIA FAPESP, 3. Proceedings of the 3rd Workshop TIDIA FAPESP, 2006. São Paulo, SP: FAPESP, 2006. p. 109-111

REFERÊNCIAS

(Abelém & Staton, 2000), Comunicação Multiponto Multimídia: Problemas e Soluções, Disponível em: http://www.cultura.ufpa.br/abelem/files/mcast_mmidia.pdf, último acesso dia 25 de maio de 2008.

(Barbosa, 2005) Matheus Q. Barbosa, Relatório TIDIA - Disponível em: <http://TIDIA-ae.incubadora.fapesp.br/portal/publications/research%20assistants%20reports/Relatorio1Matheus.pdf#search=%22video%20TIDIA%22>. Último acesso dia 22 de Agosto de 2006.

(Baset & Schulzrinne, 2004) S. Baset and H. Schulzrinne. An analysis of the skype peer-to-peer Internet telephony protocol. Technical Report CUCS-039-04, Computer Science Department, Columbia University, New York, NY, Sep 2004. Disponível em: http://www.mcs.vuw.ac.nz/courses/COMP414/2007T1/assignments/ass3/precis_radu.pdf

(Bechhofer et al., 2004) S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language reference, W3C Recommendation. <http://www.w3.org/TR/owl-ref/>, 2004.

(Berners-Lee et al., 2001) T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. Scientific American, 284(5):35 {43, Mai, 2001}.

(Bischoff & Rohrig, 2004) A. Bischoff, C. Röhrig: Streaming Audio/Video and Multiuser Virtual Reality based Environment for Collaborative Remote Experimentation. Proceedings 21st ICDE World Conference on Open Learning and Distance Education, Hong Kong 2004.

(Bowman et al., 1994) C. Mic Bowman, Peter B. Danzig, Udi Manber, and Michael F. Schwartz. Scalable Internet resource discovery: Research problems and approaches. Communications of the ACM, 37(8): 98-107, August 1994.

(Bulut et al., 2002) Hasan Bulut, Geoffrey Fox, Shrideep Pallickara, Ahmet Uyar and Wenjun Wu, Integration of NaradaBrokering and Audio/Video Conferencing as a Web Service. Proceedings of IASTED International Conference on Communications, Internet, and Information Technology, 2002. Disponível em: <http://grids.ucs.indiana.edu/ptliupages/publications/AVOverNaradaBrokering.pdf>

(Capobianco et al., 2005) D. Capobianco, F. S. Santos, M. Barbosa, C. Teixeira. "Processo de Modelagem de Resposta: Refinando Requisitos de Software de Apoio a Laboratórios de Acesso Remoto". XVI Simpósio Brasileiro de Informática na Educação (SBIE), 2005. Juiz de Fora.

(Carpenter, 2003) B. E. Carpenter. Future applications and middleware, and their impact on the infrastructure. *Future Gener. Comput. Syst.*, 19(2):191–197, Feb 2003.

(Castro et al., 2002) M. Castro, P. Druschel, A.-M. Kermarrec, and A. I. T. Rowstron. Scribe: a large-scale and decentralized application-level *multicast* infrastructure. *IEEE J. Sel. Areas Commun.*, 20(8):1489–1499, 2002.

(Comer, 2000) Douglas E. Comer, *Internetworking With TCP/IP Volume 1: Principles Protocols, and Architecture*, 4th edition, 2000. Prentice Hall.

(Conceição et al., 2006) Arlindo F. da Conceição, Jin Li, Dinei A. Florêncio Transmissão de voz sobre redes IEEE 802.11: um levantamento dos principais problemas e restrições, Webmedia 2006. Disponível em: http://www.lcpd.ime.usp.br/~arlindo/articles/doutorado/webmedia06_VoWiFi.pdf

(Cruz & Moraes, 1997) Tecnologias de comunicação e Informação para o ensino a distância na integração Universidade / Empresa. Tecnologias de comunicação e Informação para o ensino a distância na integração Universidade. ISSN 0104-4141.

(Dabek et al., 2003) F. Dabek, B. Zhao, P. Druschel, J. Kubiawicz, and I. Stoica. Towards a common API for structured peer-to-peer overlays. In *IPTPS 2003: Proc. Second International Workshop on Peer-to-Peer Systems (LNCS 2735)*, pages 33–44. Springer-Verlag, 2003.

