EDMO LOPES FILHO

ARQUITETURA DE ALTA DISPONIBILIDADE PARA FIREWALL E IPS BASEADA EM SCTP

Dissertação apresentada ao programa de Pós-graduação em Ciência da Computação da Universidade Federal de Uberlândia, para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Redes de Computadores.

Orientador: Prof. Dr. Pedro Frosi Rosa, da Universidade Federal de Uberlândia.

UBERLÂNDIA – MG

2008

Dados Internacionais de Catalogação na Publicação (CIP)

L864a Lopes Filho, Edmo, 1967-
Arquitetura de alta disponibilidade para Firewall e IPS baseada em SCTP / Edmo Lopes Filho. - 2008.
135 f. : il.

Orientador: Pedro Frosi Rosa.
Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Ciência da Computação.
Inclui bibliografia.

1. Redes de computação - Medidas de segurança - Teses. I. Rosa, Pedro Frosi. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU: 681.3-78

Elaborado pelo Sistema de Bibliotecas da UFU / Setor de Catalogação e Classificação

Edmo Lopes Filho

ARQUITETURA DE ALTA DISPONIBILIDADE PARA FIREWALL E IPS BASEADA EM SCTP

Dissertação apresentada ao programa de pós-graduação em ciência da computação da Universidade Federal de Uberlândia, para obtenção do grau de mestre em ciência da computação.
Área de concentração: redes de computadores.

Banca examinadora:

Uberlândia, 13 de fevereiro de 2008.

_____

Prof. Dr. Pedro Frosi Rosa – Orientador - UFU

_____

Prof. Dr. Paulo Roberto Guardieiro - UFU

_____

Prof. Dr. José Gonçalves Pereira Filho - UFES

AGRADECIMENTOS

À Universidade Federal de Uberlândia, em especial, à Faculdade de Computação, pela oportunidade de realizar este curso.

À CTBC, empresa na qual trabalho, pela permissão que me foi concedida para que eu pudesse me dedicar a todas as atividades do curso.

Ao meu orientador, Prof. Dr. Pedro Frosi Rosa, pela constante orientação, pelo incentivo durante a elaboração desta dissertação e pela confiança e apoio sempre prestados.

Aos demais professores do curso, que também contribuíram para minha formação.

À minha esposa Letícia e filhas: Vitória e Isadora, que sempre compreenderam a importância deste curso incentivando e apoiando-me nos momentos difíceis.

A todas as outras pessoas que, direta ou indiretamente, contribuíram para a realização desta meta.

"Que a inspiração chegue não depende de mim. A única coisa que posso fazer é garantir que ela me encontre trabalhando."
(Pablo Picasso - Artista)

SUMÁRIO

RESUMO i
ABSTRACT ii
LISTA DE FIGURAS iii
LISTA DE TABELAS v
LISTA DE ABREVIATURAS E SIGLAS vi
1 INTRODUÇÃO 16
2 MECANISMOS DE ALTA DISPONIBILIDADE 18
2.1 Introdução 18
2.2 Conceitos de Alta Disponibilidade 19
2.2.1 Tipos de mecanismos de alta disponibilidade 23
2.2.2 Implementando alta disponibilidade 27
2.3 IP Multicasting 30
2.3.1 Conceito de grupo *Multicast* 30
2.3.2 *Internet Group Management Protocol* (IGMP) 34
2.3.3 *Multicast* no ambiente de *switches* camada 2 36
2.4 Virtual Routing Redundacy Protocol (VRRP) 37
2.4.1 Formato do pacote VRRP 40
2.4.2 Requisições ARP 43
2.4.3 Considerações de segurança 43
2.5 *Keepalived* Daemon 45
2.5.1 *Linux Virtual Server* 45
2.5.2 Estrutura interna do *Keepalived* 47
3 *FIREWALL* E IPS 50
3.1 Introdução 50
3.2 Tipos de *Firewall* 52
3.2.1 Filtros de pacotes 52
3.2.2 *Application gateways* 52
3.2.3 *Circuit-Level gateways* 53
3.2.3 *Firewall* de inspeção de estado de sessão 53
3.3 Sistema de Prevenção de Intrusos 55
3.3.1 Tipos de IPS 57

3.3.2 Analisadores de protocolos 59
3.4 *Heartbeat* 60
3.5 Mecanismos de Alta Disponibilidade para *Firewall* e IPS 62
3.5.1 Tipos de alta disponibilidade 63
3.5.2 *Firewall*, IPS e VRRP 63
4 O PROTOCOLO SCTP 65
4.1 Introdução 65
4.2 Arquitetura do SCTP 66
4.2.1 Cabeçalho comum 67
4.2.2 Fatias (*Chunks*) 69
4.3 Descrição dos Tipos de Fatias 74
4.3.1 Fatia de dados (DATA) 74
4.3.2 Fatia de Reconhecimento Seletivo (SACK – *Selective Acknowledge*) 76
4.3.3 Fatia Requisição de *Heartbeat* (HEARTBEAT) 79
4.3.4 Fatia Reconhecimento de *Heartbeat* (HEARTBEAT ACK) 80
4.4 Principais Fases de Uma Associação SCTP 81
4.4.1 Inicio e concretização de uma associação 81
4.4.2 Transporte de dados em uma associação 83
4.4.3 Término de uma associação 84
4.5 Principais Vantagens do SCTP 86
4.5.1 *Head-of-line blocking* 86
4.5.2 *Multihoming* 87
5 ARQUITETURA DE ALTA DISPONIBILIDADE BASEADA EM SCTP 92
5.1 Introdução 92
5.2 Análise de Protocolos e Tráfego HA 93
5.2.1 *Firewall SunScreen* 93
5.2.2 *Firewall* ASA (*Adaptive Security Appliances*) 95
5.2.3 *Firewall* SSG (*Secure Services Gateway*) 97
5.3 Razões e Motivações para HA sobre LAN 98
5.3.1 Topologia com *clusters* de *firewalls* em HA e roteadores VRRP 100
5.3.2 Topologia com *clusters* de *firewalls* e *switches* camada 7 101
5.3.3 Topologia com *firewalls* em HA sobre LAN 102
5.4 Proposta de Alta Disponibilidade para *Firewall*/IPS Baseada em SCTP 105
5.4.1 Metalinguagem referente ao esquema 106
5.4.2 Fluxo de mensagens *heartbeat* da arquitetura 109

5.4.3 Especificação da arquitetura de HA 114
5.5 Avaliação da Proposta 120
5.6 Incrementando a Disponibilidade da LAN 125
6 CONCLUSÃO 127
REFERÊNCIAS BIBLIOGRÁFICAS 129

RESUMO

O crescimento das redes de computadores permite a comunicação eficiente e flexível entre todas as entidades que constituem um ambiente de negócio, entre as quais: consumidores, parceiros e fornecedores. Porém esta flexibilidade vem acompanhada de uma série de riscos para o negócio, incluindo a possibilidade de sua interrupção. Para tratar os problemas de segurança, administradores de rede e segurança utilizam-se, entre outras opções, de gerenciadores de políticas e regras de acesso, tais como firewall e IPS (Intrusion Prevention Systems). A dissertação aborda um dos três requisitos básicos do processo de Gestão de Segurança: a disponibilidade; através da proposta de construção de uma nova arquitetura de alta disponibilidade para firewall e IPS utilizando o protocolo SCTP.

ABSTRACT

The growth and proliferation of computer networks allow businesses efficiently to communicate with their own components as well as with their business partners, customers, and suppliers. However, the flexibility and efficiency provided by such systems come with increasing risks, including disruption of services. To address the security issues, network and security managers, among other options, often turn on network policy and access control management such as firewall and IPS (Intrusion Prevention Systems) protection. The dissertation addresses one of the three basic requirements of security management programs: "Availability" and proposes a new architecture for firewall and IPS high availability based on the SCTP protocol.

LISTA DE FIGURAS

Figura 1 – Relações entre as entidades do processo de Gestão de Risco 21
Figura 2 – Arquitetura clássica de um sistema de alta disponibilidade *dual-node* 22
Figura 3 – Diagrama de tempo *downtime* planejado e eliminado 25
Figura 4 – Transmissão *multicast*: emissor envia um único fluxo para o Grupo *Multicast* 31
Figura 5 – Formato do endereço MAC IEEE 802.3 33
Figura 6 – Mapeamento endereço IP *Multicast* para endereço MAC 34
Figura 7 – Formato das mensagens IGMP 35
Figura 8 – *Default Gateway* da rede 37
Figura 9 – Diagrama de estados do protocolo VRRP 39
Figura 10 – Formato do pacote VRRP 41
Figura 11 – Topologia VRRP com vários roteadores virtuais 44
Figura 12 – Visão global do LVS 46
Figura 13 – Estrutura interna do *Keepalived* 47
Figura 14 – *Firewall* topologia em camada simples 51
Figura 15 – Tipos de *firewall* e o modelo OSI 54
Figura 16 – NIPS em linha, técnica *Packet Scrubbing* 59
Figura 17 – Topologia HA entre *firewall*, VRRP para controle dos endereços virtuais 64
Figura 18 – Associação SCTP com múltiplos fluxos (*streams*) 66
Figura 19 – Inserção do SCTP no modelo OSI e arquitetura Internet 67
Figura 20 – Pacote SCTP (Cabeçalho e fatias) 68
Figura 21 – Parâmetros variáveis da fatia 72
Figura 22 – Fatia de Dados 74
Figura 23 – Fatia de Reconhecimento Seletivo (SACK) 77
Figura 24 – Fatia *Heartbeat Request* 79
Figura 25 – Parâmetro *Heartbeat* 80
Figura 26 – Fatia *Heartbeat ACK* 80
Figura 27 – Máquina de estado do início e concretização de uma Associação 82
Figura 28 – Diagrama de envio de Dados e *Heartbeat* em uma associação SCTP 83
Figura 29 – Máquina de estado do término de uma associação SCTP 84
Figura 30 – Problema *Head-of-line blocking* 87
Figura 31 – Conexão TCP versus associação SCTP 88
Figura 32 – Topologia *Firewall SunScreen* 94

Figura 33 – Coleta de tráfego HA entre *firewalls* ASA 5520 96
Figura 34 – Coleta de tráfego HA entre *firewalls* SSG 520 97
Figura 35 – Topologia com *cluster* de *firewalls* em HA e roteadores VRRP 101
Figura 36 – Topologia com *firewalls* em HA sobre LAN 103
Figura 37 – Esquema HA 105
Figura 38 – Topologia associada à proposta, *firewalls*/IPS com duas interfaces HA 106
Figura 39 – Fluxo de fatias associado à proposição 110
Figura 40 – Estados de transição, Mestre operando na rede e ausente 112
Figura 41 – Estados de transição, Mestre em estado de falha e retorno a operação 113
Figura 42 – Camadas da arquitetura HA proposta 115
Figura 43 – Detalhamento da camada HSOL 118
Figura 44 – Cenários *dual-homed* e duas associações 122

LISTA DE TABELAS

Tabela 1 – Níveis de alta disponibilidade 24
Tabela 2 – Exemplos de endereços *Multicast* reservados 32
Tabela 3 – Definições associadas ao VRRP 38
Tabela 4 – Tipos de autenticação do protocolo VRRP 42
Tabela 5 – *Bits* mais significativos (Tipo da Fatia) 70
Tabela 6 – Tipos de fatias (*chunks*) 71
Tabela 7 – *Bits* mais significativos (Tipo do Parâmetro) 73
Tabela 8 – *Bits* para controle de mensagens fragmentadas 75
Tabela 9 – HA sobre LAN versus HA convencional para *sites* distantes 104
Tabela 10 – Comparativo HA sobre SCTP com NSRP, SCPS e *SunScreen* 124

LISTA DE ABREVIATURAS E SIGLAS

AH – *Authentication Header*

AIMD – *Additive Increase Multiplicative Decrease*

ARP – *Address Resolution Protocol*

AS – *Autonomous System*

ASA – *Adaptive Security Appliances*

ASIC – *Application Specific Integrated Circuit*

BCP – *Business Continuity Plan*

BGP – *Border Gateway Protocol*

CBIPS – *Content Based Intrusion Prevention Systems*

CIA – *Confidentiality, Integrity and Availability*

COBIT – *Control Objectives for Information and related Technology*

DDoS – *Distributed Denial of Service*

DHCP – *Dynamic Host Configuration Protocol*

DiffServ – *Differentiated Services*

DNS – *Domain Name System*

DPI – *Deep Packet Inspection*

FTP – *File Transfer Protocol*

FWLB – *Firewall Load-Balancing*

HA – *High Availability*

HIPS – *Host Based Intrusion Prevention Systems*

HMAC – *Keyed-Hashing for Message Authentication Code*

HTTP – *HyperText Transfer Protocol*

HSRP – *Hot Standby Router Protocol*

IANA – *Internet Assigned Numbers Authority*

ICMP – *Internet Control Message Protocol*

ICNS – *Fourth International Conference on Networking and Services*

ICV – *Integrity Check Value*

IDS – *Intrusion Detection Systems*

IETF – *Internet Engineering Task Force*

IGMP – *Internet Group Management Protocol*

IMS – *IP Multimedia Subsystem*

IntServ – *Integrated Services*

IOS – *Internetwork Operating System*

IP – *Internet Protocol*

IPS – *Intrusion Prevention Systems*

IPSec – *Internet Protocol Security*

IPSTB – *IP Standby Protocol*

iSCSI – *Internet Small Computer Systems Interface*

IT – *Information Technology*

ITIL – *IT Information Library*

LAN – *Local Area Network*

LVS – *Linux Virtual Server*

MAC – *Media Access Control*

MPLS – *Multiprotocol Label Switching*

MTBF – *Mean Time Between Failures*

MTTR – *Mean Time To Repair*

MTU – *Maximum Transmission Unit*

NAT – *Network Address Translation*

NSRP – *NetScreen Redundancy Protocol*

OSI – *Open Systems Interconnection*

OSPF – *Open Shortest Path First*

OUI – *Organizationally Unique Identifier*

PDA – *Personal Digital Assistants*

PDU – *Packet Data Unit*

PIM – *Protocol Independent Multicast*

PR-SCTP – *Partial Reliability Stream Control Transmission Protocol*

QoS – *Quality of Service*

RAID – *Redundant Array of Independent Disks*

RBIPS – *Rate Based Intrusion Prevention System*

RDMA – *Direct Memory Access Protocol*

RFC – *Request for Comments*

RSVP – *Resource Reservation Protocol*

RTO – *Retransmission Time Out*

RTT – *Round Trip Time*

SCPS – *Space Communications Protocol Suite*

SCTP – *Stream Control Transmission Protocol*

SHA – *Secure Hash Algorithm*

SIMCO – *Simple Middlebox Configuration Protocol*

SIP – *Session Initiation Protocol*

SLA – *Service Level Agreement*

SNMP – *Simple Network Management Protocol*

SNTP – *Simple Network Time Protocol*

SMTP – *Simple Mail Transfer Protocol*

SPAN – *Switch Port Analyzer*

SPOF – *Single Point of Failure*

SSG – *Secure Services Gateway*

SSL – *Secure Sockets Layer*

TCB – *Transmission Control Block*

TCP – *Transmission Control Protocol*

TLV – *Tag-Length-Value*

TTL – *Time-to-Live*

UDP – *User Datagram Protocol*

URL – *Uniform Resource Locator*

UTM – *Unified Threat Management*

VLAN – *Virtual Local Area Network*

VPN – *Virtual Private Network*

VRID – *Virtual Router Identifier*

VRRP – *Virtual Routing Redundancy Protocol*

VSD – *Virtual Security Device*

VSI – *Virtual Security Interface*

1 INTRODUÇÃO

Integridade, confidencialidade e disponibilidade constituem os requisitos básicos de um programa de gestão de segurança, onde: disponibilidade é a garantia de que um sistema computacional está acessível pelos usuários autorizados quando estes requerem o acesso; integridade visa prevenir as informações e processos contra modificação intencional ou acidental não autorizadas; e confidencialidade visa prevenir a divulgação de informações para alguém que não está autorizado a acessá-las.

Entender o significado dos requisitos básicos de segurança, a denominada "Tríade CIA (*Confidentiality, Integrity e Availability*)", isto é, entender como é provida por diferentes mecanismos e como sua falta pode afetar negativamente a segurança, auxilia a identificar os problemas e gerar soluções de segurança adequadas aos ambientes de redes e sistemas.

Em um mundo de negócios cada vez mais ávido por informações e tempos de resposta menores, os requisitos de disponibilidade dos sistemas computacionais têm sido cada vez mais rigorosos. Assim, são necessários mecanismos que possibilitem que falhas de hardware e/ou software em um determinado sistema afetem o mínimo possível a capacidade de processamento do sistema ou a disponibilidade do serviço.

Os mecanismos de disponibilidade, em sua grande maioria, são calcados na redundância de *hardware*, inteligência de *software* e protocolos para identificar quando existe uma falha do sistema principal, iniciar e concluir o processo de transferência dos serviços para um sistema alternativo. Além disso, requisitos de disponibilidade geográfica, ou seja, sistemas instalados em locais distantes levaram a utilização de interfaces de monitoramento e verificação da alta disponibilidade, diretamente conectadas a LAN (*Local Area Networks*), implicando na necessidade de protocolos confiáveis para a realização do transporte destas informações.

O objetivo deste trabalho é propor e avaliar uma nova arquitetura de alta disponibilidade baseada no protocolo SCTP (*Stream Control Transmission Protocol*) [Stewart 2000] para *firewall* e IPS (*Intrusion Prevention Systems*). Portanto, para fundamentar a proposição são apresentados os mecanismos disponíveis para atender aos requisitos atuais de disponibilidade para estes equipamentos.

A dissertação está dividida em seis capítulos, da seguinte forma: capítulo 1 – introdução, motivação e objetivo do trabalho; capítulo 2 – uma breve revisão do estado da arte em conceitos e mecanismos de disponibilidade, incluindo protocolos de *multicasting* [Deering 1989], amplamente utilizados nos mecanismos atuais de disponibilidade, VRRP (*Virtual Routing Redundancy Protocol*) [Hinden 2004] e *Keepalived daemon* [Keepalived 2003], respectivamente utilizados para prover disponibilidade em ambientes de rede e servidores Linux; capítulo 3 – apresenta conceitos básicos sobre *firewalls*, IPS e os mecanismos de *heartbeat* utilizados para prover alta disponibilidade; capítulo 4 – apresenta a arquitetura do protocolo SCTP, focando especialmente em suas características de estabelecimento e término de uma associação, bem como as de *multihoming* e *multistreaming* fundamentais à nova arquitetura proposta na dissertação; capítulo 5 – propõe uma nova arquitetura de alta disponibilidade e apresenta um estudo comparativo desta arquitetura proposta em relação aos atuais mecanismos de disponibilidade; e o capítulo 6 apresenta a conclusão e indicações de trabalhos futuros.

Como fruto de pesquisas realizadas durante o curso e como base de estudos para esta dissertação, o artigo "*An IMS Control Layer PR-SCTP Based Network*" foi aceito para publicação e apresentação no ICNS 2008 (*Fourth International Conference on Networking and Services*) pp 61-66, realizado em Gosier, Guadeloupe no período de 16 a 21 de Março de 2008.

2 MECANISMOS DE ALTA DISPONIBILIDADE

2.1 Introdução

À medida que aplicações comerciais críticas migram para a Internet, o requisito de alta disponibilidade (HA - *High Availability*) torna-se ainda mais crítico para o negócio, devido à ampliação do nível de exposição destes sistemas às vulnerabilidades. Uma das maiores vantagens de um sistema em HA é que este possui redundância de *hardware*, *software* e mecanismo de HA suportado através da detecção proativa de falhas e configuração automática do conjunto de recursos, de modo que as tarefas possam ser suportadas pelos componentes restantes do sistema, em caso de falha de algum dos componentes do mesmo.

A aplicação de conceitos de alta disponibilidade amplia as capacidades de controle, produtividade e conseqüentemente de manutenção das receitas financeiras das empresas. Além disso, economiza tempo de reação através do monitoramento e diagnóstico proativo de possíveis falhas.

Porém, a existência de mecanismos de alta disponibilidade como requisitos do negócio não exime as empresas da responsabilidade de elaboração e manutenção de um Plano de Continuidade de Negócio BCP (*Business Continuity Planning*) [BSI 2006]. Ou seja, a existência destes mecanismos deve ser endereçada como a existência de meios para complementar e ampliar as possibilidades para a elaboração do plano.

A seção 2.2 descreve os conceitos de alta disponibilidade, principais mecanismos e suas características. A seção 2.3 descreve resumidamente os protocolos de IP (*Internet Protocol*) *Multicast*. A seção 2.4 descreve o protocolo VRRP (*Virtual Routing Redundancy Protocol*). Finalmente, na seção 2.5 a arquitetura do *Keepalived daemon* é apresentada.

2.2 Conceitos de Alta Disponibilidade

O termo disponibilidade descreve um sistema que provê um nível de serviço específico de acordo com necessidades estabelecidas. Em computação este termo é entendido como o período de tempo em que os serviços estão disponíveis, por exemplo: 16 horas por dia, 6 dias por semana. O nível de serviço de um sistema é o grau, a extensão de serviço que o sistema provê a seus usuários. Freqüentemente, o nível de serviço é descrito no documento Acordo de Nível de Serviço SLA (*Service Level Agreement*), seja para atender requisitos internos do negócio ou entre empresas distintas. Como exemplos de requisitos internos e entre empresas distintas, podem ser citados o SLA da rede corporativa e disponibilidade de um serviço de *hosting* em *data center*, respectivamente.

Qualquer interrupção de um serviço, planejada ou não, é reconhecida como falha (*outage*) e o tempo de duração da falha (*downtime*) pode ser medido em minutos, horas ou dias. Se o negócio requer um tempo de duração da falha igual à zero, então o requisito deste negócio é um sistema tolerante a falhas (*fault tolerance*). Sistemas de alta disponibilidade são projetados para suportar um tempo reduzido de duração da falha.

Contrastando com as soluções tolerantes a falhas, soluções de alta disponibilidade combinam investimentos e custos de manutenção menores. Além disso, por virtualmente eliminarem a possibilidade de perda de informações e conseqüente perda de receitas associadas à interrupção do negócio, estão ao alcance de qualquer empresa, que necessite implantar redes, sistemas e segurança. Como resultado direto destas características, muitas empresas estão implantando sistemas em HA para proteger o negócio ou minimizar possíveis impactos da interrupção de serviços.

Cada vez mais é necessário garantir a disponibilidade de um serviço, porém, a existência de partes mecânicas nos sistemas de informação implica na diminuição dos níveis de confiabilidade dos serviços, ampliando assim, o risco de interrupção dos mesmos. Para garantir a ausência ou minimizar a quantidade de interrupções é necessário, muitas vezes, dispor de *hardware* redundante e inteligência de *software* que entre em funcionamento automaticamente quando ocorrer falha de um dos componentes em utilização.

Um sistema de alta disponibilidade é um sistema computacional resistente à falhas de *software*, *hardware* e energia; objetivando manter os serviços disponíveis o máximo de tempo possível. A Disponibilidade de um sistema computacional, indicada por $A(t)$, é a probabilidade de que este sistema esteja funcionando e pronto para uso em um dado instante de tempo t.

Para entendermos a alta disponibilidade faz-se necessário, antes de mais nada, perceber que a alta disponibilidade não é apenas um produto ou uma aplicação que se instala, e sim um conjunto de requisitos de um sistema de informações diretamente relacionado ao risco de interrupção e às necessidades específicas de cada negócio. Portanto seu conceito não é absoluto. Por exemplo, grandes empresas com *Internet sites* espalhados pelo mundo, podem requerer acessos a seus bancos de dados em segundos. Instituições financeiras têm de estar aptas a realizar a transferência de fundos a qualquer hora do dia ou da noite, durante os sete dias da semana. Por outro lado, empresas varejistas podem necessitar de seu negócio disponível somente 18 (dezoito) horas por dia, mas durante estas dezoito horas os tempos de resposta têm que estar na escala de poucos segundos.

Muitas vezes as palavras: ameaça; vulnerabilidade; exposição; e risco são utilizadas para representar a mesma coisa, embora possuam diferentes significados e relacionamentos entre si. É muito importante entender a associação entre elas, assim, os conceitos delineados a

seguir serão utilizados nesta dissertação, para identificarmos as inter-relações aos mecanismos de disponibilidade. A figura 1 ilustra as relações entre estas entidades.

