UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES

BRUNO WALLACE DE FREITAS FRANCA
EDILSON JORGE DOS SANTOS JUNIOR
RODRIGO FERREIRA BRITTO

ANÁLISE DE TRÁFEGO E SIMULAÇÃO DE REDES MULTICAST

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2013

BRUNO WALLACE DE FREITAS FRANCA
EDILSON JORGE DOS SANTOS JUNIOR
RODRIGO FERREIRA BRITTO

ANÁLISE DE TRÁFEGO E SIMULAÇÃO DE REDES MULTICAST

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Departamento Acadêmico de Eletrônica – DAELN – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Kleber Kendy Horikawa Nabas

CURITIBA
2013

BRUNO WALLACE DE FREITAS FRANCA

EDILSON JORGE DOS SANTOS JUNIOR

RODRIGO FERREIRA BRITTO

ANÁLISE DE TRÁFEGO E SIMULAÇÃO DE REDES MULTICAST

Este trabalho de conclusão de curso foi apresentado no dia 28 de Novembro de 2012, como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Telecomunicações, outorgado pela Universidade Tecnológica Federal do Paraná. O(s) aluno(s) foi(ram) arguído(s) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado

_____

Prof. Luís Carlos Vieira
Coordenador de Curso
Departamento Acadêmico de Eletrônica

_____

Prof. Sérgio Moribe
Responsável pela Atividade de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

_____

Prof. Me. Lincoln Herbert Teixeira

_____

Prof. Dr. Kleber Kendy Horikawa Nabas
Orientador

_____

Prof. Mauricio L. S. Ramos

A Folha de Aprovação assinada encontra-se na Coordenação do Curso

RESUMO

FRANCA, Bruno Wallace de Freitas; JUNIOR, Edilson Jorge dos Santos; BRITTO, Rodrigo Ferreira. **Análise de Tráfego e Simulação de Redes *Multicast.*** 2012. 89 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas de Telecomunicações, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

Este trabalho tem por objetivo descrever os principais conceitos sobre transmissão via IP *Multicast*. Além de um estudo detalhado sobre a história, vantagens e principais características da utilização do *Multicast*, foram propostos e desenvolvidos experimentos, a fim de avaliar na prática a otimização da utilização de banda numa rede local e WAN (*Wide Area Network*) usando roteamento *Multicast*. Há uma exposição de conceitos teóricos fundamentais para a implantação e a correta configuração de todos os segmentos da rede. Os experimentos foram montados com equipamentos de rede reais, onde foi possível a verificação na prática de todos os conceitos estudados na teoria.

Palavras-chaves: *Multicast*. IGMP. PIM.

ABSTRACT

FRANCA, Bruno Wallace de Freitas; JUNIOR, Edilson Jorge dos Santos; BRITTO, Rodrigo Ferreira. **Traffic Analysis and Simulation of Multicast Networks**. 2012. 89 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas de Telecomunicações, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

This paper aims to describe the main concepts of multicast transmission. In addition to a detailed study on the history, main features and advantages of the use of multicast, were proposed and developed test scenarios in order to assess in practice the optimization of bandwidth usage on a LAN(Local Area Network) and WAN(Wide Area Network) using multicast routing. There is an exposition of theoretical concepts fundamental to the correct deployment and configuration of all network's segments. The scenarios were set up using real network equipments, where it was possible to verify in practice all the concepts studied in theory.

Keywords: Multicast. IGMP. PIM.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE SIGLAS

ARP	Address Resolution Protocol
AS	Autonomous System
CPU	Central Processing Unit
DARPA	Defense Advanced Research Projects Agency
DIFFSERV	Differentiated Services
DR	Designatede *Router*
DVMRP	Distance Vector *Multicast* Routing Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
FTP	File Tranfer Protocol
HTTP	HyperText Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGMP	Internet *Group* Management Protocol
IGRP	Interior Gateway Routing Protocol
IP	Internet Protocol
IPTV	Internet Protocol Television
IPv6	Internet Protocol versão 6
LAN	Local Area Network
MAC	Media Access Control
Mbps	*Mega Bits per Second*
MOSPF	*Multicast* extensions to Open Shortest Path First
MPLS	Multi protocol Label Switching
MRTG	Multi *Router* Traffic Grapher
MVR	*Multicast* VLAN Registration
OSPF	Open Shortest Path First
PIM	Protocol Independent *Multicast*
PIM-DM	Protocol Independent *Multicast* - Dense Mode
PIM-SM	Protocol Independent *Multicast* - Sparse Mode
QoS	Quality of Service
RFC	Request for Comments
RIP	Routing Information Protocol
RP	Rendezvous Point
RPT	Rendezvous Point Tree
RSVP	Resource reSerVation Protocol
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SSM	Source-Specific *Multicast*
TCP	Protocol-Independent *Multicast*
TOS	Type of Service
TRPB	Truncated Reverse Path *Broadcast*ing

TTL	Time to Live
UDP	User Datagram Protocol
UNIX	Universal Interactive executive
VLAN	Virtual LAN
VLC	VideoLAN (software)
WAN	Wide Area Network

SUMÁRIO

1 INTRODUÇÃO

A maior parte do tráfego da Internet é através do *Unicast*, ou seja, um transmissor envia os pacotes de acordo com a quantidade de solicitações e destinos. Por muito tempo este tipo de transmissão pareceu ser suficiente para a Internet, porém com aumento da rede mundial de computadores, também aumentou o tráfego multimídia sob demanda. Assim a transmissão *unicast* tornou-se insuficiente, principalmente na internet. Com o *broadcast*, os pacotes são enviados para este endereço e assim todas as estações na mesma rede recebem, contudo o *broadcast* não utiliza a banda da maneira mais eficiente.

Com a necessidade de criar um método de transmissão que permitisse o roteamento para diversas redes e que somente as estações de interesse recebessem os pacotes enviados, foi criado o *Multicast*, que atende esta necessidade apresentando melhor desempenho nas transmissões multimídia.

O *Multicast* está entre o *Unicast* e o *Broadcast*, e tem como característica enviar os pacotes onde os destinatários são um grupo específico, e apenas este grupo recebe o tráfego. No geral as aplicações que trabalham com *multicast* são chamadas de "Aplicações *Multicast*" ou "aplicações Multiponto" (MARQUES; CARNEIRO, 2011).

1.1 PROBLEMA

As redes de computadores com o passar do tempo se tornaram cada vez mais amplas e complexas devido ao crescimento da Internet e de soluções possíveis para a mesma. A ocupação de largura de banda é um dos grandes problemas gerados devido a essa expansão.

Tradicionalmente, quando alguém precisava transmitir uma mesma informação para vários destinos, a fonte teria de emitir para a rede, uma cópia separada para cada destino. Se a fonte fosse um *stream* (fluxo) de vídeo, a enviar a mil utilizadores simultaneamente, isto significaria que o emissor teria que criar mil cópias desse mesmo stream e enviá-lo mil vezes para a rede. Isto pressupõe um enorme esforço por parte do servidor/emissor e uma utilização claramente ineficiente da rede.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Montar experimentos com equipamentos reais, que comprovem na prática a eficiência da utilização da tecnologia Multicast.

1.2.2 Objetivos Específicos

- Realizar estudo sobre os conceitos de *Multicast*;
- Avaliar vantagens e desvantagens da utilização do IP *(Internet Protocol) Multicast*;
- Definir equipamentos a serem utilizados nos experimentos;
- Instalar software ZABBIX;
- Pesquisa e instalação de firmware que suporte *Multicast* em roteadores;
- Analisar os protocolos de roteamento DVMRP, MOSPF e PIM;
- Instalar softwares que suportem tráfego *Multicast* (VLC – *VideoLAN*);

2 REFERENCIAL TEÓRICO

As aplicações multimídia têm, em sua maioria, a característica de possuir vários participantes simultâneos, ou seja, existem um emissor e vários receptores, ou ainda, vários emissores para vários receptores. Uma forma simples de implementar estes sistemas é manter uma conexão para cada par emissor/receptor. Fica óbvio, porém, a ineficiência do sistema, uma vez que cada mensagem a ser transmitida, deverá ser replicada pelo emissor para cada um de seus receptores. Com o crescimento da necessidade de uma forma de comunicação que atingisse vários destinatários de maneira eficiente, surgiu o conceito de *multicast*. Este pode ser explicado como: forma de comunicação onde, através de uma única operação de transmissão, resulta no envio simultâneo para vários destinatários.

2.1 MULTICAST EM REDES DE COMPUTADORES

Em redes ethernet existem 3 formas de entrega de uma mensagem a um destino: *unicast*, *multicast* e *broadcast*. No *unicast*, a comunicação é 1:1, ou seja, um endereço origem e um destino. No *multicast*, é 1:n, e no *broadcast* é 1:todos. O endereçamento *multicast* permite enviar pacotes IP para um grupo determinado de usuários que previamente se cadastraram neste grupo. Cada grupo é identificado por um IP classe D, ou seja, entre 224.0.0.0 até 239.255.255.255. Para entrar, sair e se manter em grupos *multicast*, as estações utilizam o protocolo IGMP *(Internet Group Management Protocol)*, que será analisado posteriormente (ROESLER, 2001).

2.1.1 Unicast

A Microsoft descreve o *Unicast* como o reencaminhamento de tráfego destinado a uma única localização.

> O encaminhamento *unicast* é o reencaminhamento de tráfego destinado a uma única localização num conjunto de redes a partir de um anfitrião de origem para um

> anfitrião de destino, através da utilização de *router*s. Um conjunto de redes tem pelo menos duas redes ligadas por *router*s. Um *router* é um sistema intermédio de camada de rede utilizado para ligar redes com base num protocolo comum de camada de rede, tal como o TCP/IP. Uma rede é uma porção da infraestrutura de trabalho de rede (prevendo repetições, concentradores, e pontes/parâmetros de camadaº2) que se encontra ligada por *router*s e associada ao mesmo endereço de camada de rede conhecido por endereço de rede ou ID de rede (MICROSOFT, 2012).

Um quadro é enviado de um *host* e endereçado a um destino específico. Na transmissão *unicast*, há apenas um remetente e um receptor. Este tipo de transmissão é a forma predominante em redes locais e na Internet. Entre os exemplos de protocolos que usam transmissões *unicast* estão <HTTP *(Hyper Text Transfer Protocol)*, SMTP *(Simple Mail Transfer Protocol)*, FTP *(File Transfer Protocol)* e Telnet (MICROSOFT, 2012).

A figura 1 exemplifica a transmissão Unicast.

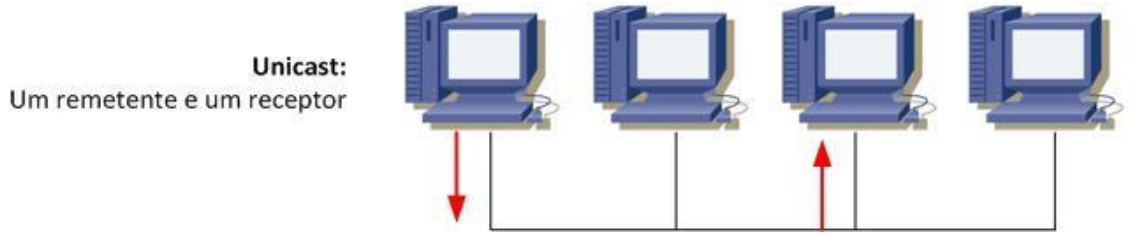

Figura 1 - Transmissão Unicast
Fonte: Autoria Própria

2.1.2 Broadcast

Um quadro é enviado de um endereço para todos os outros endereços situados na mesma rede. Nesse caso, há apenas um remetente, mas as informações são enviadas para todos os receptores conectados. A transmissão de *broadcast* é essencial durante o envio da mesma mensagem para todos os dispositivos na rede local. Um exemplo de transmissão de *broadcast* é a consulta de resolução de endereço que o protocolo de resolução de endereços ARP *(Address Resolution Protocol*) envia para todos os computadores em uma rede local (ROESLER, 2001).

A figura 2 exemplifica a transmissão Unicast.

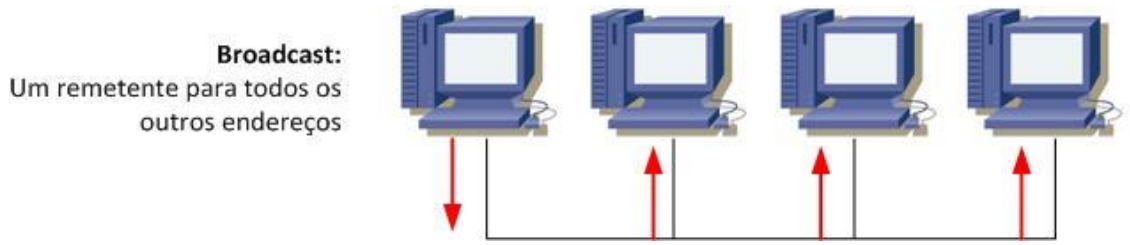

Figura 2 - Transmissão Broadcast
Fonte: Autoria Própria

2.1.3 Multicast

O multicast é uma nova forma de transmitir vídeo sobre demanda. Todos os roteadores na rede trabalham de uma forma mais inteligente. Fazem com que os pacotes de vídeo que saem de uma fonte cheguem até um destino que agora ao invés de um único usuário passa a ser um grupo de usuários cadastrados.

Em um senário de uma rede WAN semelhante a uma árvore bem ramificada, o grupo pode ser composto por usuários espalhados em subredes diferentes. E isto é possível por causa do encaminhamento dos pacotes que faz com que o vídeo chegue igualmente para todo o grupo sem exigir muito dos recursos do servidor. Diferentemente do unicast, o multicast quando transmite para mais de um receptor faz a fonte emissora transmitir o vídeo uma única vez. Os roteadores fazem a replicação dos pacotes conforme o necessário em cada nó da rede e todos os pacotes são entregues para todos os usuários do grupo.

Também diferentemente do Broadcast, o multicast não transmitirá o vídeo ao mesmo tempo para todos os hosts da subrede, incluindo os que não querem receber e Tambem se limitando apenas a rede local. Agora apenas os usuários que pertencem ao grupo multicast recebem os pacotes. Os outros nem percebem a transmissão (ROESLER,2001).

No geral o multicast exige um pouco mais dos roteadores mas garantem os mesmos padrões de qualidade do unicast ocupando menos a banda de transmissão nos "galhos da árvore". Como descreve Roesler na citação a seguir:

> Referente à qualidade de serviço (QoS), uma transmissão *multicast* é tratada da mesma forma que *unicast*, ou seja, possui as mesmas características de “melhor esforço” do IP *unicast*, sofrendo as mesmas políticas de controle de acesso e conformação de tráfego.Como a transmissão *multicast* possui o mesmo cabeçalho IP do *unicast*, os métodos de garantia de qualidade de serviço são os mesmos, e ambos

> podem usar DIFFSERV *(Diferentiated Services)*, RSVP *(Resource reServation Protocols)*, MPLS *(Multi Protocol Label Switching)*, e assim por diante (ROESLER, 2001, p.1).

A figura a seguir mostra uma comparação entre *multicast* e *unicast* em um ambiente com três switches ligados através de um roteador que suporta *multicast*. No caso, a aplicação que roda no transmissor (vídeo, por exemplo) está habilitada para gerar tráfego *multicast* na rede, e também tráfego *unicast* para cada cliente que solicita. As duas situações são mostradas na Figura 3.

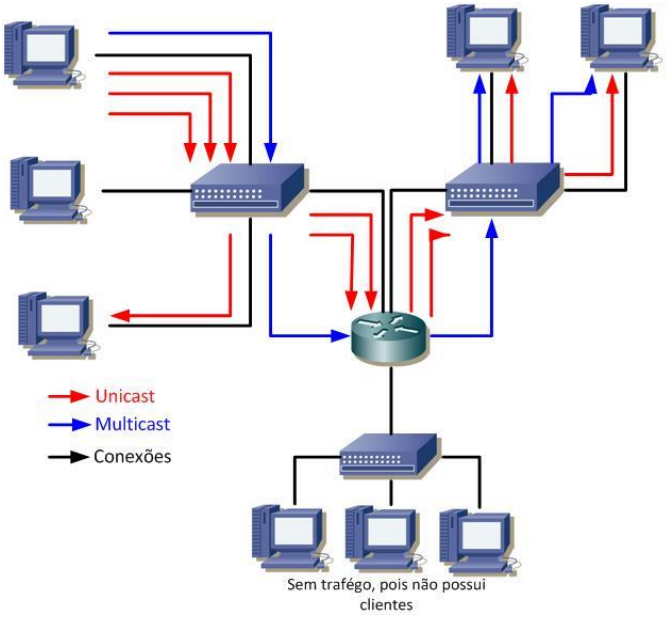

Figura 3 - Unicast x Multicast
Fonte: Autoria Própria

Multicast é a transmissão de um datagrama IP para um "grupo", ou seja, um conjunto de zero ou mais *host*s identificados por um único endereço IP de destino. Um datagrama *multicast* é entregue a todos os membros do seu grupo de destino igualmente como aos datagramas *unicast*, ou seja, o datagrama não é garantido a chegar intacto a todos os membros do grupo ou na mesma ordem em relação a outros datagramas.

A quantidade de membros de um grupo é dinâmica, ou seja, *host*s podem participar e deixar os grupos a qualquer momento. Não existe nenhuma restrição na localização ou número de membros de um grupo. O *host* pode ser membro de um ou mais grupos ao mesmo tempo. Um *host* não precisa ser membro de um grupo para enviar datagramas a ele.

Um grupo de *host*s pode ser transiente ou permanentes. Um grupo permanente tem um endereço IP fixo atribuído a ele. Este é o endereço do grupo e não dos membros, em qualquer momento um grupo permanente pode ter qualquer número de membros, até mesmo zero. Os endereços *multicast* que não são reservados para grupos permanentes, estão disponíveis para atribuições dinâmicas para grupos transientes que existem somente quando possuem membros.

O roteamento de datagramas *multicast* é tratado por "roteadores *multicast*" que podem coexistir ou não com os gateways internet. Um *host* transmite um datagrama IP que chega imediatamente a todos os membros do grupo, se o datagrama possui um TTL(*Time to Live)* maior que 1, o roteador *multicast* da rede local assume a responsabilidade de encaminhá-lo para todas as outras redes que possuem membros de destino, assim os próximos roteadores assumem a responsabilidade de entregar os datagramas *multicast* a todos os membros do grupo (ROESLER, 2001).

A figura 4 exemplifica a transmissão Unicast.

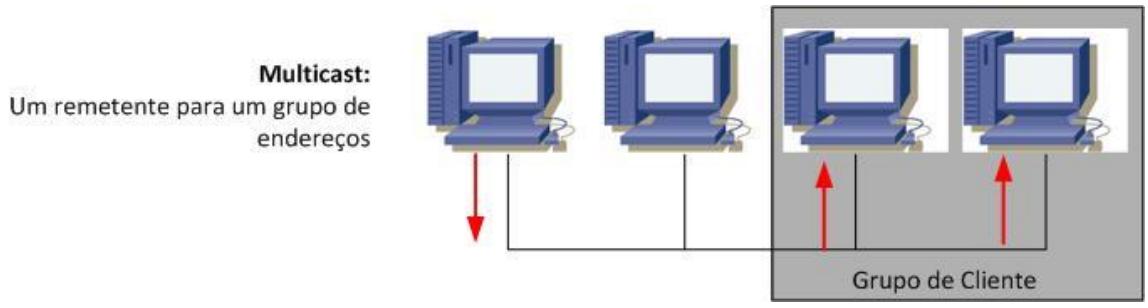

Figura 4 - Transmissão Multicast
Fonte: Autoria Própria

2.2 VANTAGENS E DESVANTAGENS

A tecnologia *multicast* oferece vantagens significativas para o sucesso de algumas aplicações avançadas. A partir do que já foi apresentado, segue abaixo vantagens e desvantagens no uso da tecnologia.

2.2.1 Vantagens

- Maior eficiência: Controla o tráfego da rede e reduz as cargas de processamento da CPU *(Central Processing Unit)* e Servidores;
- Melhor Desempenho: elimina a redundância de tráfego;
- Menos recursos, largura de banda e poder de processamento dos *host*s será necessário;
- Entrega quase simultânea está assegurada: Um único pacote é enviado simultaneamente para a rede;
- Base para ampla gama de novas aplicações que não eram possíveis no passado (teleconferências, TV, rádio, ensino à distância, treinamentos, vídeo sob demanda).
- Suporte a aplicações distribuídas: A tecnologia *multicast* é diretamente voltada para aplicações distribuídas. Aplicações multimídia como aprendizagem a distância e videoconferência podem ser utilizadas na rede de forma dimensionável e eficiente.

O uso inteligente dos recursos da rede evita replicação desnecessária de fluxos. Com isso, não há geração de tráfego indevido nos enlaces, o que ajuda a reduzir congestionamentos e, portanto, a aumentar o desempenho global da rede.

A tecnologia *multicast* é diretamente voltada para aplicações distribuídas. Aplicações multimídia como aprendizagem a distância e videoconferência podem ser utilizadas na rede de forma dimensionável e eficiente.

O custo dos recursos da rede é reduzido através de economia de banda passante nos enlaces e da economia de processamento em servidores e equipamentos da rede. Novas aplicações e serviços podem ser implantados, sem requerer renovação de recursos da rede.

O uso eficiente da rede e a redução da carga em fontes de tráfego permitem que serviços e aplicações sejam acessíveis por um grande número de participantes. Logo, serviços que rodam sobre *multicast* podem ser facilmente dimensionados, distribuindo pacotes tanto a poucos como a muitos receptores.

A economia de recursos da rede associada à redução de carga nas aplicações e servidores torna a rede menos suscetível a congestionamentos e, portanto, mais disponível para uso (MULTIREDE, 2011).

2.2.2 Desvantagens

Foram apresentadas diversas vantagens em relação ao uso do *Multicast*. Existem também algumas desvantagens. A principal delas está relacionada ao fato da entrega ser baseado em UDP *(User Datagram Protocol)*.