(Dabek et al., 2004) F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. In *SIGCOMM 2004: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 15-26, Portland, Oregon, EUA, Set 2004. ACM Press.

(Davies et. Al, 2004) NJ. Davies, D. Fensel, and M. Richardson. The Future of web services. *BT Technology Journal*, 22(1):76-82, January 2004.

(De-jen et al., 1998) LU De-jen, WANG Yu-chu, HO Jan-ming. Experience in designing a using TCP as transprot protocol VOD system over a dedicated network. *J. IEEE Transactions on Consumer Electronics*, 1998, 44(11), pages 1379-1388. Disponível em: <http://ieeexplore.ieee.org/iel4/30/15843/00735841.pdf?tp=&isnumber=&arnumber=735841>.

(Dijkstra, 1959) Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269-271.

(Ding et al., 2007) Ding, X., Erickson, T., Kellogg, W. A., Levy, S., Christensen, J. E., Sussman, J., Wolf, T. V., and Bennett, W. E. 2007. An empirical study of the use of visually enhanced voip audio conferencing: the case of IEAC. In *Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 1019-1028. DOI= <http://doi.acm.org/10.1145/1240624.1240780>

(Dunagan et al., 2004) J. Dunagan, N. J. A. Harvey, M. B. Jones, M. Theimer, and A. Wolman. Subscriber/volunteer trees: Polite, efficient overlay multicast trees. Technical Report MSR-TR 2004-131, Microsoft Research, Redmond, WA, EUA, Dec 2004. Disponível em: <http://research.microsoft.com/sn/Herald/papers/tr-2004-131.pdf>

(Egevang & Francis, 1994) K. Egevang and P. Francis. The IP network address translator (NAT). Request for Comments #1631, 1994. RFC. RFC Editor.

(Eppinger, 2005) Eppinger, J. L. TCP Connections for P2P Apps: A Software Approach to Solving the NAT Problem. Tech. Rep. CMU-ISRI-05-104, Carnegie Mellon University, Pittsburgh, PA, Jan. 2005. Disponível em: <http://reports-archive.adm.cs.cmu.edu/anon/anon/home/ftp/isri2005/CMU-ISRI-05-104.pdf>

(Flammia, 2001) G. Flammia, Peer-to-peer is not for everyone, Internet Services, IEEE Intelligent Systems May/June (2001) 78-79.

(Ford et al., 2005) Hole Punching Ford, B., Srisuresh, P., and Kegel, D. Peer-to-peer communication across network address translators. In Proceedings of the 2005 USENIX Annual Technical Conference (Anaheim, CA, Apr. 2005). Disponível em: http://www.usenix.org/publications/library/proceedings/usenix05/tech/general/full_papers/ford/ford_html/

(Ganjam & Zhang, 2004) Ganjam, A. and Zhang, H. 2004. Connectivity restrictions in overlay multicast. In Proceedings of the 14th international Workshop on Network and Operating Systems Support For Digital Audio and Video (Cork, Ireland, June 16 - 18, 2004). NOSSDAV '04. ACM, New York, NY, 54-59. DOI= <http://doi.acm.org/10.1145/1005847.1005860>

(Garcia and Toledo, 2006) Garcia, D. Z. and de Toledo, M. B. 2006. Semantics-enriched QoS policies for web service interactions. In Proceedings of the 12th Brazilian Symposium on Multimedia and the Web (Natal, Rio Grande do Norte, Brazil, November 19 - 22, 2006). WebMedia '06, vol. 192. ACM Press, New York, NY, 35-44.

(Gargolinski et al, 2005) Game Server Selection for Multiple Players Steven Gargolinski, Christopher St. Pierre, and Mark Claypool In Proceedings of the 4th ACM Network and System Support for Games (NetGames) Hawthorne, NY, USA October 2005

(Grange et al., 2000) Grange, S., T.W.Fong, and C. Baur (2000, April). Effective vehicle teleoperation on the world wide web. In IEEE International Conference on Robotics and Automation, pp. 2007-2012.

(Guerri et al., 2002) Juan C. Guerri, Carlos E. Palau, Ana Pajares, Manuel Esteve, "A Real-time E-learning System Via Satellite Based on JMF and Windows Media, ", presented at the 10th ACM Int. Conf. on Multimedia, Juan les Pins, France, 2002

(Guha et al., 2007) Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", Work in Progress, April 2007.