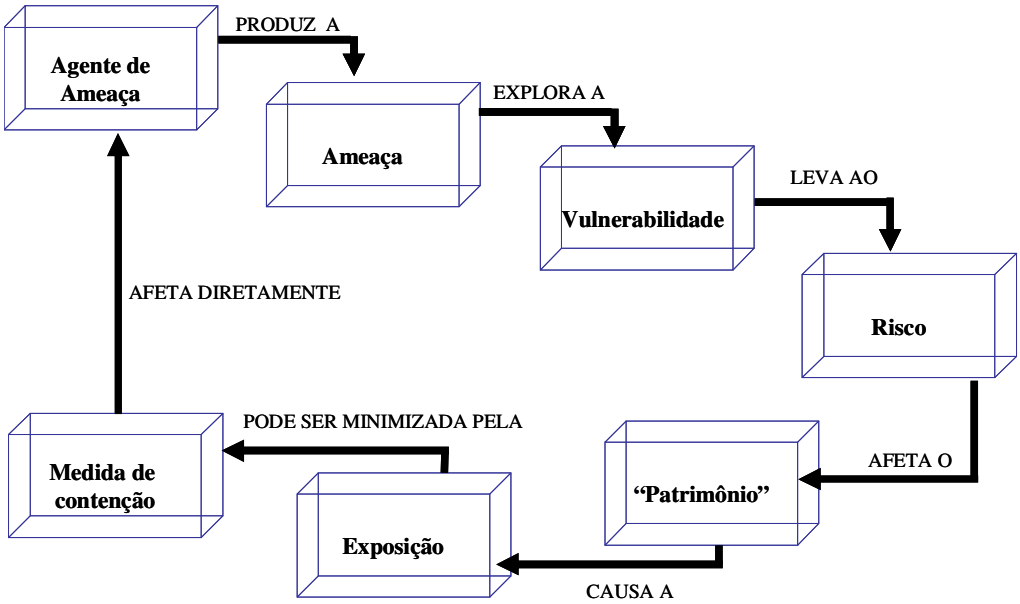

Figura 1 – Relações entre as entidades do processo de Gestão de Risco

Vulnerabilidade: é uma fraqueza do *software*, *hardware* ou processo, que provê ao agressor uma porta de entrada no computador, rede e/ou sistema. Uma vulnerabilidade pode ser uma porta ou gaveta destrancada, um serviço executando sem necessidade no servidor ou um *modem* conectado a um computador e que permite "*dial-in*".

Ameaça: é qualquer perigo potencial ao computador, rede e ou informação. Um intruso acessando a rede através de *backdoors* [Hansche 2004] caracteriza um agente de ameaça.

Risco: é a probabilidade de um agente de ameaça explorar a vulnerabilidade. É o potencial de perda gerado por esta ação. Se um *firewall* possui muitas portas abertas, existe um risco potencial de algum invasor explorar alguma destas portas, usando-a de forma indevida para penetrar na rede.

Exposição: é a ocorrência de perdas causadas pelo agente de ameaça.

Medida de contenção ou salvaguarda: é uma medida ou ação que minimiza o potencial de risco. Por exemplo: configurações de *software* adequadas, utilização de senhas mais robustas e aplicação de correções (*patches)* críticas.

Existem mecanismos e técnicas que podem ser utilizados para aumentar a disponibilidade de um sistema. Entretanto, a simples utilização destes mecanismos, não garante este aumento se não for acompanhado de uma completa análise de riscos associados ao negócio e a conseqüente proposição de topologia de disponibilidade adequada aos requisitos do negócio.

Quanto maior o nível de redundância dos componentes de uma topologia, menor será a quantidade de "pontos de falha únicos" SPOFs (*Single Point of Failure*) e menor será a probabilidade de interrupção no serviço. A evolução tecnológica e crescente concorrência de mercado têm propiciado a redução de custos para aquisição de sistemas redundantes. Surgem então os sistemas construídos com *hardware* acessível (*clusters*), modulares, com grande capacidade de desempenho e de custo mínimo. A figura 2 ilustra a configuração típica de um sistema de HA.

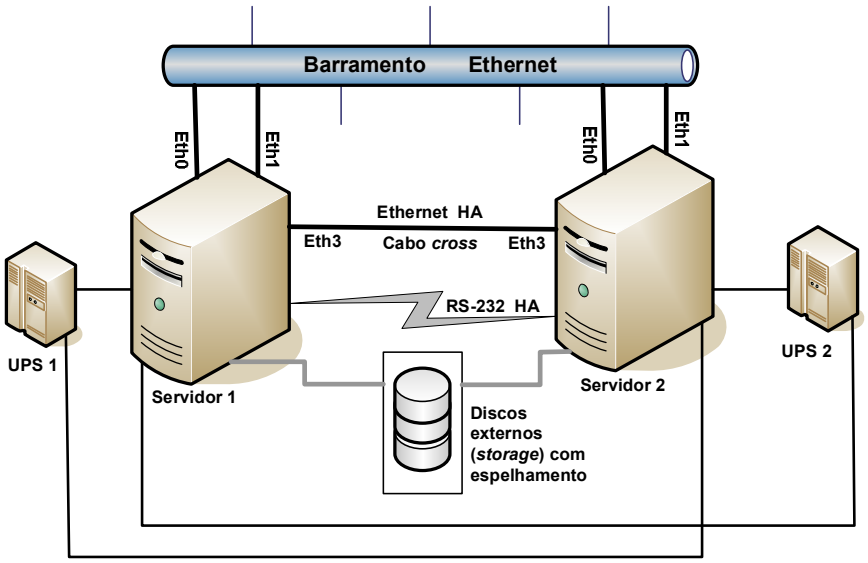

Figura 2 – Arquitetura clássica de um sistema de alta disponibilidade *dual-node*

Podemos observar que não existe um único ponto nesta arquitetura que ao falhar, implique em indisponibilidade de outro ponto qualquer (SPOF). O fato de ambos os servidores se encontrarem em funcionamento e ligados à rede não implica, porém, que desempenhem as mesmas tarefas. Essa é uma decisão do administrador e que pode ser caracterizada como balanceamento de carga.

2.2.1 Tipos de mecanismos de alta disponibilidade

O principal objetivo da alta disponibilidade é buscar uma forma de manter disponíveis os serviços prestados por um sistema a outros elementos, mesmo que o sistema em si venha a se modificar internamente por causa de uma falha. O conceito de mascarar as falhas está implícito através de redundância ou replicação.

Assim, um determinado serviço que possui requisitos de alta disponibilidade é colocado atrás de uma camada de abstração, que permita mudanças em seus mecanismos internos mantendo intacta a interação com elementos externos em caso de falha de um dos componentes de *hardware* ou *software* que suportam o serviço.

Esta camada de abstração é o coração da alta disponibilidade, uma subárea da Tolerância a Falhas, que também visa manter a disponibilidade dos serviços prestados por um sistema computacional, através da redundância de *hardware* e configuração de *software*. Em sistemas tolerantes a falhas, vários equipamentos ou componentes atuam juntos, agindo como um só, cada um monitorando os outros e assumindo seus serviços caso perceba que algum deles falhou.

Cabe salientar que alta disponibilidade não implica em tolerância a falhas, uma vez que a tolerância à falhas está caracterizada pela existência de redundância na maioria dos componentes de *hardware* e tem a habilidade de continuar os serviços independentemente da falha de *hardware* ou *software*. Entretanto, mesmo sistemas tolerantes a falhas estão sujeitos a interrupções causadas por erros humanos.

A tabela 1 ilustra um dos termos de comparação geralmente utilizados na avaliação de soluções HA (*High Availability*): níveis de disponibilidade segundo tempos de indisponibilidade (*downtime*). Esta tabela não inclui os tempos de indisponibilidade estimados para manutenções ou configurações programadas, pois são inerentes a cada solução e, portanto variáveis.

Tabela 1 – Níveis de alta disponibilidade

Nível de disponibilidade (%)	**Downtime/ano**	**Downtime/mês**
95%	18 dias 6:00:00	1 dia 12:00:00
96%	14 dias 14:24:00	1 dia 4:48:00
97%	10 dias 22:48:00	0 dias 21:36:00
98%	7 dias 7:12:00	0 dias 14:24:00
99%	3 dias 15:36:00	0 dias 7:12:00
99,9 %	0 dias 8:45:35.99	0 dias 0:43:11.99
99,99%	0 dias 0:52:33.60	0 dias 0:04:19.20
99,999%	0 dias 0:05:15.36	0 dias 0:00:25.92

Geralmente, quanto maior a disponibilidade, maior será o nível de redundância e custo das soluções. Tudo depende do tipo de serviço que se pretende disponibilizar, por exemplo, uma empresa provedora de serviços de telecomunicações requisitará certamente o nível mais elevado visando garantir níveis de disponibilidade elevados, evitando perda de clientes e perda de faturamento. É importante salientar que o nível de disponibilidade mensal não é o mesmo que o anual. Efetivamente, para se obter um nível de disponibilidade mensal de 97%, é necessário que o nível anual seja aproximadamente de 99,75%.

A disponibilidade está relacionada à taxa de ocorrência de falhas nos componentes de um sistema. Uma medida comum de confiabilidade de um componente é o tempo médio entre falhas (MTBF – *Mean Time Between Failures*), que pode ser calculado pela divisão do tempo total de operação do equipamento pelo número total de falhas MTBF = (*Total Operating Time*) / (*Total Number of Failures*). O MTBF de um sistema é obtido pela soma dos tempos

de operação de todas as unidades, incluindo as que não falharam, e dividindo pelo somatório de falhas das unidades. O tempo de operação é o somatório de horas que as unidades estavam em uso, ou seja, não estavam desligadas.

O tempo médio de reparação (MTTR – *Mean Time To Repair)* caracteriza o espaço de tempo (médio) que decorre entre a ocorrência da falha e a total recuperação do sistema ao seu estado operacional.

A disponibilidade de um sistema pode ser calculada pela fórmula:

Disponibilidade = MTBF / (MTBF + MTTR)

A figura 3 ilustra uma situação de *downtime* não planejado e outra sem interrupção.

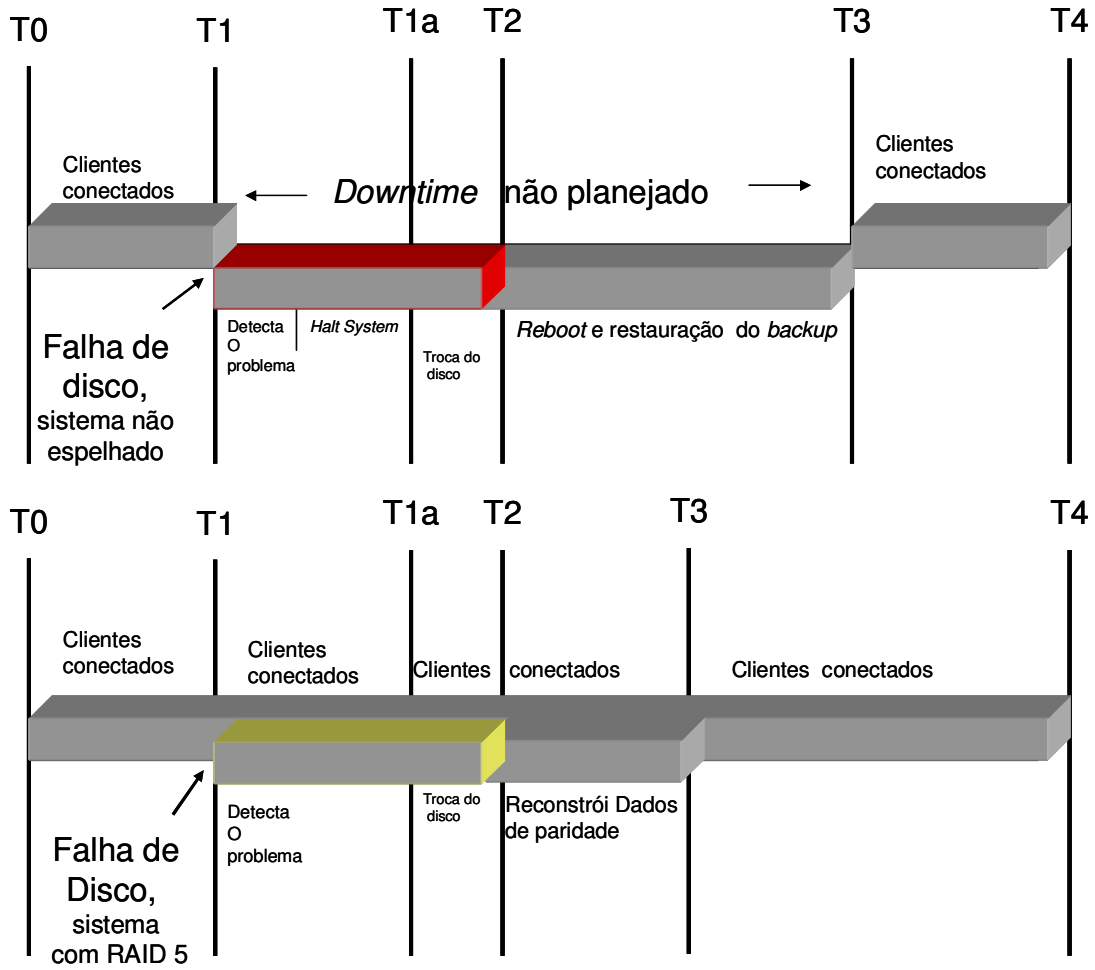

Figura 3 – Diagrama de tempo *downtime* planejado e eliminado

Na situação de *downtime* não planejado, a falha do disco ocorre em T1 e a transação do usuário é cancelada. O sistema permanece indisponível até T3, depois que o *hardware* foi substituído, o sistema restaurado e o banco de dados recuperado. Esta seqüência pode levar de horas até dias, dependendo da disponibilidade de *hardware* sobressalente. Em contrapartida, na situação de *downtime* eliminado, quando ocorre a falha de disco, outro disco assume as funções do que falhou e não existe interrupção na transação do usuário. Neste caso, o sistema de discos com RAID 5 [Ouchi 1978, Patterson 1987] (*Redundant Array of Independent Disks*), provê completa redundância para o sistema e a falha ocorre de forma transparente para o usuário final.

Os seguintes tipos de disponibilidade são descritos: básica; alta; contínua; tolerância a desastres e *E-available*.

Disponibilidade Básica: é caracterizada pela inexistência de mecanismos especiais em *software* ou *hardware* que visem de alguma forma mascarar eventuais falhas. Máquinas nesta classe apresentam uma disponibilidade de 99% a 99,9%. Ou seja, em um ano de operação a máquina pode ficar indisponível por um período de 9 horas a quatro dias. Estes dados são empíricos e os tempos não consideram a possibilidade de paradas planejadas, porém são aceitas como senso comum na literatura da área.

Alta Disponibilidade: adicionando-se mecanismos especializados de detecção, recuperação e mascaramento de falhas, pode-se aumentar a disponibilidade do sistema, de forma que este venha a se enquadrar na classe de Alta Disponibilidade. Nesta classe as máquinas tipicamente apresentam disponibilidade na faixa de 99,99% a 99,999%, podendo ficar indisponíveis por um período de pouco mais de 5 minutos até uma hora em um ano de operação.

Disponibilidade Contínua: à medida que aumentamos o nível de disponibilidade, obtém-se uma disponibilidade cada vez mais próxima de 100%, diminuindo o tempo de

inoperância do sistema de forma que este venha a ser desprezível ou mesmo inexistente. Deriva-se, então uma disponibilidade contínua, onde mesmo as interrupções planejadas e não planejadas são mascaradas, e o sistema está sempre disponível.

Tolerância a desastres: caracteriza a habilidade das instalações computacionais resistirem a múltiplas falhas ou a completa queda de sistemas de um *site* inteiro. É obtida através da replicação de recursos, utilização de estratégias de replicação e emprego de arquitetura de solução que permita um site assumir as funções do outro em caso de desastre.

Sistemas *E-available*: a expansão das atividades de negócio na Internet levou a necessidade de criação de um novo tipo de disponibilidade: *e-availability*, que caracteriza a capacidade de disponibilidade de um sistema suportar acesso rápido e de grande volume a *Web sites*. Em períodos de pico *Web* sites podem sofrer problemas de desempenho resultando em degradação ao ponto de causar a frustração de usuários e o conseqüente cancelamento de transações. *E-availability* é a combinação dos tipos de disponibilidade listados anteriormente e o planejamento de desempenho para suportar as situações de pico durante o acesso às aplicações *Web*.

2.2.2 Implementando alta disponibilidade

Na maioria das vezes o maior obstáculo para a implementação de HA não é a falha de *hardware* ou *software* em si, ou seja, é a inexistência de processos para lidar com as situações de falha. Portanto, é necessária a existência de um conjunto de processos e controles para garantir a efetividade e eficácia da gerência de disponibilidade.

As melhores práticas e padrões para Governança de Tecnologia da Informação (*Information Technology Governance*) [ITGI 2008] constituem-se em conjuntos de processos, seus relacionamentos e controles que direcionam a gestão de TI (Tecnologia da Informação), garantindo que esta possa suportar as estratégias e objetivos de negócio das empresas, incluindo-se a gestão de segurança.

O COBIT (*Control Objectives for Information and related Technology*) [ISACA 2008] constitui-se em um *framework* para governança de TI que contém objetivos de controle estruturados para permitir auditoria e quantificar os resultados de acordo com modelos de maturidade e indicadores de desempenho. Os objetivos de controle funcionam como um guia na implementação dos controles visando garantir os requisitos de eficiência, eficácia, confiança na comunicação, integridade, disponibilidade, conformidade e confiabilidade da informação.

O domínio "Entrega e Suporte" (*Delivery and Support*) do COBIT apresenta dois objetivos de controle relacionados com a gestão de segurança: Assegurar a Segurança de Sistemas e Assegurar a Continuidade de Serviços, sendo este último diretamente relacionado à disponibilidade dos serviços.

O ITIL (*IT Infrastructure Library*) [Cartlidge 2007] apresenta-se como um conjunto de melhores práticas para concepção, operação e gerenciamento de serviços de TI, tais como gerenciamento de *service desk*, incidente, mudança, capacidade, nível de serviço e segurança. O ITIL descreve as necessidades dos processos para a infra-estrutura de TI ser gerenciada de forma eficiente e eficaz, garantindo os níveis de serviço acordados com a organização e clientes.

O gerenciamento de disponibilidade é tratado no ITIL versão 3 nas publicações:

- Concepção do Serviço (*Service Design*) capítulos Gerenciamento de Disponibilidade (*Availability Management*), Gerenciamento da Continuidade dos Serviços (*IT Service Continuity Management*) e Gerenciamento de Segurança da Informação (*Information Security Management*);
- Operação do Serviço (*Service Operation*) capítulos Gerenciamento de Incidentes (*Incident Management Process*) e Gerenciamento de Problemas (*Problem Management Process*).

Está além do escopo deste trabalho detalhar os processos para a implementação e gerenciamento dos processos de alta disponibilidade, porém algumas sugestões são resumidamente listadas:

- Identificar os processos de negócio críticos e realizar análise de riscos;
- Identificar os possíveis pontos de demanda de recursos que podem sofrer picos e potenciais interrupções;
- Definir os objetivos de disponibilidade através da definição de SLAs;
- Preparar o ambiente físico adequadamente;
- Criar processos automatizados e documentação detalhada da topologia, processos e configurações;
- Utilizar ambientes de desenvolvimento e de testes separados do ambiente de produção;
- Manter estoque de partes sobressalentes;
- Realizar contratos de aquisições de capacidade sobre demanda;
- Criar processos de informação eficiente para escalonamento de problemas;
- Treinar a equipe de suporte;
- Realizar simulações de falhas e identificar possíveis melhorias no processo; e
- Preparar-se para desastres.

2.3 IP Multicasting

IP *multicasting* [Deering 1989] consiste em uma abordagem conservadora de banda, pois reduz o tráfego através da entrega simultânea de um único fluxo de dados para vários destinatários. Entre as aplicações que utilizam esta vantagem incluem-se: videoconferência, ensino a distância, comunicações corporativas, distribuição de *software* e de notícias. A maioria destas aplicações tem o desempenho como requisito principal em detrimento da confiabilidade.

IP *multicasting* entrega o tráfego de origem para múltiplos destinatários sem acrescentar carga adicional ao emissor ou destinatários, utilizando-se da menor quantidade de banda de rede necessária ao transporte dos pacotes. Os pacotes *multicast* são replicados na rede por roteadores, usualmente através do protocolo IGMP (*Internet Group Management Protocol*) [Cain 2002], sendo que outros protocolos podem ser utilizados, como por exemplo, o protocolo PIM (*Protocol Independent Multicast*) [Fenner 2006]. Todas as alternativas requerem que o emissor envie mais de uma cópia da mesma informação e algumas requerem até mesmo que o emissor envie uma cópia individual para cada receptor. A figura 4 ilustra o mecanismo de distribuição de informação através de IP *multicasting*.

2.3.1 Conceito de grupo *Multicast*

Multicast é baseado em conceito de grupo, onde um conjunto arbitrário de receptores expressa interesse em receber um determinado fluxo de dados. Este grupo não possui limitação geográfica e os componentes podem estar localizados em qualquer ponto da Internet. Os receptores interessados em fazer parte de um grupo devem ingressar no grupo através do protocolo IGMP. O receptor tem de obrigatoriamente fazer parte do grupo para receber as informações.

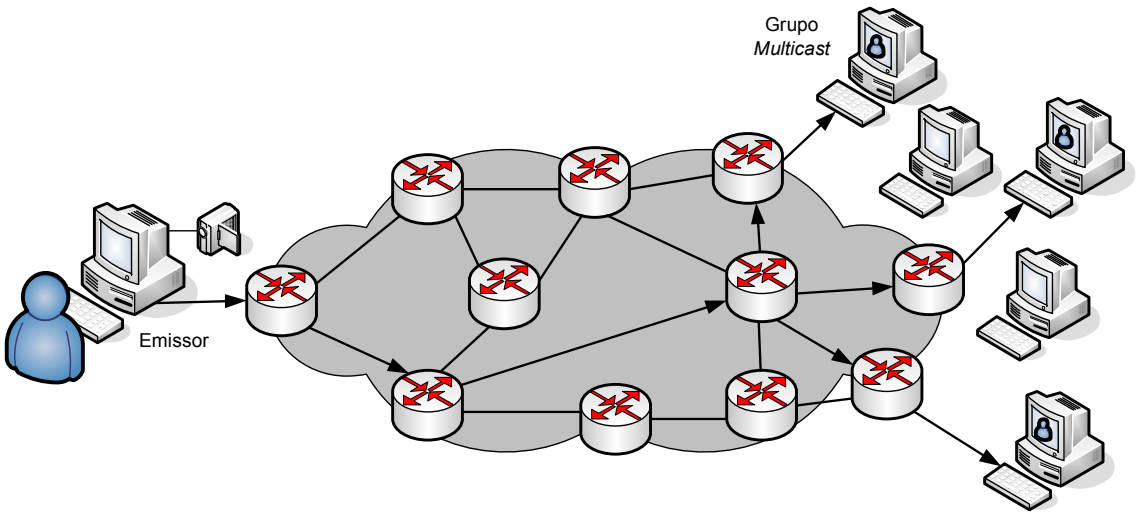

Figura 4 – Transmissão *multicast*: emissor envia um único fluxo para o Grupo *Multicast*

2.3.1.1 Endereçamento IP *Multicast*

O endereçamento *Multicast* especifica um grupo arbitrário de *hosts* que ingressaram no grupo e desejam receber as informações enviadas para tal grupo. O IANA (*Internet Assigned Numbers Authority*) controla a liberação de endereços IP *Multicast*. O espaço de endereçamento da classe D foi designado para este uso, o que significa que todos os endereços IP *Multicast* devem figurar no intervalo: 224.0.0.0 – 239.255.255.255. Note-se que este intervalo de endereços está designado apenas para os receptores da informação, ou seja, o endereço de origem (emissor) será um endereço *unicast*.

O IANA reservou o intervalo de endereçamento 224.0.0.0 – 224.0.0.255 para ser utilizado por protocolos no segmento de rede local. Pacotes com este endereçamento não devem ser repassados por roteadores e geralmente são transmitidos com um TTL (*Time-To-Live*) igual a 1. Os protocolos de rede utilizam este intervalo de endereçamento para descoberta automática de outros roteadores na rede e para comunicar alterações importantes nas informações das tabelas de rotas. Por exemplo, o protocolo OSPF (*Open Shortest Path*

First) [Moy 1998] utiliza os endereços 224.0.0.5 e 224.0.0.6 para trocar informações de estado de conexão (*link state*). A tabela 2 ilustra alguns endereços conhecidos:

Tabela 2 – Exemplos de endereços *Multicast* reservados

Endereço	**Utilização**
224.0.0.1	Todos os sistemas nesta sub-rede
224.0.0.2	Todos os roteadores nesta sub-rede
224.0.0.5	Roteadores OSPF
224.0.0.6	Roteadores OSPF designados
224.0.0.12	Servidores DHCP (*Dynamic Host Configuration Protocol*) ou Agente de repasse (*relay*)

O IANA reservou o intervalo de endereços de 224.0.1.0 – 238.255.255.255 para o Escopo de Endereçamento Global (*Globally Scoped Address*), utilizado para *multicast* de dados entre organizações através da Internet. Alguns endereços estão reservados para o IANA, como por exemplo, 224.0.1.1 para o protocolo SNTP [Mills 2006] (*Simple Network Time Protocol*).

O intervalo de endereçamento de 239.0.0.0 – 239.255.255.255 é chamado Escopo de Endereçamento Limitado (*Limited Scope Addresses* ou *Administratively Scoped Addresses*). São definidos pela RFC 2365 [Meyer 1998] para serem restritos a um grupo local ou organização. Roteadores são configurados com filtros para evitar que tráfego *multicast* neste intervalo saia dos domínios do Sistema Autônomo (AS – *Autonomous System*) [Hawkinson 1996] ou qualquer domínio definido pelo usuário.

A RFC 2770 [Meyer 2001] propõe a utilização do intervalo 233.0.0.0/8 (*Glop Addressing*), para organizações que possuem um número de AS reservado. O número do AS é embutido nos segundo e terceiro octetos deste intervalo para utilização por aplicações específicas do usuário.