Segue abaixo as desvantagens pelas características do UDP:

- Entrega baseada em melhor esforço: quedas podem ser esperadas. Aplicações *multicast* não devem esperar dados confiáveis como os baseados em TCP *(Transmission Control Protocol)* .
- Não evita congestionamento: A falta de janelas TCP e do mecanismo "slow-start" pode resultar em um congestionamento na rede, pois o roteador pode ficar sem espaço de *buffer* ou aplicar alguma política de descartar pacotes. Se possível, aplicações *multicast* devem tentar detectar e evitar condições de congestionamento.
- Duplicatas: Algumas aplicações *multicast* podem gerar pacotes duplicados ocasionalmente.
- Entrega desordenada de pacotes: Mudanças na topologia da rede podem afetar a ordem de entrega dos pacotes.

No geral pode-se perceber que confiabilidade é um ponto a ser ter cuidado na transmissão de tráfego *Multicast*. Segurança é outra questão que não foi suficientemente resolvida no passado com o desenvolvimento da tecnologia.

Soluções adequadas para as questões acima podem abrir grandes oportunidades para diversas aplicações comerciais, como por exemplo, entrega de dados financeiros (MULTIREDE, 2011).

3 ENDEREÇAMENTO IP

O endereço IP, na versão 4 do IP (IPv4), é um número de 32 bits oficialmente escrito com quatro octetos (Bytes) representados no formato decimal como, por exemplo, "192.168.1.3". A primeira parte do endereço identifica uma rede específica na internet, a segunda parte identifica um *host* dentro dessa rede. Devemos notar que um endereço IP não identifica uma máquina individual, mas uma conexão à internet. Assim, um gateway conectando à n redes tem n endereços IP diferentes, um para cada conexão.

Os endereços IP podem ser usados tanto para nos referir a redes quanto a um *host* individual. Por convenção, um endereço de rede tem o campo identificador de *host* com todos os bits iguais a 0 (zero). Podemos também nos referir a todos os *host*s de uma rede através de um endereço por difusão, quando, por convenção, o campo identificador de *host* deve ter todos os bits iguais a 1 (um). Um endereço com todos os 32 bits iguais a 1 é considerado um endereço por difusão para a rede do *host* origem do datagrama. O endereço 127.0.0.1 é reservado para teste (loopback) e comunicação entre processos da mesma máquina. O IP utiliza três classes diferentes de endereços. A definição de tipo de classes de endereços deve-se ao fato do tamanho das redes que compõem a internet variarem muito, indo desde redes locais de computadores de pequeno porte, até redes públicas interligando milhares de *host*s.

Existe outra versão do IP, a versão 6 que utiliza um número de 128 bits. Com isso dá para utilizar 25616 endereços diferentes.

O endereço de uma rede (não confundir com endereço IP) designa uma rede, e deve ser composto pelo seu endereço (cujo último octeto tem o valor zero) e respectiva máscara de rede (ROESLER, 2001).

A figura 5 mostra o quadro com as classes de endereço IP e seus intervalos de endereçamento.

Classe A	0	netID (7bits)	hostID (24 bits)	0.0.0.0 até 127.255.255.255
Classe B	10	netID (14bits)	hostID (16 bits)	128.0.0.0 até 191.255.255.255
Classe C	110	netID (21bits)	hostID (8 bits)	192.0.0.0 até 223.255.255.255
Classe D	1110	netID (7bits)		224.0.0.0 até 239.255.255.255

Figura 5 - Intervalo de Endereços
Fonte: Autoria própria

3.1 ENDEREÇAMENTO *MULTICAST* NIVEL 3

O *multicast* utiliza a classe D de endereçamento da Internet, ou seja, de 224.0.0.0 a 239.255.255.255. A diferença no cabeçalho IP de um pacote *unicast* e um *multicast* é apenas o endereço. A classe D permite até 28 bits (268 milhões de endereços), como mostra a figura6.

1	**1**	**1**	**0**	**Grupo Multicast (28 bits)**

Figura 6 - Endereço IP Multicast
Fonte: Autoria Própria

Existem alguns endereços *multicast* reservados pelo IANA (*Internet Assigned Numbers Authority),* como o 224.0.0.0, que é a base, e não deve ser usado. Já os endereços de 224.0.0.1 a 224.0.0.255 são reservados para protocolos de roteamento, como, por exemplo, "todos os sistemas na subrede" (224.0.0.1), "todos os roteadores nesta subrede" (224.0.0.2), "todos roteadores DVMRP" (224.0.0.4), "todos roteadores MOSPF" (224.0.0.6). Veremos mais a diante esses IPs quando abordarmos a utilização de cada um.

Outros são reservados para determinadas aplicações, como o "IETF1-áudio" (224.0.1.11) e o IETF1-vídeo (224.0.1.12). O conjunto de endereços de 239.0.0.0 a 239.255.255.255 é reservado para uso local, podendo ser usado em intranets.

O quadro 1 mostra os principais intervalos de endereçamento *Multicast* IPv4. Vale salientar que existem mais intervalos definidos pela IANA (ROESLER, 2001).

Intervalo de Endereçamento	**Descrição**
224.0.0.0 - 239.255.255.255	Intervalo total para *Multicast* (244.0.0.0/8)
224.0.0.0 - 224.0.0.255	Intervalo reservado para controle *Multicast* e escopo Local
224.0.1.0 - 224.0.1.255	Intervalo conhecido como *Internetwork Control Block*, utilizado para o tráfego que deve ser encaminhado através da Internet pública, como NTP e MDHCPDISCOVER
224.1.0.0 - 224.1.255.255	*Spanning Tree Multicast Group*
224.0.2.0 - 224.0.255.255, 224.3.0.0 - 224.4.255.255 e 233.252.0.0-233.255.255.255	Endereços atriubuidos pela IANA para AD-HOC(Bloco I, II e III).
224.0.1.0 - 238.255.255.255	Intervalo escopo global
239.0.0.0 - 239.255.255.255	Intervalo reservado para uso Local

Quadro 1 - Principais endereços Multicast
Fonte: Autoria Própria

3.2 ENDEREÇAMENTO *MULTICAST* NIVEL 2

Dentro de uma rede local, existe um conjunto de endereços MAC *(Media Access Control)*, destinado ao *multicast*. Assim, o protocolo de nível 2 e, conseqüentemente, o *host* do cliente, sabe se a mensagem é destinada a ele ou não.

Esse conjunto de números foi reservado pelo IANA, e compreende todos os endereços de 01-00-5E-00-00-00 a 01-00-5E-7F-FF-FF (somente 23 bits, comparado aos 28 do endereço *multicast*). Existe um processo de mapeamento entre o IP *multicast* e o Ethernet *multicast*, que é simplesmente substituir os 23 bits menos significativos do Ethernet pelos 23 bits menos significativos do IP. Assim, quando uma estação cliente se cadastra em determinado grupo *multicast*, automaticamente o *driver* da placa de rede passa a receber mensagens MAC com endereçamento correto (ROESLER, 2001).

A figura 7 exemplifica a como o endereço IP(nível 3) é convertido para o endereço de multicast.

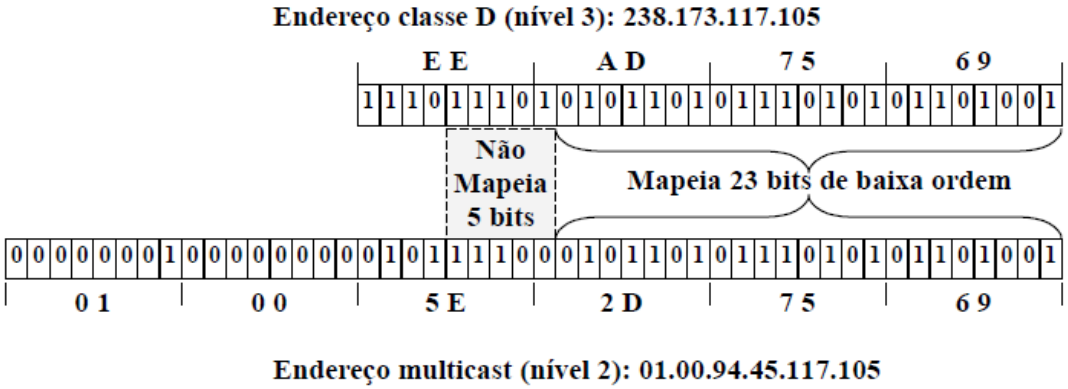

Figura 7 - Mapeamento endereço nível 2 e 3
Fonte: ROESLER (2009, p. 4)

A maioria dos *switches* Ethernet atuais, quando recebem um número MAC *multicast*, tratam como se fosse endereço *broadcast*, ou seja, enviam para todas suas portas. Assim, qualquer cliente cadastrado no grupo vai receber o pacote e jogar para o nível de cima, e qualquer cliente não cadastrado vai receber o pacote e ignorá-lo. Existem alguns switches que já entendem IGMP *Snooping*, e são úteis para minimizar o tráfego desnecessário em redes nível 2, mas isso é um tópico que será abordado mais a diante (ROESLER, 2001).

4 MROUTER

Um *Mrouter*, ou roteador *multicast*, é um programa de roteador que faz a distinção entre os pacotes *multicast* e *unicast* e determina como devem ser distribuídos ao longo da Internet *Multicast* (às vezes conhecido como o *Multicast* Backbone ou MBone que será visto posteriormente). Usando um algoritmo apropriado, um m*router* diz a um dispositivo de comutação o que fazer com o pacote *multicast*.

Mrouters compõem atualmente "ilhas" no MBone separados por roteadores *unicast*. Assim, um m*router* pode disfarçar pacotes *multicast* para que eles possam atravessar roteadores *unicast*. Isto é feito fazendo com que cada pacote *multicast* pareça um pacote *unicast*, o endereço de destino é o m*router* seguinte. Este processo é chamado de IP tunneling.

Os dois principais modelos de roteamento *multicast* que m*router*s usam para distribuir pacotes são *dense-mode routing* e *sparse-mode routing*. O protocolo utilizado é determinado pela largura de banda disponível e da distribuição de usuários finais através da rede. Se a rede tiver muitos usuários finais e bastante largura de banda, *dense-mode routing* é usado. No entanto, se a largura de banda é limitada e os usuários são mal distribuídos, é usado o *sparse-mode routing*.

Sem um *mrouter* em algum ponto da rede, não é possível termos o *Multicast* funcionando adequadamente (GALIANO, 2011).

5 IGMP (Internet Group Management Protocol)

O IGMP é o principal protocolo de *multicast*. É ele o responsável por gerenciar os grupos e todas as mensagens referentes à ações *multicast* na rede. É um protocolo que trabalha entre o roteador e os *hosts*.

Através do IGMP um *host* envia um relatório ao roteador mais próximo dizendo que deseja fazer parte de um determinado grupo *multicast*, essa solicitação é chamada de *Membership Report*. Esta mensagem também é usada para sinalizar que o *host* ainda pertence ao grupo e tem interesse em continuar recebendo o tráfego especifico. Mensagens do tipo *Membership Report* são enviadas ao endereço do grupo.

Os roteadores também enviam mensagens na rede, essas são chamadas de *Membership Query*, e têm por finalidade determinar quais os grupos("*host groups*") têm membros nas suas redes diretamente conectadas.

Em outras palavras essas mensagens são nada mais que interrogações para que os roteadores saibam quais *host*s pertencentes às suas redes fazem parte de algum grupo *multicast*, ou ainda se existem grupos nas suas redes, visto que os grupos podem estar vazios, caso não haja mais nenhum *host* associado a ele. Quando o roteador envia essa mensagem o destino no cabeçalho do pacote é o endereço 224.0.0.1.

Estas mensagens *Membership Query*, são periodicamente transmitidas na rede. Caso haja algum *host* em algum grupo ou algum interessado em receber a informação de um fluxo *multicast*, este *host* responde com uma *Membership Report* informando para o roteador qual grupo *multicast* ele pertence, visto que cada *host* pode participar de mais de um grupo.

Vale salientar que para evitar o congestionamento de mensagens *Membership Report*, cada estação espera um tempo aleatório para dar a resposta. Antes de um *host* enviá-la, ele "olha" para o meio de transmissão para ver os relatórios que já foram enviados ao roteador, caso algum *host* do seu grupo já tenha enviado a *Membership Report*, o seu tempo de espera expira e cancela o envio. Assim assegura-se que o roteador receberá apenas uma *Report* de cada grupo, evitando congestionamento desnecessário na rede.

De acordo com os relatórios recebidos, os roteadores sabem como determinar o tráfego *multicast* que deve ser encaminhado para cada grupo.

Ns figures 8 e 9 é possível ver um exemplo de roteador enviando a mensagem de *Query* para a rede e a resposta de *Report* pelos host(apenas uma resposta por grupo) (DIG, 2012).

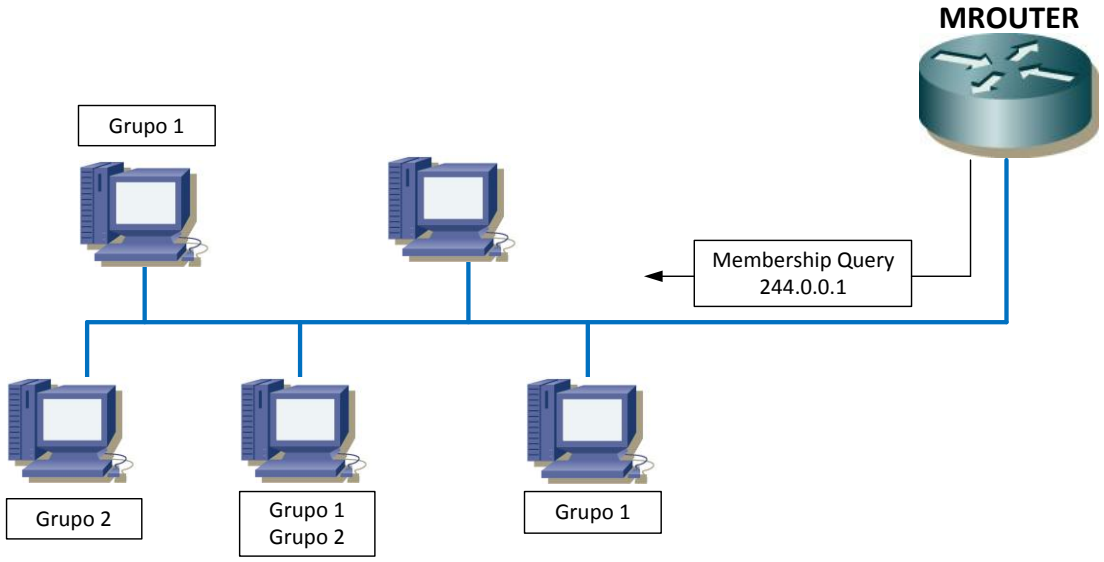

Figura 8 - Roteador enviando Membership Query.
Fonte: Autoria Própria

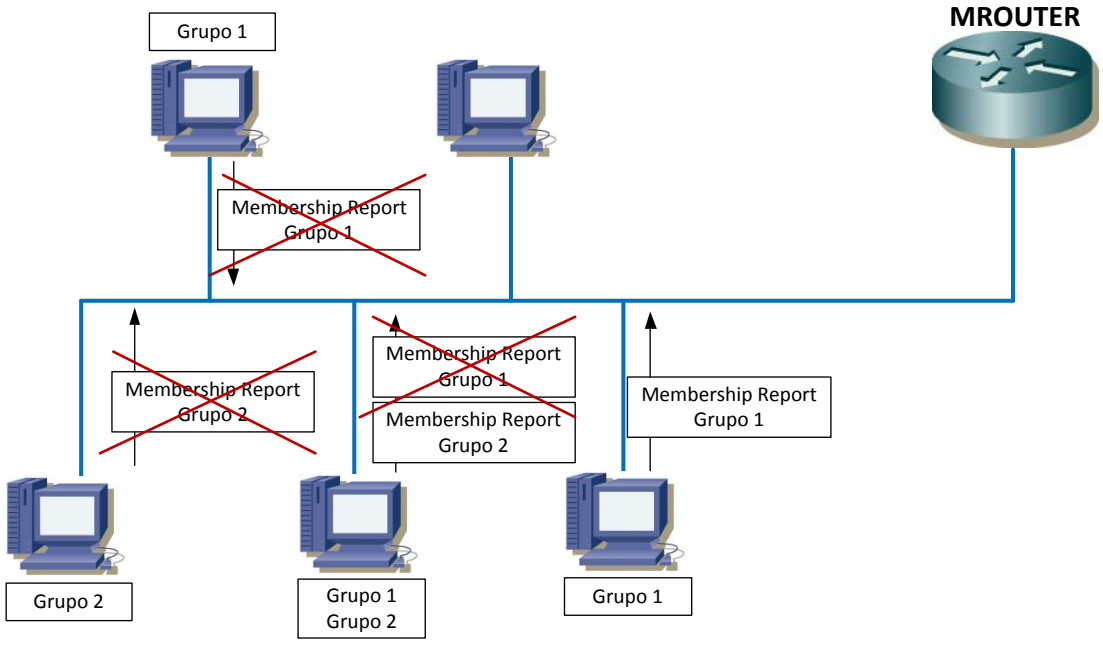

Figura 9 - Hosts enviando Membership Report
Fonte: Autoria Própria

Quando o software da aplicação multimídia pede ao software de rede da estação (*host*) para esta se juntar a um grupo *multicast*, uma mensagem IGMP é enviada ao *router* mais próximo (se o *host* não for já um membro do grupo). Ao mesmo tempo, o endereço *multicast* de classe D do grupo ao qual se junta é mapeado como um endereço de baixo nível e a interface da rede é programada para aceitar pacotes para esse endereço.

Por exemplo, se uma estação passa a integrar um grupo num interface Ethernet, os 23 bits mais baixos do endereço de classe D são mapeados aos 23 bits mais baixos do endereço Ethernet. Devido a esta filtragem de endereços *multicast* por hardware, um *router* não necessita manter uma lista detalhada das estações que pertencem a cada endereço de grupo, mas apenas esse membro, pelo menos, do grupo, está presente na sub-rede à qual se encontra vinculado (DIG, 2012).

5.1 ESTRUTURA DO QUADRO

Como o ICMP*(Internet Control Message Protocol*), as mensagens IGMP são encapsuladas e enviadas em datagramas IP no Payload. Já o mapeamento de endereços IP para endereços de rede local é considerado como responsabilidade dos módulos de rede local. A figura 10 exemplifica o encapsulamento:

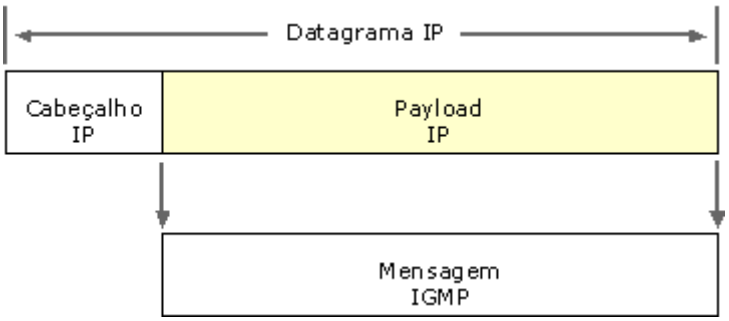

Figura 10 - IGMP encapsulado no IP
Fonte: Autoria Própria

O protocolo IGMP apresenta 3 versões. A seguir será comentado cada um deles.

5.2 IGMPv1

A versão 1 apresentava fraquezas, principalmente quanto a latência elevada associada com o término de sessões *multicast*. Depois do último membro de um grupo *multicast* numa sub-rede ter abandonado o grupo, os outros *router*s não eram imediatamente notificados para deter a propagação de tráfego para o grupo. Esta demora era causada pelo IGMPv1, esperando até que várias interrogações indicassem que não restavam membros na sub-rede, de um grupo em particular. Assim, indesejavelmente, tráfego desnecessário era encaminhado para a sub-rede. O custo deste envio inútil podia ser elevado, particularmente num segmento da Internet com largura de banda restrita. Abaixo é mostrada a estrutura da primeira versão (DIG, 2012).

Na tabela abaixo esta a estrutura do quadro da versão 1.

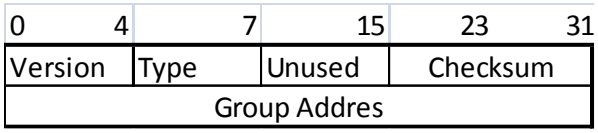

Figura 11 - Diagrama IGMP
Fonte: Autoria Própria

Os campos são:

- Version: É colocada a versão do protocolo (existem 3 versões)
- Type: Tipo de mensagem que esta sendo transmitida
- *Host Membership Query*
- *Host Membership Report*
- Unused: Campo não utilizado, zerado quando enviado e ignorado quando recebido.
- Checksum: Algoritmo de verificação de integridade da mensagem IGMP.
- *Group Address*: Este campo depende do conteúdo do campo type.

Quando a mensagem é do tipo 1(*Membership Query*), esse campo é zerado quando enviado e zerado quando recebido.

Quando a mensagem é do tipo 2(*Membership Report*), esse campo contém o endereço do grupo sendo *report*ado, ou que o *host* deseja participar (DIG, 2012).

5.3 IGMPv2

IGMP versão 2 funciona basicamente da mesma maneira que a versão 1. A principal diferença é que há uma mensagem de sair do grupo (*leave group*). Com esta mensagem os *host*s podem comunicar aos roteadores locais que pretendem deixar um grupo e assim não receber aquele tráfego.

É uma importante mudança, pois depois do último membro de um grupo *multicast* numa sub-rede ter abandonado o grupo, os outros roteadores não eram imediatamente notificados que não havia mais nenhum *host* naquele grupo.

Esta demora era causada pelo IGMPv1 esperando até que várias mensagens indicassem que não restavam membros na sub-rede, de um grupo em particular. O custo deste envio inútil era a principal desvantagem do IGMPv1.

Por esse motivo a adição da mensagem sair em grupo IGMP versão 2 reduz a latência deixada em relação ao IGMP versão 1, tráfego indesejado e desnecessário pode ser interrompido mais cedo.