(Guha et al, 2006) GUHA, S., DASWANI, N., AND JAIN, R. An Experimental Study of the Skype Peer-to-Peer VoIP System. In P. Druschel, E. G. Sirer, and B. Y. Zhao, editors, IPTPS 2006: Proceedings of the 5th International Workshop on Peer-to-Peer Systems, page 6, Santa Barbara, CA, EUA, Feb 2006.

(Guha, 2004) S. STUNT - Simple Traversal of UDP Through NATs and TCP too. Network Working Group Internet-Draft, Expires: June 11, 2005. [online] <http://nutss.gforge.cis.cornell.edu/pub/draft-guha-STUNT-00.txt>
Último acesso dia 25 de Maio de 2008

(Gummadi et al., 2002) K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: estimating latency between arbitrary Internet end hosts. In IMW 2002: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment, pages 5 -18, Marseille, France, Nov 2002. ACM Press.

(Guo et al., 2002)M. Guo, M. Ammar, and E. Zegura, "Selecting among replicated batching video-on-demand servers, "Proceedings of the 12th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV 2002), Miami, Floria, USA, 2002, pp 155-163.

(Hlavacs & Buchinger, 2008) Hlavacs, H. and Buchinger, S. 2008. Hierarchical video patching with optimal server bandwidth. ACM Trans. Multimedia Comput. Commun. Appl. 4, 1 (Jan. 2008), 1-23. DOI=<http://doi.acm.org/10.1145/1324287.1324295>

(Hama et al., 2006) J. J. van der Hama, F. Dijkstra, F. Travostinob, H. M. A. Andreea, and C. T. A. M. de Laat. Using RDF to describe networks. Future Generation Computer Systems, 22(8):862-867, Out 2006.

(Hung et al., 2007) Hung, H., Jayagopi, D., Yeo, C., Friedland, G., Ba, S., Odobez, J., Ramchandran, K., Mirghafori, N., and Gatica-Perez, D. 2007. Using audio and video features to classify the most dominant person in a group meeting. In Proceedings of the 15th international Conference on Multimedia (Augsburg, Germany, September 25 - 29, 2007

(Huston, 2006) G. Huston. Exploring autonomous system numbers. Internet Protocol J., 9(1):2-23, Mar 2006.

(Iana, 2002) Iana. Special-use IPv4 addresses. Request for Comments #3330, 2002.

(Jardim et al., 2005) Jardim, C. H.; Martelini JR, A.; Freire, J.; Silva, E. Q.; Lara, S. S. A.; Santos, F. S.; Kudo, T. N.; Fortes, R. P. M.; Pimentel, M. G. C.. Whiteboard: uma Ferramenta de Apoio ao Ensino e Aprendizado com Uso de Anotação Eletrônica. In: XVI Simpósio Brasileiro de Informática na Educação, 2005, Juiz de Fora MG, Brasil.

(Kamolphiwong et al., 2002) T. Kamolphiwong, S. Kamolphiwong, and C. Siriuenyong, A Design Framework of Interactive Distance Learning in Distributed Systems, Proceedings of the IEEE International Conference on Computers in Education (ICCE'02), 2002.

(Karp et al., 2004) B. Karp, S. Ratnasamy, S. Rhea, and S. Shenker. Spurring adoption of DHTs with OpenHash, a public DHT service. In IPTPS 2004: Proc. 3rd International Workshop on Peer-to-Peer Systems (LNCS 3279), pages 195–205. Springer-Verlag, 2004.

(Klyne & Carroll, 2004) G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and abstract syntax, W3C Recommendation. <http://www.w3.org/TR/rdf-concepts/>, 2004.

(Kostić et al., 2008) Kostić, D., Snoeren, A. C., Vahdat, A., Braud, R., Killian, C., Anderson, J. W., Albrecht, J., Rodriguez, A., and Vandekieft, E. 2008. High-bandwidth data dissemination for large-scale distributed systems. ACM Trans. Comput. Syst. 26, 1 (Feb. 2008), 1-61. DOI=<http://doi.acm.org/10.1145/1328671.1328674>

(Kurose & Ross, 2005), James F. Kurose e Keith W. Ross, Redes de computadores e a Internet, Uma abordagem top-down, 2005; Edição: 3, Editora: Pearson.