2.3.1.2 Endereçamento *Multicast* camada 2

Normalmente as interfaces de rede, em um segmento de LAN, só recebem os pacotes destinados ao seu endereço MAC (*Media Access Control*) (*unicast*) ou para o endereço MAC de *broadcast*. Neste caso é necessário um mecanismo para que múltiplos *hosts* possam receber o mesmo pacote e possam diferenciá-lo entre os diferentes grupos *multicast*. No padrão IEEE 802.3 [Metcalfe 1975] o *bit* 0 (zero) do primeiro octeto é utilizado para indicar um *frame broadcast* e/ou *multicast*. A figura 5 ilustra a localização do *bit Broadcast/Multicast* no *frame Ethernet*.

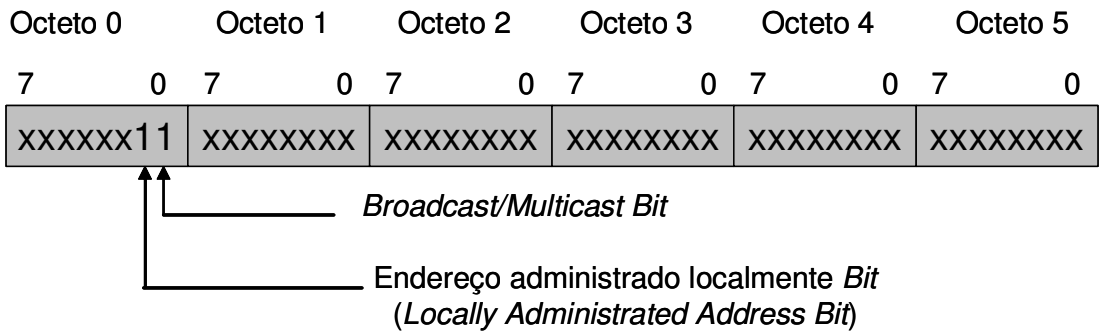

Figura 5 – Formato do endereço MAC IEEE 802.3

Este *bit* indica se o *frame* é destinado para um grupo arbitrário de máquinas ou para todas as máquinas na rede, no caso de endereço *broadcast* (0xFFFF.FFFF.FFFF). IP *Multicast* utiliza esta capacidade para transmitir pacotes IP para um grupo de máquinas em um segmento de rede.

2.3.1.3 Mapeamento para o endereço *Ethernet* MAC

Metade do bloco de endereçamento *Ethernet* (MAC) que se inicia com o endereço 01:00:5E em hexadecimal está reservado para endereços *multicast* pelo IANA. Isto cria o intervalo 0100.5e00.0000 – 0100.5e7f.ffff como disponível para endereços *Ethernet* MAC. Esta alocação permite que 23 *bits* no endereço *Ethernet* correspondam ao endereço do grupo

IP *Multicast*. Este mapeamento coloca os 23 *bits* de nível mais baixo nos 23 bits do endereço *Ethernet*. A figura 6 ilustra este mapeamento.

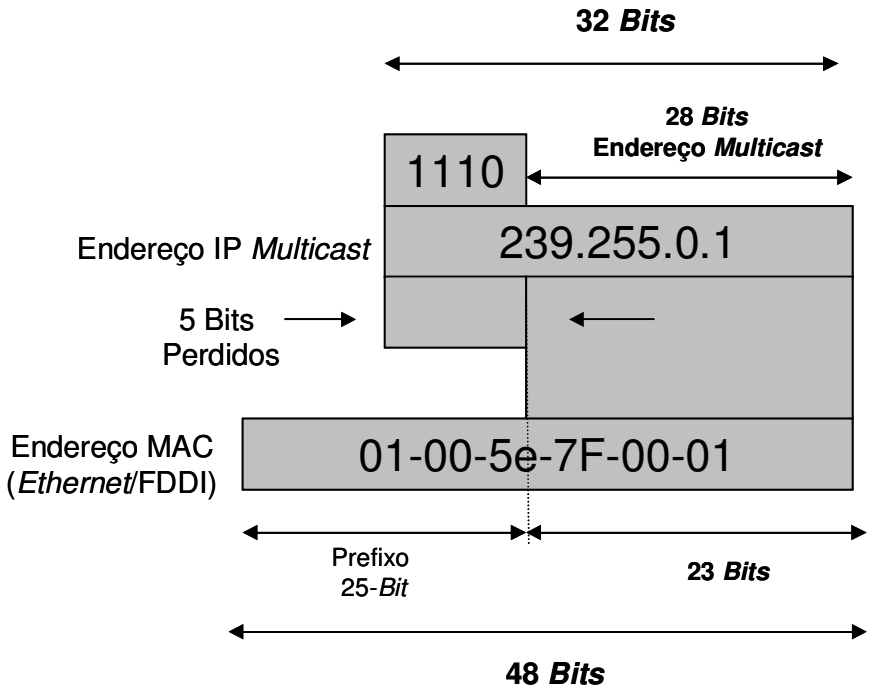

Figura 6 – Mapeamento endereço IP *Multicast* para endereço MAC

Por causa da perda dos 5 bits superiores no mapeamento, o endereço resultante não é único. De fato, 32 IDs (identificadores) de grupos *multicast* são mapeados no mesmo endereço *Ethernet*, o que pode levar a sobreposição de grupos deixando a responsabilidade de decisão sobre a aceitação do pacote para o *driver* da interface (descarte através de filtros) ou para o módulo IP.

2.3.2 *Internet Group Management Protocol* (IGMP)

IGMP [Cain 2002] é utilizado para registrar dinamicamente os participantes (*hosts*) de um grupo *multicast* em uma rede. Os *hosts* identificam os membros de um grupo através do envio de mensagens IGMP para os roteadores locais. Os roteadores recebem as mensagens e enviam outras mensagens periodicamente para verificar que grupos estão ativos ou inativos em uma sub-rede. A figura 7 ilustra o formato dos pacotes IGMP versões 1 e 2.

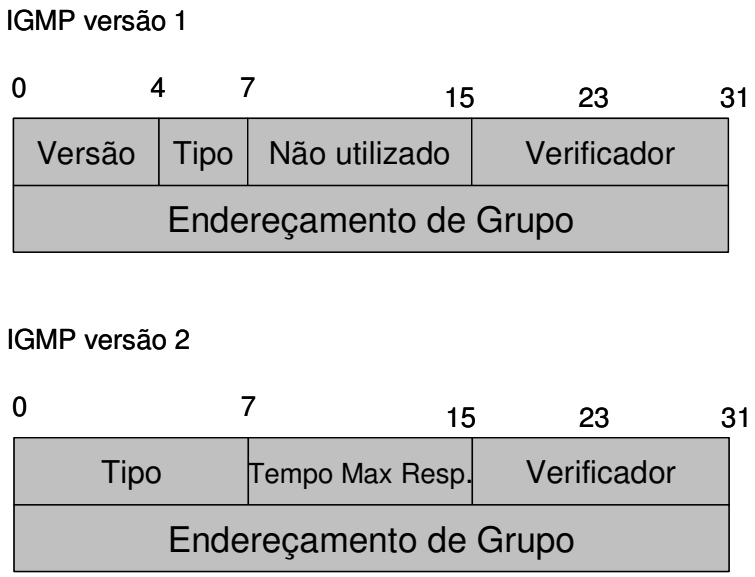

Figura 7 – Formato das mensagens IGMP

Na versão 1 existem dois tipos de mensagens IGMP: Pesquisa de Membro (*Membership Query*) e Resposta de Membro (*Membership Report*). *Hosts* enviam para a rede mensagens do tipo *Resposta de Membro* para indicar que têm interesse em participar de um determinado grupo *multicast*. O roteador envia periodicamente mensagens IGMP do tipo *Pesquisa de Membro*, para verificar se pelo menos um *host* da sub-rede está interessado em receber tráfego destinado ao grupo *multicast*. Quando não existe resposta para três mensagens de pesquisa consecutivas, o roteador suspende as atividades do grupo e pára de enviar tráfego para o mesmo.

Na versão 2 existem quatro tipos de mensagens IGMP:

- Pesquisa de membro;
- Resposta de membro versão 1;
- Resposta de membro versão 2; e
- Deixar o grupo (*Leave Group*).

A versão 2 funciona basicamente da mesma forma que a versão 1, a grande diferença está na mensagem "Deixar o Grupo". Os *hosts* podem comunicar ao roteador sua decisão em

deixar o grupo através do envio desta mensagem. O roteador envia mensagens de pesquisa para determinar se existem *hosts* remanescentes interessados em receber tráfego do grupo *multicast*. Quando não existe resposta para as mensagens de pesquisa, o roteador suspende as atividades do grupo e pára de enviar tráfego para o mesmo. Desta forma é possível reduzir a latência gerada pela versão 1 do protocolo IGMP.

2.3.3 *Multicast* no ambiente de *switches* camada 2

O comportamento padrão de um *switch* de rede camada 2 é replicar o tráfego *multicast* para todas as portas que pertençam à rede. Isto vai contra o princípio básico de um *switch* que é limitar o tráfego para as portas que realmente necessitam receber o tráfego. Uma técnica para tratar esta situação consiste em o *switch* examinar algumas informações de camada 3 nos pacotes IGMP enviados entre roteadores e *hosts*.

Quando o *switch* identifica um pacote IGMP do tipo "Resposta de Membro" de um *host* pertencente a um grupo *multicast*, o *switch* adiciona o número da porta do *host* a uma tabela de controle associada com o grupo *multicast*. Quando o *switch* identifica uma mensagem IGMP do tipo "Deixar o Grupo", ele remove o número da porta do *host* da tabela de controle. Esta técnica é denominada IGMP *Snooping*.

Uma vez que as mensagens de controle IGMP são transmitidas como pacotes *multicast*, elas não são diferenciadas de pacotes de dados *multicast* na camada 2. Logo, um *switch* executando IGMP *Snooping* tem de verificar todos os pacotes de dados *multicast* para verificar se eles contêm alguma mensagem de controle.

Portanto, se IGMP *Snooping* é configurado em *switches* de baixo desempenho, em situações de tráfego de dados *multicast* intenso, podem ocorrer sérios impactos no desempenho da rede. Uma alternativa é implementar IGMP *Snooping* em *switches* de desempenho considerável, em especial aqueles que possam executar esta função em

processadores ASIC (*Application Specific Integrated Circuits*) dedicados. Assim, a verificação é realizada diretamente em *hardware* não afetando a desempenho da rede.

2.4 Virtual Routing Redundacy Protocol (VRRP)

As redes constituem um importante elemento em um sistema de comunicações. Portanto, interrupções causadas por falhas nas redes têm que ser evitadas ou minimizadas. O protocolo VRRP (*Virtual Router Redundancy Protocol*) [Hinden 2004] foi desenvolvido para eliminar o SPOF (*single point of failure*) ponto de falha único, quando clientes utilizam um único endereço IP como *gateway* padrão na rede LAN. A figura 8 ilustra a topologia clássica de *gateway* padrão (*default gateway*) da rede.

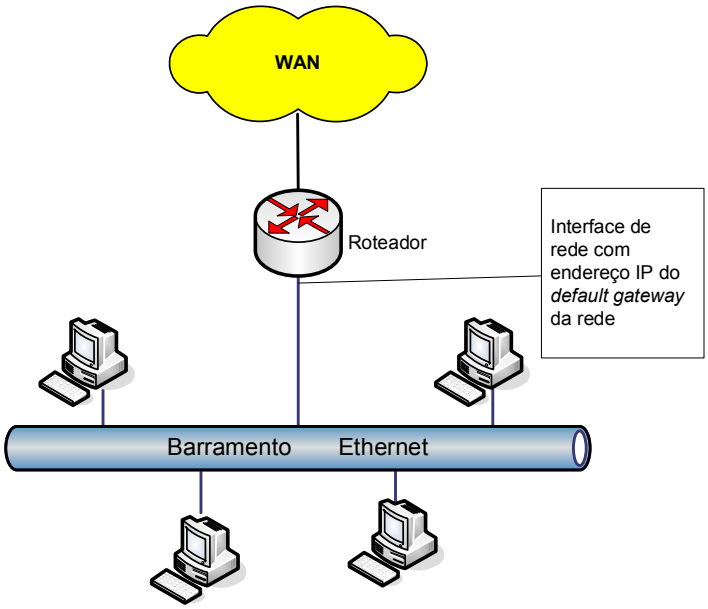

Figura 8 – *Default Gateway* da rede

Nesta topologia, se o roteador falhar a comunicação externa será interrompida. Para minimizar o risco de interrupção, outro roteador poderia ser inserido na rede, mas todas as estações cliente teriam de ser configuradas novamente no caso de falha do roteador principal.

O VRRP utiliza o conceito de roteador virtual e provê redundância sem a necessidade de intervenção manual ou configurações adicionais nos *hosts*. O VRRP especifica um protocolo eletivo, que dinamicamente atribui a responsabilidade de roteador virtual a um dos elementos da rede. A tabela 3 resume os principais papéis associados ao VRRP.

Tabela 3 – Definições associadas ao VRRP

Escopo	**Definições**
Roteador VRRP	O roteador que executa o protocolo VRRP. Pode participar de um ou mais roteadores virtuais.
Roteador virtual	Objeto abstrato administrado pelo VRRP, que atua como roteador padrão (*default router*) para *hosts* de uma rede. Consiste de um VRID (*Virtual Router Identifier*) e um conjunto de endereços IP. Um roteador VRRP pode ser *backup* de um ou mais roteadores virtuais.
Dono do endereço IP	O roteador VRRP que possui o endereço IP do roteador virtual como endereço real das interfaces.
Endereço IP primário	Um endereço IP selecionado a partir do conjunto de endereços atribuídos às interfaces. Um algoritmo possível é sempre selecionar o primeiro endereço.
Mestre virtual	O roteador VRRP que assume a responsabilidade de repassar os pacotes enviados para o endereço IP associado ao roteador virtual, e responder às requisições ARP (*Address Resolution Protocol*) para este endereço IP.
Backup virtual	O conjunto de roteadores VRRP disponíveis para assumir a responsabilidade de repasse de pacotes, caso o mestre corrente falhe.

Este mecanismo eletivo define um grupo de roteadores denominado Grupo VRRP, que designa um dos roteadores como mestre e os outros como secundários (*backup)*. O roteador

mestre é o roteador que controla os endereços IP associados ao roteador virtual. O roteador virtual é caracterizado pelo identificador do roteador virtual VRID (*Virtual Router Identifier*) e um conjunto de endereços IP. O processo eletivo do roteador mestre é determinado por uma prioridade previamente estabelecida. Após o processo eletivo, para minimizar tráfego de rede, somente o roteador mestre envia periodicamente mensagens de anúncio para a rede.

O roteador *backup* somente se torna mestre se este falhar e tornar-se indisponível, ou se a prioridade do *backup* for alterada para um valor maior que a prioridade definida para o mestre. Este processo é definido pelos estados Início, Mestre e *Backup*, conforme ilustrado na figura 9, sendo que:

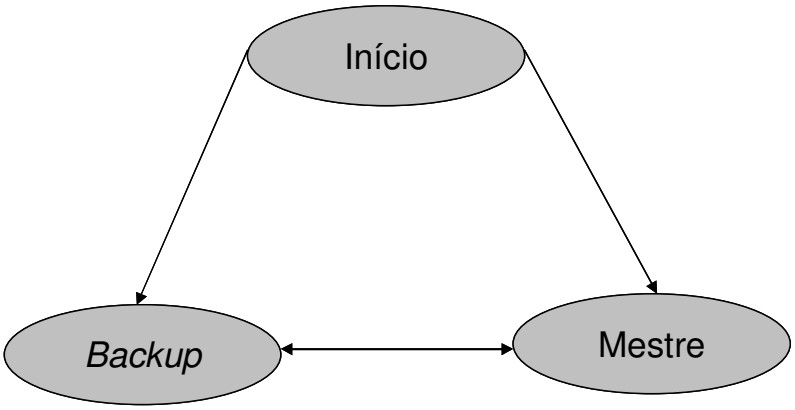

Figura 9 – Diagrama de estados do protocolo VRRP

- Início – estado no qual o roteador detecta seu próprio estado de acordo com pacotes de anúncio recebidos;
- Mestre – estado no qual o roteador envia pacotes de anúncio, responde às requisições ARP (*Address Resolution Protocol*) [Plummer 1982] destinadas ao endereço do roteador virtual e repassa pacotes com endereço de destino MAC igual ao endereço MAC do endereço do roteador virtual; e
- *Backup* – o roteador identifica os pacotes de anúncio, compara sua prioridade com a prioridade divulgada, monitora o estado do mestre e não responde às requisições ARP destinadas ao endereço do roteador virtual.

O VRRP provê funções similares aos protocolos proprietários HSRP (*Hot Standby Router Protocol*) [Plummer 1982] e IPSTB (IP *Standby Protocol*) [Higginson 1997].

O VRRP utiliza IP *multicasting* para enviar as mensagens. Cada roteador virtual possui um endereço MAC bem definido, que é utilizado como endereço de origem para todas as mensagens VRRP enviadas pelo roteador mestre periodicamente para habilitar a identificação na rede (*bridge learning*). O endereço MAC do roteador virtual está associado ao endereço IEEE MAC 00-00-5E-00-01-{VRID}, onde os três primeiros octetos são derivados do IANA OUI (*Organizationally Unique Identifier*) e os dois octetos seguintes (00-01) indicam o bloco de endereço reservado para o VRRP. {VRID} é o identificador do roteador virtual. Este mapeamento permite a utilização de até 255 roteadores VRRP em uma rede.

2.4.1 Formato do pacote VRRP

Os pacotes VRRP são transportados pelo protocolo IPv4 (ainda não existe especificação para IPv6) com endereços de destino IP *multicast*, onde o endereço de origem é o endereço primário atribuído à interface, o endereço de destino é o endereço atribuído pelo IANA: 224.0.0.18; e o TTL padrão é 255 (outros valores indicam que o pacote deve ser descartado). O número de protocolo atribuído pelo IANA para o VRRP é 112 (decimal). A figura 10 ilustra o formato dos campos do pacote VRRP e em seguida são apresentadas as descrições destes campos.

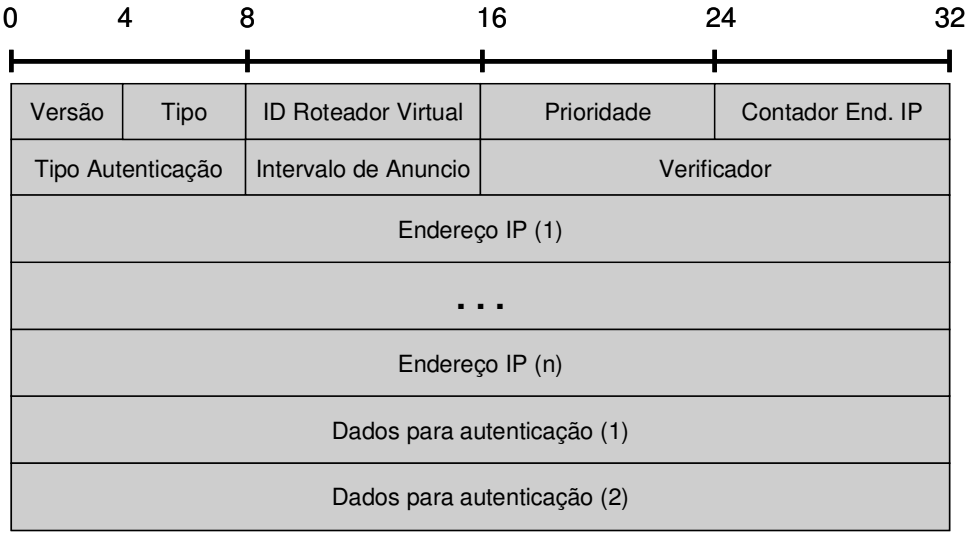

Figura 10 – Formato do pacote VRRP

- Versão – este campo especifica a versão do protocolo VRRP que a RFC 3768 [Hinden 2004] define como 2;
- Tipo – especifica o tipo do pacote VRRP. O único tipo definido é 1, que indica um pacote de anúncio. Pacotes com tipos diferentes devem ser descartados;
- Identificador do Roteador Virtual (VRID) – identifica o roteador virtual para o qual o pacote está carregando informações de estado. O intervalo de definição é um número decimal que varia de 1 a 255;
- Prioridade – inteiro sem sinal de 8 *bits*, especifica a prioridade do roteador emissor dos pacotes VRRP. Quanto maior o valor, maior será a prioridade. O valor da prioridade do roteador VRRP que suporta os endereços IP associados com o roteador virtual tem de ser 255 (decimal). Os roteadores *backup* têm de utilizar uma prioridade que varia de 1 a 254 e o valor *default* é 100 (decimal). O valor zero tem um significado especial indicando que o mestre corrente deixou de participar do grupo VRRP. É utilizado para acionar a transição do *backup* para mestre de forma rápida sem esperar pelo período de silêncio (*timeout*) do roteador mestre.

- Contador de Endereços IP – indica a quantidade de endereços IP contidas no anúncio VRRP;
- Tipo de Autenticação – inteiro de 8 *bits* que identifica o método de autenticação utilizado. O tipo de autenticação é único em um roteador virtual. Pacotes cujo tipo de autenticação é desconhecido ou não são idênticos ao método local de autenticação devem ser descartados. A versão de VRRP especificada na RFC 2338 [Knight 1998] tinha vários tipos de autenticação definidos, que foram removidos na especificação da RFC 3768, porque não agregavam a segurança desejada e permitiam que múltiplos mestres fossem criados em uma mesma rede para o mesmo VRID. Os tipos de autenticação atuais são definidos pela tabela 4:

Tabela 4 – Tipos de autenticação do protocolo VRRP

Tipo de Autenticação	**Função**
0	Sem autenticação.
1	Reservado – definido para manter compatibilidade com a RFC 2338.
2	Reservado – definido para manter compatibilidade com a RFC 2338.

- Intervalo de anúncio – indica o intervalo em segundos entre pacotes de anúncio. O padrão é 1s. Este campo é utilizado na pesquisa de problemas com roteadores com erros de configuração;
- Verificador – é utilizado para verificar a integridade da mensagem VRRP;
- Endereços IP – um ou mais endereços que estão associados com o roteador virtual e é utilizado para verificação de problemas em roteadores erroneamente configurados; e
- Dados para autenticação – um campo do tipo *string* utilizado para manter compatibilidade com a RFC 2338. Deve ser iniciado com zero na transmissão e ignorado no recebimento.

2.4.2 Requisições ARP

Quando um *host* envia requisições ARP para um dos endereços IP do roteador virtual, o roteador mestre virtual tem de responder a requisição com o endereço MAC do roteador virtual. O mestre não pode responder as requisições ARP com seu endereço MAC físico. Este esquema permite que o cliente sempre utilize o mesmo *gateway*, ou seja, o mesmo endereço MAC, independentemente de qualquer que seja o roteador mestre em operação.

No processo de início do roteador VRRP ou carga do sistema (*boot*) ele não deve enviar mensagens ARP contendo seu endereço MAC físico, para os endereços IP, os quais ele é responsável. Portanto, deve enviar somente mensagens ARP que incluem o endereço MAC virtual. Isto implica em:

- Durante o processo de configuração de uma interface, o roteador VRRP deve realizar um *broadcast* de mensagens de requisição de ARP gratuito (*gratuitous* ARP), contendo o endereço MAC do roteador virtual. Este processo deve ser realizado para cada endereço IP da interface; e
- Durante o processo de carga do sistema operacional e etapa de operacionalização das interfaces para VRRP, o roteador deve atrasar as requisições de ARP gratuito e respostas até que os endereços IP e endereço MAC virtual estejam ambos configurados.

2.4.3 Considerações de segurança

O VRRP não inclui em sua versão corrente nenhum mecanismo de autenticação. Os mecanismos incluídos em versões anteriores se mostraram ineficientes em termos de segurança.

De acordo com a natureza do protocolo VRRP, mesmo que as mensagens sejam codificadas através do emprego de técnicas de criptografia, não é possível prevenir que

roteadores hostis se comportem como se fossem o mestre VRRP da rede, criando condições de múltiplos mestres.

A autenticação de mensagens VRRP pode prevenir que um roteador hostil interfira na rede e force com que os roteadores migrem para o estado *backup*. Entretanto, múltiplos mestres na rede podem causar tanta interrupção quanto nenhum roteador disponível, o que o processo de autenticação não evita. Mesmo que um roteador hostil não consiga interromper o VRRP, é possível interromper as mensagens ARP e criar o mesmo efeito dos roteadores migrando para o estado *backup*.

Cabe salientar que estes tipos de ataques não deveriam ter grande severidade, pois têm que ser realizados internos à rede, devido à natureza do VRRP (TTL=255 verificado no recebimento do pacote) e aos respectivos controles de segurança geralmente empregados nas redes corporativas. Entretanto, com o advento de vermes [Hansche 2004] é possível que uma máquina interna da rede seja contaminada e execute os ataques, ampliando o grau de severidade do mesmo.

A figura 11 ilustra uma topologia com mais de um roteador virtual.