Outra característica que apresenta formas de reduzir o overhead do protocolo são as mensagens de interrogação dirigidas a grupos específicos (*Group* Specific *Query* Message) , ou seja permitem ao roteador interrogar grupos específicos nas redes onde estão diretamente vinculados em vez de serem forçados a interrogar todos os grupos indiscriminadamente. Em outras palavras, quando o roteador recebe uma mensagem de *leave*, ele envia uma *Group* Specific *Query* Message para o grupo especifico deste *host* que solicitou sair, caso ele não receba nenhuma resposta deste grupo, o roteador sabe que não existem mais indivíduos interessados no tráfego, assim o *router* notifica outros *router*s para cessarem o encaminhamento de tráfego para a sub-rede dirigido ao grupo.

Com relação ao diagrama do pacote, não apresenta grandes diferenças em relação à primeira versão, as mudanças são dentro dos campos com as melhorias já citadas(*Leave group* e *Group*-Specific *Query*) (DIG, 2012).

A figura 12 mostra a estrutura do quadro IGMPv2.

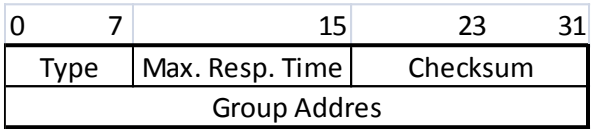

Figura 12 - Diagrama IGMPv2
Fonte: Autoria Própria

No campo TYPE pode existir 4 tipos de mensagens IGMP:

- 0x11= *Host Membership Query*;

  General *Query*

  Group-Specif *Query*

- 0x12=Version 1 *Membership Report*(Usado para compatibilidade com a versão 1);
- 0x16=Version 2 *Membership Report*(Usado para versão 2);
- 0x17=*Leave Group*;

O campo Max Resp. Time corresponde ao máximo tempo antes de um *host* mandar uma *Report* na rede (DIG, 2012).

5.4 IGMPv3

A versão 3 do IGMP vai mais longe na redução do overhead. A largura de banda será conservada pela mensagem *Group*-Source *Report* que permitirá às estações receber tráfego de fontes específicas de um grupo *multicast*. Nas versões anteriores do IGMP, o tráfego de todas as fontes tinha de ser encaminhado para uma sub-rede mesmo se as estações estivessem apenas interessadas em receber tráfego de fontes específicas. As mensagens *Leave-Group* apresentadas em primeira instância pela versão 2 foram também aperfeiçoadas para permitir às estações largar um grupo inteiro ou para especificar a fonte a que queriam renunciar.

Resumidamente no IGMPv3 o usuário tem 2 opções: receber pacotes de apenas um endereço específico (conhecido como SSM, *Source-Specific Multicast*), ou receber pacotes de todos, exceto de um endereço específico.

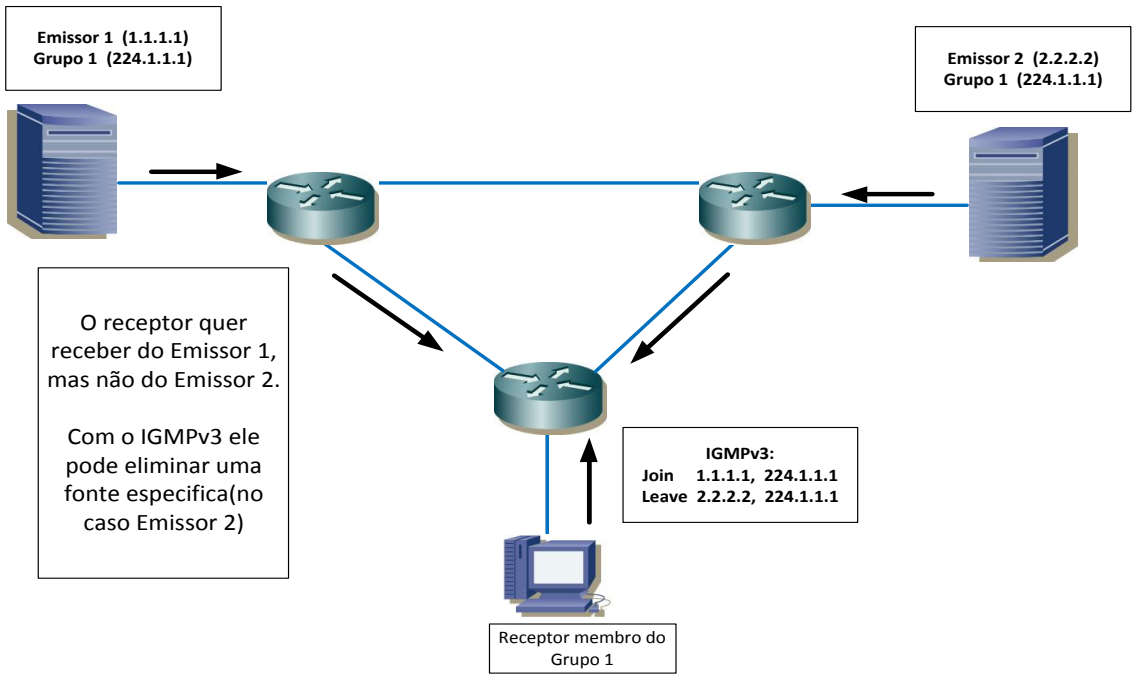

Figura 13 - IGMPv3 filtrando origem de pacote
Fonte: Autoria Própria

A figura 13 exemplifica o funcionamento da filtragem de pacotes. Temos o cenário com dois servidores (Emissor 1 e Emissor 2) enviando tráfego para o Grupo 1. O receptor quer entrar no grupo 1 e receber o tráfego do Emissor 1, mas não do Emissor 2. Utilizando o IGMPv3 ele pode eliminar uma origem.

Pelos métodos acima mencionados, os roteadores *multicast* estão habilitados a manter, por interface, uma tabela atualizada contendo os grupos cujo tráfego tem interesse, após a recepção de pacotes *multicast*, os roteadores sabem para que interfaces os pacotes devem ser encaminhados.

Levando-se em conta que as versões mais recentes do IGMP podem reduzir o tráfego desnecessário, otimizando a utilização deste protocolo, deve ser favorecida a escolha de sua utilização em detrimento das anteriores quando possível (DIG, 2012).

5.5 IGMP SNOOPING

Numa rede baseada em switch(Layer2), todo tráfego *multicast* recebido é mandado para todo o domínio de *broadcast*. Isto pode prejudicar muito o desempenho da rede, pois

consome banda desnecessária da rede, principalmente se existirem muitos servidores mandando tráfego para a rede.

A função básica do IGMP Snooping é restringir o tráfego *multicast* em uma rede baseada em switch. Com o IGMP Snooping, o switch "ouve" os relatórios de IGMP entre o roteador e os *hosts* ligados a ele, assim consegue mapear quais portas devem ser liberadas para cada tráfego *multicast* que estiver sendo enviado à rede e bloqueando as outras portas para estes pacotes, montando assim uma tabela semelhante à ARP. Quando ele "ouve" mensagem de *leave*, imediatamente ele remove a porta do *host* que enviou a mensagem da lista que deseja receber o tráfego.

Quando um *host* manda uma requisição para entrar em um grupo, o switch intercepta a mensagem de IGMP *Membership Report* e cria uma entrada para trafego *multicast* desse grupo na porta onde está o *host*, depois envia uma mensagem para todas as portas do roteador, assim o roteador atualiza sua tabela de roteamento.

Quando um segundo ou terceiro *host* quer participar do mesmo grupo, o switch novamente intercepta a mensagem de IGMP *Membership Report*, mas não envia a todas as portas do roteador, ele encaminha relatórios IGMP para as portas do roteador, usando relatórios Proxy. Como os *router*s enviam em intervalos de 60 segundos relatórios para saber se ainda existem membros nos grupos, quando essa mensagem é recebida no switch, o segundo *host* ou os outros que solicitaram participação no grupo, começam a receber o tráfego desejado.

Se um dos *host*s que esta recebendo o fluxo *multicast* deseja deixar o grupo, o switch envia uma mensagem para todos do grupo da porta de onde recebeu a mensagem de *leave.* Caso ele não receba mais nenhuma requisição de *leave*, ele bloqueia para o tráfego apenas a porta que enviou a solicitação, sendo assim ele não encaminha a mensagem para o *router*. Se outro *host* envia solicitação para deixar o grupo, o processo se repete. Quando não tiver mais nenhum membro no grupo, o switch captura a mensagem de *leave* do primeiro *host* que entrou no grupo e envia para todas as portas do roteador e remove a entrada de sua tabela (DIAS, 2002).

5.5.1 MVR *(MULTICAST VLAN REGISTRATION)*

Por padrão em uma rede Layer 2, um fluxo *multicast* recebido em uma VLAN *(Virtual Local Área Network)* nunca é distribuído às interfaces externas(outras VLANs). Se os *host*s em várias VLANs distintas desejam receber um mesmo fluxo *multicast*, uma cópia separada do fluxo *multicast* é distribuído para cada VLAN que está solicitando.

Com isso mais uma vez caímos no problema de largura de banda desnecessária sendo consumida, visto que o mesmo tráfego está sendo enviado várias vezes para cada VLAN.

O MVR*(Multicast VLAN Registration)*, é a resposta para esse problema. A idéia principal é o conceito de uma VLAN fonte de *multicast*, ou seja, os *host*s de diferentes VLANs interessados no fluxo *multicast*, compartilham o mesmo tráfego através do MVR, mas sem mudar suas VLANs de origem.

Em outras palavras uma única VLAN *multicast* pode ser compartilhada na rede enquanto os *host*s permanecem em VLANs separadas.

Um serviço que tira muito proveito do MVR é o IPTV *(Internet Protocol Television)* , onde vários assinantes podem solicitar assistir um canal através da rede, assim estes estarão dividindo o mesmo fluxo, visto que todos estarão recebendo o tráfego da VLAN/Fonte.

O MVR é compatível com o IGMPv2, sendo assim através do mensagens de *join* e *leave* do IGMP a porta de um assinante(*host*) pode requisitar participação em um grupo ou deixar ele. Como é um protocolo de camada 2 ele pode rodar paralelamente com o IGMP Snooping. Quando os 2 estão ativos na mesma rede, o MVR controla a entrada e saída apenas em grupos que estão configurados sob o próprio MVR. Entrada e saída de todos os outros grupos não configurados serão gerenciadas pelo IGMP Snooping.

Como citado acima, os grupos *multicast* devem ser associados ao MVR, assim o switch aloca os grupos na sua tabela de encaminhamento. Através de interceptações de mensagens IGMP ele modifica a tabela incluindo ou removendo os assinantes do grupo receptor da VLAN/Fonte, mesmo eles estando em VLANs diferentes. Este comportamento de encaminhamento seletivo permite o tráfego ser passado entre diferentes VLANs e isolando o fluxo dessas diferentes VLANs garantindo mais banda e segurança (CISCO SYSTEM, 2011).

Na figura 14 um exemplo de um segmento de rede com MVR.

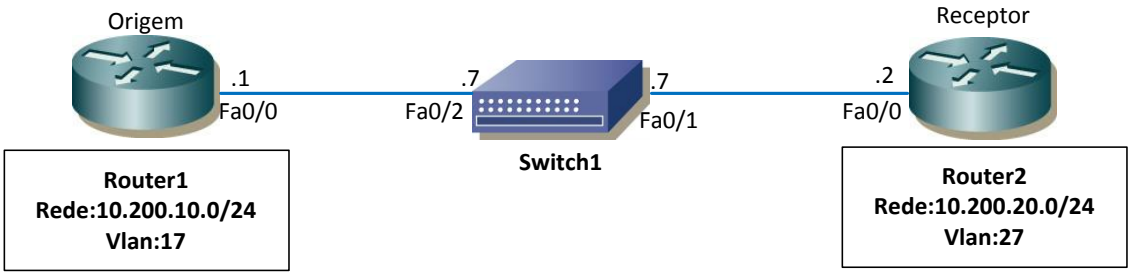

Figura 14 - Rede com MVR
Fonte:Autoria Própria

No exemplo, uma rede rodando PIM no Modo-Denso, o *Router*1 é a fonte *multicast* e o *Router*2 o receptor. Usando um IP 235.1.1.1 para o grupo *multicast* em questão, o MVR é configurado no Switch1 com a VLAN17 sendo a VLAN/Fonte para os assinantes da Vlan 27 pertencentes ao grupo.

Segue abaixo como ficaria a configuração nos roteadores:

*Router*1

*Router*1# interface FastEthernet0/0

*Router*1# ip *Address* 10.200.10.1 255.255.255.0

*Router*1# ip pim dense-mode

*Router*2

*Router*2# interface FastEthernet0/0

*Router*2# ip *Address* 10.200.10.2 255.255.255.0

*Router*2# ip pim dense-mode

Como mostrado, o MVR não participa da configuração de camada 3, ele deve ser configurado e roda em camada 2. Segue abaixo como seria a configuração no Switch.

Switch1

Switch1# no ip *multicast*-routing distributed

Switch1#!

Switch1#mvr vlan 17

Switch1#mvr mode dynamic

Switch1#mvr *group* 235.1.1.1
Switch1# mvr
Switch1# !
Switch1# interface FastEthernet0/1
Switch1#switchport access vlan 17
Switch1# switchport mode access
Switch1#mvr type source
Switch1#mvr vlan 17 *group* 235.1.1.1
Switch1# !
Switch1# interface FastEthernet0/2
Switch1#switchport access vlan 27
Switch1# switchport mode access
Switch1# mvr type receiver
Switch1# mvr vlan 17 *group* 235.1.1.1
Switch1# !
Switch1#interface Vlan17
Switch1#ip *Address* 10.200.10.7 255.255.255.0
Switch1#ip pim dense-mode
Switch1#!
Switch1#interface Vlan27
Switch1#ip *Address* 10.200.20.7 255.255.255.0
Switch1#ip pim dense-mode

Como mostrado na configuração Switch1, a VLAN/Fonte é associada ao grupo *multicast* e suas portas são configuradas para saber em quais portas o tráfego da VLAN pode ser encaminhado. Pode-se perceber também que as portas do switch devem ter IPs pertencentes à rede que está conectada e estas devem estar habilitadas para rodar o protocolo de roteamento, no caso o PIM Dense-Mode (CISCO SYSTEM, 2011).

6. MBONE

6.1 HISTÓRIA

O Mbone, ou *Multicast* Backbone, teve sua origem de uma pesquisa conjunta entre a Universidade da Califórnia do Sul, Instituto de Tecnologia de Massachussetts, Centro de pesquisa Xerox Palo Alto, Laboratório nacional Lawrence Berkeley e outras unidades patrocinadas pela Agencia de Defesa de pesquisa avançada dos EUA.

Em 1990 esta comunidade criou uma extensa rede de pesquisa chamada DARPA *(Defense Advanced Research Projects Agency)*, consistindo de estações de trabalho UNIX *(Universal Interactive Executive)* servindo com roteadores interconectados via links T1. Inicialmente no projeto, a comunidade desenvolveu uma versão preliminar do IP *Multicast* sobre rede. Esta foi a primeira oportunidade de experimentar o *multicast* em larga escala.

O próximo passo importante ocorreu quando a comunidade DARTNet desenvolveu várias aplicações real-time usando IP *Multicast* para estudar os problemas em grupos de conferencia sobre redes comutadas por pacotes. As reuniões semanais da DARTNet começam a gerar interesse na expansão da infra-estrutura *Multicast* além dos limites de laboratório. Entretanto os fabricantes de roteadores não suportavam o trafico *Multicast*, porem os designers do IP *Multicast* tinha habilitado os roteadores *Multicast* a encaminhar pacotes e trocas mensagens de rotas sobre links virtuais.

A habilidade de construir subredes *Multicast* usando túneis tornou-se um ambicioso experimento em Março de 1992, 32 sites *Multicast* isolados cobriam 4 países que foram configurados com uma grande rede virtual *Multicast*, a qual foi usada para transmitir a "23ª *Internet Engineering Task Force* Meeting". Esta interligação virtual usada para juntas as sub-redes *Multicast* foi chamada de "*Multicast* Backbone". Apesar de varias dificuldades técnicas o experimento foi um sucesso. Em Março de 1997, foram inclusas 3.400 redes *Multicast* ao mesmo tempo. Em poucos anos, o MBone de uma pequena pesquisa tornou-se uma infra-estrutura de comunicação usada em grande escala (GALIANO, 2011).

6.2 IP MULTICASTING

Do ponto de vista da rede, o *Multicast* simplesmente executa operações que resultam em cópias de dados entregues a vários receptores, esta pode ser implementada de duas formas:

- Transmissor usa uma conexão de transporte *Unicast* para cada receptor, ou seja, a mensagem é duplicada no transmissor e enviada individualmente a cada receptor.
- Transmissor envia um único datagrama, e cada roteador se torna responsável a enviar a seus receptores, melhorando assim o consumo de banda na rede.

O IP *Multicast* já esta mudando a forma das empresas fazerem negócios. Através do IP *Multicast* na internet, aplicações corriqueiras a empresas, como videoconferências, podem tomar escala global facilmente (STARDUST, 1999).

6.3 LIMITAÇÕES E FUTURO DO MBONE.

Desde que o Mbone cresceu, este tem sofrido uma série de problemas, que vem ocorrendo com constante freqüência.

- Escalabilidade: A mais importante razão para isto é a dificuldade de crescimento da gerencia das topologias virtuais flexíveis. Como o MBone cresceu, seu tamanho torno-se um problema, em termos de igualdade de rotas e erros de configuração. Grandes, as redes são inerentemente instáveis. Em picos, o Mbone tem quase 10.000 roteadores.
- Gerenciamento: Como o MBone cresce randomicamente, tornou-se difícil de gerencias. O MBone não possui um gerenciamento central, a maioria das tarefas foram manipuladas por site. A maioria da coordenação ocorre através da lista de discussão MBone. Dois tipos de ineficiências comumente são observados:
- Gerenciamento de Túnel (Topologia Virtual): O MBone é caracterizado como um conjunto de ilhas *multicast* conectadas por túneis. O objetivo tem sido conectar estas ilhas da maneira mais eficiente, mas constantemente túneis abaixo do ideal têm sido criados.
- Gestão de políticas entre domínios: Limites de domínio são outra fonte de problemas ao tentar gerenciar uma topologia plana. O modelo na Internet de hoje é estabelecer AS’s *(Autonomous System)* limites entre os domínios da Internet. AS’s são

comumente geridos ou detidos por diferentes organizações. Entidades em um AS são tipicamente não confiáveis por entidades em outro AS. Como resultado, o intercâmbio de encaminhamento de informações através das AS's fronteiras, é tratado com muito cuidado.

O primeiro dos problemas é a complexidade e instabilidade das grandes topologias flexíveis. O segundo problema é que não existe nenhum mecanismo para construir uma topologia hierárquica de roteamento *Multicast*. Na tentativa de resolver estes dois problemas criou-se a primeira tentativa de implantar o inter domínio *multicast*.

O Interdomain *multicast* evoluiu a partir da necessidade de fornecer escalabilidade e hierarquiabilidade, em toda a internet *multicast*. Protocolos que fornecem a funcionalidade necessária têm sido desenvolvidos, mas a tecnologia é relativamente imatura. Estes protocolos estão sendo consideradas pelo IETF e, simultaneamente, sendo avaliado por meio da implantação extensiva (STARDUST, 1999).

7. ÁRVORES DE DISTRIBUIÇÃO

Roteadores criam árvores de distribuição que permitem controlar o caminho do tráfego *multicast*. Existem dois tipos básicos: Source *Tree* e Shared *Tree* (FRANCISCO, 2000).

7.1 *SOURCE TREE (Shortest Path Trees)*

Utiliza mais memória por parte dos roteadores (S, G), onde S(*Source*) é o IP da fonte e G (*Group*) é o endereço do grupo (existe uma árvore para cada par S , G).

Embora utilize mais memória o método *Source Tree* percorre caminhos otimizados, minimizando atraso na entrega dos pacotes. A raiz da árvore é a origem do tráfego, onde se calcula um *Spanning Tree* até os receptores. Essa árvore usa o menor caminho para atingir o destino (também conhecida como *Shortest Path Tree* ou simplesmente STP).

O caminho escolhido por uma rede é exemplificado na figura 15, onde temos o emissor enviado os pacotes para a rede. O STP escolhe o caminho mais curto (FRANCISCO, 2000).

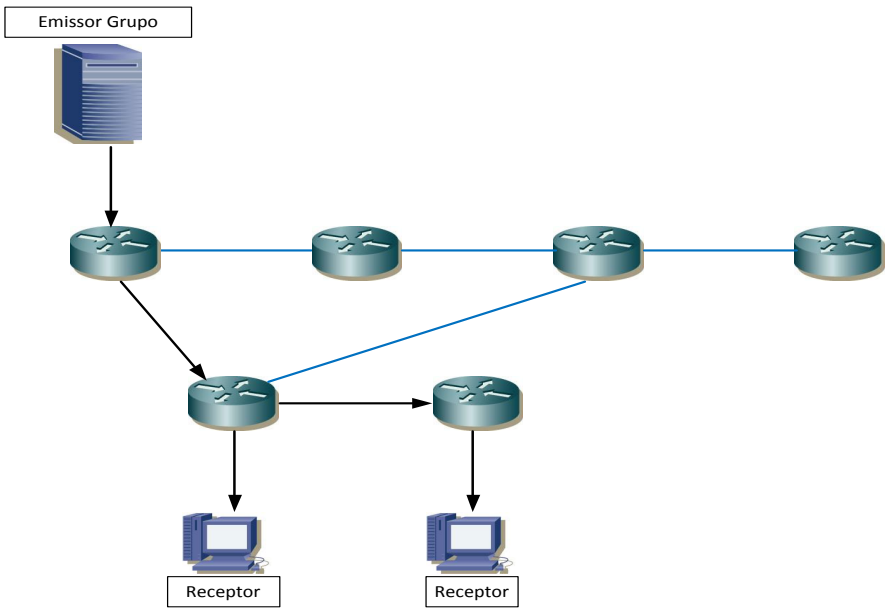

Figura 15 – Source*Tree*
Fonte: Autoria Própria

7.2 SHARED *TREE (Rendezvous Point Trees)*

Ao contrário do *Source Tree* as *Shared Tree* utiliza menos memória(G), contudo percorre caminhos não otimizados podendo introduzir atrasos na entrega dos pacotes.