(Li & Sollins, 2004) J. Li and K. Sollins. Exploiting autonomous system information in structured peer-to-peer networks. In ICCCN 2004: Proceedings of 13th International Conference on Computer Communications and Networks, pages 403-408, Chicago, IL, EUA, Out 2004. IEEE CS Press.

(Liu et al., 2008) Jiangchuan Liu Rao, S.G. Bo Li Hui Zhang, Simon Fraser Univ., Vancouver; Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast, Proceedings of the IEEE, Subject Category: General/Other (Interdisciplinary), Jan. 2008, Volume: 96, Issue: 1 On page(s): 11-24, ISSN: 0018-9219, Digital Object Identifier: 10.1109/JPROC.2007.909921

(Lobato et al., 2005) Lobato, Daniel Corrêa; Baldochi JR, Laércio Augusto; PIRES, Daniel Facciolo; Pessoa, Thiago Rufim de Moraes; GASPARIN, Renato; Montoro, Flávio Aldrovani; Teixeira, Cesar Augusto Camillo. A

Multimedia Instant Messenger for an e-Learning Environment. In: II Workshop TIDIA, 2005, São Paulo, SP.

(Lobato et al, 2006) Daniel C. Lobato, Mário B. Avelino, Matheus Q. Barbosa, Cesar A. C. Teixeira, Maria da Graça C. Pimentel, Audio and Video Streaming into the TIDIA-Ae Instant Messenger and Tools for Accessing Resources remotely, disponível em: <http://coweb.dc.ufscar.br/mostra.php?ident=4.6>, acessado em: 31 de outubro de 2006.

(Lobato et al, 2006a) Lobato, D. C.; Barbosa, M. Q.; Pimentel, M. G.; Teixeira, C. A. C. A Channel-driven Communication Architecture for Applications Deployed on the Internet. In: III Workshop TIDIA FAPESP, 2006, p. 109-111.

(Lobato et al., 2007) Lobato, D. C.; Barbosa, M. Q.; Pimentel, M. G.; Teixeira, C. A. C. MOJOHON: A Channel-driven Communication Architecture for Applications Deployed on the Internet. In: The 22nd Annual ACM Symposium on Applied Computing, 2007, Seoul, Korea. v. 2. p. 1061-1066

(Lobato et al., 2007a) Lobato, D. C.; Bulcão Neto, R.; Barbosa, M. Q.; Pimentel, M. G. C.; Teixeira, C. A. C. S-MOJOHON: Towards a semantic architecture for message exchanging. In: IEEE International Conference on Semantic Computing, 2007, Irvine, CA.

(Lobato et al., 2008) Lobato, D. C.; Bulcão Neto, R.; Barbosa, M. Q.; Pimentel, M. G. C.; Teixeira, C. A. C. Exploiting Semantic Information on a Message Exchanging Middleware. In: Proceedings of the 23rd ACM Symposium of Applied Computing. EUA : ACM Press, 2008. (poster)

(Martins, 2001) Luciano Martins, Transmissão de Vídeo usando IPV6 e Multicasting em Redes de Alto Desempenho. Dissertação de Mestrado apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC – USP – São Carlos, fevereiro de 2001.

(Maximilien and Singh, 2004) E. Michael Maximilien and Munindar P. Singh. Toward autonomic web services trust and selection. In: ICSOC'04: Proceedings of the 2nd international conference on Service oriented computing, pages 212-221, New York, NY, USA, 2004. ACM Press

(Mendona and Silva, 2005) N.C. Mendona and J.A.F. Silva, “An Empirical Evaluation of Client-side Server Selection Policies for Accessing Replicated Web Services”, Proceedings of ACM SAC, pp. 1704-1708, 2005.

(Muller, 1998) Nathan J. Muller, Improving and Managing Multimedia Performance Over TCP/IP Networks, International Journal Of Network Management, Volume 8, pp. 356-367, 1998.

(Natarajan et al., 2006) Preethi Natarajan, Janardhan R. Iyengar, Paul D. Amer, Randall Stewart, SCTP: An innovative transport layer protocol for the

web. Proceedings of the 15th international conference on World Wide Web table of contents. Pages: 615 - 624, 2006.