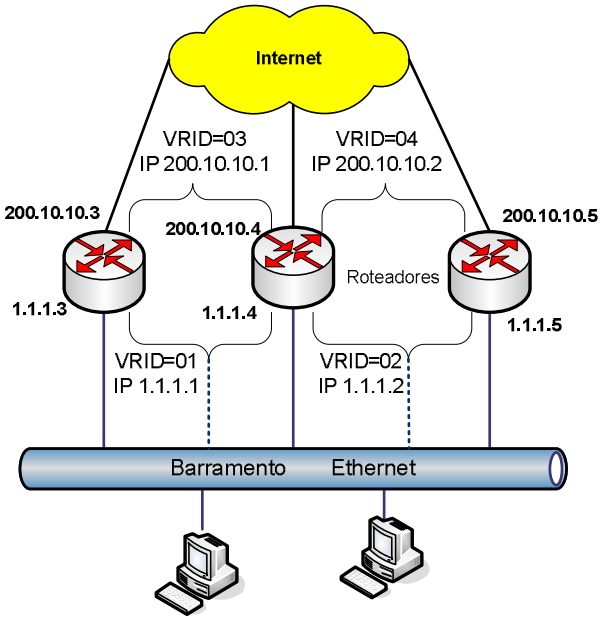

Figura 11 – Topologia VRRP com vários roteadores virtuais

Mesmo com as vulnerabilidades inerentes ao protocolo, o VRRP é largamente utilizado para prover redundância de *gateway* para as redes e como protocolo de suporte a implementação de alta disponibilidade em estruturas as mais variadas, como por exemplo, a estrutura do *Keepalived daemon*.

2.5 *Keepalived* Daemon

Keepalived [Keepalived 2003] constitui-se em um conjunto robusto de componentes de *software*, para implementação de protocolo de alta disponibilidade e a verificação de saúde entre máquinas *Linux* de uma mesma rede. Estas duas características permitem a implementação de estrutura para manipular conjuntos (*cluster*) de servidores *Linux* virtuais LVS (*Linux Virtual Server*), através da adição ou remoção de servidores no conjunto de acordo com as informações de decisão derivadas dos verificadores de saúde (*helth-checkers*). O *Keepalived* também é utilizado para implementar ambientes de alta disponibilidade estruturados sob o sistema operacional Linux para *firewall*.

2.5.1 *Linux Virtual Server*

O LVS é uma versão aprimorada de *Kernel* do Linux que adiciona facilidades para balanceamento de carga. Atua como uma *bridge* de rede através da tradução de endereços NAT (*Network Address Translation*) [Egevang 1994] para balancear os fluxos de pacotes. Os componentes do LVS são:

- Interface WAN – interface da rede WAN que será acessada pelos clientes;
- Interface LAN – interface de rede LAN que gerencia e comunica-se com os servidores cuja carga será balanceada;

- Linux *Kernel* – uma versão de *kernel* funcionando como roteador com o módulo de LVS mais recente;
- VIP (*Virtual* IP) – o endereço IP virtual que será acessado pelos clientes;
- Servidores reais – caracteriza os servidores disponíveis para executar as aplicações a serem balanceadas;
- *Server pool* – conjunto de servidores que suporta o balanceamento de um determinado serviço. São definidos através da escolha de servidores no conjunto de servidores reais;
- Servidor virtual – ponto de acesso de um *Server pool*;
- Serviço virtual – fluxos de protocolos associados com o VIP.

A figura 12 ilustra a visão global do LVS.

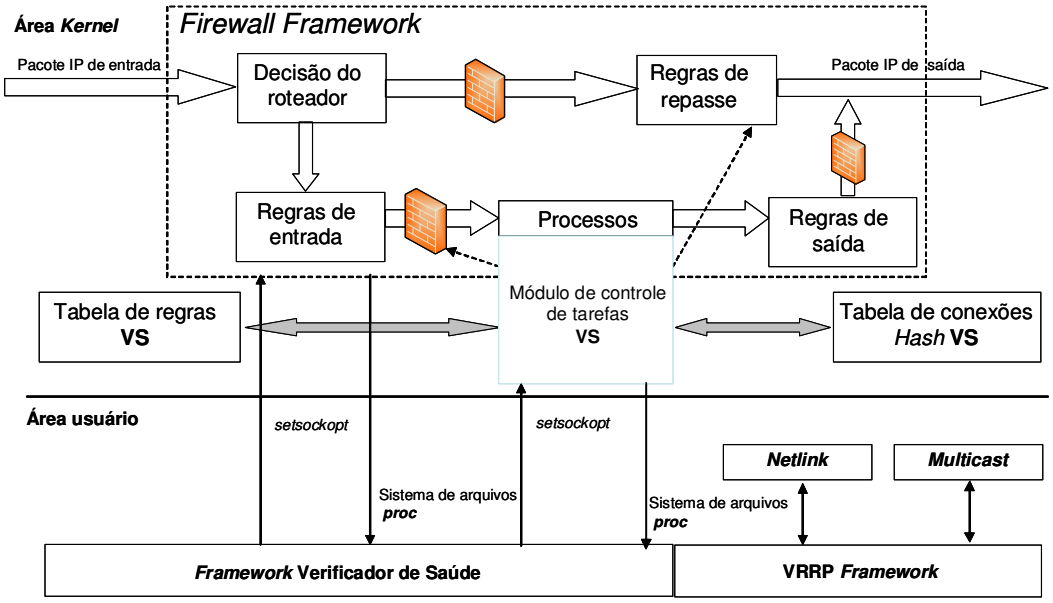

Figura 12 – Visão global do LVS

A arquitetura do *Keepalived* utiliza os seguintes componentes de *kernel*:

- LVS *kernel framework* – utiliza a chamada de sistema *setsockopt* para o *kernel* versão 2.2 e *setsockopt netfilter* para o *kernel* 2.4. A função *setsockopt* manipula opções em múltiplos níveis, associadas com um *socket*;

- *IPCHAINS* ou *firewall* framework – para o kernel 2.2, este módulo é responsável pelas regras de controle, tradução de endereços NAT e controle de fluxo de pacotes. No kernel 2.4 as regras de NAT são gerenciadas pelas chamadas *netfilter*;
- Interface *NETLINK* – esta interface é utilizada para configurar e remover endereços IP virtuais relacionados ao protocolo VRRP;
- *MULTICAST* – as mensagens de anúncio são enviadas para um grupo *Multicast*. Instâncias sincronizadas de VRRP são monitoradas e controladas pelo mestre (*Real Load Balancer*) do conjunto LVS.

2.5.2 Estrutura interna do *Keepalived*

O *Keepalived* utiliza uma estrutura *multithreaded* baseada em um multiplexador central de entrada e saída (I/O) e seus principais componentes são: o "Verificador de Saúde"; e "Gerador de Pacotes VRRP". A figura 13 ilustra a estrutura interna do *Keepalived daemon*.

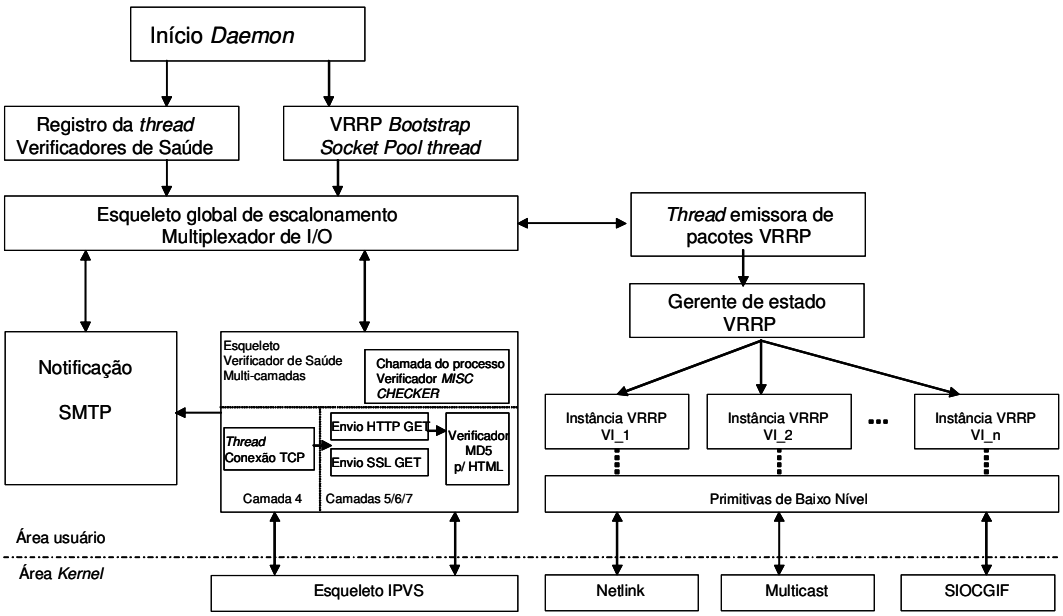

Figura 13 – Estrutura interna do *Keepalived*

O "Verificador de Saúde" controla o estado operacional de cada um dos componentes do *cluster* LVS e pode utilizar uma das seguintes opções:

- Verificador TCP – executa na camada 4 a verificação padrão do protocolo TCP (*Transmission Control Protocol*) [DARPA 1981, Stevens 2000], através de conexões TCP *nonblocking/timed-out* e se o servidor remoto não responde a uma requisição TCP após um determinado tempo, o teste é considerado falho e o servidor é removido do conjunto;
- HTTP (*HyperText Transfer Protocol*) GET – executa na camada 5 o teste através do envio de mensagens HTTP [Fielding 1999] GET para uma URL (*Uniform Resource Locator*) definida, então o conteúdo da mensagem GET é verificado através de um algoritmo MD5 [Rivest 1992]. Sendo que se o resultado não é idêntico a um valor esperado, então o teste é considerado falho e o servidor é removido do conjunto;
- SSL (*Secure Sockets Layer*) GET – executa o mesmo teste da mensagem HTTP GET, porém utilizando o protocolo SSL [Ylonen 2006]; e
- MISC CHECK – esta opção permite que um programa, definido pelo usuário, seja utilizado como verificador, onde o resultado tem de ser 0, que indica teste falho, ou 1 que indica teste positivo.

O "Gerador de Pacotes VRRP" (*packet dispatcher*) administra as instâncias de VRRP para controlar o processo de passagem de atividades de um nó falho para um nó ativo (*failover*), quando o nó principal falhar. Ele também gerencia o sincronismo das instâncias VRRP, realiza a verificação de integridade dos pacotes de anúncio utilizando IPSec AH ICV (*IP Security Protocol – Authentication Header – Integrity Check Value*) [Doraswamy 1999], realiza chamadas do sistema e realiza o processo de retorno para as funções de mestre (*fallback*), quando este retorna de um processo de falha e existe *backup* executando as atividades primárias.

Os componentes "Verificador de Saúde" e "Gerador de Pacotes VRRP" utilizam as seguintes primitivas de baixo nível:

- Notificação SMTP (*Simple Mail Transfer Protocol*) – permite ao *Keepalived* enviar notificações assíncronas através de *e-mail*, protocolo SMTP [Klensin 2001];
- IPVS *framework* – constitui a interface para manipulação dos servidores (*real server pool*) através de métodos de balanceamento de carga;
- *Netlink* – interface para roteamento dos pacotes VRRP, administra os endereços virtuais VIPs (*Virtual* IP) do protocolo VRRP;
- *Multicast* – utilizado para enviar anúncios do protocolo VRRP;
- *IPCHAINS framework* – utilizado para fins de compatibilidade com o *kernel* versão 2.2, realiza tradução de endereços NAT. Versões mais recentes utilizam o módulo *NETFILTER*;
- *SYSLOG* – todas as mensagens de notificação do *Keepalived* são armazenadas em *logs* através do *syslog daemon*.

O *Keepalived* foi construído com o objetivo de ser genérico e flexível, permitindo a inclusão de outros verificadores de saúde para o conjunto.

3 *FIREWALL* E IPS

3.1 Introdução

O *firewall* consiste em um método de proteção de uma rede contra outra não confiável (*untrusted*) [Hansche 2004], através do controle de tráfego não autorizado. O método utilizado pode variar, porém um *firewall* é constituído por dois componentes: um para o bloqueio de tráfego e outro para tráfego autorizado. O *firewall* se comporta como um *gateway* que repassa o tráfego de um domínio de rede para outro, desde que o tráfego esteja em conformidade com uma regra de segurança específica para o tipo de tráfego em análise.

Um sistema de prevenção de intrusos IPS (*Intrusion Prevention System*) [Plato 1990] é constituído por *hardware* e *software* que monitora as atividades de uma rede e/ou sistema buscando encontrar atividades maliciosas e/ou comportamento indesejável. Caso alguma atividade não autorizada seja detectada, ele pode reagir em tempo real para bloquear ou prevenir a concretização destas atividades. IPSs de rede, por exemplo, operam em linha com a rede para monitorar todo o tráfego de rede buscando encontrar, deter código malicioso e ataques. Quando um ataque é detectado ele pode excluir os pacotes ofensores, enquanto permite outros fluxos de pacotes.

Firewalls e IPS são componentes importantíssimos na arquitetura de segurança das empresas, pois protegem contra acesso não autorizado e evitam atividades maliciosas, respectivamente. Além disso, se empregados em conjunto, ampliam exponencialmente as capacidades de proteção e defesa do negócio.

Como elementos cruciais na gestão de segurança, eles têm requisitos próprios de alta disponibilidade, que são delineados pelas necessidades do negócio. Entretanto, não existe padronização na implementação de mecanismos de HA para *firewall* e IPS. O que pode ser observado conforme capítulo 5 e seção 2 é uma miscelânea de protocolos e utilização de mecanismos proprietários.

A figura 14 ilustra uma topologia em camada simples para *firewall*. Entretanto podem ser empregados em duas ou três camadas, [Hansche 2004] dependendo das necessidades do negócio. A discussão do emprego em camadas, suas vantagens e desvantagens estão além do escopo deste trabalho.

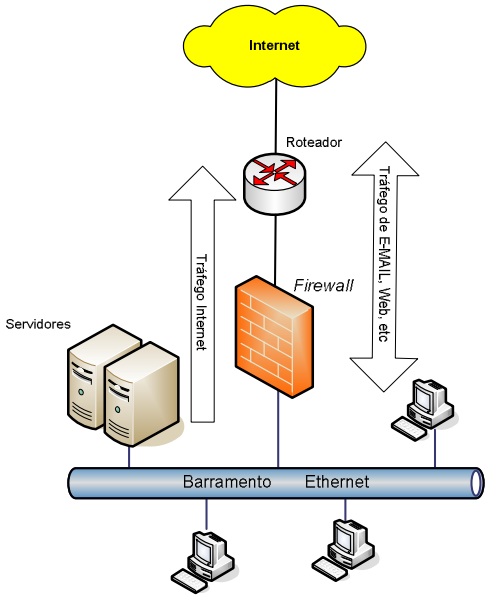

Figura 14 – *Firewall* topologia em camada simples

As demais seções deste capítulo estão estruturadas da seguinte forma: seção 3.2 descreve os tipos de *firewall* e suas aplicações. A seção 3.3 expõe os conceitos e tipos de IPS. A seção 3.4 descreve o conceito de *heartbeat*. Finalmente a seção 3.5 descreve mecanismos de alta disponibilidade para *firewall* e IPS.

3.2 Tipos de *Firewall*

Firewall pode ser definido como uma barreira de proteção, que controla o tráfego de dados entre uma rede e a Internet ou entre redes distintas. Seu objetivo é permitir somente a transmissão e a recepção de fluxos de dados autorizados. Os *firewalls* são classificados tipicamente como: filtros de pacote, *proxies* de circuito, *gateways* de aplicação, *firewall* de inspeção de estado de sessão (*stateful firewall)*, ou uma combinação destes.

3.2.1 Filtros de pacotes

Filtro de pacotes (*packet filtering*) característica geralmente disponível para ser configurada em roteadores está centrada na análise de pacotes, comparando-os contra um conjunto de regras para determinar se o pacote deve ser permitido ou não. As regras podem incluir o endereço IP de origem, endereço IP de destino, horário do dia, a porta endereçada, autenticação do usuário e quantidade de conexões simultâneas na rede. Uma regra de política de segurança que permite um pacote passar através de um filtro de pacotes é denominada *pinhole*.

3.2.2 *Application gateways*

Proxy ou *Application Gateway* comporta-se como um intermediário no processo de conexão, operando na camada 7 (aplicação) do modelo de referência OSI (*Open Systems Interconnection*). A sessão do usuário estabelece uma conexão com o *proxy*, que por sua vez estabelece uma conexão com o endereço de destino. A resposta para o pedido é recebida e analisada antes de ser entregue para o solicitante original. Como resultado, duas conexões são estabelecidas adicionando custo de desempenho para a rede.

É necessária uma camada de *proxy* para cada protocolo que será verificado. O processo de verificação e decisão sobre o pacote, na camada de aplicação, requer que o *proxy*

conheça profundamente os serviços dos protocolos e suas peculiaridades. Alguns protocolos, como HTTP, não se comportam bem através de *proxies*, por causa do tempo necessário para a autenticação e estabelecimento das sessões. Este tipo de *firewall* introduz perda de desempenho para as aplicações.

3.2.3 *Circuit-Level gateways*

Este tipo de *firewall* opera na camada 5 (sessão) do modelo de referência OSI, estabelecendo conexões entre *hosts* confiáveis e clientes. Similarmente ao *firewall* do tipo *proxy*, não ocorre conexão direta entre sistemas de origem e destino. Além disso, a aplicação tem de suportar o mecanismo intermediário de repasse dos pacotes. Um exemplo de mecanismo de repasse compatível é o SOCKS [Leech 1996].

SOCKS, abreviatura da palavra *sockets*, é um protocolo utilizado para manusear tráfego TCP através de um servidor intermediário (*gateway*). Seu propósito é habilitar aplicações cliente de um lado do *gateway* SOCKS, conseguir acesso aos servidores de aplicação localizadas do outro lado do *gateway* SOCKS, sem a necessidade de conexão IP direta. Constitui-se em um mecanismo de *firewall* simples, porque verifica os pacotes entrantes, saintes e esconde os endereços IP de aplicações clientes.

3.2.3 *Firewall* de inspeção de estado de sessão

Firewall de Inspeção de Estado de Sessão (*stateful inspection*) é um tipo de *firewall* bem versátil que opera em todas as camadas do modelo de referência OSI. Entretanto, a maioria é operada nas camadas 3 e 4 deste modelo.

Independentemente do modelo de operação, os estados das comunicações e aplicações, derivados das sete camadas, são utilizados para construir uma tabela de estado geral. Esta tabela de estado é consultada e o resultado da consulta é utilizado no processo decisório para permitir ou eliminar um pacote. Ou seja, o *firewall* guarda o estado de todas as últimas transações efetuadas e inspeciona o tráfego para evitar pacotes ilegítimos. Por

exemplo, se a resposta de um pacote é recebida no *firewall*, tem de haver uma entrada na tabela de estado contendo informações do pacote requisitante da sessão. Se não existir, o pacote será descartado. A maioria dos *firewalls* comercialmente disponíveis está baseada neste modelo.

A figura 15 ilustra os tipos de *firewall* e relacionamento com as camadas do modelo de referência OSI.

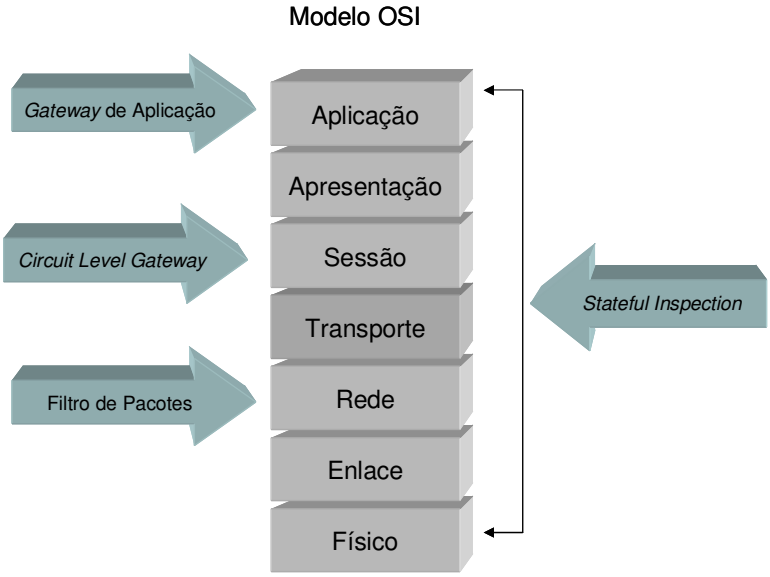

Figura 15 – Tipos de *firewall* e o modelo OSI

Um novo conceito de *firewall* denominado Gerência de Ameaças Unificada UTM (*Unified Threat Management*), ou *Firewall* de Próxima Geração, apareceu recentemente no mercado como uma opção ainda mais robusta para segurança das redes. O UTM enseja em seu conceito um conjunto de múltiplas proteções de segurança incluídas em uma única plataforma.

Um UTM típico disponibiliza como serviços de segurança: *firewall* de inspeção de estado de sessão, antivírus [Hansche 2004], filtro de *Web* [Hansche 2004], VPN (*Virtual Private Network*) [Doraswamy 1999], anti-spam [Hansche 2004], e serviços de prevenção de

intrusos IPS; oferecendo um mecanismo de gerência centralizada sem impactar no desempenho da rede.

3.3 Sistema de Prevenção de Intrusos

Os sistemas de prevenção de intrusos IPS (*Intrusion Prevention Systems*) são considerados uma extensão da tecnologia de Detecção de Intrusos IDS (*Intrusion Detection Systems*) [Hansche 2004]. IDS são caracterizados por monitorar o tráfego de forma passiva, detectar anomalias e gerar alarmes. Não realizam nenhuma ação para impedir tráfego malicioso.

IPS foram criados no final dos anos 90 para resolver ambigüidades no monitoramento passivo de rede, instalando-se sistemas de detecção em linha com a rede. São considerados uma evolução sobre *firewalls*, pois tomam suas decisões de controle de acesso baseados no conteúdo da aplicação, em vez de controlar por endereçamento de origem, destino e portas, como *firewalls* tradicionais o fazem. Como sistemas de IPS são originalmente uma extensão de IDS, eles continuam relacionados, porém o IPS tem a capacidade de ser reativo e bloquear tráfego malicioso, através da técnica conhecida como análise detalhada de pacotes DPI (*Deep Packet Inspection*) [Screen 2006, Light 2006].

Um problema relacionado com IPS e IDS é a detecção de falsos positivos, ou seja, a identificação de um tráfego como malicioso erroneamente. Portanto, têm de ser otimizados de acordo com as características das aplicações para manter-se uma baixa taxa de falsos positivos.

Alguns IPSs podem prevenir até mesmo quanto a ataques ainda não identificados ou conhecidos, como os causados por *buffer overflow* [Hansche 2004]. O papel de um IPS na rede é geralmente confundido com o de um *firewall* de aplicação. Apesar de haver similaridades entre estas tecnologias, a abordagem de proteção é fundamentalmente diferente.

Enquanto *firewalls* de aplicação funcionam como intermediários (*proxies*) na rede, o IPS praticamente pode trabalhar de forma invisível, ou seja, sem endereço IP. Um IPS normalmente não possui endereço IP para o segmento de rede que está monitorando e não responde diretamente a nenhum tráfego de rede. Em vez disso, monitora de forma silenciosa o tráfego de rede.

Alguns produtos de IPS do mercado possuem funções de *firewall* embutidas em seu conjunto, porém estas funções aparecem como uma conveniência de produto e não como uma característica central. Além disso, a tecnologia de IPS oferece uma visão profunda das operações da rede provendo informações sobre os *hosts* ativos, processos de autenticação não autorizados, conteúdo impróprio e muitas outras informações das camadas de aplicação e rede.

Por outro lado, *firewalls* de aplicação podem possuir algumas funções semelhantes às executadas por IPS, como por exemplo, impor regras de especificação de protocolos assim como definidos nas respectivas RFCs, ou seja, qualquer utilização do protocolo que extrapole o que está definido em sua RFC, levará ao descarte do pacote. Outro tipo de funcionalidade, que pode estar presente, é a utilização de bancos de assinaturas, para suportar análise em tempo real e bloquear tráfego indevido. Ao contrário do IPS, um *firewall* de aplicação possui endereçamento IP no segmento de rede que monitora e pode ser diretamente endereçado através deste.

Apesar das similaridades existentes entre ambos, o *firewall* de aplicação tem foco em suas funcionalidades de *firewall*, apresentando as de IPS como um agregado de

funcionalidade da solução. Assim, não há como comparar de forma concreta estas duas tecnologias.

3.3.1 Tipos de IPS

Os sistemas de detecção e prevenção de intrusos podem ser classificados da seguinte forma: IPS de *Host*; IPS de rede; IPS de conteúdo; *Rate based* IPS.

3.3.1.1 HIPS (IPS de *Host*)

Neste tipo de IPS a aplicação de prevenção é instalada no equipamento (*host*) e compartilha o uso de processadores e memória com outras aplicações. Geralmente são instalados como uma aplicação de segurança em servidores da rede. Este tipo de IPS monitora apenas o tráfego de rede endereçado ao servidor em questão.

3.3.1.2 IPS de rede

Em um IPS de rede (NIPS – *Network* IPS) a capacidade de detecção e prevenção de intrusos está instalada em um equipamento específico para esta função. Consiste em um equipamento instalado na rede para monitorar o tráfego de um ou mais segmentos desta.

O IPS de rede pode ser instalado na frente de um *firewall*, ou seja, entre o *firewall* e a Internet; ou entre este e a rede interna. A topologia dependerá das características de segurança da aplicação que se deseja proteger e de outros fatores, tais como desempenho do IPS, volume de tráfego, topologia da rede e quantidade de sessões simultâneas.

Um IPS de rede é um conjunto de *hardware* e *software* de propósito específico, são estruturados para inspecionar o tráfego de rede, analisar, detectar e baseado na configuração ou regras de segurança, podem descartar o tráfego de rede malicioso.