Estas árvores também são conhecidas como *Rendezvous Point Trees* ou simplesmente RPT. Sua árvore de encaminhamento *multicast* tem como ponto central ou raiz, o roteador *Rendezvous Point*(RP). O RP sempre é o ponto principal da rede, mas nem sempre essa é a melhor maneira de se utilizar a banda disponível. Sendo assim, as árvores podem estar configuradas para possibilitar, após o estabelecimento do fluxo de dados *multicast*, uma otimização para a uma *Shortest Path Trees* (DIG: Enterprise Campus Topology, 2012).

A figura 16 mostra a mesma rede usada para a exemplificação do *source tree*, mas com a *shared tree* o tráfego necessita passar pelo RP da rede, assim nem sempre utiliza o caminho mais curto (FRANCISCO, 2000).

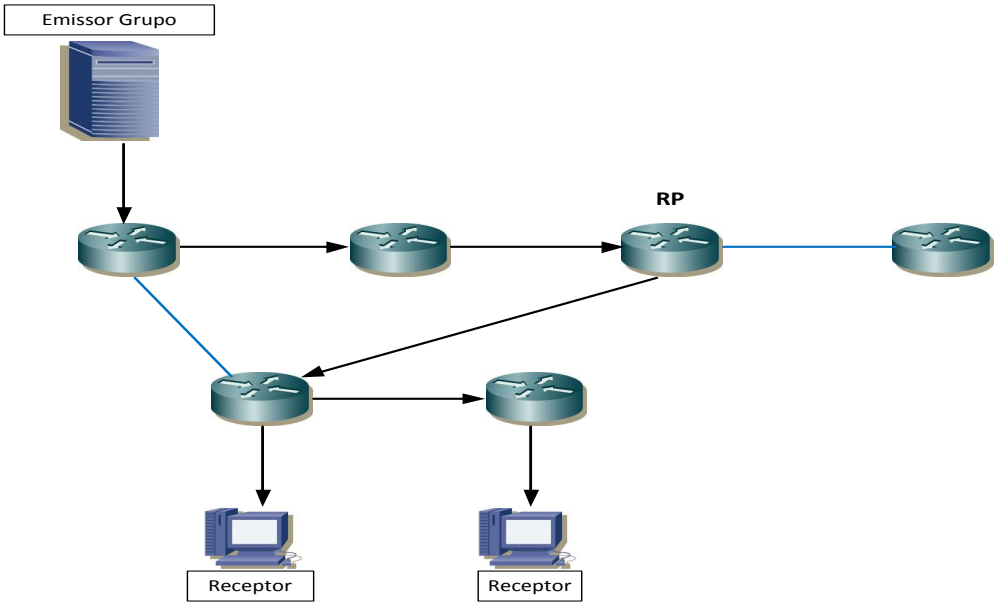

Figura 16 - Shared *Tree*
Fonte: autoria Própria

7.3 COMPARAÇÃO ENTRE OS TIPOS DE ÁRVORES

Source *Tree*s e Shares *Tree*s não apresentam situações de loop, mas cada uma tem suas vantagens e desvantagens.

As Source *Tree*s tem a vantagem de criar um caminho melhor para alcançar os destinos, isto garante uma baixa latência para o roteamento dos pacotes na rede.

No entanto, os roteadores devem manter informações sobre o caminho para cada fonte. Logo, em uma rede grande torna-se um problema devido ao consumo dos recursos do roteador.

As Shared *Tree*s mantêm poucas informações de estado em cada roteador que é uma ótima vantagem, porém com ela temos a existência de caminhos não otimizados. Isso pode gerar atrasos e má utilização dos recursos da rede.

A figura 17 mostra a comparação das duas árvores (FRANCISCO, 2000).

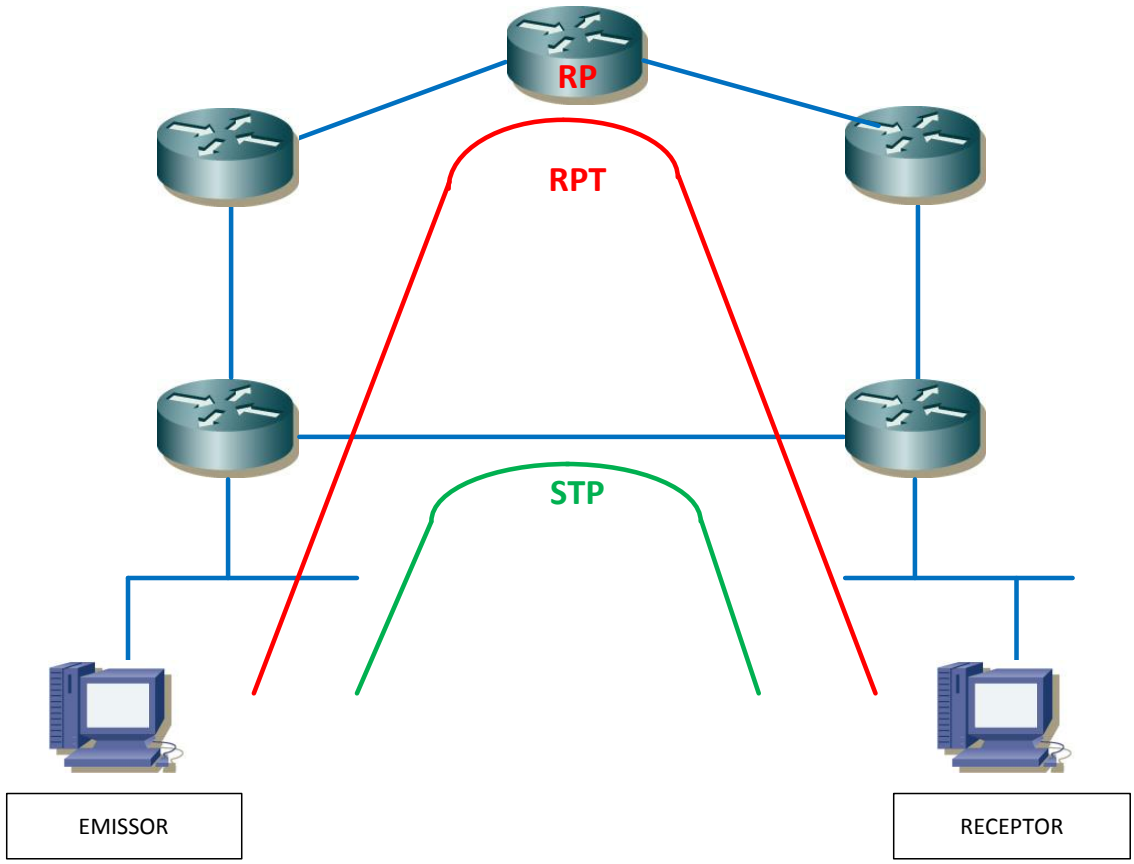

Figura 17 - Comparação RPT e STP
Fonte: Autoria Própria.

8. PROTOCOLOS DE ROTEAMENTO MULTICAST.

Antes de estudarmos os protocolos de roteamento que trabalham com transporte de datagramas *multicast*. Devemos ter em mente que a palavra *multicast* surgiu de um conceito criado pela necessidade da otimização da estrutura TCP/IP atual para suportar trafego multimídia em diversos nodos da rede sem obstruir por congestionamento os locais que são desnecessários. Porem quando este conceito surgiu, à rede TCP/IP já estava implantada e já havia toda a estrutura da internet criada. Tentar adequar todos os protocolos para suportar ao mesmo tempo *multicast* era inviável. Então a solução foi criar protocolos que operam em conjunto com o roteamento normal. Todo o roteamento *multicast* é feito via software porem operando de uma forma dependente de outros protocolos de roteamento normal. Existe o roteamento com os protocolos mais conhecidos como RIP, OSPF *(Open Shortest Path First)*, EIGRP *(Enhanced Interior Gateway Routing Protocol)*, e acima destes vem o roteamento *multicast* com os protocolos que veremos. Portanto devemos ter sempre em mente esta separação e operacionalmente configurar os dois níveis de roteamento porque eles trabalham independentemente. Isto ocorre porque os protocolos de roteamento *multicast* precisão que toda a árvore de enlaces esteja previamente montada para poder funcionar adequadamente.

O roteamento *multicast* consistem em enviarmos uma única mensagem do servidor, e está, seja replicada apenas quando for necessária para diminuir o consumo de banda. A mensagem só pode ser duplicada em regiões adequadas para garantir a entrega dos pacotes aos *host*s sem que ocorra uma multiplicação excessiva.

Outra questão é que como tudo surgiu através da necessidade foi necessária uma solução rápida. Vários desenvolvedores ao redor do mundo tentaram resolver este problema. A primeira solução definitivamente implementada foi o protocolo DVMRP (Distance Vector *Multicast* Routing Protocol) ao qual veremos agora (ARMSTRONG, 1992).

8.1 DVMRP (Distance Vector Multicast Routing Protocol)

Este protocolo como o nome já diz é um protocolo que toma como fator decisivo para criação da rota, a distância do vetor. Todos os protocolos que utilizam este método. Calculam a menor distância à rede remota para definição do melhor caminho. Cada vez que um pacote

passa por um *router* chamamos este fenômeno de hop (salto). Tudo o que os roteadores têm a fazer é determinar o menor número de hops até o destino. Alguns dos protocolos que pertencem a esta classe (Distance Vector) são o RIP e o IGRP (FILIPPETTI, 2009).

Um detalhe é que mesmo que o IGRP *(Interior Gateway Routing Protocol)* trabalhe levando em conta também a largura de banda e a métrica default e não apenas a contagem de saltos, ele é classificado como distance vector por considerar primeiramente a quantidade de hops e depois outros fatores decisivos para critério de desempate (FILIPPETTI, 2009).

O funcionamento deste processo só é permitido porque os roteadores vizinhos estão sempre mandando tabelas de roteamento. Estas tabelas são misturadas as próprias tabelas armazenadas internamente ao roteador para formar um mapa completo da rede. É um processo chamado de *routing by rumour* porque os roteadores apenas aceitam e anexam as informações sem nenhuma verificação. No entanto um destino pode ter mais de um único caminho com o mesmo número de saltos. Neste caso o RIP *(Routing Information Protocol)* por trabalhar apenas levando em conta a contagem de hops ao se deparar com este tipo de situação executará um processo chamado *round-robin load balance* e distribuirá a carga de pacotes alternadamente entre os dois ou mais caminhos encontrados, podendo realizar este processo com até seis links simultaneamente. Mas no caso do DVMRP será escolhido o caminho do roteador que tiver o menor endereço IP (WAITZMAN, 1988).

Por curiosidade na mesma situação o IGRP ainda levantará a largura de banda e a métrica para tomar a decisão adequada ao invés de dividir os pacotes balanceadamente entre os caminhos concorrentes.

No nosso caso o DVMRP será mais semelhante ao RIP pelo fato do DVMRP ter sido criado com base na implantação do protocolo RIP, mas com esta pequena diferença em casos de iguais caminhos. O DVMRP nada mais é do que uma adaptação do RIP para operar em *multicast*. Mas lembrando de que o DVMRP não fará o roteamento de pacotes normais, apenas os de *multicast*, portanto é necessário implantar um roteamento normal junto ao DVMRP. E para este caso, especificamente, devemos utilizar o RIP para construção da árvore devido ao fato do DVMRP ter sido feito com base no RIP. Sem a utilização do protocolo RIP os *router*s não podem construir sua árvore interna com a tipologia da rede.

O funcionamento do DVMRP é muito simples. Ele basicamente transforma a solicitação de envio de datagramas *multicast* em *unicast* e então repassa. O protocolo combina o funcionamento do RIP com o algoritmo *Truncated Reverse Path Broadcasting* (TRPB). Que faz um mapeamento dos *host*s pertencentes a um grupo *multicast* e então direciona os pacotes para os mesmos. Caso exista uma rede onde nenhum *host* pertence a um grupo

multicast e ele é o único *Mrouter* ele cancela o envio de *multicast* para aquela determinada sub-rede. Para reestabelecer a transmissão caso entre um *host* em um local onde a transmissão foi cancelada é rapidamente enviada uma mensagem de *Graft* e então a transmissão é reestabelecida.

O DVMRP permite a criação de túneis que servem para estabelecer a comunicação *multicast* entre dois *mrouters* interligados por um roteador comum que não possui nenhuma configuração de *multicast*.

Também foi acrescentado ao protocolo o conceito de TTL (*time-to-live*) para evitar que os pacotes muito antigos e fora do tempo de exibição fiquem propagando na rede. Assim os pacotes com o TTL muito altos são descartados.

A troca de datagramas de roteamento é feita com a utilização do protocolo IGMP, em uma pequena porção de cabeçalho IGMP de comprimento fixo, e outra porção composta por um fluxo de dados codificados (WAITZMAN, 1988).

A figura 18 mostra o datagrama DVMRP.

4	8	16	24	32
Version	Type	Sub-Type	Checksum	DVMRP Data stream

Figura 18- Datagrama DVMRP
Fonte: Autoria Própria.

Em *version*, a versão será sempre “1”. Em *Type*, o tipo para DVMRP é “3”. O subtipo pode ser:

Response. A mensagem fornece rotas para um destino. valor 1;

Request. A mensagem solicita rotas para um destino. valor 2;

Non-membership report. A mensagem fornece relatórios de não membros;

Non-membership cancellation. A mensagem serve para o cancelamento de prévios relatórios de não membros;

O Checksum é um stream resultante de uma soma da mensagem inteira excluindo o cabeçalho IP. No momento da computação o checksum é zerado.

O resto da mensagem DVMRP é um fluxo de dados etiquetados (*Tagged Data*). O motivo da utilização da Tagged Data é propiciar extensibilidade ao código. Graças a isso é

possível criar comandos novos a partir de novas tags, e também reduzir a quantidade de mensagens redundantes.

Os elementos do fluxo, como comandos de cancelamento, são múltiplos de 16 bits para ajustar adequadamente o pacote. Os comandos são como um código numérico de oito bits.

Todo o mapeamento de *hosts* e grupos com sessões abertas é feito através da junção desses pacotes quando os mesmo se referem a uma nova estrutura encontrada na rede. Cada vez que ocorre uma alteração é enviada uma tabela com a árvore atual da rede. E este processo pode ocupar certa banda de transmissão.

Na figura 18, podemos ver um exemplo de transmissão onde cada seta azul representa um pacote sendo transmitido. Basicamente é transmitido um pacote para cada *host*. Exceto para os *host*s ligados a switch que está conectada em R3. Neste caso apenas um pacote foi transmitido e a switch multiplicou para todos os *host*s interessados. Isto ocorre porque no caso deste exemplo a switch está com o IGMP configurado. Então o *router* R3 encaminhou um único pacote para o grupo *multicast*. O switch conhecendo todos os *host*s interessados pertencentes ao grupo fez a distribuição correta.

Se caso não existisse *host* pertencente a um grupo *multicast* em R4, a transmissão seria cancelada em R4 e só seria restabelecida após a mensagem de graft no momento em que um *host* entrasse no grupo (WAITZMAN, 1988).

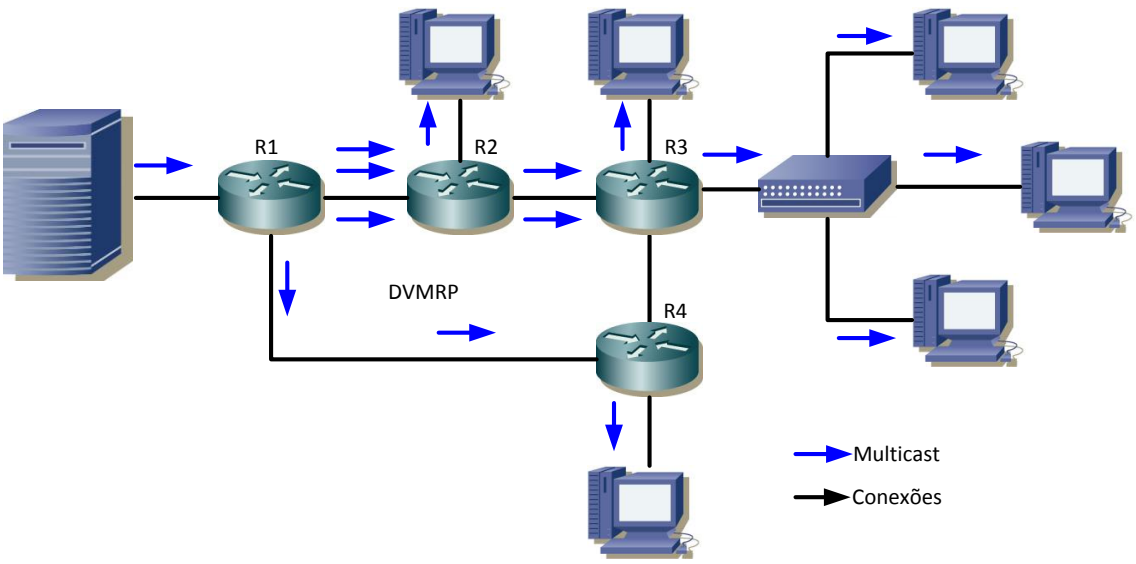

Figura 19 - Encaminhamento com DVMRP
Fonte: Autoria Própria.

Se este mesmo laboratório fosse realizado com uma transmissão *unicast* (exemplo na Figura 20) e não estivesse configurado o IGMP no switch a banda seria ainda maior como podemos ver na figura 19. O único ganho obtido com relação aos dois exemplos (figura 18 e figura 19) se deu através do switch por causa do IGMP. Se tivéssemos uma rede onde muitos *host*s estivessem ligados a switch com IGMP teríamos até certo ganho.

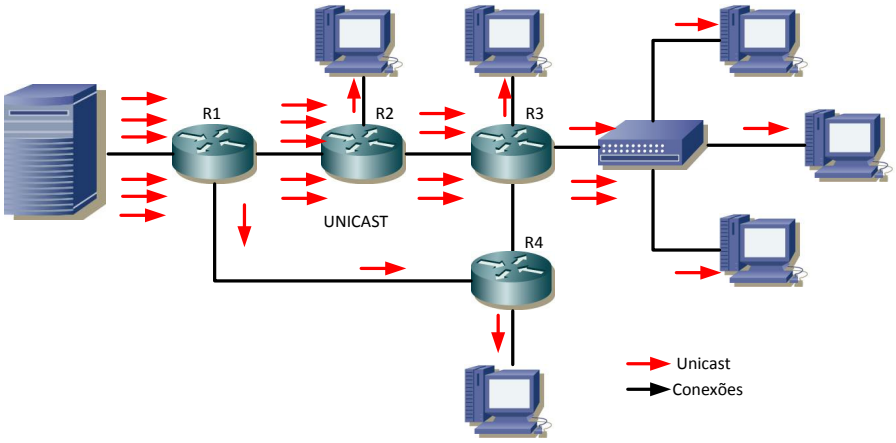

Figura 20 - Encaminhamento Unicast
Fonte: Autoria Própria.

Com estas informações podemos deduzir antecipadamente que o DVMRP atende a necessidade de controle de tráfego em demanda para grupos específicos, mas não reduz significativamente o consumo de banda em redes grandes, apenas na banda do servidor. A necessidade de um protocolo que atendesse ao crescimento do tráfego em demanda de vídeo continuou e então surgiram outros protocolos mais eficientes como veremos a seguir (WAITZMAN, 1988).

8.2 MOSPF (Multicast Open Shortest Path First)

Um grande avanço se deu na história do *multicast* com o surgimento do MOSPF que foi concebido através de uma extensão do tradicional protocolo OSPF. Como o OSPF não é proprietário e a maior parte dos equipamentos suporta este protocolo, o MOSPF ganhou uma boa aceitação e comprovou sua eficiência com relação à implementação *multicast* citada anteriormente. Em uma estrutura Mbone rodando DVMRP podemos ter sistemas autônomos

rodando MOSPF sem problema nenhum por que o MOSPF é compatível com o DVMRP e tal compatibilidade facilita muito a sua implantação.

A forma de operação do MOSPF é muito semelhante ao OSPF tradicional por ser a expansão do mesmo. Uma curiosidade é que o OSPF utiliza o conceito de *multicast* para trafegar seus datagramas de controle e informar atualizações na rede como apresentado por Marco Fillippetti.

> Esse protocolo já foi criado tendo-se em vista redes de grande porte. Ele não trabalha com propagação via *broadcast*, mas via *multicast*, ou seja, apenas routers que estejam rodando esse protocolo receberão as atualizações necessárias. Uma interface que não esteja rodando OSPF conectada a ela não receberá as mensagens de atualização via *multicast.* (FILIPPETTI, 2009, p. 268).

Os pacotes OSPF são transmitidos apenas aos roteadores que estão configurados para operar com este protocolo. Neste caso já podemos ver uma vantagem com relação ao protocolo anterior. No caso do DVMRP nós tínhamos uma situação onde cada alteração da rede (saída ou entrada de um *host*, alteração na tipologia ou em uma rota) provocava a circulação de tabelas contendo uma relação estrutural completa para todas as sub-redes conhecidas, gastando a banda e fazendo com que pacotes de controle percorressem locais desnecessários. Sem contarmos ainda que o aumento da rede faz estas tabelas ficarem proporcionalmente mais pesadas para serem transmitidas.

Outro avanço do MOSPF é que ele também passará a enviar apenas as alterações ocorridas e não mais a tabela completa a cada mudança. E agora os pacotes de controle não são mais enviados para todas as sub-redes e sim apenas para os locais de interesse.

A grande diferença é que o MOSPF não trabalha com o conceito de hops. Agora a tomada de decisão é feita levando em consideração o custo do meio. Isto ocorre porque é atribuído um valor de métrica a cada conexão. O roteador localiza todos os caminhos e escolhe onde a soma de todas as métricas for menor. Parece simples, mas nem sempre o melhor caminho é o mais curto. Podem ocorrer situações onde a banda pode compensar o tempo gasto na replicação dos pacotes. Isto também ajuda muito nas regiões onde a banda é mais estreita porque o tráfego multimídia será desviado e passará a ser suportado para atender apenas os *host* conectados aquele nó com mais qualidade. Por essas e outras características é que descreveremos melhor as principais propriedades do encaminhamento de pacotes MOSPF (MOY, 1994).