(Obrackza & Silva, 2000) K. Obraczka and F. Silva. Network latency metrics for server proximity. In GLOBECOM 2000: Proceedings of the 43th IEEE Global Telecommunications Conference 2000, volume 1, pages 421-427, São Francisco, CA, EUA, Dez 2000.

(Oki et al., 1993) B. Oki, M. Pfluegl, A. Siegel, and D. Skeen. The information bus: an architecture for extensible distributed systems. SIGOPS Oper. Syst. Rev., 27(5):58-68, 1993.

(Ostuni & Barlow, 2002) A. Terrazas, J. Ostuni, and M. Barlow. JAVA Media API: Cross-Platform Imaging, Media and Visualization. Sams, 2002.

(Pimentel et al., 2005) Pimentel, Maria da Graça Campos; Eleutério, Pedro Monteiro da Silva; Santos, Felipe Silva; Teixeira, Cesar Augusto Camillo. A Synchronous Communication Service API using J2EE and JMS. In: Simpósio Brasileiro de Sistemas Multimídia e Hiperemídia, 2005, Poços de Caldas. Proceedings: posters, 2005. v. 1. p. 1-3.

(Qazzaz, 2003) B. Qazzaz, R. Suppi, F. Cores, A. Ripoll, P. Hernandez, and E. Luque. Providing interactive video on demand services in distributed architecture. In 29th Euromicro Conference (EUROMICRO'03), page 215, September 2003.

(Reynolds & Postel, 1994) J. Reynolds and J. Postel. Assigned numbers. Request for Comments #1700, 1994.

(Ribeiro & Leung, 2005), Eduardo Parente Ribeiro, Victor C. M. Leung, Asymmetric Path Delay Optimization in Mobile Multi-homed SCTP Multimedia Transport. 1st ACM workshop on Wireless multimedia networking and performance modeling, Pages: 70 - 75, 2005.

(Rosenberg et al., 2008) Rosenberg, J., Mahy, R., and P. Matthews, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", Work in Progress, January 2008.

(Rowstron & Pastry, 2001) A. I. T. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In Proc. IFIP/ACM Middleware, LNCS 2218, pages 329-350, 2001.

(Salmito et al., 2004) T. Salmito, J. P. Farias, G. Elias, G. L. de Souza, and L. E. Leite. A distributed hierarchical architecture for a Video-on-Demand distribution service (in Portuguese). In Proc. WebMedia & LA-Web 2004 Joint Conference, pages 104-111, 2004.

(Sayal et al., 1998) M. Sayal, Y. Breibart, P. Scheuermann, and R. Vingralek, "Selection algorithm for replicated web servers, " presented at the Workshop Internet Server Performance (WISP/SIGMETRICS), Madison, WI, June 1998.

(Shankar and Schaar, 2007) Shankar, N.S.; van der Schaar, M., Performance Analysis of Video Transmission Over IEEE 802.11a/e WLANs, Vehicular Technology, IEEE Transactions on Volume 56, Issue 4, July 2007 Page(s):2346 - 2362 , DOI: 10.1109/TVT.2007.897646, acessado em: 26 de Maio de 2008, disponível em: <http://medianetlab.ee.ucla.edu/papers/029.pdf>

(Sharma et al., 2006) P. Sharma, Z. Xu, S. Banerjee, and S.-J. Lee. Estimating network proximity and latency. SIGCOMM Comput. Commun. Rev., 36(3):39-50, 2006.

(Smith & Jonsson, 2004) Ipv6, Smith, C. Jonsson, K. Peer to peer [Internet Protocol], Green Hills Software. Communications Engineer Publication, Volume: 2, Issue: 2, page(s): 24- 27.

(Srinivasan et. Al, 2006) N. Srinivasan, M. Paolucci, and K. Sycara. Semantic web service discovery in the owl-s ide. In Proceeding of the 39th Hawaii International Conference os System Sciences, 2006

(Srisuresh et al., 2008) P. Srisuresh, Kazeon Systems, B. Ford, D. Kegel, State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs), Network Working Group Request for Comments: 5128, Category: International, Março, 2008. Disponível em: <http://www.ietf.org/rfc/rfc5128.txt>, acessado em 2 de Junho de 2008.