3.3.1.3 IPS de conteúdo

CBIPS (*Content Based* IPS) inspeciona o conteúdo dos pacotes, buscando determinadas seqüências ou padrões de informações denominados assinaturas. A detecção de assinaturas

ajuda a prevenir contra ataques conhecidos, tais como propagação de vermes e exploração de vulnerabilidades de protocolos.

3.3.1.4 RBIPS (*Rate based* IPS)

São primariamente concebidos para prevenir interrupção de serviços e ataques distribuídos de interrupção de serviços DDoS (*Distributed Denial of Service*) [Hansche 2004]. Através do monitoramento em tempo real do tráfego de rede, este IPS armazena informações estatísticas sobre os padrões de comportamento de tráfego da rede em análise. Assim, pode identificar taxas anormais de tráfego para certos tipos de tráfego, por exemplo, TCP, UDP (*User Datagram Protocol*) [Postel 1980] e pacotes ARP, considerando a quantidade de sessões por segundo, número de pacotes por conexão, quantidade de pacotes por porta específica, e etc. Os ataques são identificados quando limites são extrapolados. Estes limites são dinamicamente ajustados de acordo com a hora do dia, dia da semana e à medida que o tráfego de rede evolui.

Tráfego de rede não usual, porém legítimo pode criar situações de falso positivo, ou seja alarmes falsos. Portanto, a efetividade do RBIPS está relacionada à especificidade das informações estatísticas coletadas e armazenadas. Uma vez que um ataque é detectado, várias técnicas de prevenção podem ser aplicadas, entre elas: limitação da banda disponível (*rate-limiting*), identificação e bloqueio da origem e validação da origem.

A figura 16 ilustra a topologia de um IPS de rede (NIPS) em linha utilizando a técnica de reescrita de pacotes (*packet scrubbing*). Nesta técnica o pacote agressor é identificado, analisado e reescrito com informação não nociva. O pacote agressor também poderia ser descartado como forma de proteção da rede.

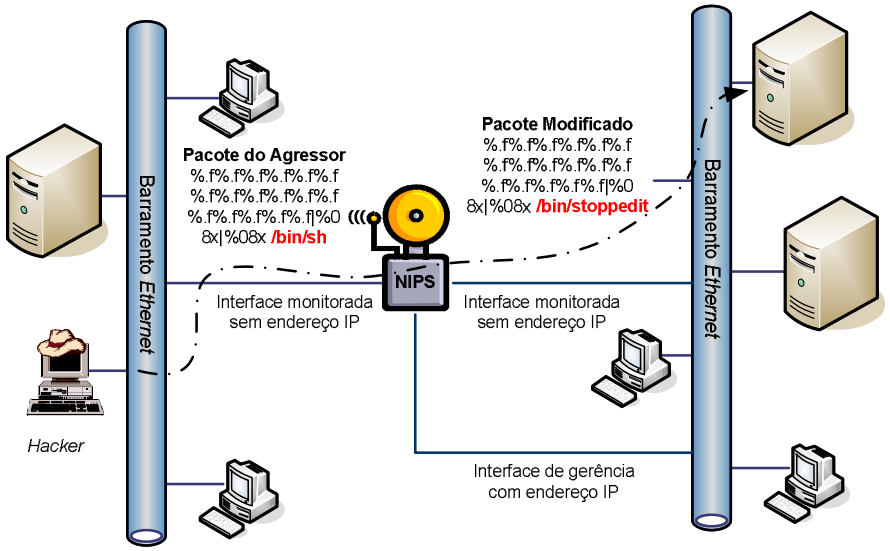

Figura 16 – NIPS em linha, técnica *Packet Scrubbing*

3.3.2 Analisadores de protocolos

No desenvolvimento de IPS um fator chave é o uso de analisadores de protocolos, os quais podem nativamente decodificar protocolos da camada de aplicações, tais como HTTP e FTP [Postel 1985]. Uma vez que os protocolos são decodificados, o IPS pode analisar diferentes partes do pacote, identificar possíveis anomalias ou código para exploração de vulnerabilidades. Por exemplo, a existência de um arquivo binário grande no campo agente de usuário (*User-Agent*) de uma requisição do serviço do protocolo HTTP, não é usual e pode indicar na maioria das vezes uma tentativa de intrusão. O analisador de protocolos pode identificar esta anomalia e rapidamente excluir os pacotes ofensores.

Nem todos os sistemas de detecção e prevenção de intrusos (IPS) possuem a funcionalidade completa de analisador de protocolos. Alguns produtos estão calcados no reconhecimento de padrões (assinaturas), o que pode ser suficiente na maioria dos casos. Entretanto, cria uma fraqueza nas capacidades de detecção, pois várias vulnerabilidades possuem dezenas ou milhares de variações de exploração da mesma e, assim, IPS baseados em assinaturas de ataques podem não detectar uma nova variação do ataque. Produtos

baseados na análise de protocolos podem bloquear variações de ataques, pois monitoram a rede em busca de vulnerabilidades específicas do protocolo.

3.4 *Heartbeat*

Um *heartbeat* é um tipo de mensagem trocada entre máquinas em um intervalo de tempo regular, na ordem de segundos ou milissegundos. Se uma mensagem *heartbeat* não é recebida após um intervalo de tempo, a máquina que deveria ter enviado a mensagem é declarada em estado de falha. Um protocolo de *heartbeat* é geralmente utilizado para negociar e monitorar a disponibilidade de um recurso. Tipicamente, quando um processo de *heartbeat* se inicia em uma máquina, ocorre um processo eletivo com outras máquinas da mesma rede HA, para determinar qual será a responsável primária pelos recursos.

Em redes HA com mais de dois equipamentos, é muito importante considerar a questão da partição, onde as partes de uma rede podem estar funcionando, porém sem comunicação entre elas. Em situações como esta é salutar que os recursos tenham apenas um equipamento como responsável primário por eles, e não um responsável em cada parte da rede.

Se em cada partição da rede existem equipamentos tentando tomar o controle dos recursos, temos uma situação denominada "cérebro partido" (*split-brain*). O rompimento das conexões HA pode resultar neste tipo de problema dependendo do fabricante e modelo de equipamento em questão, pois no rompimento o tráfego de *heartbeat* entre os equipamentos pode ser suspenso. Algumas técnicas para evitar o problema incluem: utilizar mais de uma interface para *heartbeat;* e utilizar as interfaces não dedicadas para *heartbeat* como alternativa

ao envio de tráfego em caso de falha da interface dedicada, esta técnica é denominada caminho secundário (*secondary path*).

O problema oposto ao "cérebro partido" é denominado "acéfalo" (*no-brain*) e ocorre quando nenhum dos *firewalls* ou IPS assumem o papel de mestre na rede na existência de uma falha. Este problema é resultante de falhas múltiplas na rede, como por exemplo, falha de *switches* independentes ao mesmo tempo (queda de energia), onde as interfaces de rede dos *firewalls*/IPS estão conectadas. Assim, ambos os equipamentos detectarão a falha e entrarão em estado inoperante.

Já que uma mensagem de *heartbeat* é utilizada para verificar a saúde de elementos da rede, é extremamente importante que o protocolo de transporte desta mensagem seja o mais confiável possível. Comutar de um sistema em estado *backup* para um ativo, devido a um falso alarme, pode ser desastroso para o recurso e para as aplicações que o utilizam.

Também é importante reagir rapidamente em caso de falhas, assim, temos mais uma razão para que o protocolo de transporte seja confiável. Por isso, é comum encontrar topologias de HA onde as mensagens de *heartbeat* podem trafegar por mais de uma interface de rede, por exemplo, uma interface serial e outra *Ethernet*, geralmente através de conexão direta entre os equipamentos através de cabo de pares entrelaçados (*crossover*).

Entretanto requisitos de disponibilidade para o negócio baseados em sua localização geográfica, por exemplo, necessidade de equipamentos instalados separadamente em prédios distintos, com distâncias acima de 100 metros (tamanho máximo de um cabo de rede *Ethernet*); aliados ao aumento de tráfego das redes e crescentes requisitos de desempenho, tornaram imprescindível a necessidade de conexões HA sobre LAN e até mesmo WAN.

Atualmente, conexões HA podem ser estruturadas em interfaces *fast* e *gigabit Ethernet*, através de *switches* em LANs. Neste caso, o tráfego de controle de HA e transferência de estado estão sujeitos a toda sorte de problemas associados com LAN, tais

como congestionamento, disponibilidade de própria rede, desempenho e *loops* de rede. Assim, são necessários alguns cuidados relevantes a este tipo de implementação que são descritos no capítulo 5 desta dissertação.

3.5 Mecanismos de Alta Disponibilidade para *Firewall* e IPS

Os mecanismos de controle de disponibilidade para *firewall* e IPS possuem requisitos de confiabilidade rigorosos e existem diferentes mecanismos de controle da alta disponibilidade. Alguns destes mecanismos são suportados pelo protocolo VRRP and IP *Multicast*.

Independentemente do tipo de *firewall* ou IPS utilizado na rede, para prover alta disponibilidade são necessários dois mecanismos:

- O primeiro mecanismo é o *heartbeat*, utilizado para verificar a funcionalidade dos elementos componentes da estrutura de HA; e
- O segundo mecanismo é utilizado para enviar as informações de estado do equipamento primário paras os outros definidos como *backup*.

Durante o funcionamento do segundo mecanismo, as informações estáticas, tais como configurações de endereços IP, regras de controle de acesso e informações dinâmicas, tais como as conexões ativas, funcionalidades em operação, tabelas de tradução de endereços DNS (*Domain Name System*) [Mockapetris 1987] e tabelas de mapeamento de endereço NAT são sincronizadas entre os elementos da estrutura de HA.

3.5.1 Tipos de alta disponibilidade

O objetivo mais importante de uma arquitetura HA é eliminar pontos únicos de falha (SPOF) da arquitetura. A funcionalidade de HA disponibilizada na maioria dos *firewalls* e IPS está calcada nos seguintes modelos: Ativo/Passivo; e Ativo/Ativo.

3.5.1.1 Ativo/Passivo

Neste modelo um dos equipamentos atua como principal (mestre) e os demais atuam como *backup*. O mestre envia todas as informações de estado para o(s) *backup*(*s*). Se o mestre falhar o *backup* é promovido a mestre e assume as sessões previamente estabelecidas.

3.5.1.2 Ativo/Ativo

Neste modelo ambos os equipamentos são configurados para o estado ativo, compartilhando as sessões entre eles através de balanceamento de carga. Se um dos equipamentos falha o outro assume as sessões e tráfego deste. Neste caso, um trabalho prévio e constante de avaliação e divisão de carga por parte do administrador de segurança é fundamental, para que não ocorra sobrecarga em situações de contingência.

3.5.2 *Firewall*, IPS e VRRP

A implementação de mecanismos de alta disponibilidade para *firewall* e IPS é mais complexa do que a implementação destes mecanismos para roteadores, porque além das questões relacionadas ao tráfego de *heartbeat* e permutação de endereçamento IP no caso de falhas; é necessária a transferência de informações para manutenção do estado das conexões, tabelas de tradução de endereços (NAT) e configurações. Para *firewalls* do tipo "Inspeção de Estado de Sessão" a transferência e controle do estado das sessões é vital no processo de comutação (*failover*) de *backup* para mestre, visando minimizar ao máximo a perda de sessões já estabelecidas.

Adicionalmente, se o endereçamento IP do *firewall*/IPS está atrelado ao endereço físico (MAC) da interface, a aplicação genuína do VRRP não resolve este e os outros problemas mencionados no parágrafo anterior. Portanto, são necessárias variações de implementação derivadas do VRRP, ou a utilização deste como componente direto de uma das camadas de HA.

A figura 17 ilustra uma topologia de HA para *firewall,* onde o VRRP é utilizado para controle da camada de endereçamento virtual.

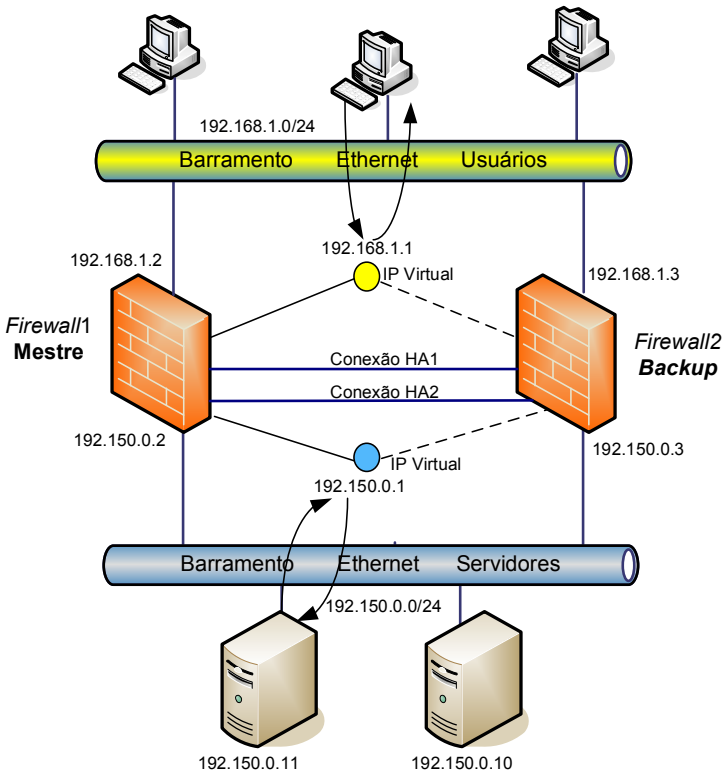

Figura 17 – Topologia HA entre *firewall*, VRRP para controle dos endereços virtuais

Nesta topologia a transferência de *heartbeat* e informações de estado das conexões não é realizada pelo VRRP. As conexões dedicadas HA1 e HA2 são utilizadas para este fim, através de protocolos, tais como TCP e UDP. Note-se que estas conexões dedicadas não utilizam à estrutura LAN disponibilizada pelos *switches*.

4 O PROTOCOLO SCTP

4.1 Introdução

O SCTP (*Stream Control Transmission Protocol*) é um protocolo de transporte orientado a conexão fim-a-fim, confiável e que provê serviços até então não suportados por outros protocolos de transporte clássicos, que atualmente suportam a Internet, tais como o confiável e orientado a conexão TCP (*Transmission Control Protocol*), ou o não confiável e não orientado a conexão UDP (*User Datagram Protocol*). Suas especificações estão contidas na RFC 2960 [Stewart 2000] do IETF (*Internet Engineering Task Force*).

Este capítulo tem o objetivo de apresentar os aspectos conceituais e arquiteturais do SCTP, necessários ao suporte da arquitetura de alta disponibilidade proposta no capítulo 5, bem como as características que o tornam imprescindível, mesmo com os protocolos TCP e UDP já amplamente utilizados na Internet.

Na seção 4.2 é apresentada a arquitetura do SCTP. A seção 4.3 descreve as fatias de DADOS, HEARTBEAT e SACK. A seção 4.4 apresenta as principais fases de uma associação SCTP. Finalmente a seção 4.5 descreve as vantagens do SCTP em relação a outros protocolos.

4.2 Arquitetura do SCTP

O SCTP é um protocolo de transporte, orientado a conexão, com controle de fluxo, entrega confiável, ordenada ou não entre dois pontos. Oferece vantagens sobre o TCP, tais como múltiplos fluxos (*multistreaming*) e múltiplos endereços (*multihoming*), que ampliam as capacidades de disponibilidade.

SCTP introduz o conceito de associação, que caracteriza a existência de uma conexão entre duas entidades de aplicação. Esta conexão suporta múltiplos e independentes fluxos (*streams*) de dados em uma mesma associação. Um fluxo é unidirecional e transporta mensagens ordenadas ou desordenadas. A figura 18 ilustra uma associação com múltiplos fluxos (*streams*).

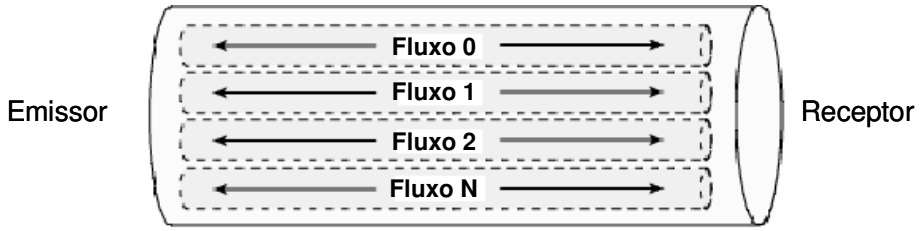

Figura 18 – Associação SCTP com múltiplos fluxos (*streams*)

Como protocolo de transporte, o SCTP apresenta-se como um dos possíveis componentes desta camada, que se situa entre as camadas de aplicação e de rede. Porém, opera independentemente dos demais protocolos componentes da mesma camada. A figura 19 ilustra a equivalência entre as camadas do modelo OSI [ISO 1979] (*Open Systems Interconnection*), a arquitetura Internet e inserção do protocolo SCTP em ambos. O protocolo da camada de rede considerado neste trabalho é o IP (*Internet Protocol*) [DARPA 1981].

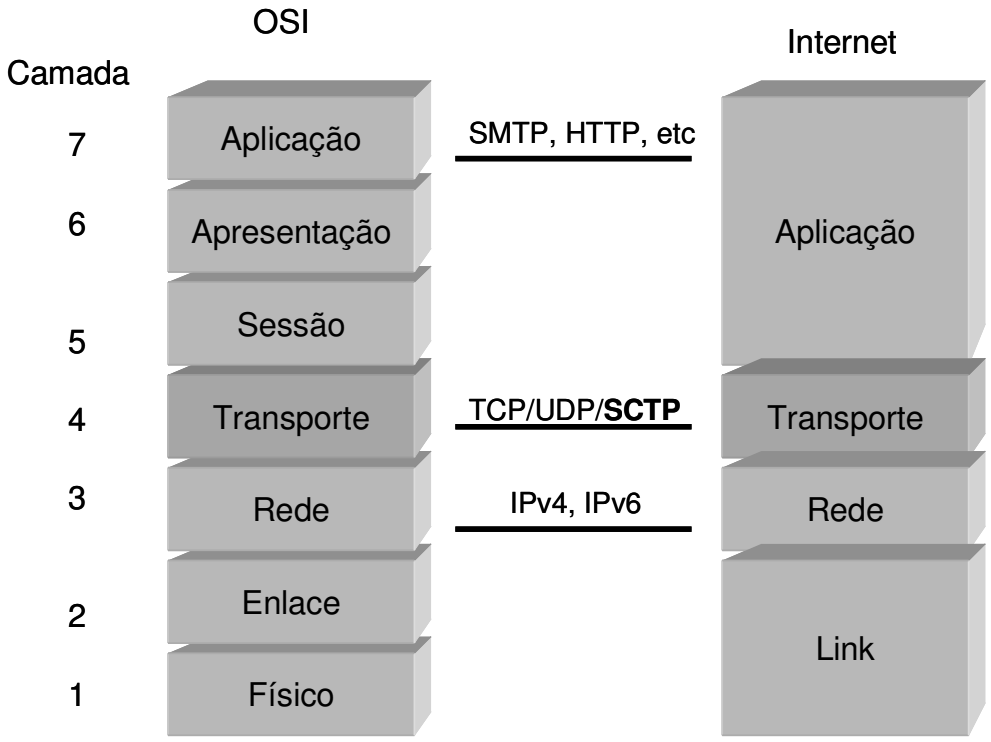

Figura 19 – Inserção do SCTP no modelo OSI e arquitetura Internet

A unidade de dados PDU (*Packet Data Unit*), unidade de informação trocada pelas entidades pares do protocolo SCTP são chamados de pacotes SCTP. Estes são compostos de um cabeçalho comum e fatias (*chunks*) de dados ou controle. A figura 20 ilustra o formato de um pacote SCTP. O tamanho do pacote SCTP é calculado em função do parâmetro MTU (*Maximum Transmission Unit*) [Mogul 1990, Stevens 2000]. O SCTP mantém controles para RTT (*Round Trip Time*) e RTO (*Retransmission Time Out*) através de método que calcula aproximação de números inteiros derivado do método criado por Van Jacobson [Jacobson 1990] para o protocolo TCP, ou através do uso do mesmo modelo.

4.2.1 Cabeçalho comum

O cabeçalho comum consiste de 12 octetos. Para a identificação de uma associação o SCTP usa o mesmo conceito de porta utilizado pelo TCP e UDP. Para a detecção de erros na transmissão e verificação de integridade, cada pacote é protegido por uma soma de controle (*checksum*) de 32 bits, gerada pelo algoritmo CRC32c [Stone 2002], que na RFC 3309

substituiu o algoritmo Adler-32 definido na RFC 2960 e até então utilizado. O algoritmo CRC32c é mais robusto que a soma de controle de 16 bits do TCP ou UDP. Pacotes com somas de controle inválidas são descartados. O cabeçalho comum também contém uma etiqueta de verificação (*verification tag*) de 32 bits. Esta é específica da associação e é trocada entre as entidades no começo da associação.

O mecanismo de controle de congestionamento provido pelo SCTP é idêntico ao provido pelo TCP, onde a janela de controle de congestionamento é reduzida pela metade quando ocorre perda de pacotes, seguindo o algoritmo AIMD [Floyd 2000] (*Additive Increase/Multiplicative Decrease*).

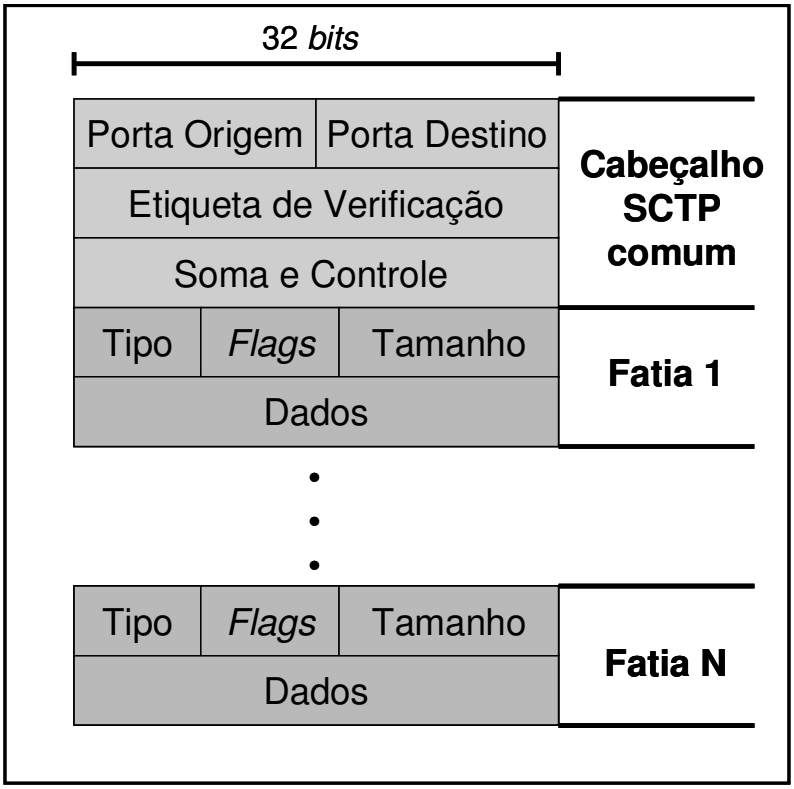

Figura 20 – Pacote SCTP (Cabeçalho e fatias)

Os campos Porta Destino e Porta Origem ocupam um número inteiro sem sinal de 16 *bits* e se referem ao número das portas da camada de transporte de origem e destino. A Porta Origem combinada com o endereço IP fornece a identificação da associação à qual o pacote

pertence. A Porta Destino permite ao protocolo identificar para qual entidade de aplicação no destino o pacote deve ser entregue.

A Etiqueta de Verificação (*Verification Tag*) é um número inteiro sem sinal de 32 *bits* gerado de forma randômica e permite ao receptor validar se um pacote pertence a uma associação corrente. O valor da Etiqueta de Verificação é negociado no início da associação. Pacotes recebidos sem o valor esperado são descartados. O receptor do pacote SCTP utiliza a Etiqueta de Verificação para validar o emissor. Na transmissão, o valor deste campo tem de ser definido como o valor da etiqueta de início recebido do emissor, excetuando-se as seguintes condições:

- Pacotes contendo a fatia INIT têm o valor zero neste campo;
- Pacotes contendo a fatia SHUTDOWN COMPLETE com o T-Bit ativado, têm este número copiado da fatia SHUTDOWN ACK; e
- Pacotes contendo a fatia ABORT podem ter a Etiqueta de Verificação copiada do pacote que causou o envio do ABORT.

4.2.2 Fatias (*Chunks*)

Cada fatia se inicia com um campo de Tipo que é usado para distinguir fatias de dados ou controle, seguido por *flags* específicos e pelo campo de Tamanho, uma vez que podem ter tamanho variável. São identificadas por um número de seqüência de transmissão TSN (*Transmission Sequence Number*). O TSN é independente do fluxo (*stream*) ou da mensagem à qual pertence sendo obrigatório em todas as fatias de dados.

O pacote SCTP pode ser composto de múltiplas fatias de dados e de controle permitindo utilização eficiente da MTU. As exceções são as fatias INIT, INIT ACK, e SHUTDOWN COMPLETE, as quais devido às suas funcionalidades peculiares não podem

compartilhar o pacote SCTP ou estar em uma posição que não seja como a primeira fatia logo após o cabeçalho comum.