O encaminhamento dos pacotes *multicast* depende da fonte emissora e do destino *multicast*. Este roteamento é chamado Fonte/Destino (*Source/Destination*) e trabalha em

contraste com os algoritmos *unicast* (Ex. RIP, OSPF e EIGRP) que roteiam com base exclusivamente no destino. Com isto os pacotes sempre percorrem o melhor caminho.

Os pacotes enviados entre a fonte e o destino devem percorrer o caminho com o menor custo. O custo pode ser expresso, por exemplo, pelo delay na transmissão. E então o pacote percorrerá o caminho com o menor delay. A métrica pode ser configurada de acordo com a preferência. Mas de qualquer forma, um valor de métrica é atribuído a cada interface de saída do roteador e representará o seu custo.

O MOSPF toma a maior vantagem possível de caminhos comuns entre a fonte e o destino. Algumas vezes os *hosts* de um grupo estão muito espalhados na rede e faz com que os pacotes precisem ser replicados varias vezes. A idéia é tentar baixar o número de replicações em alguns nós, por isso o MOSPF replica os pacotes o mínimo possível em casos específicos.

Para um dado datagrama, todos os *routers* montam uma árvore shortest-path idêntica. Existe apenas um caminho entre a fonte e o destino (que pode ser um membro de um grupo particular). E ao contrário dos protocolos *unicast*, não há provisão de custo idêntico para múltiplos caminhos.

Em cada hop os pacotes multimídia são enviados como links de dados *multicast* diretamente. Existem duas exceções. Primeira, em redes não *broadcast*, que não suportam serviços de link de dados *multicast*/*broadcast*, e então os datagramas são encaminhados para o MOSPF vizinho. Segunda, Os roteadores MOSPF podem ser configurados para encaminhar os datagramas como links de dados *unicast* em redes específicas para evitar a Replicação excessiva em certas situações anômalas.

O encaminhamento de pacotes é um mecanismo baseado no conteúdo de um cache de dados. Este cache de dados é chamado de cache de encaminhamento e existe uma entrada separada para cada combinação de fonte/destino. A entrada serve para indicar ao nó vizinho de onde o pacote pode ter vindo (*Upstream Node*) e para onde deve ser encaminhado (*Downstream Interfaces*).

O cache de encaminhamento é formado por dois componentes. O primeiro componente é chamado de tabela do grupo local e é construído pelo protocolo IGMP. O segundo é chamado de árvore do caminho mais curto do datagrama. Esta árvore é enraizada na fonte emissora do datagrama e habilita sua entrega às redes distantes.

Os datagramas são marcados com a sua classificação de *Type of Service* (TOS), que podem variar em cinco valores mutuamente exclusivos: *minimize delay; maximize throughput; maximize reliability; minimize monitary cost e normal service.* O caminho do

pacote no MOSPF pode variar de acordo com a classificação TOS utilizada. Por exemplo, um tráfego *multicast* sensitivo ao delay (retardo) pode seguir rotas diferentes de uma aplicação *multicast* de alto *throughput* (vazão). A classificação TOS no protocolo MOSPF é, como no OSPF, opcional, e roteadores que a suportam podem ser misturados livremente como os que não a suportam.

A tabela do grupo local mantém o controle da participação no grupo nas redes diretamente ligadas ao roteador. Cada entrada da tabela do grupo local é um par que indica qual das redes locais tem um ou mais membros conectados ao IP de um grupo *multicast* formando uma árvore. Isto ajuda o roteador a decidir para qual rede local encaminhar o pacote e assim garantir que o mesmo foi entregue completamente a todos os membros de um grupo *multicast*.

Quando existem caminhos comuns na arvore, o MOSPF elimina a necessidade de transmitir vários pacotes (uma para cada *host* semelhante ao *unicast*). Por tanto, apenas uma pacote passa e quando chaga em um nó ele é multiplicado para ser entregue a um *host* ou reencaminhado a outro *router*. Esta técnica é uma das principais redutoras do consumo da banda seguindo a filosofia do *multicast*. Podemos visualizar melhor na figura 21, considerando que a métrica entre R1 e R2 somada à métrica entre R2 e R3 seja mais favorável do que entre R1 e R4. Podemos notar o enorme ganho na rede inteira (MOY, 1994).

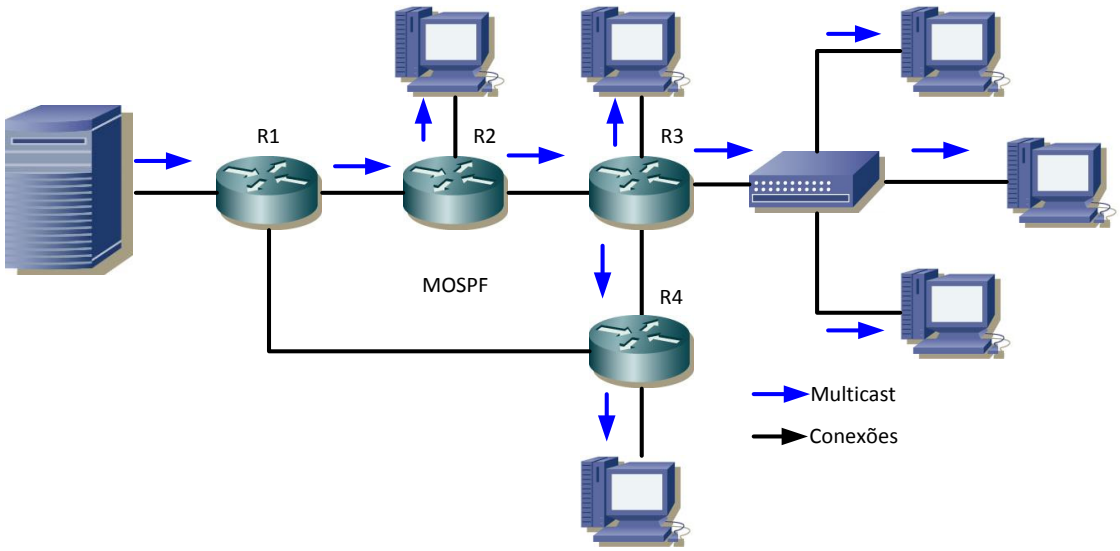

Figura 21 - Encaminhamento via MOSPF
Fonte: Autoria Própria.

O único problema de se implantar um sistema com MOSPF é que ele depende do protocolo OSPF. E caso a rede possua vários setores com protocolos diferentes (RIP, IGRP, EIGRP) ele não será tão eficiente (MOY, 1994).

8.3 PROTOCOL INDEPENDENT MULTICAST (PIM)

Ainda no início da década de 90 surgiram protocolos independentes que juntos formaram uma família denominada coletivamente por *Protocol Independent Multicast* (PIM, RFC5384). Este nome foi atribuído porque agora estes protocolos passaram a ser totalmente independente dos protocolos *unicast*. Nas situações antigas se implantássemos uma rede com MOSPF, seriamos obrigados a configurar também o OSPF normal para auxiliar com a construção da tabela de roteamento. Agora, com a utilização do PIM este problema acabou. O PIM utiliza a tabela de roteamento existente no roteador sem se importar com a forma com que ela foi elaborada. Se for usando RIP, IGRP, OSPF, EIGRP ou qualquer outro, funcionará normalmente. A implantação, por tanto, ficou muito mais fácil (ADAMS, 2005).

8.3.1 PIM-DM (Protocol Independent Multicast – Dense Mode)

Das famílias PIM existentes, as mais conhecidas e implantadas são a PIM-DM (PIM *Dense-Mode*, RFC 3973) e a PIM-SM (PIM *Sparse-Mode*). Apesar de serem protocolos independentes cada um possui características diferentes. No caso do PIM-DM, A forma de trabalhar é muito semelhante ao DVMRP. Os pacotes são enviados igualmente para todos os *router*s e segmentos da rede de forma *unicast*.

Este protocolo não é muito eficiente em grandes redes pelos mesmos motivos do DVMRP. No entanto algumas diferenças foram feitas e deixaram o PIM-DM um pouco melhor. Uma delas foi a utilização do RPF para prevenir loops devido a replicação excessiva dos pacotes.

Outra diferença principal é que a transmissão pode ser podada nos locais onde não existirem membros de um grupo *multicast*. Essa poda possui uma vida útil que quando expirada a transmissão é restabelecida para aquele local. Este processo pode ser reiniciado várias vezes. Quando ocorre um estado de poda, também chamado em inglês por *Prune-State*, este estado é associado a um par (S, G) e quando um usuário entra em um grupo G no mesmo momento, é feito um “enxerto” e os pacotes passam a ser encaminhados da fonte S para o usuário do grupo G. transformando o ramo podado em um ramo de encaminhamento (ADAMS, 2005).

Podemos ver um exemplo com o mesmo laboratório feito anteriormente na figura 21. Vamos considerar que neste caso nenhum *host* pertence a um grupo no *routers* R4. Portanto ele está em estado de poda (figura 22). Caso entre um *host* em R4. Automaticamente é feito um enxerto e os pacotes passam a ser reencaminhados como mostra a figura 23 (ADAMS, 2005).

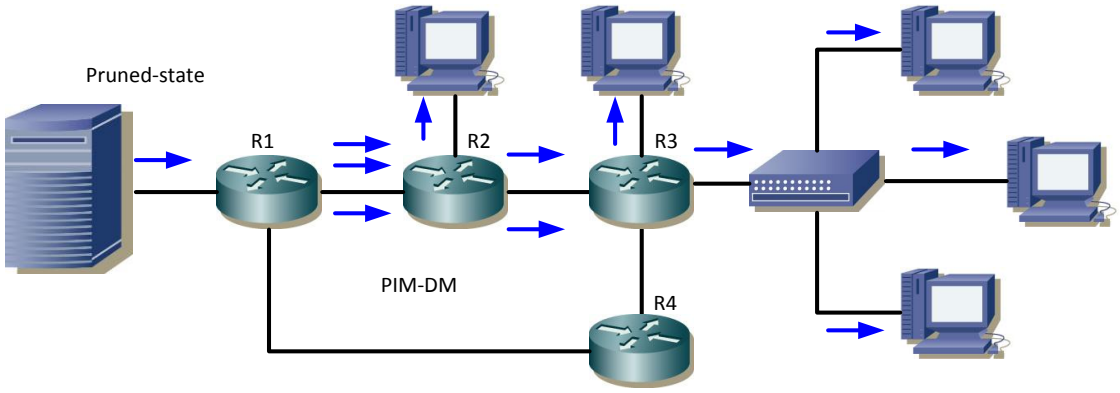

Figura 22 - Encaminhamento com PIM-DM em poda no router R4
Fonte: Autoria Própria.

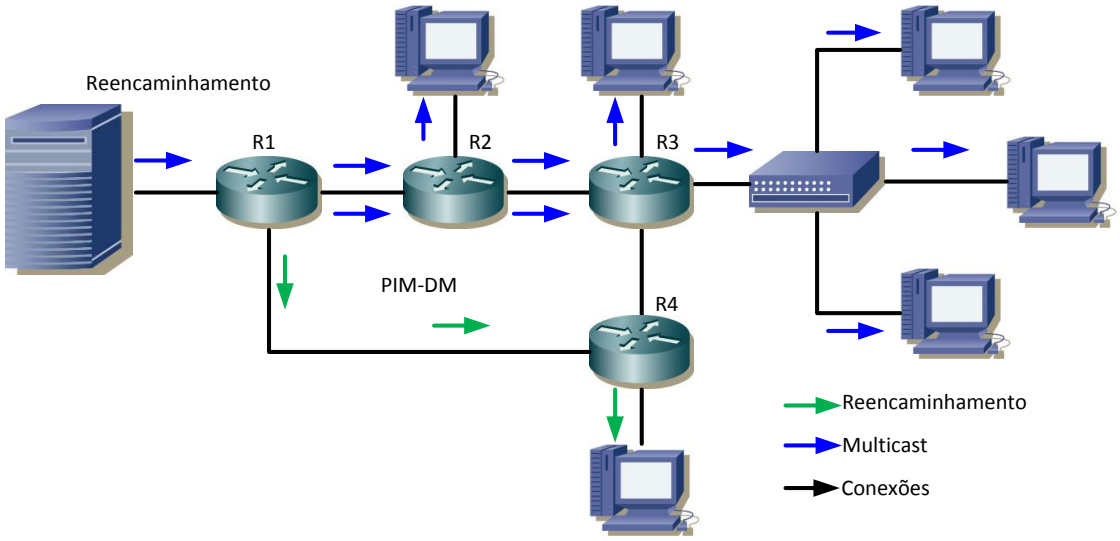

Figura 23 - Encaminhamento com PIM-DM
Fonte: Autoria Própria.

8.3.2 PIM-SM (Protocol Independent Multicast – Sparse Mode)

O protocolo PIM *Sparse-mode* foi construído para operar em redes muito dispersas. Uma grande diferença para os protocolos de modo denso é que agora os roteadores devem manifestar interesse em receber o tráfego *multicast*. Nos outros protocolos citados anteriormente era considerado que todos os roteadores do mapa deveriam receber o tráfego a menos que fosse enviada uma mensagem de desligamento.

Outra diferença é que agora foi introduzido o conceito de "ponto de encontro" ou *Rendavous-Point* (RP). Em cada domínio existe um conjunto de roteadores atuando como RP's, mas cada grupo deve ter um único RP para não gerar conflito.

Todos os *host*s devem estar configurados com um roteador designado (DR-*Designatede Router)* é um roteador com o maior numero IP da sub-rede. Quando alguém entra em um grupo enviando uma solicitação IGMP ao DR, o DR começa a buscar o RP daquele grupo. Esta busca é feita utilizando um espalhamento determinístico sobre todos os RP's pertencentes ao domínio. Depois de encontrado o RP, o DR envia uma solicitação de inclusão ao RP do grupo (Figura 24). O RP passará então a enviar os pacotes *multicast* referentes ao grupo para o DR que por sua vez entregará à sub-rede que contiver o *host* que acabou de entrar no grupo (FEANER, 2006).

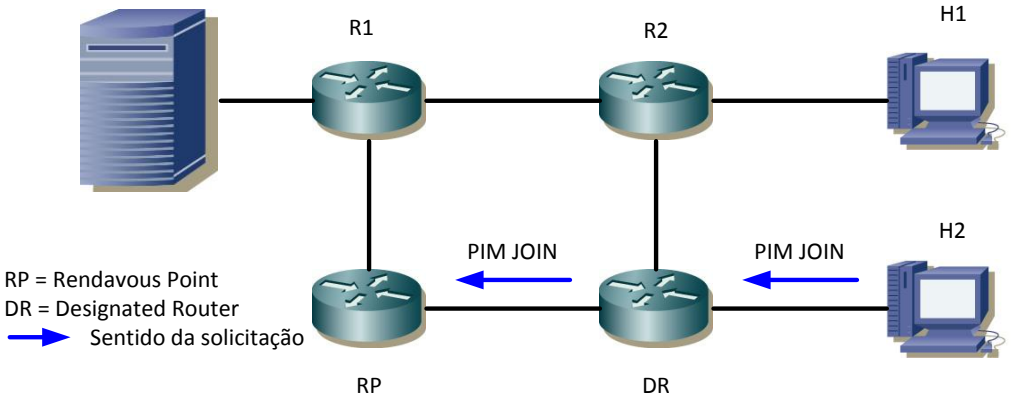

Figura 24 - Entrada de um novo host com PIM-SM
Fonte: Autoria Própria.

Quando um servidor inicia a transmissão *multicast* para um grupo, o DR do servidor encapsula o primeiro pacote em um pacote de registro PIM-SM chamado de *PIM-Register* e envia ao RP diretamente por *unicast*. Ao receber esta mensagem o RP devolve um PIM-*Join* ao DR do servidor. Feito isso os roteadores intermediários adicionam em sua tabela de replicação *multicast* o par (S, G) facilitando o envio dos próximos pacotes multimídia direto para o RP. É interessante notar que depois que a fonte começa a mandar os pacotes multimídia para o RP, estes pacotes são encaminhados de forma *unicast* e só depois que chegam ao RP é que são enviados para os *host*s de forma *multicast*. Como mostra a figura 25.

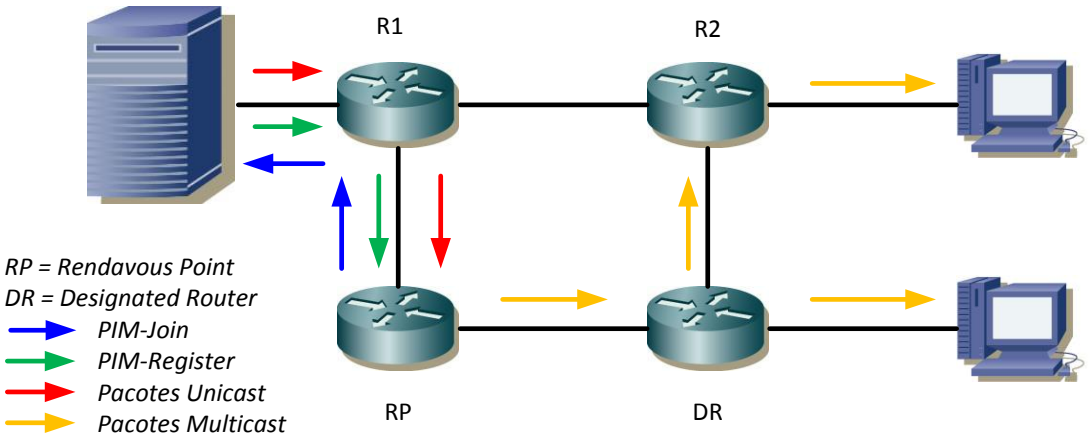

Figura 25 - Entrada de uma Fonte e encaminhamento PIM-SM
Fonte: Autoria Própria.

Este processo parece ser capaz de resolver a maior parte dos problemas de tráfego. Mas em grandes redes também podemos ter muitos usuários conectados. Fazendo com que o *router* RP fique sobrecarregado de multiplicar pacotes e controlar tudo sozinho. Então o PIM além de trabalhar com a árvore RP também pode trabalhar ao mesmo tempo com a árvore do caminho mais curto para tentar desviar o tráfego. A utilização da árvore do caminho mais curto pode ser aplicada separadamente.

Quando a arvore do menor caminho é encontrada, o roteador pode mandar uma mensagem de desligamento para o RP e continuar apenas a mandar as mensagens pelo caminho mais curto. Como mostra na figura 26, temos uma situação onde o *host* H1 passou a receber tráfego multimídia pelo caminho mais curto. E se caso o *host* H2 saísse da transmissão, seria enviada uma mensagem de desligamento para o RP e os pacotes passariam a ser enviados diretamente pelo caminho mais curto até o *host* H1 (FEANER, 2006).

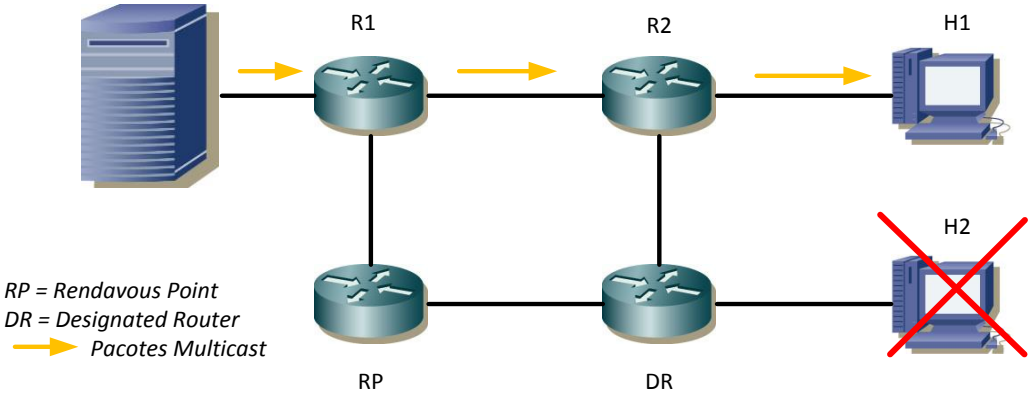

Figura 26 - Encaminhamento PIM-SM pelo método da árvore mais curta
Fonte: Autoria Própria.

9. EXPERIMENTOS

Para ter contato com a tecnologia *multicast*, foram propostos experimentos, com intuído de possibilitar um melhor entendimento e avaliação do funcionamento desse tipo de transmissão.

Num primeiro momento foram propostos cenários virtuais, fazendo-se uso de simuladores de rede para emular a rede rodando *multicast*. O software escolhido havia sido o GNS3. Ao decorrer do desenvolvimento do trabalho, houve a possibilidade de realizar os testes com equipamentos reais. Sendo assim todos os cenários aqui propostos foram desenvolvidos em equipamentos de rede reais (roteadores, switches, computadores e cabos).

9.1 ZABBIX

Ao decorrer do trabalho, vários softwares foram testados para serem usado como padrão para avaliar o tráfego e construir gráficos de ocupação de banda nas redes propostas. Dentre eles estavam o MRTG *(Multi Router Traffic Grapher)*, CACTI, ZABBIX. Escolhemos o Zabbix devido a sua facilidade de instalação e operação.

> O Zabbix é um software que monitora diversos parâmetros de uma rede como a integridade e desempenho dos servidores. Oferece excelentes relatórios e visualização de dados de recursos com base nos dados armazenados. Usa um mecanismo de notificação flexível que permite aos usuários configurar um e-mail com alertas para qualquer evento, o que permite uma reação rápida para os problemas de uma rede de dados ou até mesmo de um servidor.
> Corretamente configurado, o Zabbix pode desempenhar um papel importante no controle da infraestrutura de TI. Isto é igualmente verdade para as pequenas organizações com alguns servidores e para grandes empresas com um grande número de servidores (DRIEMEYER, 2012).

Para o nosso ambiente fizemos o download do Zabbix na forma de Appliance sobre a plataforma em Linux, ou seja, um único arquivo que quando importado em algum aplicativo gerenciador de máquinas virtuais (VMware Workstation), cria uma VM com o sistema operacional Linux, e junto ao SO uma versão pré-instalada e pré configurada do Zabbix, sendo necessários apenas alguns ajustes de configuração para o funcionamento do servidor.

Na figura 27 a VM do Zabbix criada no VMware Worstation após importação.