(Terrazas et al., 2002) A. Terrazas, J. Ostuni, and M. Barlow. JAVA Media API: Cross-Platform Imaging, Media and Visualization. Sams, 2002.

(Tomos & Frossard, 2007) Thomos, N. and Frossard, P. 2007. Raptor network video coding. In Proceedings of the international Workshop on Workshop on Mobile Video (Augsburg, Bavaria, Germany, September 28 - 28, 2007). MV '07. ACM, New York, NY, 19-24. DOI= <http://doi.acm.org/10.1145/1290050.1290056>

(TIOBE, 2007) TIOBE Programming Community Index, Disponível em: <http://www.tiobe.com/tpci.htm>, ranking das linguagens de programação, acessado em 25 de Maio de 2008.

(Touch, 2002) J. D. Touch. Those pesky nats. IEEE Internet Computing, 6(4):96, Jul 2002.

(Tulu et al., 2003) Tulu, B. Samir Chatterjee Tarun Abhichandani Li, H. Secured video conferencing desktop client for telemedicine. In: Enterprise Networking and Computing in Healthcare Industry, 2003. Healthcom 2003.

(Tsuchiya et al., 2006) Tsuchiya, T. Yoshinaga, H. Koyanagi, K. Honda, A. Minami, A. STARCast: streaming collaboration platform using the overlay technology. IEEE SAINT Workshops 2006. 4 pages. DOI 10.1109/SAINT-W.2006.37

(Uyar & Fox, 2005) Ahmet Uyar and Geoffrey Fox. Investigating the Performance of Audio/Video Service Architecture I: Single Broker Proceedings of the IEEE International Symposium on Collaborative Technologies and Systems CTS 2005, pages 120-127.

(Uyar & Fox, 2005b) Ahmet Uyar and Geoffrey Fox. Investigating the Performance of Audio/Video Service Architecture II: Broker Network. Proceedings of the IEEE International Symposium on Collaborative Technologies and Systems CTS 2005 pages 128-135.

(Venkatachalam, 2006) G. Venkatachalam. Developing P2P protocols across NAT. Linux J., 2006(148):1-8, 2006.

(Vianna et al., 2006) Bruno de Azevedo Vianna, Nilmax Teones Moura, Célio Vinicius Neves de Albuquerque, Vinod E. F. Rebello, Cristina Boeres. adaMOS: Algoritmo MOS-Adaptativo para Fontes VoIP. Webmidia 2006.

(Wang et al., 2004) Tcp Connenctions B.Wang, J. Kurose, P. Shenoy, and D. Towsley. Multimedia streaming via TCP: An analytic performance study. In Proceedings of ACM Multimedia. October 2004.

(Wellman et al., 2001). Barry Wellman, Anabel Quan Haase, James Witte and Keith Hampton. Does the Internet Increase, Decrease, or Supplement Social Capital?. Research Bulletin #6. Disponível em: <http://www.urbancenter.utoronto.ca/pdfs/researchbulletins/06.pdf> Último acesso dia 25 de Maio de 2008.

(Zamora et al., 2007) Zamora, N. H., Kao, J., and Marculescu, R. 2007. Distributed power-management techniques for wireless network video systems. In Proceedings of the Conference on Design, Automation and Test in Europe (Nice, France, April 16 - 20, 2007). Design, Automation, and Test in Europe. EDA Consortium, San Jose, CA, 564-569.

(Zamora et al., 2008) Zamora, N. H., Hu, X., Ogras, U. Y., and Marculescu, R. 2008. Enabling multimedia using resource-constrained video processing techniques: A node-centric perspective. ACM Trans. Des. Autom. Electron. Syst. 13, 1 (Jan. 2008), 1-27. DOI= <http://doi.acm.org/10.1145/1297666.1297684>

(Zegura et al., 2000) E. W. Zegura, M. H. Ammar, Z. Fei, and S. Bhattacharjee. Application-Layer Anycasting: A Server Selection Architecture and Use in a Replicated Web Service. IEEE/ACM Trans. on Networking, 8(4):455--466, Aug. 2000.

(Zimmermann et al., 2004) R. Zimmermann, M. Pawar, D. A. Desai, M. Qin, and H. Zhu. High resolution live streaming with the HYDRA architecture. *Comput. Entertain.*, 2(4):16–30, 2004.