No início de uma associação, a fatia INIT é enviada pela entidade iniciadora e o INIT ACK é retornado pela entidade parceira para confirmar o recebimento do INIT. Quando do envio das fatias INIT e INIT ACK ainda não existe conexão, por isso não faz sentido que outras fatias estejam inseridas no mesmo pacote SCTP. A fatia SHUTDOWN COMPLETE marca o fim da conexão e assim, também não faz sentido transmitir mais nenhuma fatia. A tabela 6 apresenta os tipos de fatias. O processo de início e término de uma associação será descrito na seção 4.4.

4.2.2.1 Tipo da Fatia

O campo Tipo é um número inteiro sem sinal de 8 *bits* e de acordo com os dois *bits* de mais alta ordem deste campo, tomar-se-á a decisão do que fazer com as fatias que não são reconhecidas pela entidade SCTP destinatária de acordo com a tabela 5.

Tabela 5 – *Bits* mais significativos (Tipo da Fatia)

Bits	**Atitude do receptor**
00	Pare o processamento deste pacote, descarte-o e não processe nenhuma outra fatia dentro deste pacote.
01	Mesma atitude dos "*Bits* 00" e informe que um parâmetro não reconhecido foi encontrado.
10	Ignore esta fatia e continue o processamento.
11	Mesma atitude dos "*Bits* 10" e informe que uma fatia não reconhecida foi encontrada.

Tabela 6 – Tipos de fatias (*chunks*)

Número	**Binário**	**Tipo da fatia (*chunk*)**
0	00000000	Dados (DATA)
1	00000001	Início de conexão (INIT)
2	00000010	Reconhecimento de Identificação (INIT ACK) - O recebimento do INIT ACK estabelece a associação
3	00000011	Reconhecimento seletivo (SACK) - Reconhece o recebimento de fatias de dados (DATA).
4	00000100	Requisição de *Heartbeat* (HEARTBEAT)
5	00000101	Reconhecimento de *Heartbeat* (HEARTBEAT ACK)
6	00000110	Aviso de fim de conexão abrupta (ABORT)
7	00000111	Fim de conexão (SHUTDOWN)
8	00001000	Reconhecimento de fim de conexão (SHUTDOWN ACK)
9	00001001	Erro de operação (ERROR)
10	00001010	Situação *Cookie* (*COOKIE* ECHO)
11	00001011	Reconhecimento do *Cookie* (*COOKIE* ACK)
12	00001100	Reservado para notificação explicita de congestionamento (ECNE)
13	00001101	Reservado para redução da janela de congestionamento (CWR)
14	00001110	Fim de conexão completa (SHUTDOWN COMPLETE)
15 a 62		Reservado pelo IETF (*Internet Engineering Task Force)*
63	00111111	Definido pelo IETF para extensões de fatias
64 a 126		Reservado pelo IETF
127	01111111	Definido pelo IETF para extensões de fatias
128 a 190		Reservado pelo IETF
191	10111111	Definido pelo IETF para extensões de fatias
192 a 254		Reservado pelo IETF
255	11111111	Definido pelo IETF para extensões de fatias

4.2.2.2 *Flags* da Fatia

O campo *Flags* é um campo tipo inteiro sem sinal de 8 *bits* e sua representação dependerá do tipo de fatia a que pertence. Se a fatia não necessitar destas opções (*flags*) estes oito *bits* serão preenchidos com o valor zero e serão ignorados pelo destinatário.

4.2.2.3 Tamanho da Fatia

Tamanho da fatia inclui o cabeçalho e todas as opções (*flags*). É um campo tipo inteiro sem sinal de 16 *bits* e representa o tamanho total em *bytes*. O tamanho total da fatia é obrigatoriamente um múltiplo de 4 *bytes*. Se o tamanho não for um número de 4 *bytes*, a fatia receberá os *bytes* necessários para este requisito preenchidos com o valor zero. Estes *bytes* adicionais serão ignorados pelo destinatário.

4.2.2.4 Dados da Fatia

Neste campo estão as informações que a fatia carrega. Este campo varia de acordo com o tipo, que determina a funcionalidade que esta fatia exerce (controle, transmissão de dados, verificação de linha, etc.). Também, serão encontrados neste campo parâmetros opcionais e parâmetros de tamanho variável que fazem parte do dado sendo enviado pela fatia. Os parâmetros possuem um cabeçalho específico conforme pode ser visualizado na figura 21.

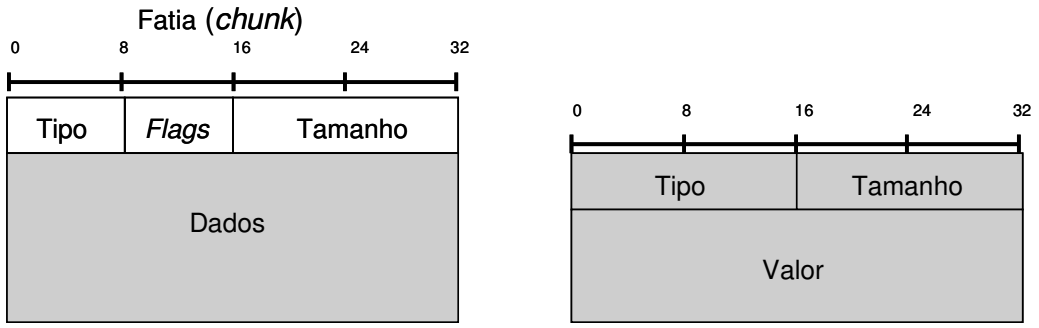

Figura 21 – Parâmetros variáveis da fatia

Descrição dos campos dos parâmetros:

- Tipo – é um inteiro sem sinal, possui 16 bits, identifica o parâmetro variando de 0 a 65534. O valor 65535 é reservado pelo IETF para definição de extensões;

- Tamanho – é um inteiro sem sinal de 16 *bits* e contém o tamanho do parâmetro em *bytes*, incluindo os campos: Tamanho, Tipo e Valor do parâmetro, excluindo-se *bytes* de alinhamento. Assim, se o valor do parâmetro for nulo, este campo possuirá um tamanho mínimo de 4 *bytes*; e

- Valor – neste campo estão as informações que serão transmitidas pela fatia. O total do tamanho dos parâmetros precisa ser múltiplo de 4 *bytes*. Se o tamanho total não se encaixar neste quesito a entidade SCTP de origem deve inserir no final da fatia os *bytes* necessários e preenchê-los com zero. Estes *bytes* de preenchimento não são contabilizados pelo campo tamanho do parâmetro e são ignorados pelo ponto receptor. Os dois *bits* mais significativos do Tipo do Parâmetro, assim como nos tipos das fatias, orientam a atitude do ponto receptor quando o parâmetro não é reconhecido como válido de acordo com a tabela 7.

Tabela 7 – *Bits* mais significativos (Tipo do Parâmetro)

Bits	**Atitude do receptor**
00	Pare o processamento deste pacote, descarte-o e não processe nenhuma outra fatia dentro deste pacote.
01	Mesma atitude dos "*Bits* 00" e informe que um parâmetro não reconhecido foi encontrado.
10	Ignore este parâmetro e continue o processamento.
11	Mesma atitude dos "*Bits* 10" e informe que um parâmetro não reconhecido foi encontrado.

4.3 Descrição dos Tipos de Fatias

Nesta seção são apresentados os formatos das fatias de Dados (DATA), Reconhecimento Seletivo (SACK), HEARTBEAT e HEARTBEAT ACK. Suas respectivas descrições e funcionalidades também serão descritas. Estas fatias serão utilizadas na arquitetura proposta para transportar as informações de controle de estado e verificação de saúde dos *firewalls* e IPS, respectivamente.

4.3.1 Fatia de dados (DATA)

A figura 22 apresenta a estrutura da fatia de Dados, que é utilizada para transportar as informações inseridas pela aplicação.

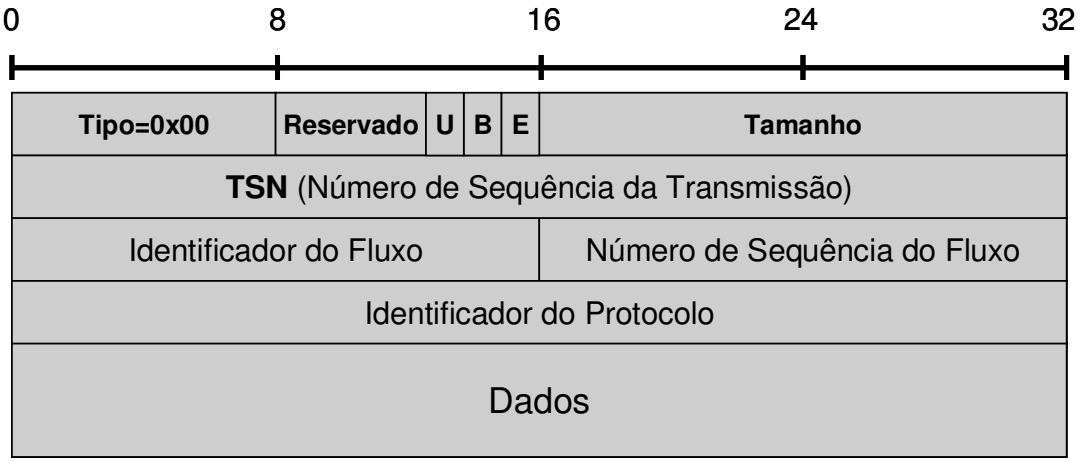

Figura 22 – Fatia de Dados

Descrição dos campos da fatia de dados:

- Tipo – na fatia de dados assume valor 0 (zero);
- Reservado – este campo deve ser sempre preenchido com zeros e ignorado pela entidade SCTP receptora;

- *Flags* (U,B,E) – este campo possui 3 *bits* que se referem aos campos U, B e E respectivamente. O campo U (*Unordered*) indica se os dados sendo enviados pela fatia devem ou não ser ordenados na chegada. Este *bit* é atribuído pela aplicação. O campo B (*Beginning*) indica se estes dados são o inicio de uma mensagem fragmentada, e o campo E (*Ending*) indica se estes dados são o fim de uma mensagem fragmentada.

Uma mensagem não fragmentada deve ter os *bits* B e E preenchidos com 1 (um). Ativar ambos com 0 (zero) indica um fragmento intermediário de uma mensagem com múltiplos fragmentos conforme indicado na tabela 8. Quando uma mensagem é fragmentada em múltiplas fatias, o campo TSN é utilizado pelo receptor para remontar a mensagem. Isto significa que para cada parte de uma mensagem fragmentada, o TSN tem de ser estritamente seqüencial.

Tabela 8 – *Bits* para controle de mensagens fragmentadas

B	**E**	**Descrição**
1	0	Primeira parte de uma mensagem fragmentada
0	0	Parte intermediária de uma mensagem fragmentada
0	1	Última parte de uma mensagem fragmentada
1	1	Mensagem não fragmentada

- Tamanho – é um campo tipo inteiro sem sinal de 16 *bits*. Contém o tamanho total da fatia, incluso os dados de cabeçalho e excluindo *bytes* de alinhamento. Isto significa que uma fatia de dados tem tamanho mínimo de 16 *bytes*;

- TSN – representa o número de seqüência da fatia de dados. Os números válidos de TSN estão no intervalo $0 \leq TSN \leq (2^{32} - 1)$. Quando o TSN chega ao seu número limite ele volta a zero;
- Identificador do Fluxo – identifica a qual fluxo este dado pertence;
- Número de Seqüência do Fluxo – contém o número de identificação da mensagem. Os números válidos para este campo variam de 0 a 65535. Este número de identificação é usado quando uma mensagem foi fragmentada. Todos os fragmentos de uma mensagem contêm o mesmo Número de Seqüência do Fluxo. Assim, quando os fragmentos chegam ao destino são ordenados em uma mesma mensagem usando o Número de Seqüência do Fluxo e o TSN;
- Identificador do Protocolo – a aplicação preenche este campo com o número de identificação do protocolo que esta sendo transmitido pelo campo de dados. Esta identificação não é usada pelo SCTP, mas é transmitida para a entidade de aplicação pela entidade SCTP destino e é enviado mesmo nos fragmentos. O número zero neste campo indica que ele não esta sendo usado; e
- Dados – contém as informações em processo de transmissão pelo pacote de dados.

4.3.2 Fatia de Reconhecimento Seletivo (SACK – *Selective Acknowledge*)

Esta fatia é enviada para uma entidade SCTP para informar reconhecimento de fatias de dados (DATA) recebidos corretamente e informar possíveis buracos (intervalos) nas subseqüências de fatias de dados recebidas, representadas por seus TSNs. A fatia SACK tem de conter os parâmetros *Cumulative TSN Ack* e *Advertised Receiver Window Credit* (a_rwnd). Por definição o valor do parâmetro *Cumulative TSN Ack* é o último TSN recebido, antes da ocorrência de quebra na seqüência de TSNs recebidos. Este parâmetro confirma o recebimento de todos os TSNs menores ou iguais ao seu valor.

Uma possível falha de envio é um bloco de uma ou mais fatias de dados que deveriam ter sido recebidos. Estas falhas são detectadas porque o TSN é incremental. Se a fatia de dados é recebida com uma falha na seqüência do TSN, e após um determinado tempo as fatias desta falha não chegaram, a entidade SCTP receptora assume que houve uma falha de envio.

A fatia SACK enviará o TSN da última fatia de dados de uma série de fatias recebidas com sucesso (*Cumulative TSN*). Este TSN reconhece automaticamente todas as fatias enviadas antes dela como tendo sido recebidas corretamente. Esta fatia também informa o tamanho da janela de recepção da entidade SCTP receptora no momento que a fatia SACK foi enviada. A figura 23 ilustra esta fatia.

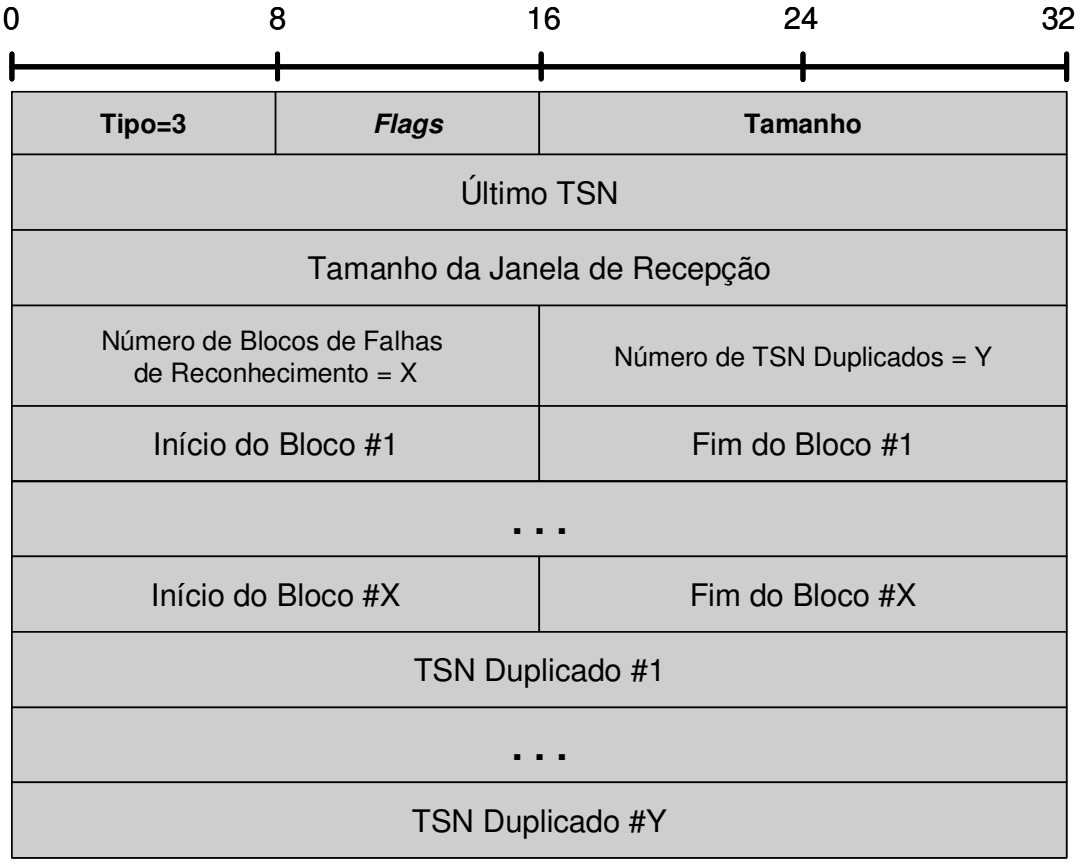

Figura 23 – Fatia de Reconhecimento Seletivo (SACK)

Os seguintes campos compõem a fatia SACK:

- *Flags* – inteiro de 8 *bits* sem sinal preenchido com zeros e ignorado pela entidade SCTP receptora;
- Último TSN (*Cumulative TSN*) – número inteiro sem sinal de 32 *bits*, contém o TSN do último pacote de uma série recebida sem falhas;
- Tamanho da Janela de Recepção – número inteiro sem sinal de 32 *bits*, contém o tamanho em *bytes* do espaço de memória reservada para recepção de fatias no momento que a fatia SACK foi enviada;
- Número de Blocos de Falhas de Reconhecimento – número inteiro sem sinal de 16 *bits*, indica o número de blocos de falhas;
- Numero de TSN Duplicados – número inteiro sem sinal de 16 *bits*, indica o número de TSNs duplicados recebidos.
- Início de Bloco #1 (Início do bloco de falhas número 1) – número inteiro sem sinal de 16 *bits*, marca o início de um bloco. Um bloco é um intervalo de fatias. Este campo contém o *offset* do início deste bloco. Para calcular este número, usa-se o TSN do último TSN (*Cumulative TSN*) e adiciona-se este *offset*. O início do bloco de falhas é o primeiro TSN do bloco de fatias de dados tidos como perdidos ou não enviados. O início do bloco de repetições é o primeiro TSN do bloco de fatias de dados que foram recebidos mais de uma vez.
- Fim do Bloco #1 (fim do bloco de falhas número 1) – número inteiro sem sinal de 16 *bits*. Contém o *offset* do final do bloco de falhas ou repetições. A obtenção de seu valor é semelhante ao Início do bloco de falhas.
- TSN Duplicados – número inteiro sem sinal de 32 *bits*. Indica quantas vezes um TSN foi recebido desde a última fatia SACK enviada. Cada vez que um TSN

duplicado é recebido ele é adicionado à lista de TSN duplicados. Quando o SACK é enviado, esta lista é descontinuada. Quando um novo TSN duplicado chega, ela é iniciada novamente.

4.3.3 Fatia Requisição de *Heartbeat* (HEARTBEAT)

Uma entidade SCTP deve enviar esta fatia à entidade parceira para verificar se esta pode ser "alcançada" (*reachability*) através de um endereço IP específico definido na associação corrente. O campo Parâmetro contém as informações de *Heartbeat*, que se caracteriza por uma estrutura de dados "opaca" compreensível somente pelo emissor. A figura 24 ilustra a estrutura da fatia *Heartbeat Request*.

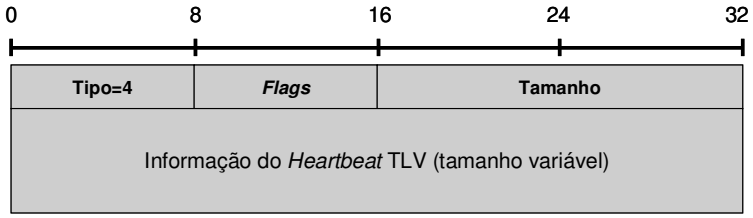

Figura 24 – Fatia *Heartbeat Request*

Segue a descrição dos campos da fatia requisição de *Heartbeat*:

- *Flags* – número inteiro sem sinal de 8 *bits*. São colocados zeros no envio e é ignorado pelo receptor;
- Tamanho – número inteiro sem sinal de 16 *bits*. Tamanho da fatia em *bytes*, incluindo o cabeçalho e as informações do *Heartbeat*; e
- Informação do Heartbeat (*Heartbeat Information*) – campo de tamanho variável e obrigatório, que segue a formato ilustrado na figura 25.

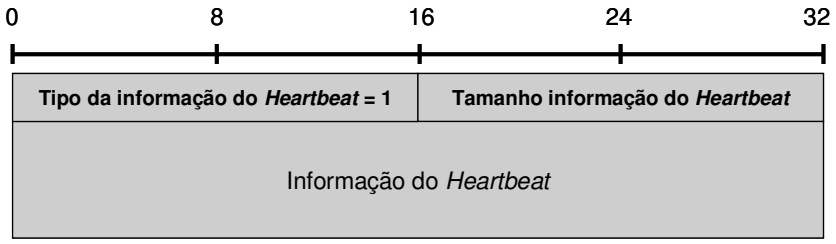

Figura 25 – Parâmetro *Heartbeat*

4.3.4 Fatia Reconhecimento de *Heartbeat* (HEARTBEAT ACK)

A entidade SCTP que recebe a fatia de requisição de HEARTBEAT deve responder com a fatia HEARTBEAT ACK. Esta fatia é sempre enviada para o endereço IP destino contido no pacote SCTP que continha a fatia requisição de *Heartbeat*. A figura 26 ilustra seu formato.

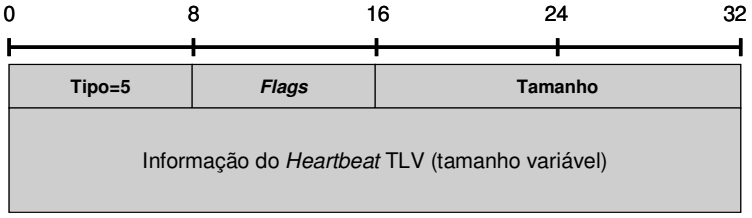

Figura 26 – Fatia *Heartbeat ACK*

Segue a descrição dos campos da fatia *Heartbeat ACK*:

- *Flags* – número inteiro sem sinal de 8 *bits*, preenchidos com zero e ignorados na recepção;
- Tamanho – número inteiro sem sinal de 16 *bits*, contém o tamanho da fatia em bytes, incluindo cabeçalho e parâmetros; e
- Informação do *Heartbeat* (*Heartbeat Information*) – campo de tamanho variável. É um campo obrigatório que contém informação copiada do parâmetro da fatia requisição de *Heartbeat*. A informação de retorno comprovará a autenticidade do receptor, evitando que terceiros assumam a associação indevidamente.

4.4 Principais Fases de Uma Associação SCTP

Esta seção descreverá as principais fases de uma associação SCTP.

4.4.1 Inicio e concretização de uma associação

No início de uma associação o requisitante gera uma fatia INIT e a envia para a entidade receptora ou respondedora. O emissor inicia o temporizador do INIT e cria o bloco de controle da transmissão TCB (*Transmission Control Block*). Se o nó receptor quiser aceitar a associação, ele gera um INIT ACK que inclui um *cookie*. O nó emissor recebe o INIT ACK e pára o temporizador do INIT. Ele gera, então, um COOKIE ECHO e o envia ao receptor, iniciando o temporizador do *cookie*. Dados também podem ser inseridos neste pacote. O receptor verifica a validade do *cookie* e se este for válido, ele envia um COOKIE ACK ao emissor. O emissor recebe COOKIE ACK e começa a próxima fase de transmissão de dados. A figura 27 ilustra este processo.

Durante o ciclo de uma associação SCTP uma série de eventos podem causar mudanças de estados da associação como resposta a eles, dentre os quais podemos destacar:

- Chamadas de primitivas da entidade de aplicação: a entidade de aplicação pode causar uma mudança de estado fazendo uso das chamadas disponibilizadas pelo SCTP; e
- Recepção de primitivas como INIT, COOKIE ECHO, ABORT, etc. e fatias de controle.

O *cookie* é um HMAC (*Keyed-Hashing for Message Authentication Code*) [Krawczyk 1997] geralmente um SHA-1 (*Secure Hash Algorithm*) [Eastlake 2001] gerado pela entidade que envia a fatia INIT ACK. A troca desta informação (*cookie*) tem o objetivo de confirmar a autenticidade da entidade SCTP parceira. Por isso são usados os endereços de origem dos

pacotes IP que transportam as fatias COOKIE e COOKIE ECHO. Este mecanismo provê proteção contra ataques de SYN *flooding* [Ferguson 2000], através de um *four way handshake* sendo que as duas últimas etapas do processo podem transportar dados nos pacotes, para acelerar o processo.