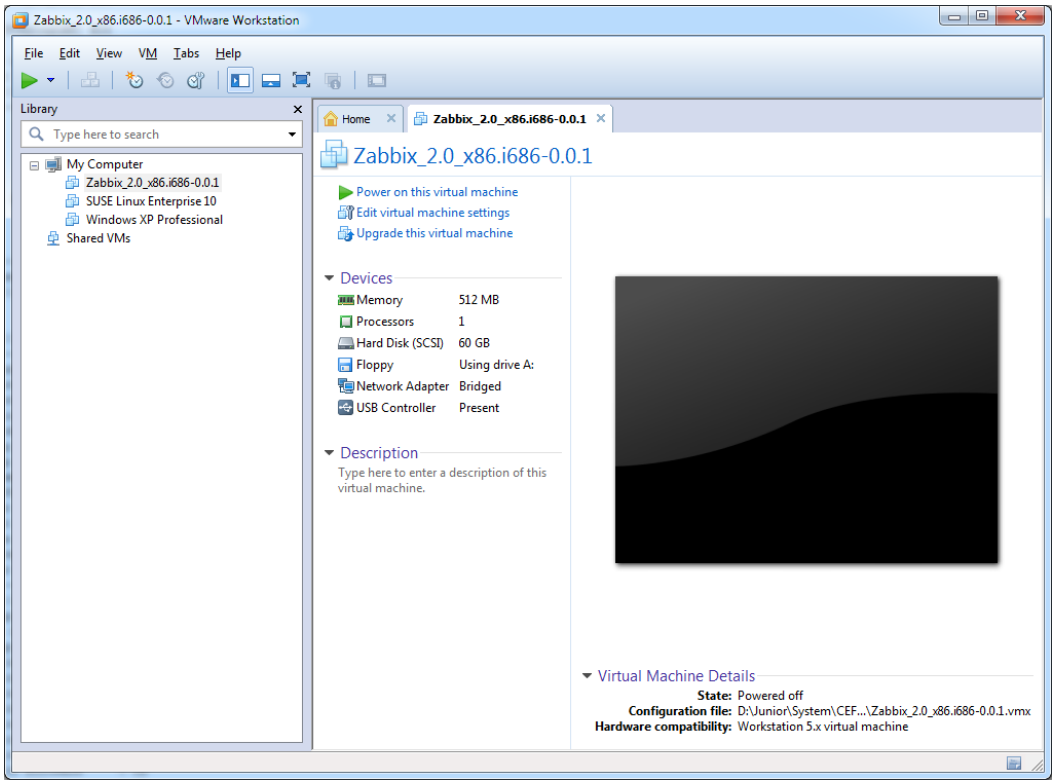

Figura 27 - VM do Zabbix
Fonte: Autoria própria

Depois de importada, a VM funciona igual a um servidor físico. As configurações do zabbix foram feitas de duas maneiras: através da tela de console do Linux, como mostra na figura 28 e através da interface web como mostra na figura 29.

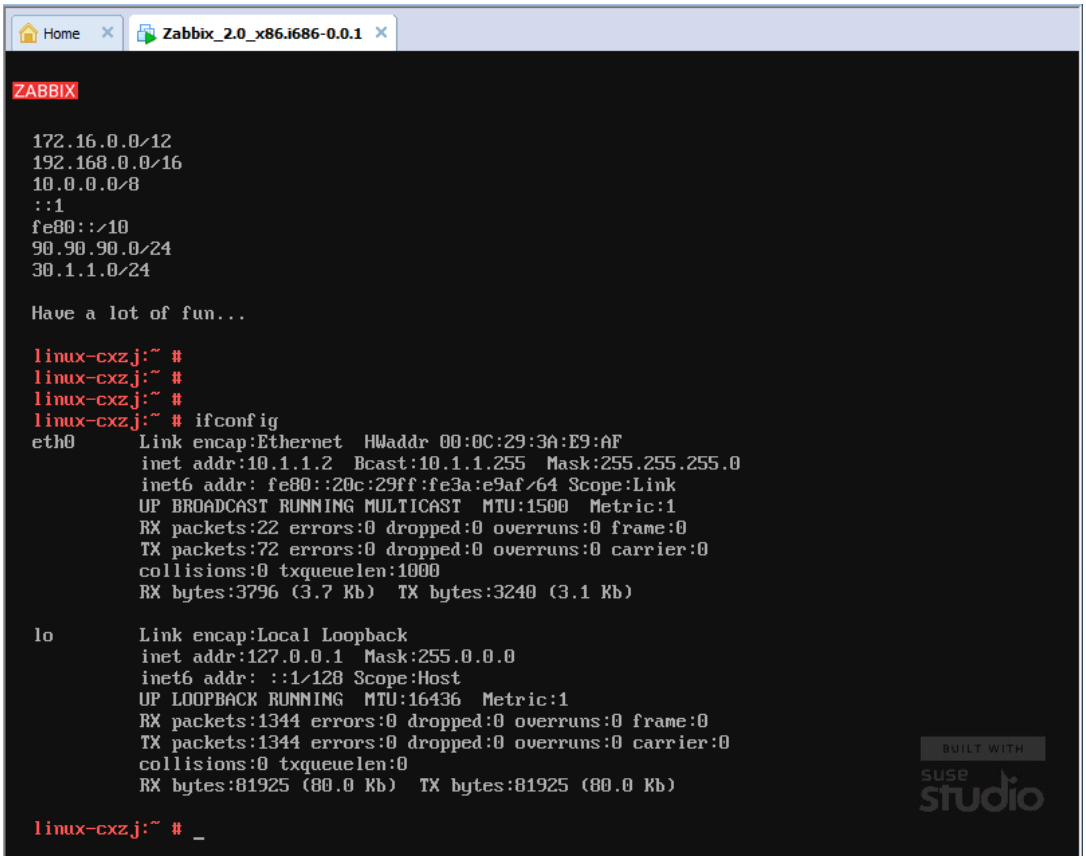

Figura 28 - Zabbix tela do console
Fonte: Autoria própria

Figura 29 - Zabbix interface Web
Fonte: Autoria própria

Nosso experimento foi todo configurado para trabalhar através do SNMP *(Simple Network Management Protocol)*, para isto necessitamos efetuar as configurações no Zabbix

para que este pudesse ser capaz de receber e verificar todo tipo de evento SNMP. As configurações foram feitas basicamente como mostra a figura 30.

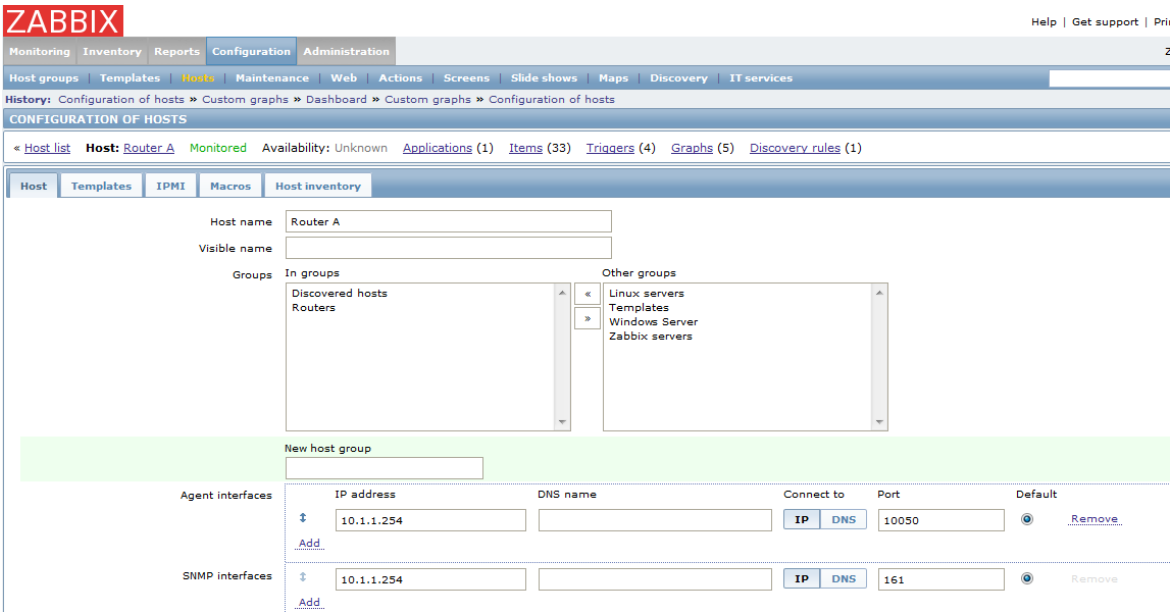

Figura 30 - Configurações Zabbix
Fonte: Autoria própria

Após os eventos SNMP terem sido recebidos, estes são interpretados e armazenados. Com o acumulo destes eventos os gráficos podem ser gerados para posterior analise, como na Figura 31

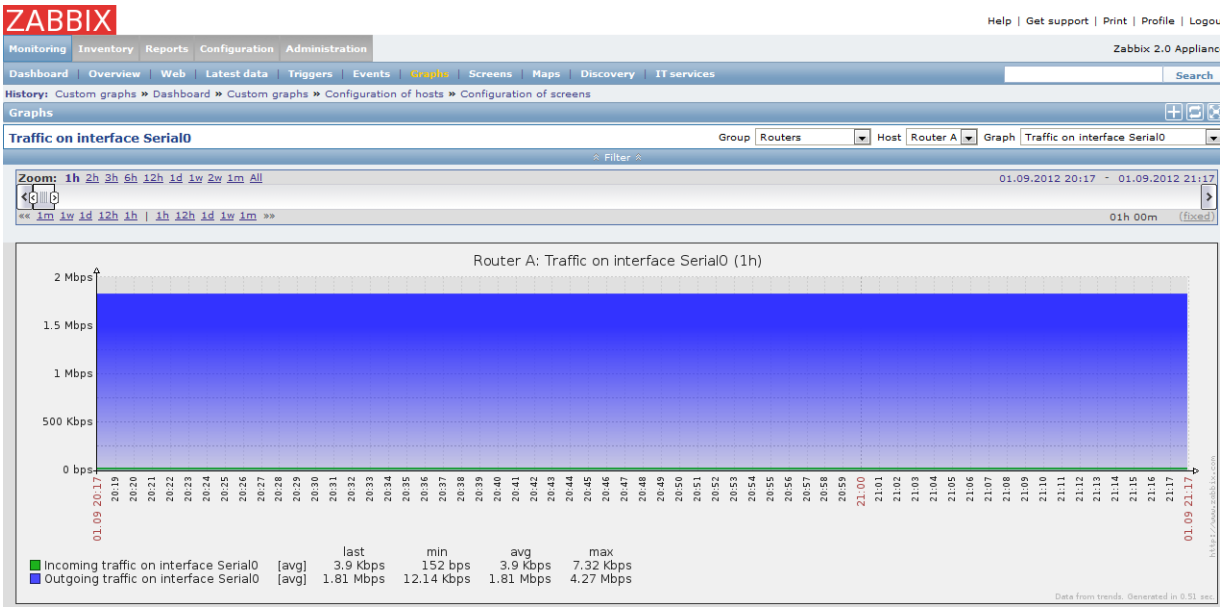

Figura 31 - Zabbix gerando gráficos a partir de pacotes SNMP
Fonte: Autoria própria

9.2 EXPERIMENTO 1 – MULTICAST EM UMA REDE LAN

9.2.1 Recursos Utilizados

Figura 32 - Roteador Cisco
Fonte: (CERTIFICATION KITS, 2012)

1x – Roteador Cisco 1751V;

Versão do firmware: Version 12.2(1r)XE1, RELEASE SOFTWARE (fc1)

Figura 33 - Switch 3COM
Fonte: (ZDTRONIC, 2012)

1x – Switch 3Com 4500 SuperStack 3
Versão do firmware: 3Com OS V3.03.02s168p15

Foi montado o experimento de uma rede local conforme a Figura 34. A partir deste foi feita a transmissão de um stream de vídeo nos modos ponto-a-ponto, *broadcast* e *multicast*, assim foi possível comparar a utilização de banda na interface do servidor do vídeo.

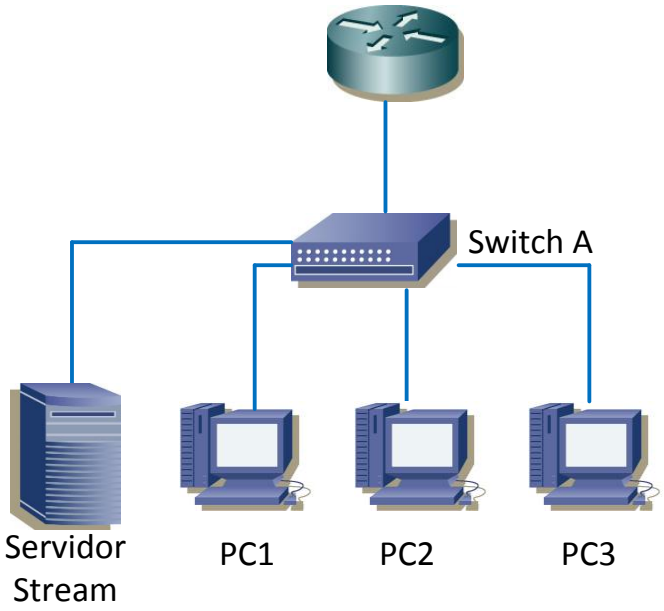

Figura 34 - Experimento
Fonte: Autoria Própria

Todas as configurações foram feitas a partir da VLAN 90, e os IPs foram distribuídos conforme a tabela 2.

Equipamento	IP	Mascara	GW	Interface
Router A	90.90.90.254	255.255.255.0	N/A	FastEthernet0/0
Switch A	90.90.90.253	255.255.255.0	90.90.90.254	Ethernet1/0/24
Servidor Stream	90.90.90.2	255.255.255.0	90.90.90.254	Ethernet1/0/2
PC 1	90.90.90.1	255.255.255.0	90.90.90.254	Ethernet1/0/1
PC 2	90.90.90.3	255.255.255.0	90.90.90.254	Ethernet1/0/3
PC 3	90.90.90.4	255.255.255.0	90.90.90.254	Ethernet1/0/4

Quadro 2 - IPs Utilizados
Fonte: Autoria Própria

9.2.2 Configuração Switch

```
#vlan 90
#
interface Vlan-interface90
 ip Address 90.90.90.253 255.255.255.0
#

#
interface Ethernet1/0/1
 port access vlan 90
#
interface Ethernet1/0/2
 port access vlan 90
#
interface Ethernet1/0/3
 port access vlan 90
#
interface Ethernet1/0/4
 port access vlan 90

#
```

```
interface Ethernet1/0/24
 port link-type trunk
 port trunk permit vlan all
#
#
 ip route-static 0.0.0.0 0.0.0.0 90.90.90.254 preference 60
#
```

9.2.3 Configuração Roteador

```
hostname TCC_UTFPR
!
ip multicast-routing
!
interface FastEthernet0/0.90
 description *** VLAN 90 TCC ***
 encapsulation dot1Q 90
 ip Address 90.90.90.254 255.255.255.0
!
snmp-server community public RO
snmp-server trap-source FastEthernet0/0.90
snmp-server source-interface informs FastEthernet0/0.90
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps ds1
snmp-server enable traps eigrp
snmp-server enable traps tty
snmp-server enable traps xgcp
snmp-server enable traps aaa_server
snmp-server enable traps isdn call-information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-not-avail
```

snmp-server enable traps isdn ietf
snmp-server enable traps icsudsu
snmp-server enable traps hsrp
snmp-server enable traps config
snmp-server enable traps entity
snmp-server enable traps cpu threshold
snmp-server enable traps config-copy
snmp-server enable traps flash insertion removal
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change shamlink interface-old
snmp-server enable traps ospf cisco-specific state-change shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps syslog
snmp-server enable traps cnpd
snmp-server enable traps rtr
snmp-server enable traps atm subif
snmp-server enable traps pim neighbor-change rp-mapping-change invalid-pim-message
snmp-server enable traps ipmulticast
snmp-server enable traps mvpn
snmp-server enable traps msdp
snmp-server enable traps rsvp
snmp-server enable traps pppoe
snmp-server enable traps l2tun session
snmp-server enable traps bgp
snmp-server enable traps ipmobile
snmp-server enable traps dial

snmp-server enable traps dsp card-status
snmp-server enable traps event-manager
snmp-server enable traps voice poor-qov
snmp-server enable traps voice fallback
snmp-server enable traps dnis
snmp-server host 90.90.90.1 public

Foi necessário habilitar o serviço SNMP no roteador para conseguirmos capturar os pacotes da sua interface e assim gerar os gráficos de ocupação.

9.2.4 Avaliando Pacotes IGMP na Rede

Utilizamos o PC3(90.90.90.4) com o Wireshark para coletar os pacotes IGMP que estavam trafegando na rede durante todos os testes.

Em 2 minutos de coleta, foram capturados pacotes de IGMP como o *Membership Query*(224.0.0.1) enviado pelo *router* em intervalos de 1 minuto conforme figura 35.

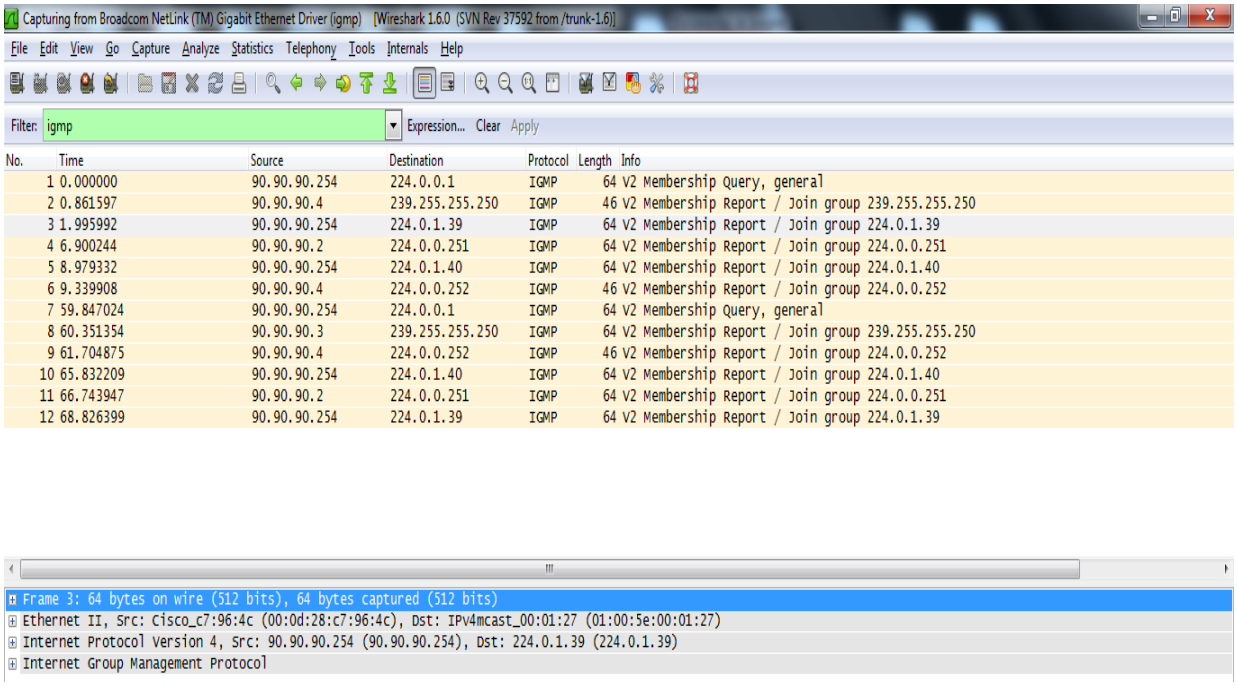

Figura 35 - Pacotes capturados antes dos testes

Fonte: Autoria Própria

Também foram capturados os pacotes de LLMNR - *Link-Local Multicast Name Resolution* (224.0.0.252), protocolo de resolução de nomes similar ao DNS. Este protocolo já resolve endereços em IPv6, diferentemente do NETBIOS que só resolve IPv4. Com o LLMNR o próprio *host* resolve os nomes numa rede sem DNS, utilizando a porta UDP 5355 com um endereço IPv6 especifico, já o NETBIOS usa *broadcast*. O LLMNR já vem nos seguintes sistemas operacionais Win Vista, Win 7 e Win Server 2008.

Outro pacote capturado foi o SSDP – *Simple Service Discovery Protocol* (239.255.255.250), protocolo este utilizado para descoberta de serviços na rede como UPnP (*Universal Plug enad Play*).

A partir dessa rápida coleta já percebemos que pacotes *multicast* trafegam em nossa rede a partir de serviços já definidos pelos próprios softwares e Sistemas Operacionais instalados em nossas máquinas.

Com o comando *show arp – a* a partir da mesma máquina, podemos verificar os endereços IPs da rede e seus respectivos MACs, conforme figura 36.

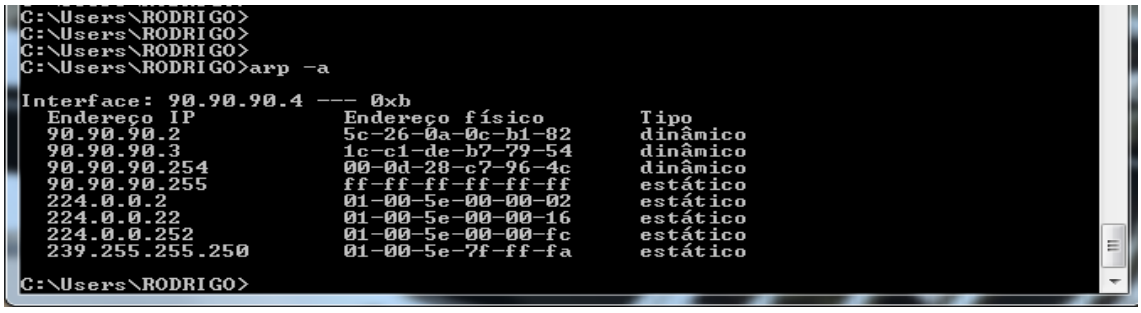

Figura 36 - Endereços Físicos
Fonte: Autoria Própria

Nos endereços *multicast* o mapeamento do MAC é montado a partir do prefixo *multicast* (01-00-5E) mais os 23 bits menos significativos do IP, como já explicado anteriormente. Na figura 37 é mostrado um exemplo do mapeamento, foi utilizado o endereço 239.255.255.250 (SSDP) para este exemplo.