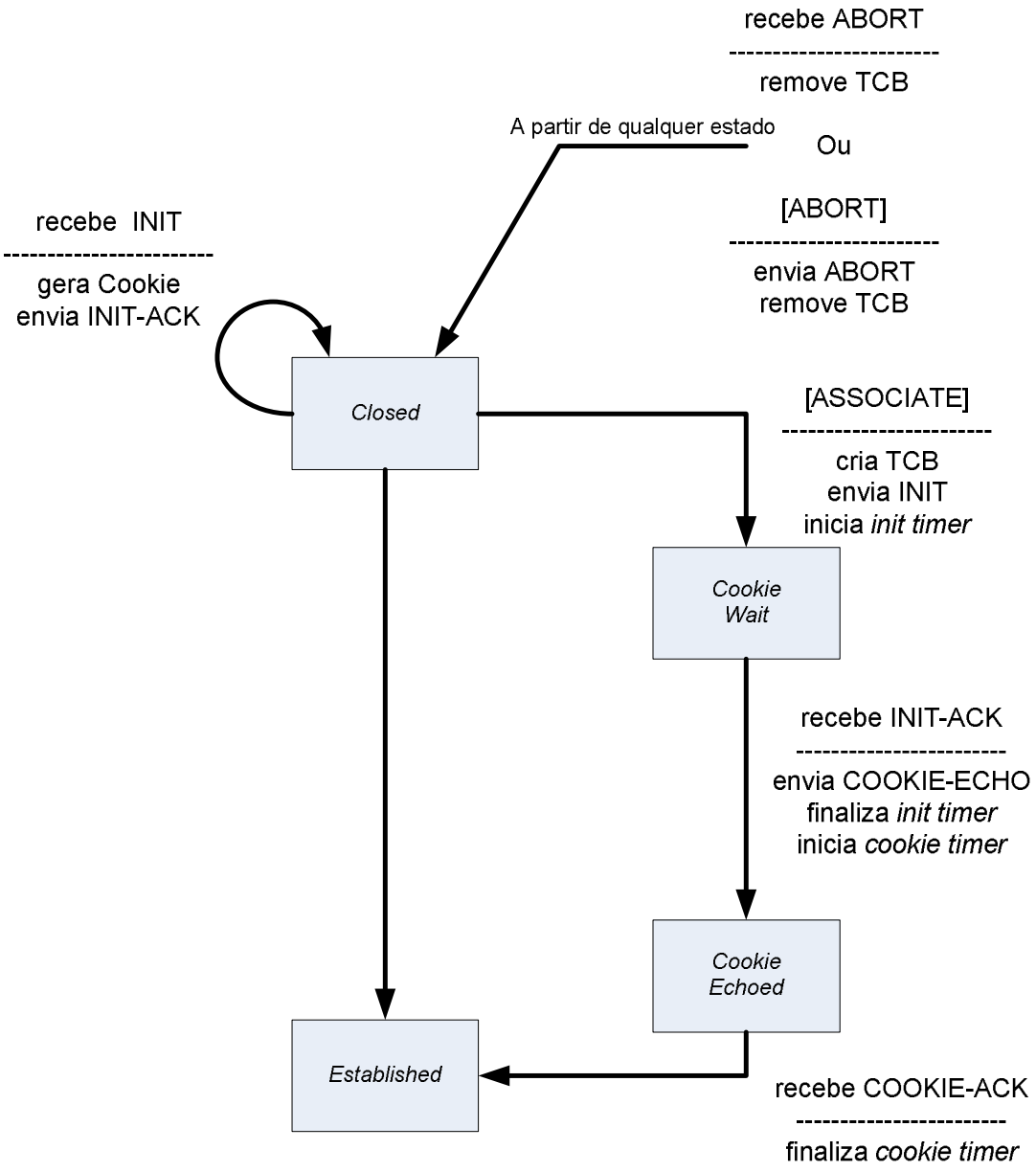

Figura 27 – Máquina de estado do início e concretização de uma Associação

4.4.2 Transporte de dados em uma associação

Durante todo o processo de transmissão, fatias HEARBEAT e HEARTBEAT ACK são trocadas entre as entidades em intervalos de tempo regular, o padrão é a cada 30 segundos. Estes testam a conectividade entre os pontos terminais preservando a validade da transmissão de dados.

O emissor e receptor trocam fatias de dados e após o recebimento de cada fatia de dados, os pontos terminais retornam SACK para confirmar o recebimento. Dados são transmitidos até que um dos pontos terminais decida por encerrar a associação, através do envio de uma fatia SHUTDOWN. A figura 28 ilustra este processo.

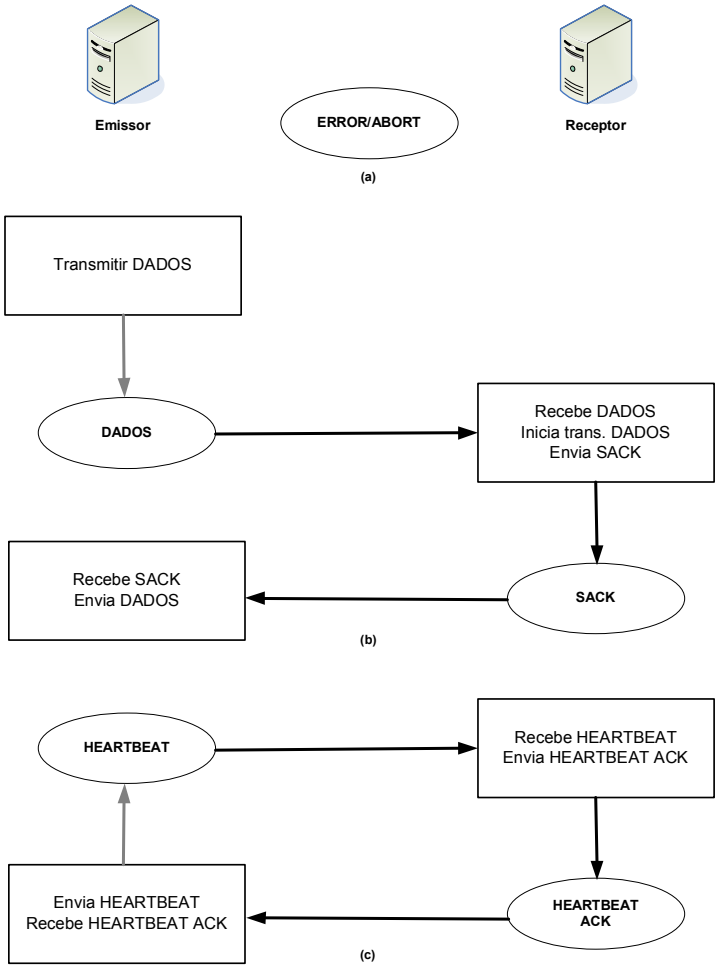

Figura 28 – Diagrama de envio de Dados e *Heartbeat* em uma associação SCTP

4.4.3 Término de uma associação

Ambas as entidades participantes de uma associação podem decidir terminá-la por diversas razões e o podem fazer praticamente em qualquer momento. Existe a possibilidade de um encerramento via fatia SHUTDOWN, assegurando que nenhum dado é perdido, ou este pode ser feito de forma abrupta, via fatia ABORT. A figura 29 ilustra o diagrama de estados de encerramento de uma associação.

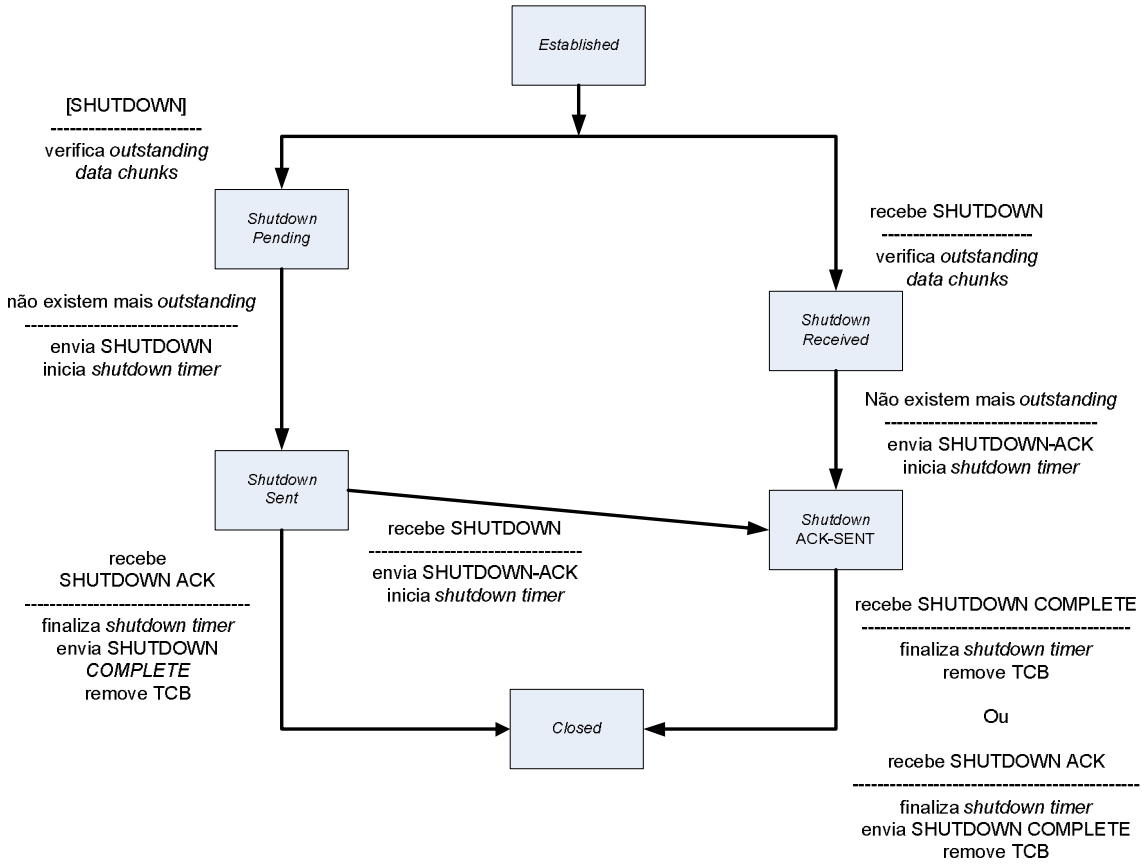

Figura 29 – Máquina de estado do término de uma associação SCTP

<u>Encerramento SHUTDOWN</u> (*Graceful Termination*) – após o recebimento da primitiva SHUTDOWN do processo do usuário da camada superior, uma instância SCTP deve parar de aceitar dados deste processo e enviar um SHUTDOWN. Este processo é garantido por um temporizador. O par receberá a fatia SHUTDOWN e enviará o SHUTDOWN ACK. Quando o par que iniciou o procedimento de

encerramento recebe o SHUTDOWN ACK, ele interromperá o temporizador, enviará um SHUTDOWN COMPLETE removendo todos os dados relacionados com aquela associação, entrando no estado CLOSED. O par que recebe o SHUTDOWN COMPLETE também poderá remover todos os registros daquela associação entrando também no estado CLOSED. Caso este SHUTDOWN COMPLETE seja perdido, o par continuará enviando SHUTDOWN ACKs até que um contador de erros seja excedido, indicando que o outro par está inacessível.

Abortando a Associação – um ponto terminal pode também decidir abortar uma associação. Ele tem de preencher o rótulo de verificação no pacote de saída e não deve associar nenhuma fatia de dados neste pacote. O receptor não responde, mas valida esta fatia de controle e remove a associação se o ABORT contém a etiqueta de verificação correta, informando o encerramento aos processos das camadas superiores.

Durante os estados de SHUTDOWN *Pending* e *Recebe* SHUTDOWN, ocorre a verificação para identificar se existe envio de fatias de dados pendente (*outstanding data chunks*).

4.5 Principais Vantagens do SCTP

Em 1998 um grupo de trabalho do IETF (SIGTRAN) foi constituído para criar um mecanismo confiável para transporte de sinalização de controle de chamadas através da Internet. O objetivo primário do grupo era criar um complemento para o sistema de sinalização de telefonia SS7 (*Signaling System Seven*) [Ong 1999] utilizando o protocolo IP. Durante os trabalhos, dois problemas críticos relacionados ao protocolo TCP foram considerados: *Head-of-line blocking*; e *Multihoming*.

4.5.1 *Head-of-line blocking*

Este problema ocorre durante o envio de mensagens independentes utilizando uma mesma conexão TCP com entrega ordenada. Quando uma mensagem é perdida, as mensagens transmitidas após esta mensagem têm de aguardar no *buffer* da camada de transporte do receptor até que a mensagem perdida seja retransmitida e chegue ao receptor. Este atraso pode ser muito significativo para aplicações sensíveis a atraso, tais como aplicações de controle de sinalização telefônica.

O SCTP suporta múltiplos e independentes fluxos (*streams*) de mensagens (*multistreaming*) em uma associação. Cada mensagem enviada em uma associação SCTP é atribuída a um fluxo particular. Os dados componentes de um fluxo são enviados ordenadamente, respeitando outros dados enviados no mesmo fluxo. Dados em fluxos diferentes não possuem nenhum requisito de ordem entre si, resultando em fluxos ordenados e paralelos. Esta característica provê uma instância de entrega parcialmente ordenada. O serviço de múltiplos fluxos do SCTP é que possibilita evitar o problema do *Head-of-line blocking*. A figura 30 ilustra este problema.

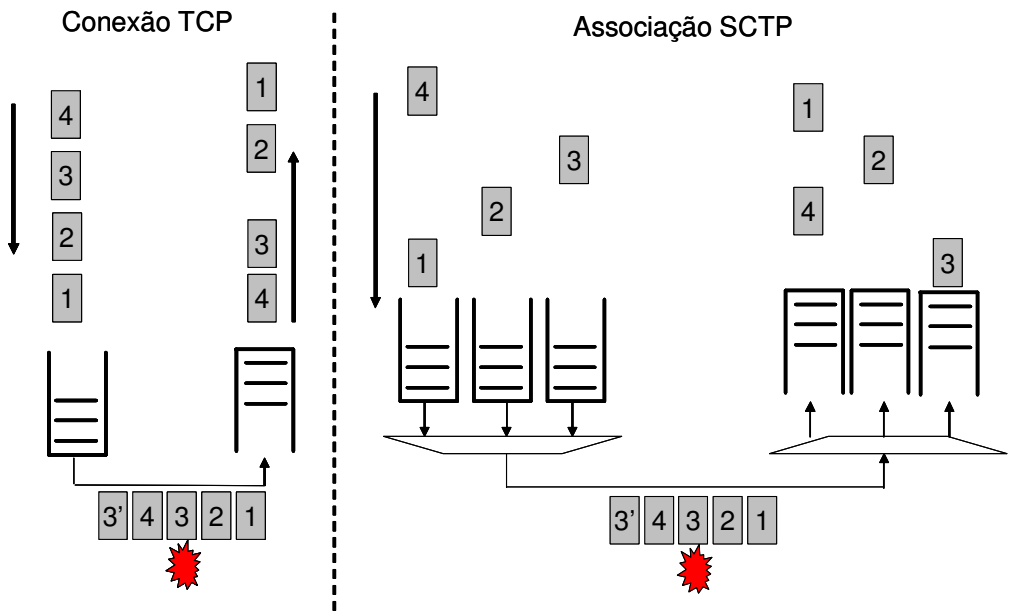

Figura 30 – Problema *Head-of-line blocking*

O SCTP assegura a entrega ordenada no mesmo fluxo. Se uma mensagem é perdida ou corrompida na rede e necessita ser retransmitida, somente o fluxo correspondente está sujeito ao problema do *Head-of-line blocking*. Assim a mensagem 4 pode ser enviada antes da mensagem 3 ser retransmitida, pois estão em fluxos distintos. A ordenação de mensagens pode ser completamente desativada através da ativação do *flag* (opção) *unordered*.

4.5.2 *Multihoming*

Onde um *host* com múltiplos enlaces (interfaces de rede e endereços IP distintos) à Internet, para propósitos de redundância não pode esperar pela convergência de rotas em caso de problemas na rede, por exemplo, falha de um roteador do caminho e/ou falha da interface de rede de um servidor. Uma vez que esta convergência pode levar minutos, o envio de mensagens críticas para o parceiro da associação pode ser comprometido. O tempo de convergência de rotas em uma rede com problemas pode ser fatal para aplicações sensíveis a atraso.

A solução disponibilizada pelo SCTP para este problema estabelece que os dois pontos de uma associação podem especificar múltiplos endereços de conexão durante a fase

inicial da associação, que permitirão o envio automático de dados através de um endereço alternativo quando falhas ocorrerem, e o mais importante: este envio ocorrerá de forma transparente para a aplicação, que não terá conhecimento da ocorrência de uma falha de nível inferior.

Esta característica de tolerância a falhas não está presente no TCP, onde cada conexão está associada a um único endereço e em caso de falhas a aplicação deve contabilizar um tempo de espera e terminar anormalmente (*ABORT*). A figura 31 ilustra a diferença entre a conexão TCP e a possibilidade de múltiplos endereços no SCTP.

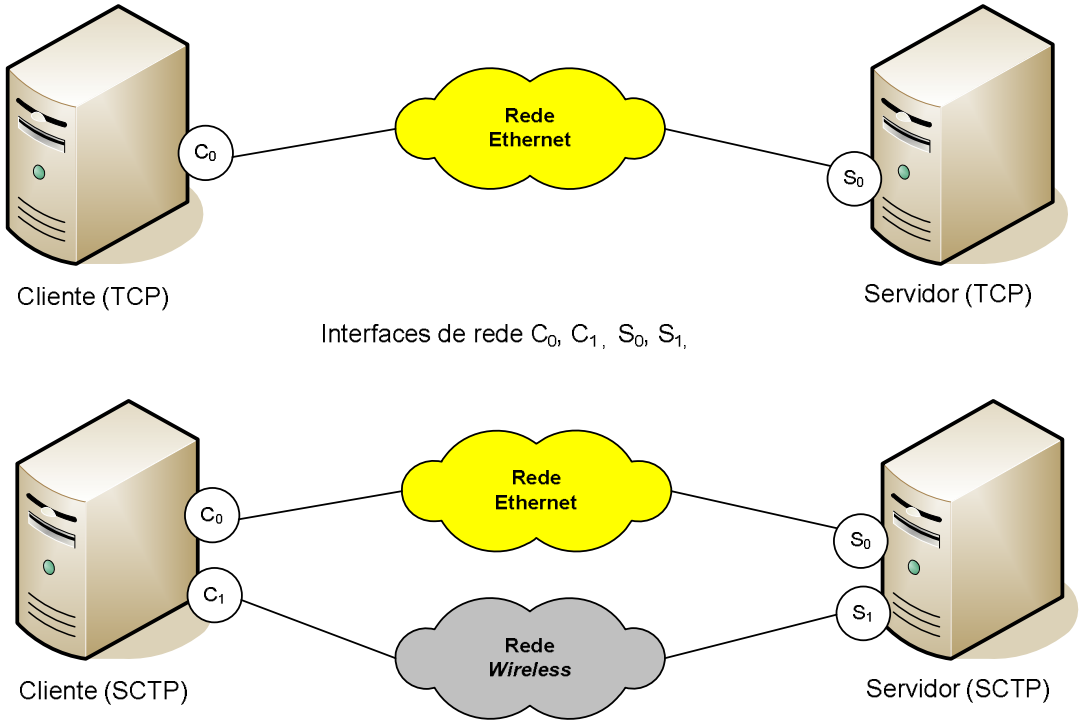

Figura 31 – Conexão TCP versus associação SCTP

Para identificar falhas de enlace, o SCTP utiliza dois métodos: fatias *heartbeat* e condição limite de retransmissão de dados. No início de uma associação contendo múltiplos

endereços IP, um deles é eleito como primário e o mecanismo de monitoramento de *heartbeat* funcionará da seguinte forma:

- A fatia HEARTBEAT é enviada para qualquer endereço de destino que não tenha contabilizado resposta por um período (*timeout*) maior que o do contador de tempo do *heartbeat*;
- A contabilização de resposta utiliza fatias que atualizam o RTT, usualmente fatias DATA e HEARTBEAT. Assim, um endereço de destino é considerado inativo se nenhuma destas fatias foi enviada para ele;
- O contador de tempo do *heartbeat* é iniciado novamente toda vez que uma fatia de DATA ou HEARTBEAT é enviada;
- O receptor responde com uma fatia HEARTBEAT-ACK;
- Toda vez que uma fatia HEARTBEAT é enviada, a variável *Destination Error* para o destino específico é incrementada;
- Toda vez que a fatia HEARTBEAT-ACK é recebida, o contador de erros *Destination Error* é zerado;
- Toda vez que uma fatia de dados (DATA), que foi enviada para o destinatário, é confirmada (SACK) o contador de erros é zerado;
- Toda vez que ocorre um intervalo de tempo sem resposta (*timeout*) associado à fatia de dados (DATA T3-rtx) no destino, o contador de erros é incrementado;
- Toda vez que o contador de erros exceder um limite preestabelecido, usualmente 5 (cinco), o destino é declarado como não acessível;
- Se o endereço de destino primário é marcado como não acessível e se existir endereço alternativo, este será escolhido e utilizado;
- Fatias HEARTBEAT continuarão a ser enviadas para os endereços não acessíveis. Se ocorrer resposta o contador de erros é zerado e o destino é marcado como acessível;

- Se este destino (endereço não acessível) é o endereço IP eleito como primário no início da associação e não ocorreu nenhuma intervenção do usuário, a comunicação é restaurada para este endereço.

Considerando os problemas *head-of-line blocking* e de múltiplos endereços, os esforços do grupo SIGTRAN resultaram em um novo protocolo de transporte de propósito geral, que outras aplicações também podem utilizar. Assim nasceu o SCTP. Outras melhorias vinculadas ao SCTP incluem:

Serviço de orientação de mensagens – em uma conexão TCP se necessário, a aplicação tem de prover a divisão de mensagens em mensagens menores (*framing*). No SCTP, as bordas das mensagens são preservadas, por exemplo: se uma aplicação envia uma mensagem de 100 *bytes*, o destinatário receberá os 100 *bytes* em uma única leitura, nem mais e nem menos. O protocolo UDP também provê o serviço de orientação de mensagens, porém sem a confiabilidade que o SCTP oferece.

Serviço de envio de mensagens não ordenado – em uma conexão TCP todas as mensagens são enviadas para o destino exatamente na mesma ordem que a aplicação ou emissor geraram. Além do serviço de mensagens ordenadas, o SCTP oferece a possibilidade de envio de mensagens de forma não ordenada. O Protocolo UDP também suporta esta característica, porém sem a confiabilidade do SCTP. O serviço de envio de mensagens de forma não ordenada é útil para aplicações que já provêem este serviço e não necessitam do ordenamento provido pelo protocolo TCP e, conseqüentemente, não sofrerão as conseqüências de desempenho para o ordenamento desnecessário (*overhead*). Aplicações que suportam serviços de armazenamento de dados através da LAN, tais como suporte a iSCSI [Satran 2004] e RDMA [Romanow 2005], constituem exemplos de aplicações que já controlam o ordenamento independentemente do protocolo de transporte.

Capacidade de extensão – um pacote TCP está limitado a 40 *bytes* (*for options*). Os pacotes SCTP podem ser expandidos através do uso do campo TLV (*Tag-Length-Value*). O SCTP possui, incorporado em sua estrutura de campos TLV, procedimentos de gestão da compatibilidade mantendo-o funcional mesmo quando um dos lados suporta um conjunto de funções mais avançadas que o outro.

Time-to-Live – o SCTP possui uma opção de se especificar o TTL, ou seja, o tempo de vida da mensagem na rede. Aplicações podem definir quanto tempo uma mensagem será útil na rede. Se o tempo expirar antes da mensagem chegar ao receptor, o emissor pode parar de tentar enviar, ou mesmo ignorar a mensagem. Esta característica é chamada de confiabilidade parcial (*partial reliability*), PR-SCTP [Stewart 2004]. Aplicações de jogos *on-line* na Internet e comunicações móveis, onde o estado da localização corrente ou ambiente tem duração efêmera e é substituído por um estado revisado. Nesta situação a aplicação pode descartar as informações obsoletas, economizar banda, evitar congestionamento e perda de pacotes.

SYN cookies – O SCTP utiliza um mecanismo para estabelecer uma associação baseado na troca de mensagens (*four-way handshake*) com *cookies* assinados. Nenhum estado de conexão é mantido pelo destinatário, nem recursos são reservados, até que o emissor prove que realmente possui o endereço IP iniciador da associação. *Syn cookies* previnem que o uso de técnicas de mascarar endereços IP (*spoofing*) para realizar ataques de interrupção de serviço (DDoS), especificamente *SYN Flooding* [Ferguson 2000] tenham efeito.

Serviços TCP avançados – serviços avançados incorporados ao TCP, tais como SACK [Mathis 1996], *Appropriate Byte Counting* [Allman 2003] e *Explicit Congestion Notification* [Ramakrishnan 2001] foram também incorporados ao SCTP através de projetos específicos e implementação nativa.

5 ARQUITETURA DE ALTA DISPONIBILIDADE BASEADA EM SCTP

5.1 Introdução

O capítulo 2 evidencia protocolos de alta disponibilidade como o VRRP e também mecanismos como o *Keepalived daemon*. No capítulo 3 são relacionados os tipos de *firewall* e IPS, bem como os mecanismos de disponibilidade aplicados correntemente.

Os mecanismos de disponibilidade para *firewall*/IPS operam em sua grande maioria através de conexões diretas e dedicadas, através de cabos de rede específicos para tal. Esta forma de conexão caracteriza-se como uma evolução das antigas conexões seriais RS232 [EIA 1969] e visa evitar possíveis problemas associados com LANs.

É notório que alguns mecanismos não estão totalmente estruturados para operar diretamente na LAN compartilhada. Mesmo o VRRP, que fora estruturado para operar em segmentos de redes locais e suportar redundância de roteamento, não está preparado para suportar completamente os mecanismos de HA para *firewall* e IPS. Além disso, possui vulnerabilidades que podem comprometer sua estabilidade e conseqüentemente a de todo o ambiente.

A proposta de arquitetura de alta disponibilidade objetiva a proposição de um mecanismo que atenda aos requisitos de disponibilidade de negócios voltados às novas necessidades de mercado, tais como disponibilidade geográfica, *sites* redundantes e redução de custos.