Endereço nivel 3			239	255	255	250
HEXA			EF	FF	FF	FA
Binario			1 1 1 0 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 0 1 0
			5 bits perdidos	23 menos significativos		
HEXA	01	00	5E	7F	FF	FA
Binário	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 1 0 1 1 1 1 0	0 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 0 1 0
Endereço multicast nivel 2	01	00	94	127	255	250

Figura 37 - Mapeamento de endereço para Multicast
Fonte: Autoria Prórpia

Avaliando no roteador temos apenas alguns IPs de *Multicast* na sua tabela de roteamento conforme figura 38.

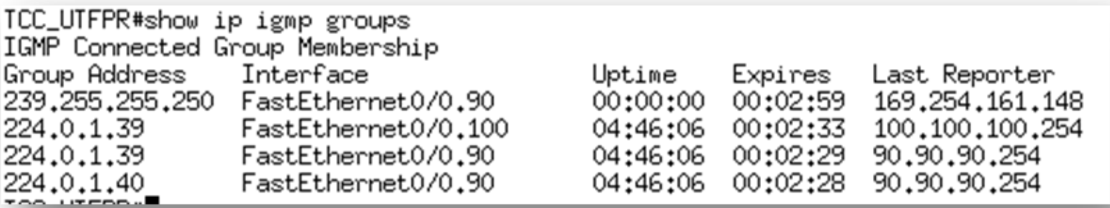

Figura 38 - Grupos IGMP no Roteador
Fonte: Autoria Própria

9.2.5 Transmissão *Unicast*

Para esse experimento foram criados 3 streams de vídeo a partir da máquina servidora(90.90.90.2). Cada stream teve como endereço de destino o IP de um *host* da rede (PC1, PC2 e PC3).

Monitorando a interface ethernet do servidor por cerca de 20 minutos, tivemos um pico de 4,45 Mbps *(Mega Bits per Second)* e uma média de 2,5 Mbps durante o intervalo de teste. A figura 39 mostra a coleta neste intervalo.

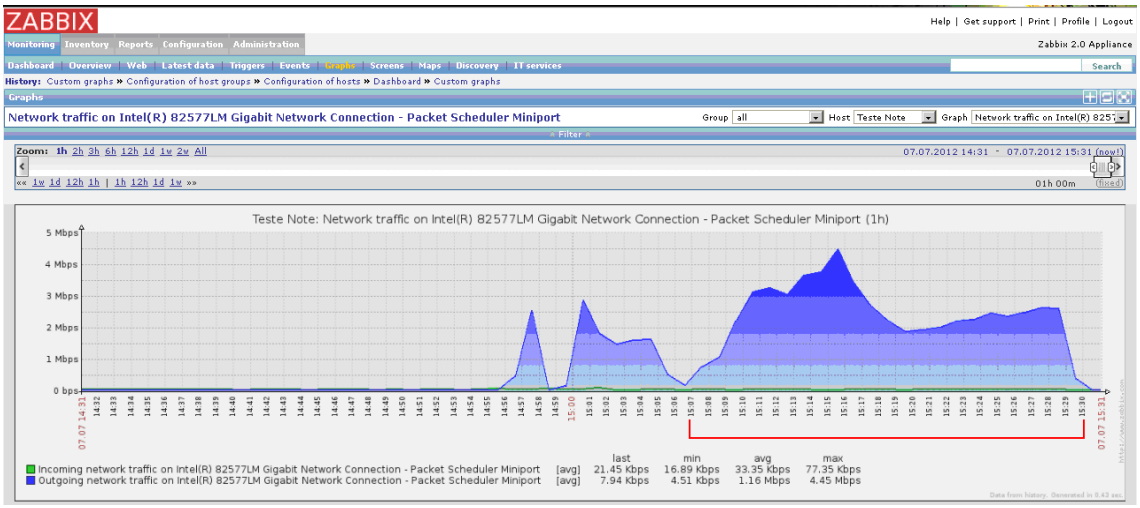

Figura 39 - Ocupação da banda em Unicast
Fonte: Autoria Própria

9.2.6 Transmissão *Broadcast*

Para esse experimento foi criado um único stream de vídeo a partir da máquina servidora (90.90.90.2), com o IP de destino 90.90.90.255(*broadcast* da rede).

Monitorando a interface ethernet do servidor por cerca de 20 minutos, tivemos um pico de 1,2 Mbps e uma média de 0,6 Mbps durante o intervalo de teste.

A figura 40 mostra a coleta neste intervalo.

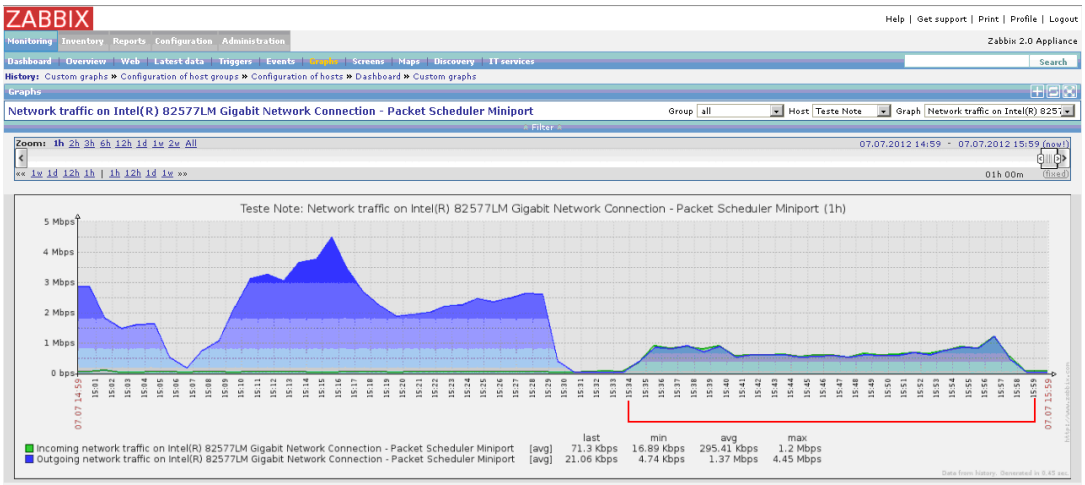

Figura 40 - Ocupação da banda em Broadcast.
Fonte: Autoria Própria

9.2.7 Transmissão *Multicast*

Para esse experimento foi criado um único stream de vídeo a partir da máquina servidora(90.90.90.2), com o IP de destino 239.100.100.100(endereço do intervalo de IPs *Multicast*). Vale salientar que iniciamos o s*Tree*am com o IGMP Snooping desabilitado no switch.

Monitorando a interface ethernet do servidor por cerca de 20 minutos, tivemos um pico de 1,65 Mbps e uma média de 1 Mbps durante o intervalo de teste.

A figura 41 mostra a coleta neste intervalo.

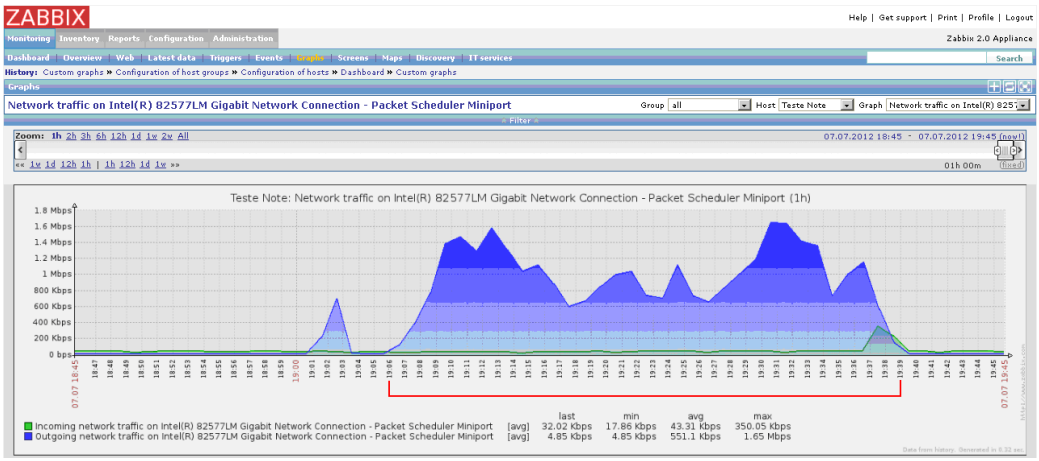

Figura 41 - Ocupação da banda em multicast.
Fonte: Autoria Própria

Na figura 42 é possivel ver no wireshark os *host*s requisitando entrada no grupo *multicast* do IP 239.100.100.100, ou seja as mensagens de *join*. Conseguimos ver todos requisitando entrada no grupo pois o IGMP Snooping não está habilitado, sendo assim as mensagens IGMP são tratadas como *broadcast* pelo switch.

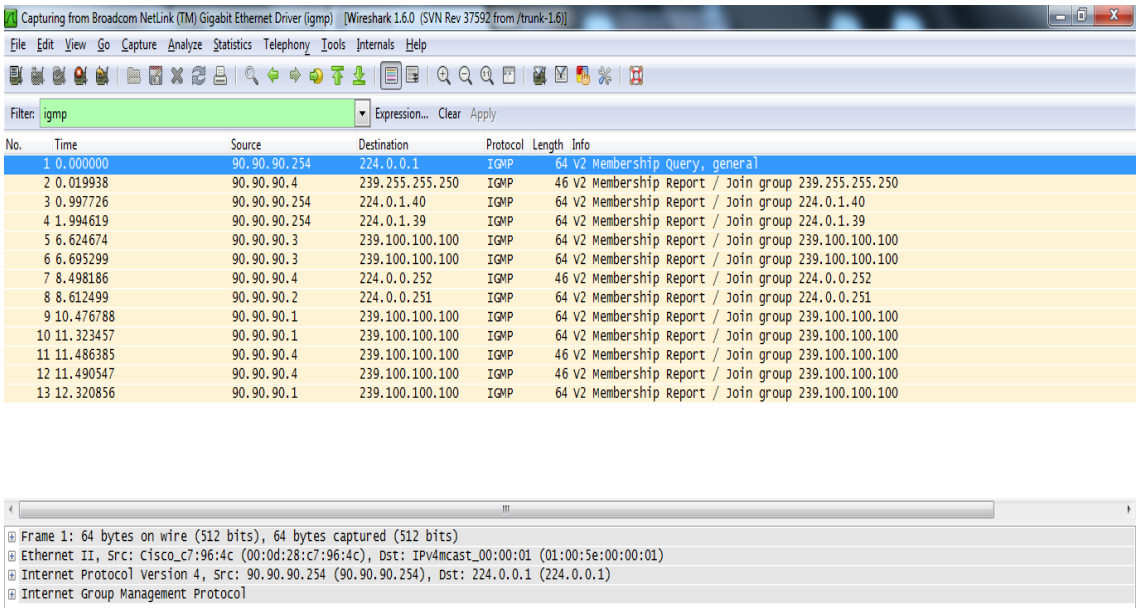

Figura 42 - Pacotes IGMP capturados no WireShrk durante o teste Multicast
Fonte: Autoria Própria

Verificado na figura 43 o grupo *multicast* criado no roteador.

```
TCC_UTFPR#show ip igmp groups
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
239.255.255.250  FastEthernet0/0.90       01:15:12  00:02:49  90.90.90.2
239.100.100.100  FastEthernet0/0.90       00:04:14  00:02:49  90.90.90.4
224.0.1.60       FastEthernet0/0.90       00:15:10  00:02:49  90.90.90.3
224.0.1.39       FastEthernet0/0.100      08:58:17  00:02:58  100.100.100.254
224.0.1.39       FastEthernet0/0.90       08:58:17  00:02:51  90.90.90.254
224.0.1.40       FastEthernet0/0.90       01:16:27  00:02:50  90.90.90.254
```

Figura 43 - Grupos criados no roteador(Destacado em vermelho o grupo criado)
Fonte: Autoria Própria

Após alguns minutos de teste foi habilitado o IGMP globalmente no switch com o comando *igmp-snooping enable* e individualmente em cada vlan com o mesmo comando.

Na figura 44 a partir do intervalo número 132 foi habilitado o IGMP Snooping. Percebe-se em vermelho que quando o *router* envia a mensagem de *query*, o *host* que está capturando os pacotes no wireshark consegue ver as mensagens IGMP dos outros *hosts* da rede. A partir do intervalo número 133 o IGMP Snooping foi habilitado, assim quando o *router* envia a mensagem de *query*, apenas a resposta da própria interface é mostrada, ou seja, com o IGMP Snooping os *hosts* não ficam recebendo tráfego desnecessário de outros pontos da rede. Pode-se perceber esse comportamento destacado em verde na imagem.

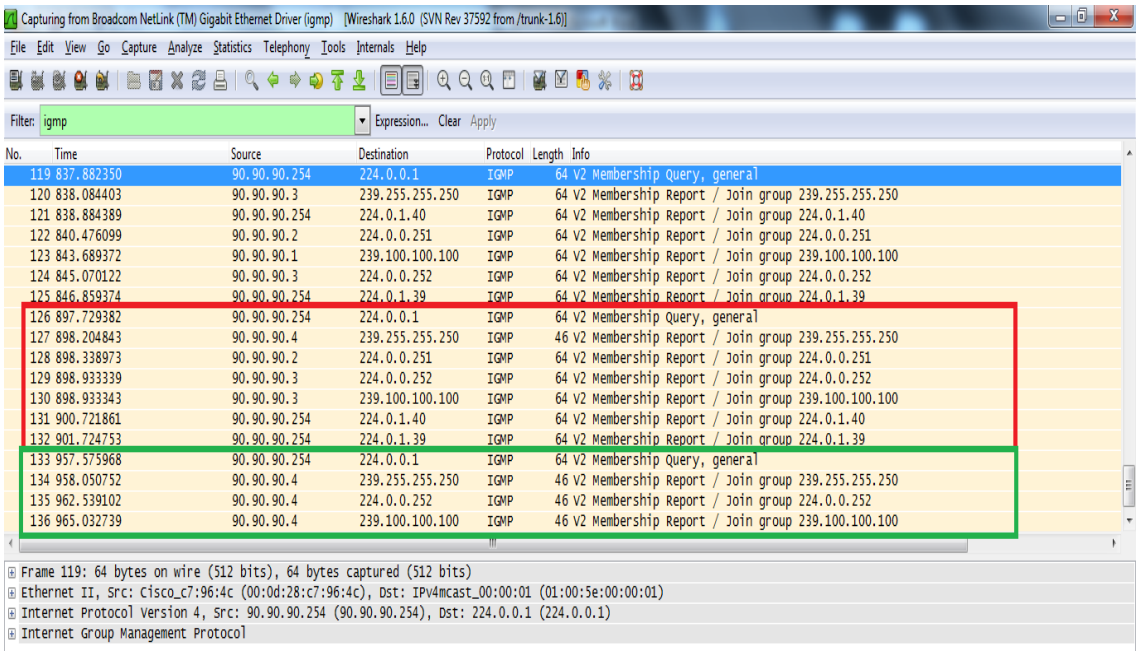

Figura 44 - Captura após IGMP Snooping habilitado
Fonte: Autoria própria

Na figura 45 é mostrado o Grupo *Multicast* criado no switch quando igmp snooping está habilitado.

```
[4500]display igmp-snooping group
  Total 5 IP Group(s).
  Total 5 MAC Group(s).

  Vlan(id):90.
    Total 5 IP Group(s).
    Total 5 MAC Group(s).
    Static Router port(s):

    Dynamic Router port(s):
                    Ethernet1/0/24
    IP group(s):the following ip group(s) match to one mac group.
        IP group address:239.100.100.100
        Static host port(s):
        Dynamic host port(s):
                    Ethernet1/0/2                Ethernet1/0/4
```

Figura 45 - IGMP Snooping no Switch
Fonte: Autoria Própria

9.2.8 Comparação Final dos 3 Tipos de Transmissão.

A tabela 3 e a figura 46 comparam a ocupação na porta do servidor nos três tipos de transmissão.

Tabela 1 - Tabela Comparativa dos 3 tipos de Transmissão

Tipo de Transmissão	Ocupação Média (Mbps)	Pico (Mbps)
Ponto-a-Ponto	2,5	4,45
Broadcast	0,6	1,2
Multicast	1	1,65

Fonte: Autoria Própria

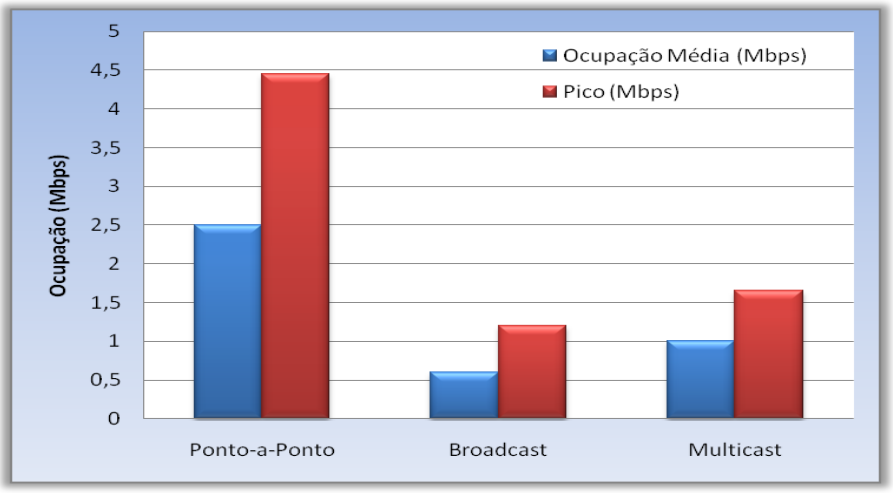

Figura 46 - Gráfico comparativo dos 3 tipos de transmissão
Fonte: Autoria Própria

Podemos ver claramente a eficiencia do *Broadcast* e *Multicast* em comparação com a tranmissão Ponto-a-Ponto. Se levarmos em conta que tinhamos apenas três *host*s requisitando o video já percebemos uma grande difereça na ocupação.

A diferença seria muito maior se tivessemos centenas de *host*s requisitando o mesmo stream.

Pudemos concluir que numa rede local a transmissão *broadcast* e *multicast* tiveram numeros bem próximos, mas devemos salientar que tivemos uma qualidade de imagem e

sincronismo muito superior quando transmitimos em Multicas se comparado com o *broadcast*.

Se somarmos a eficiencia da utilização do IGMP Snooping na rede, demostramos uma eficiencia melhor que a do *broadcast* em relação a pacotes que estão trafegando entre os *host*s.

Na figura 47 é mostrado o cenário fisico do experimento.

Figura 47 - Foto do experimento em funcionamento
Fonte: Autoria Própria

9.3 EXPERIMENTO 2 – *MULTICAST* EM REDES DISTINTAS

9.3.1 Recursos Utilizados

Equipamentos Utilizados:

Figura 48 - Roteador Cisco
Fonte: (CERTIFICATION KITS, 2012)

2x – Roteador Cisco 1751V;

Versão do firmware: Version 12.2(1r)XE1, RELEASE SOFTWARE (fc1)

Figura 49 - Switch 3COM
Fonte: (ZDTRONIC, 2012)

1x – Switch 3Com 4500 SuperStack 3

Versão do firmware: 3Com OS V3.03.02s168p15

Foi montado o experimento com duas redes locais separadas por 2 roteadores conforme a Figura 50. A partir deste foi testado a transmissão de um stream de vídeo nos modos ponto-a-ponto e *multicast*, assim foi possível comparar a utilização de banda na interface do servidor do vídeo. Foi testada a transmissão do vídeo fazendo uma limitação de banda entre os roteadores para avaliar se o *multicast* poderia realmente ser uma solução para alta ocupação do link. A tabela 4 mostra os IPs usados.

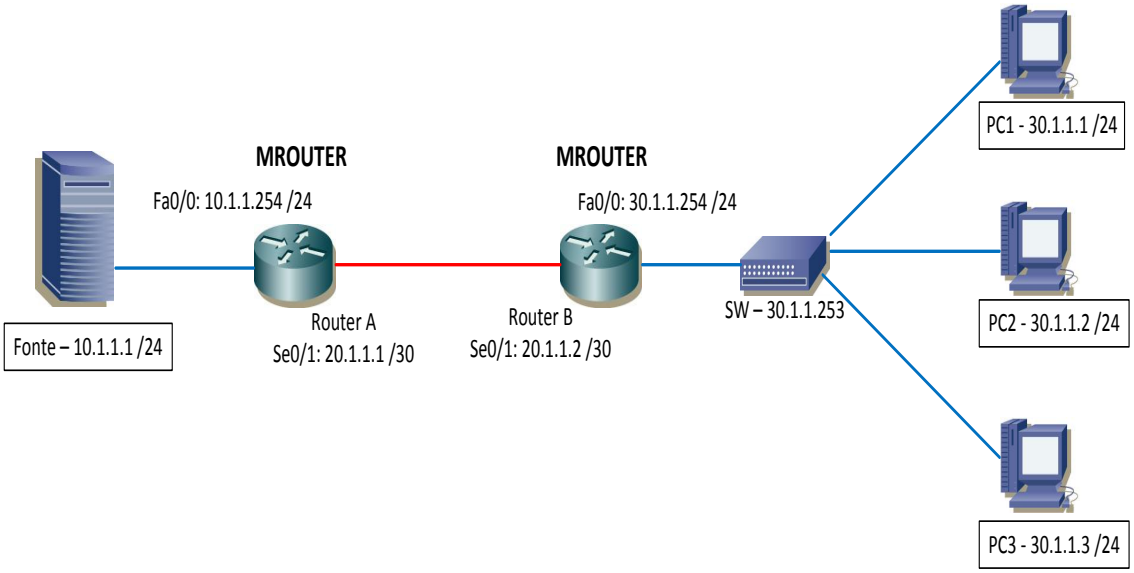

Figura 50 - Experimento montado
Fonte: Autoria Própria

Equipamento	IP	Mascara	GW	Interface
Router A	20.1.1.1	255.255.255.252	N/A	Serial
	10.1.1.254	255.255.255.0	N/A	FastEthernet0/0
Servidor Stream	10.1.1.1	255.255.255.0	10.1.1.254	ligado do *router*
Equipamento	**IP**	**Mascara**	**GW**	**Interface**
Router B	20.1.1.2	255.255.255.252	N/A	Serial
	30.1.1.254	255.255.255.0	N/A	FastEthernet0/0
Switch	30.1.1.253	255.255.255.0	30.1.1.254	Ethernet1/0/24
PC 1	30.1.1.1	255.255.255.0	30.1.1.254	Ethernet1/0/1
PC 2	30.1.1.2	255.255.255.0	30.1.1.254	Ethernet1/0/2
PC 3	30.1.1.3	255.255.255.0	30.1.1.254	Ethernet1/0/3

Quadro 3 - IPs Utilizados - Experimento 2
Fonte: Autoria Própria

9.3.2 Configuração Switch.

#vlan 90

interface Vlan-interface90

```
 ip Address 90.90.90.253 255.255.255.0
#
interface Ethernet1/0/1
 port access vlan 90
#
interface Ethernet1/0/2
 port access vlan 90
#
interface Ethernet1/0/3
 port access vlan 90
#
interface Ethernet1/0/4
 port access vlan 90
#
interface Ethernet1/0/24
 port link-type trunk
 port trunk permit vlan all
#
 ip route-static 0.0.0.0 0.0.0.0 90.90.90.254 preference 60
#
```

9.3.3 Configuração Roteador A.

```
hostname Router_A
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$kicc$VHruPIgAcwi02zEH6aBvk/
!
no aaa new-model
ip cef
```

```
!
ip multicast-routin
!
interface Ethernet0
 no ip Address
 shutdown
 half-duplex
!
interface FastEthernet0
 ip Address 10.1.1.254 255.255.255.0
 ip pim dr-priority 42949672
 ip pim sparse-mode
 speed auto
!
interface Serial0
 ip Address 20.1.1.1 255.255.255.252
 ip pim dr-priority 4294967
 ip pim sparse-mode
 encapsulation ppp
 ip ospf network broadcast
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.0 0.0.0.255 area 0
 network 20.1.1.0 0.0.0.3 area 0
!
ip forward-protocol nd
!
no ip <http server
no ip <http secure-server
ip pim rp-Address 10.1.1.254
ip pim autorp listener
ip pim send-rp-announce Serial0 scope 32
ip pim send-rp-announce FastEthernet0 scope 32
```

```
ip pim register-source Serial0
!
snmp-server community public RO
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps ds1
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps flash insertion removal
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps aaa_server
snmp-server enable traps atm subif
snmp-server enable traps bstun
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps dlsw
snmp-server enable traps entity
snmp-server enable traps event-manager
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps ipmobile
snmp-server enable traps ipmulticast
snmp-server enable traps msdp
snmp-server enable traps mvpn
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
```

```
snmp-server enable traps ospf cisco-specific state-change nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change shamlink interface-old
snmp-server enable traps ospf cisco-specific state-change shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps pim neighbor-change rp-mapping-change invalid-pim-messa
ge
snmp-server enable traps pppoe
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps rtr
snmp-server enable traps stun
snmp-server enable traps syslog
snmp-server enable traps l2tun session
snmp-server host 10.1.1.2 public
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 no login
!
end!
```

9.3.4 Configuração Roteador B

```
hostname Router_B
!
boot-start-marker
```

```
boot-end-marker
!
enable secret 5 $1$f1HV$9pDS/z4NJScY6gfunfVEr1
!
no aaa new-model
!
resource policy
!
memory-size iomem 15
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
!
ip cef
ip multicast-routing
!
interface FastEthernet0/0
 description ***INTERFACE LOCAL B***
 ip Address 30.1.1.254 255.255.255.0
 ip pim sparse-mode
 speed auto
!
interface Serial0/0
 ip Address 20.1.1.2 255.255.255.252
 ip pim sparse-mode
 encapsulation ppp
 ip ospf network broadcast
 clock rate 2000000
!
router ospf 1
 log-adjacency-changes
 redistribute connected
 network 20.1.1.0 0.0.0.3 area 0
```

```
 network 30.1.1.0 0.0.0.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 no auto-summary
!
no ip <http server
!
ip pim rp-Address 10.1.1.254
ip pim autorp listener
ip pim send-rp-announce Serial0/0 scope 32
ip pim send-rp-announce FastEthernet0/0 scope 32
!
!control-plane
!
line con 0
line aux 0
line vty 0 4
 no login
!
end
```

Novamente foi necessário habilitar o servidor SNMP no roteador A para conseguirmos capturar os pacotes da sua interface e avaliar a banda utilizada entre os links.

9.3.5 Transmissão Ponto-a-Ponto – Banda 2Mbps

Para esse experimento assim como no experimento1 (LAN), foram criados 3 streams de vídeo a partir da máquina servidora(10.1.1.1). Cada stream teve como endereço de destino o IP de um *host* da rede 30.1.1.0/24, forçando assim o roteamento *multicast* entre o RouteA e *Router* B.

Neste experimento a banda foi limitada entre os roteadores (interfaces Seriais) em 2 Mbps. Monitorando a interface Serial do *Router*A por cerca de 20 minutos, foi obtido o pico de 2 Mbps durante todo o intervalo do experimento, ou seja, foi utilizado todo o recurso do link. Foi observado que a qualidade do vídeo nos receptores foi muito ruim. Apresentou uma quantidade muito alta de macroblocos na imagem e muito atraso entre som e imagem.

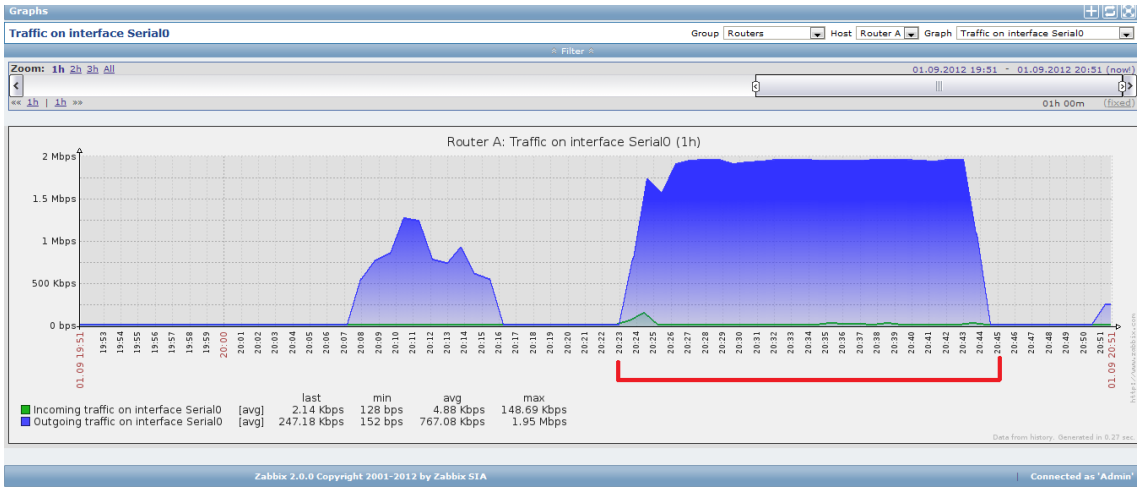

Figura 51 - Ocupação da banda em Unicast banda de 2Mbps – Pico 2Mbps
Fonte: Autoria Própria

Destacado em vermelho na figura 51 o tempo do teste. Foi observado que toda a banda do link estava sendo usada.

9.3.6 Transmissão Multicast – Banda 2 Mbps

Para esse experimento foi criado um único *stream* de vídeo a partir da máquina servidora(10.10.10.1), com o IP de destino 239.100.100.100(endereço do intervalo de IPs *Multicast*).

Monitorando a interface Serial do *Router*A por cerca de 20 minutos, tivemos o pico de aproximadamente 1.5 Mbps durante todo o intervalo de teste, ou seja, para este caso o *Multicast* otimizou a utilização do link que ainda tinha cerca de 500 kbps livres.

Além da melhora na utilização da banda foi percebida uma ótima qualidade de som, imagem e sincronismo nos *streams* em cada *host* da rede que estava recebendo o tráfego.

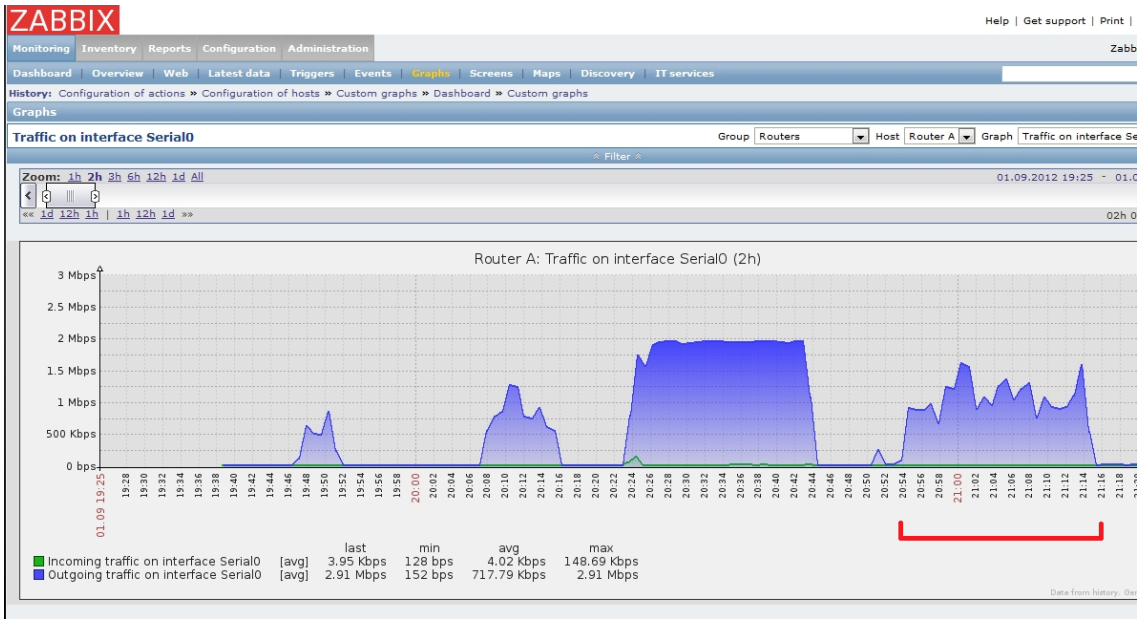

Figura 52 - Ocupação da banda em Multicast banda 2 Mbps – Pico 1,5Mbps
Fonte: Autoria Própria

Destacado em vermelho na figura 52 o tempo do teste. Foi observado melhora na utilização da banda.

9.3.7 Transmissão Ponto-a-Ponto – Banda 8 Mbps

Para esse experimento aumentamos a banda do link entre os roteadores para 8 Mbps com o comando *clock rate 8000000*, na interface Serial do *Router*A.

Foram criados novamente 3 *streams* de vídeo a partir da máquina servidora(10.1.1.1). Cada *stream* teve como endereço de destino o IP de um *host* da rede 30.1.1.0/24, forçando assim o roteamento *multicast* entre o RouteA e *Router* B.

Monitorando a interface Serial do *Router* A por cerca de 20 minutos, tivemos o pico de aproximadamente 4,5 Mbps durante todo o intervalo de teste, ou seja, no primeiro teste tivemos aproximadamente 2,5 Mbps de banda reprimida(4,5 – 2), devido a limitação do link.

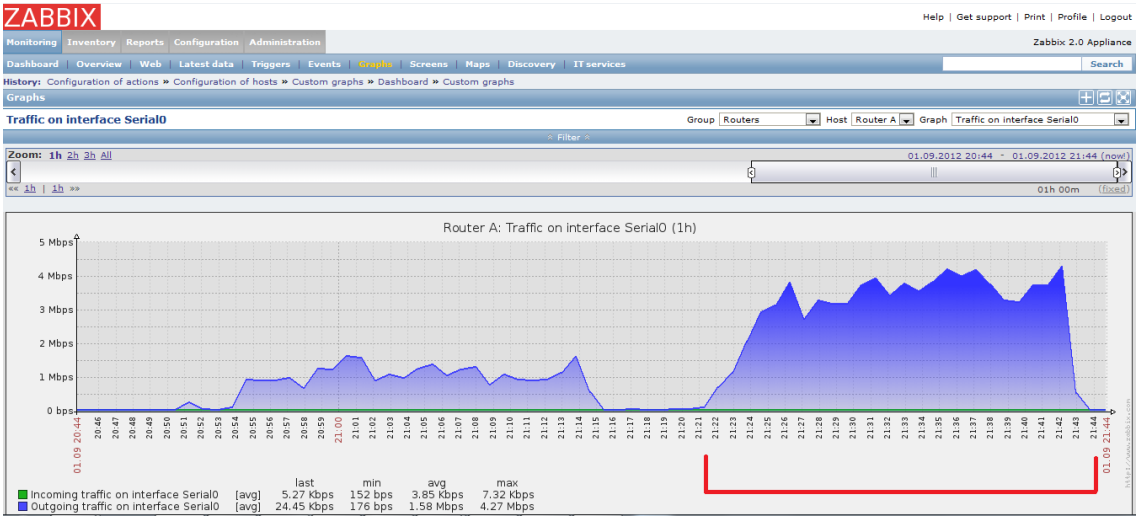

Figura 53 - Ocupação da banda em Unicast banda de 8Mbps – Pico 4,5Mbps
Fonte:Autoria Própria

Destacado em vermelho na figura 53 o tempo do teste.

9.3.8 Comparação Final dos 3 tipos de Transmissão.

Ao final deste experimento ficou muito claro a eficiência que o *Multicast* pode trazer para uma rede, melhorando significativamente a utilização de banda entre 2 links.

Pode-se perceber que a soma do tráfego dos 3 streams de vídeos era de aproximadamente 4,5 Mbps, uma banda muito maior do que a estabelecia inicialmente entre os 2 roteadores (2 Mbps). Neste caso a transmissão via *Multicast* seria uma ótima alternativa para realizar a entrega, principalmente quando diversos usuários tem um link compartilhado com baixa capacidade.

Novamente se levarmos em conta que tinhamos apenas três *host*s requisitando o video em um link de 2 Mbps, a banda não foi suficiente para a demanda, se tivessemos centenas de *host*s requisitando o mesmo stream, a qualidade seria infinitamente pior do que a já apresentada.

Podemos concluir que a transmissão *multicast* seria uma solução bastante apropriada, visto que indenpendente do numero de usuários requisitando o video, a banda utilizada seria de apenas 1,5 Mbps. Valores são comparados na tabela 5.

Tabela 2 - Tabela comparativa da ocupação dos links durante os experimentos

Tipo de Transmissão	Banda do Link (Mbps)	Pico (Mbps)	Ocupação do Link(%)
Ponto-a-Ponto	2	2	100
Multicast	2	1,5	75
Ponto-a-Ponto	8	4,5	56,25

Fonte: Autoria Própria

10. CONCLUSÃO

Através deste estudo foi possível familiarizar-se com uma nova tecnologia de transmissão de dados que não constava na ementa do curso e vem sendo cada vez mais utilizada no mercado de telecomunicações. Hoje vemos nas empresas o crescimento da utilização do *multicast* principalmente quando falamos sobre serviço de IPTV, que é um assunto cada vez mais comum quando se trata de novos serviços nas operadoras (TV ao vivo, Radio no Desktop, ensino a distancia, treinamentos, videoconferência, entrega de dados em tempo real, transferência de arquivos, transmissões corporativas e vídeo sobre demanda).

A partir dos experimentos propostos, foi possível ter um contato direto com a implantação, operação e gerenciamento de uma rede trabalhando com os serviços utilizando-se do *multicast*. Com estes foi possível avaliar na prática toda a eficiência apresentada na teoria que este modo de transmissão pode proporcionar.

Com a conclusão dos experimentos ficaram muito claras as melhorias dos serviços rodando a partir do *multicast*, tanto avaliando a rede que teve seus recursos melhor utilizados quanto para o usuário final, que teve uma melhor qualidade de áudio, vídeo e sincronismo recebido.

Mesmo com todas as vantagens, é importante ressaltar que algumas desvantagens podem ser significantes quando se está planejando uma readequação em uma rede para suportar o *Multicast*. A primeira delas é que todos os equipamentos devem ter firmwares com versões que suportem o *multicast*, caso contrário será necessário um melhor estudo sobre métodos de tunelamento para possibilitar o envio de forma adequada. Outra desvantagem é que a tecnologia trabalha com pacotes UDP, ou seja, a entrega é baseada em melhor esforço, sendo assim a confiabilidade na entrega pode deixar a desejar assim como a questão da segurança.

Todas as desvantagens citadas acima abrem um leque de possibilidades para estudos e desenvolvimentos de melhorias futuras nesse ramo.

Ao fim deste trabalho foram adquiridos os conhecimentos necessários para o planejamento de uma solução que atenda a necessidade de alto tráfego em uma rede com certa limitação de banda, bem como a adequação de uma rede existente para suportar a esta nova tecnologia.

11. REFERÊNCIAS

ADAMS, A. **Request for Comments: 3973.** Disponível em: <http://tools.ietf.org/rfc/rfc3973.txt>. Acesso em: 28 jul. 2012.

An Introduction to IP Multicast. Disponível em: <http://ganges.cs.tcd.ie/undergrad/4ba2/*multicast*/>. Acesso em: 9 mar. 2012.

ARMSTRONG, FREIER. **Multicast Transport Protocol.** Disponível em: <http://datatracker.ietf.org/doc/rfc1301/>. Acesso em: 30 jul. 2012.

CERTIFICATIONS KITS. Disponível em: <http://www.certificationkits.com/images/17212.jpg>. Acesso em: 24 ago. 2012.

Cisco Systems, ***Multicast VLAN Registration*** **(MVR) on a Catalyst 3750 Sample Configuration**. Disponível em: <http://www.cisco.com/en/US/products/hw/switches/ps5023/products_configuration_example09186a008076c6e0.shtml>. Acesso em: 25 jul. 2012.

DIG: Enterprise Campus Topology. **IP Multicast Technology Overview**. Disponível em: <http://www.cisco.com/en/US/docs/ios/solutions_docs/ip_*multicast*/White_papers/mcst_ovr.html. Acesso em: 9 mar. 2012.

DRIEMEYER, Marco. **Guia de Instalação do Zabbix 1.8.x no CentOS 6**. Disponível em: <https://docs.plenatech.com.br/linux/zabbix_centos>. Acesso em: 20 mar. 2012.

DIAS, Beethovem Zanella. **Projeto Multicast**. Disponível em: <http://mesonpi.cat.cbpf.br/mcast>. Acesso em: 13 jul. 2012.

FEANER, BILL. **Protocol Independent Multicast - Sparse Mode (PIM-SM).** Disponível em: <http://www.rfc-editor.org/rfc/rfc4601.txt>. Acesso em: 25 abr. 2012.

FILIPPETTI, Marco Aurélio. **CCNA 4.1 Guia Completo de Estudo.** 1.ed. Florianópolis: Visual Books, 2008.

FRANCISCO, Antonio João Ferreira. **MULTICAST**. Disponível em: <http://www.ime.usp.br/~ajoaoff/mac499/monografia/*multicast*.html>. Acesso em: 23 set. 2011.

GALIANO, Herbert Luna. **Soluções Multicasting na Internet.** Disponível em: <http://penta2.ufrgs.br/rc952/trab2/hl_intro.html>. Acesso em 05 set. 2011.

GASPARY, Luciano Paschoal . **Multicast Tutorial**. Disponível em: <http://penta.ufrgs.br/redes296/*multicast*/tutorial.html>. Acesso em: 03 set. 2011.

GROSSMANN, Jeremy; EROMENKO Alexey. **What is GNS3?**. Disponível em: <http://www.gns3.net/ >. Acesso em: 10 set. 2011.

HENRIQUE, Julio. **Monitoração de tráfego com MRTG**. Disponível em: <http://www.vivaolinux.com.br/artigo/Monitoracao-de-trafego-com-MRTG>. Acesso em: 17 set. 2011.

KIM, Hyung. **VLC media player**. 2010, Disponível em: <http://www.videolan.org/vlc>. Acesso em: 17 set. 2011.

LIN, David. **IP MULTICAST**. Disponível em: <http://www.cisco.com/en/US/products/ps6552/products_ios_technology_home.htm>. Acesso em: 03 set. 2011.

MARQUES, Vera; CARNEIRO, Jorge. **Multicast**. Disponível em: <http://www.ipg.pt/user/~sduarte/rc/trabalhos/*Multicast*/*Multicast*.htm#Autores%20%C2%A0 >. Acesso em: 10 set. 2011

MICROSOFT. **DESCRIÇÃO GERAL DO ENCAMINHAMENTO *UNICAST***. Disponível em <http://technet.microsoft.com/pt-pt/library/cc786079(v=ws.10).aspx>. Acesso em: 20 mar. 2012.

MOY, J. **Request for Comments: 1584. 1994.** Disponível em: <http://tools.ietf.org/rfc/rfc1584.txt>. Acesso em: 28 jul. 2012

MULTIREDE , **MCAST-BKIP:** Multicast em Backbones IP. São Paulo, 2011

ROESLER , Valter. **Transmissão Multicast versão resumida**. 2001. 15f. Tese - Universidade do vale do Rio do Sinos Unisinos, São Leopoldo, 2001.

STARDUST. **MCAST2000 -** A Survey of the History of Internet Multicast. California: Campbell, 1999.

TANENBAUM, Andrew S. **Redes de computadores**. 4. ed. Rio de Janeiro: Campus, 2003.

WAITZMAN, D; PARTRIDGE, C. **Request For Comments: 1075**. 1988
Disponível em: <http://www.ietf.org/rfc/rfc1075.txt>. Acesso em: 21 jul. 2012.

WILLIAMSON, Beau. **Developing IP Multicast Networks – Cisco Press Publications**. 2003

ZDTRONIK. Disponível em:
<http://www.zdtronic.com/images/3CR17561-91.jpg>. Acesso em: 24 ago. 2012