Esta proposta melhora os mecanismos de disponibilidade e permite a utilização da estrutura de HA sobre LANs. Um estudo de avaliação dessa proposta também é relatado, evidenciando que o SCTP se adapta naturalmente aos requisitos de confiabilidade necessários aos mecanismos de HA.

A seção 5.2 apresenta exemplos de protocolos, análise de tráfego e topologias em HA para *firewall* e IPS. A seção 5.3 apresenta as razões e motivações para HA sobre LAN. A seção 5.4 descreve e detalha a proposição. Na seção 5.5 é apresentada a avaliação da arquitetura. Finalmente, a seção 5.6 apresenta outras medidas que podem ser implementadas de forma a complementar os níveis de disponibilidade da rede.

5.2 Análise de Protocolos e Tráfego HA

Durante o desenvolvimento deste trabalho, devido à rarefação de exemplos comparativos em documentações relacionadas de tráfego HA e visando aprimorar o entendimento do funcionamento dos mecanismos atuais, foram realizadas coletas e análises de tráfego de *firewall* e IPS operando em HA e em ambientes de teste, devidamente estruturados no *Data Center* da empresa CTBC Telecom em Uberlândia – MG. Também, é relatada a experiência do autor em ambientes HA de *firewall* em projetos realizados antes do desenvolvimento deste trabalho.

5.2.1 *Firewall SunScreen*

Em meados do ano de 2002 foi instalada a primeira topologia de *firewall* no *Data Center* operando em HA que constituia-se de duas máquinas executando o sistema operacional SunOs 5.8 e o pacote de *firewall SunScreen* versão 3.2 da Sun Microsystems.

O mecanismo de HA do *firewall SunScreen* 3.2 [Sun Microsystems 2001] requer em seu processo de configuração uma máquina como primária (ativa) e outra como secundária (passiva), exige que uma interface de rede *Ethernet* seja dedicada para *heartbeat* e que as interfaces de roteamento sejam conectadas a *hubs*. A figura 32 ilustra um exemplo desta topologia.

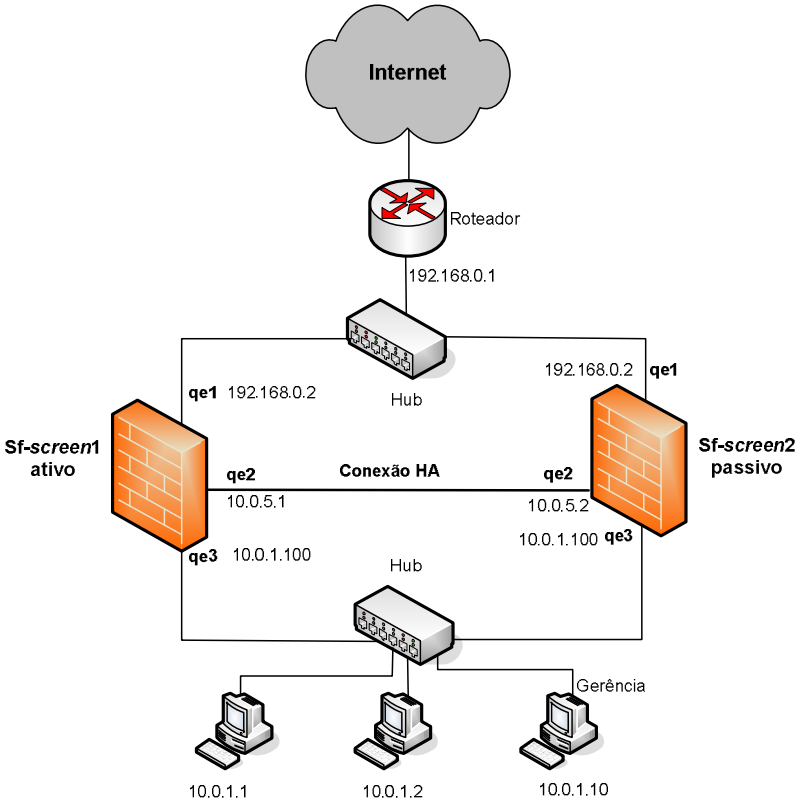

Figura 32 – Topologia *Firewall SunScreen*

As interfaces de roteamento do cluster HA *Sunscreen* têm o mesmo nome e endereçamento IP e quando o *firewall* secundário passa para o estado primário devido a alguma falha do *firewall* primário, os endereços MAC de cada interface de roteamento do *firewall* secundário são alterados para o mesmo endereço MAC das interfaces do *firewall* primário.

As interfaces de roteamento do *firewall* secundário, que opera como passivo, encontram-se em modo promíscuo, ou seja, todo o tráfego de rede replicado pelos *hubs* é

capturado e processado. Assim, o *firewall* passivo duplica o estado das conexões ativas e da máquina de filtros através da análise em tempo real do fluxo de pacotes; não havendo troca da tabela de estado como em mecanismos de HA mais modernos. As interfaces do *firewall* secundário operando em modo promíscuo justificam a exigência de conexão destas a *hubs*, pois estes, ao contrário de *switches*, replicam o tráfego para todas as portas.

As informações de configuração de rede e do sistema operacional não são divulgadas entre os componentes do *cluster* HA. As configurações de regras e chaves secretas são trocadas entre os membros do *cluster* HA em texto claro através da interface de *heartbeat* utilizando o protocolo TCP. A verificação de estado operacional entre as máquinas também é realizada através da interface *heartbeat* utilizando o protocolo ICMP (*Internet Control Message Protocol*) [Stevens 2000].

5.2.2 *Firewall* ASA (*Adaptive Security Appliances*)

A figura 33 ilustra parte do tráfego de HA capturado no dia 10/12/2007 entre dois *firewalls* modelo ASA 5520, fabricante CISCO Systems Inc., conectados em regime de alta disponibilidade, modo ativo/passivo e versão de sistema operacional IOS (*Internetwork Operating System*) 7.2.(3).

A conexão HA utiliza uma LAN, pois os equipamentos localizam-se em prédios distintos visando atender requisitos para mitigar riscos de interrupção do negócio, envolvendo duplicidade de instalações físicas. Para a compreensão do tráfego coletado, a opção de codificação (criptografia) foi desativada.

Para captura do tráfego foi utilizado o mecanismo de SPAN (*Switch Port Analyzer*) [Cisco 2008], que copia o tráfego de interfaces do *switch* ou VLANs (*Virtual Local Area Networks*) [IEEE 2006] de origem para interfaces de destino. Além disso, foi utilizado o *software* analisador de protocolo *Wireshark* [Wireshark 2008] versão 0.99.6a, para identificar o padrão de tráfego.

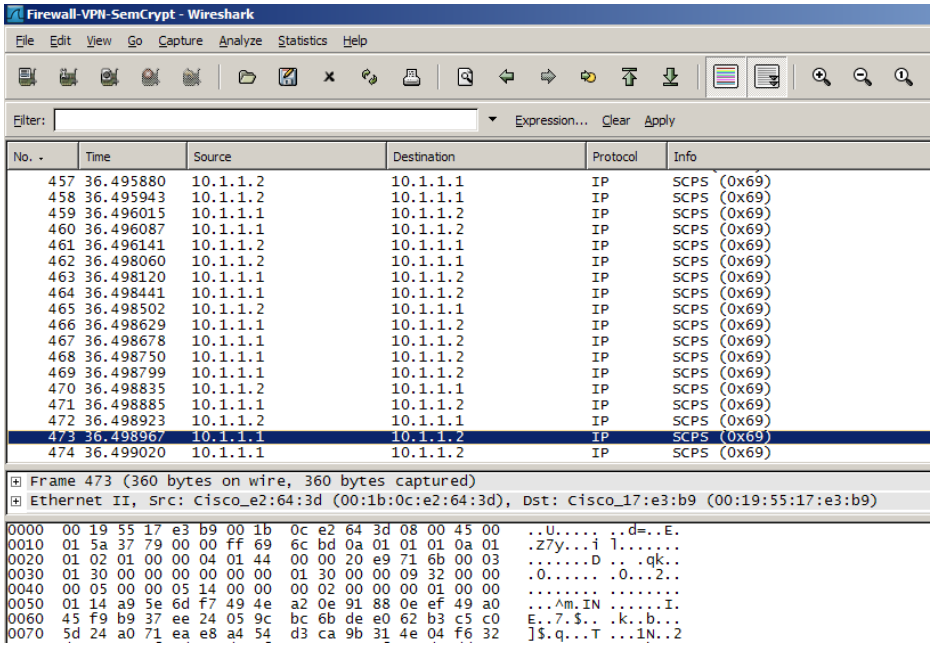

Figura 33 – Coleta de tráfego HA entre *firewalls* ASA 5520

Na topologia analisada, um mecanismo proprietário de HA baseado no protocolo SCPS (*Space Communications Protocol Suite*) [SCPS 1997] é utilizado tanto para o envio de mensagens contendo *heartbeat*, quanto para o envio das informações de estado do *firewall* ativo para o *backup*.

Apesar da conexão HA entre os *firewalls* encontrar-se estruturada sobre rede LAN, durante a coleta de tráfego foi possível verificar através de simulação que em caso de falha na conexão de *heartbeat*, por exemplo, rompimento do cabo, o *firewall* passivo não assumirá a topologia em caso de falha do *firewall* ativo.

Caso o *firewall* passivo esteja operando como primário e ocorra sua reinicialização, ele, não conseguindo comunicar-se com o *firewall* primário (em estado de falha) assumirá suas funções de passivo, ou seja, não assumirá os serviços de controle de segurança, caracterizando uma situação de acéfalo. Assim, estes testes evidenciam a não adequação deste mecanismo para operar HA sobre LAN.

5.2.3 *Firewall* SSG (*Secure Services Gateway*)

A figura 34 ilustra parte do tráfego de HA capturado no dia 13/01/2008 entre dois *firewalls*, modelo SSG 520, fabricante Juniper Networks Inc., conectados em regime de alta disponibilidade, modo ativo/passivo, opção de caminho secundário e versão de sistema operacional NetScreen 5.1.0r4.

Para captura do tráfego foram utilizados mecanismos de SPAN e o *software* analisador de protocolo *Wireshark* versão 0.99.6a para identificar o padrão de tráfego.

Durante a coleta de tráfego foi possível identificar que o tráfego em análise poderia ser coletado em quaisquer portas do *switch* que estivessem na mesma VLAN das interfaces HA, pois o tráfego de *heartbeat* do protocolo NSRP (*NetScreen Redundancy Protocol*) [Cameron 2007] é propagado em camada 2 através de *broadcast* dos pacotes.

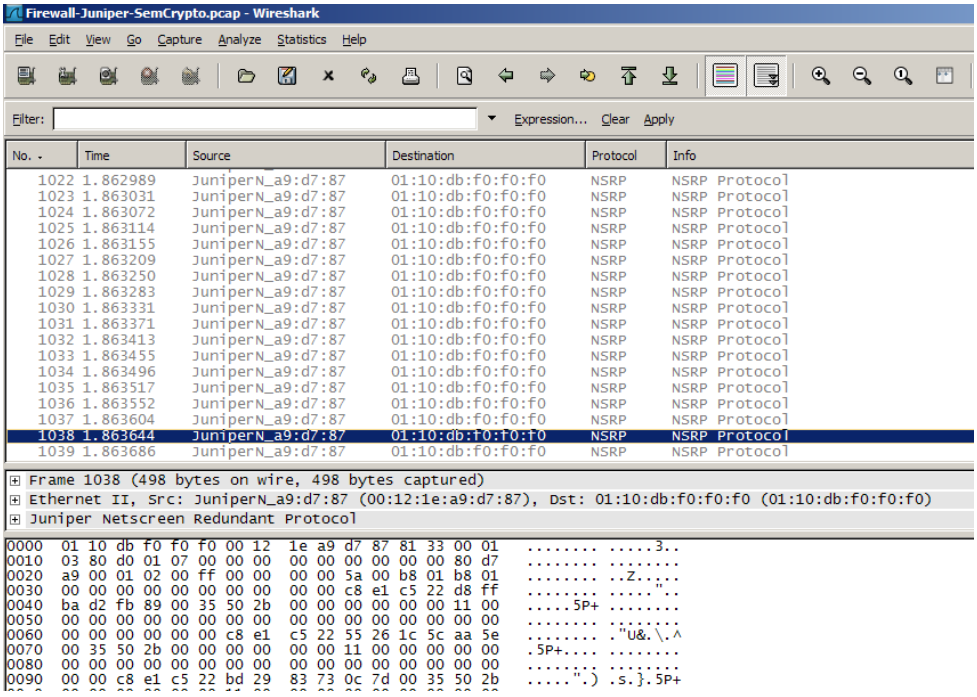

Figura 34 – Coleta de tráfego HA entre *firewalls* SSG 520

O NSRP é um protocolo proprietário utilizado para a troca de mensagens de *heartbeat* e informações de estado entre *firewalls*/IPS do fabricante Juniper Networks Inc. Assim, como o protocolo VRRP, o protocolo NSRP utiliza o conceito de *firewall* virtual através do VSD (*Virtual Security Device*) e interface virtual de rede VSI (*Virtual Security Interface*). Em um grupo VSD, um dos *firewalls* é designado como mestre e o outro como *backup*.

Assim como no protocolo VRRP, o processo eletivo é realizado através da definição de prioridades e os seguintes estados se aplicam: mestre; *backup* primário; *backup*; início; desqualificado (requer intervenção manual); e inoperante.

A forma de propagação do NSRP induz a possíveis problemas de desempenho na rede, por isso é expressamente recomendada sua utilização somente através de cabos diretamente conectados às interfaces HA.

5.3 Razões e Motivações para HA sobre LAN

Os mecanismos atuais de disponibilidade para *firewall*/IPS operam em sua grande maioria através de conexões diretas e dedicadas, através de cabos de rede específicos para tal, limitando-se a uma distância máxima de conexão de 100 metros. *Hubs* não são mais utilizados na topologia em HA devido aos problemas inerentes de desempenho e segurança, tais como *broadcasting* e ataques através da utilização de *sniffers* de rede.

A evolução das aplicações em quantidade e volume de acessos, necessidade de mobilidade dos usuários e conseqüente aumento do tráfego de rede praticamente aboliram a

forma de conexão HA através das conexões seriais RS232, dando lugar as conexões *fast* e *gigabit Ethernet*.

A constante necessidade de maximização de retorno do investimento, redução de custos e ampliação dos requisitos de HA para o negócio, constituem-se necessidades antagônicas, pois quanto maior o nível de disponibilidade requerido maior será o valor de investimento na solução final de HA.

Outro fator de análise, a ser considerado na topologia de HA, é a localização dos *sites* de processamento das informações, pois quanto mais concentrados os recursos de rede, servidores e segurança, maior será a probabilidade de interrupção do negócio no caso de ocorrência de incidentes com o local.

O advento de VPNs BGP/MPLS (*Border Gateway Protocol/Multiprotocol Label Switching*) [Semeria 2001, Armitage 2000] permitiu a extensão das redes locais das empresas através de *backbones* distintos e através da própria Internet, ampliando as possibilidades de distribuição dos recursos de tecnologia da informação mesmo em grandes distâncias.

Assim, a evolução das características de conectividade das LANs, construídas através de *switches* de alto desempenho com conexões via fibras ópticas, provendo segmentação, VPNs BGP/MPLS e capacidade de priorização de tráfego, possibilitam a descentralização do negócio em um nível antes praticamente impossível.

Neste aspecto, as soluções de HA para *firewall* e IPS não acompanharam esta evolução em sua completude, pois os protocolos utilizados não possuem a confiabilidade necessária para lidar com possíveis problemas na rede. Além disso, os mecanismos de HA não foram estruturados em sua concepção original para operar sobre rede, sendo na maioria dos casos desaconselhada nos próprios manuais de configuração, a sua utilização sobre LAN.

Para exemplificar as vantagens de HA sobre LAN considerar-se-á o cenário onde uma empresa possui dois *sites* situados a 10 quilômetros de distância entre si, interligados por fibra óptica com redundância e por caminhos distintos. Além disso, os dois *sites* possuem conectividade com a Internet e capacidade de processamento das informações no caso de falha de um deles, pois os servidores estão distribuídos. Abstraindo-se das características necessárias de alta disponibilidade para as aplicações e considerando-se a necessidade de investimento em *firewall*/IPS para manter a conectividade com a Internet nos dois sites, as seguintes possibilidades podem ser implementadas:

- Topologia com *clusters* de *firewalls* em HA e roteadores VRRP em cada *site*;
- Topologia com *clusters* de *firewalls* e *switches* camada 7 em cada *site*; e
- Topologia com um *firewall* em cada site com suporte a HA sobre LAN.

5.3.1 Topologia com *clusters* de *firewalls* em HA e roteadores VRRP

A topologia apresentada na figura 35 caracteriza um esquema clássico de alta disponibilidade para *firewall*/IPS considerando *sites* redundantes. Nesta topologia, devido à distância entre os *sites* é necessário considerar um conjunto de *firewall*/IPS para cada *site* visando manter a confiabilidade da estrutura.

Como os mecanismos de HA não estão preparados para suportar roteamento através do *default gateway* da rede entre *sites* distintos, é necessário incluir camadas de roteadores na rede operando com o protocolo VRRP e garantindo a disponibilidade de roteamento. Também é necessária uma integração estrita entre as camadas de roteadores e o protocolo de roteamento do *firewall*/IPS para que falhas nas interfaces de rede sejam detectadas e comunicadas ao *cluster*.

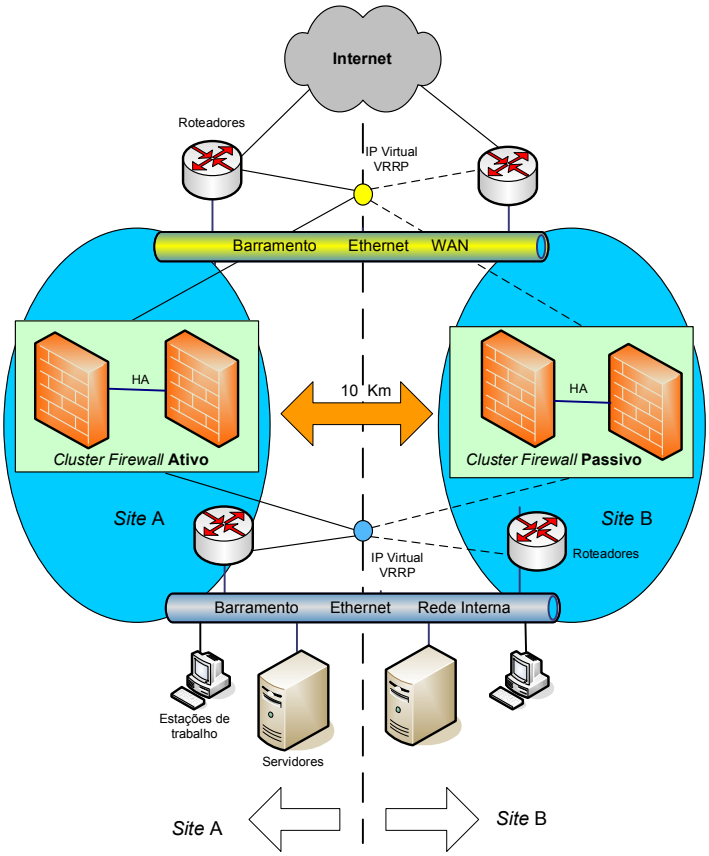

Figura 35 – Topologia com *cluster* de *firewalls* em HA e roteadores VRRP

5.3.2 Topologia com *clusters* de *firewalls* e *switches* camada 7

Uma alternativa à camada de roteadores VRRP é a utilização de *switches* camada 7 com recursos de balanceamento de carga ou *switches* específicos para balanceamento de carga entre *firewalls*, os denominados FWLB (*Firewall Load-Balancing*) [Khan 2006]. Está além do escopo deste trabalho, discutir as características e modos de implementação de FWLB, porém é notório que topologias que os utilizam possuem maior complexidade de configuração e administração.

Assim, as opções de topologia descritas nas seções 5.3.1 e 5.3.2 possuem como principais desvantagens:

- Alto investimento, pois há a necessidade de aquisição de quantidade maior de *firewalls*/IPS e aquisição de roteadores ou FWLBs;

- Perda das conexões estabelecidas no caso de interrupção total do *site* ativo para a opção que utiliza roteadores, pois estes não estão estruturados para manter tabelas de conexões ativas;
- Complexidade na gestão de configuração, uma vez que existem mais equipamentos na topologia para serem configurados e integrados;
- Complexidade na gestão de incidentes, pois a presença de mais equipamentos na rede amplia os pontos de falha e o processo de detecção que não estiver estruturado sobre ferramentas adequadas de gerência poderá apresentar tempos de detecção destas falhas não condizentes com as expectativas do negócio.

5.3.3 Topologia com *firewalls* em HA sobre LAN

A figura 36 ilustra a topologia com *firewalls* em HA sobre LAN. Ao contrário da topologia da figura 35, nesta topologia são necessários dois *firewalls* para prover alta disponibilidade entre os *sites* e não são necessários os roteadores para a rede interna. Portanto, os investimentos e pontos de falha na rede são menores.

Porém, a implementação desta topologia utilizando os recursos e *hardware* disponíveis correntemente apresentam os problemas identificados na seção 5.2, tais como *broadcasting* indevido na rede e possibilidade de comportamentos incondizentes com as necessidades e requisitos de disponibilidade. Assim, a necessidade de um mecanismo mais elaborado para o controle de HA nesta topologia faz-se necessário. Este será delineado a partir da seção 5.4.

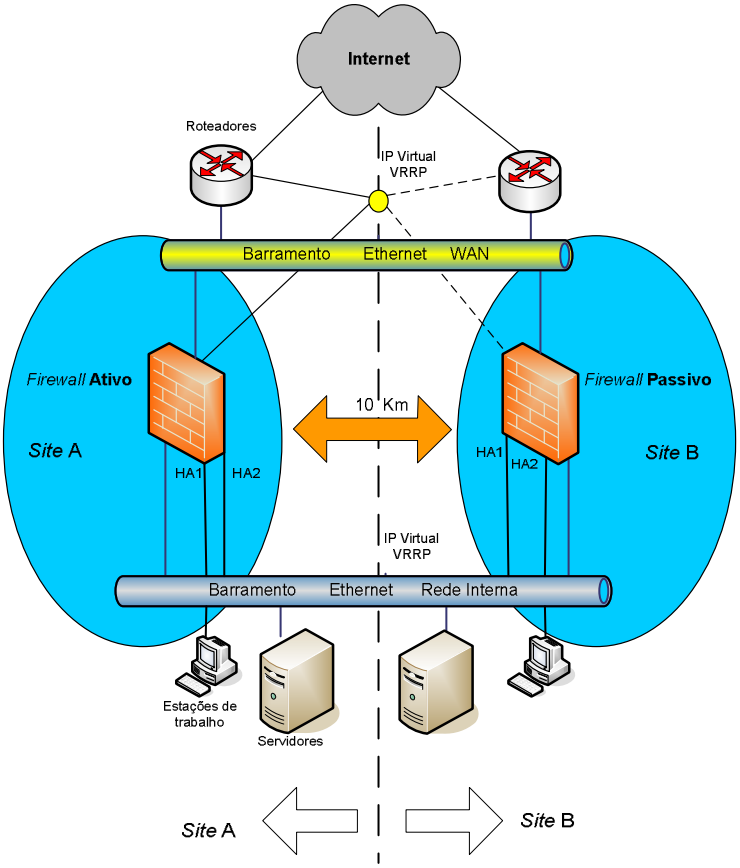

Figura 36 – Topologia com *firewalls* em HA sobre LAN

Embora atualmente possua problemas relacionados à implementação, esta topologia apresenta as seguintes vantagens:

- Necessidade de investimentos inferiores aos investimentos das opções anteriores;
- Média complexidade de configuração e administração;
- Não há perda das conexões estabelecidas em caso de queda do *site* principal; e
- Menor complexidade na gestão de incidentes.

A tabela 9 compara a opção de HA para *firewall*/IPS sobre LAN versus a opção de HA convencional para implementação de *sites* geograficamente separados.

Tabela 9 – HA sobre LAN versus HA convencional para *sites* distantes

	HA sobre LAN	**HA convencional**
Investimentos na topologia	Médio	Alto
Complexidade da topologia	Média	Alta
Complexidade de configuração	Média	Alta
Complexidade na gestão de incidentes	Baixa	Média – Alta
Confiabilidade da estrutura	Alta	Média
Possibilidade de perda das conexões estabelecidas em caso de queda do *site* principal	Baixa	Alta – Com uso de VRRP; Média – Com uso de FWLBs.

Durante o desenvolvimento deste trabalho não foram identificadas outras iniciativas de proposição de HA para *firewall*/IPS sobre LAN e nem comparativos entre os mecanismos correntes de HA. Portanto, este trabalho não comenta artigos relacionados.

Assim, além dos benefícios de redução de custos, ampliação da flexibilidade e dos níveis de disponibilidade, a proposição de um mecanismo de HA para *firewall*/IPS sobre LAN enseja como principal razão e motivação a inexistência desta opção e possibilidade de padronização do mecanismo proposto.

5.4 Proposta de Alta Disponibilidade para *Firewall*/IPS Baseada em SCTP

A proposta da arquitetura HA para *firewall* e IPS está estruturada para suportar o modelo Ativo/Passivo, onde só pode existir um equipamento como mestre e outro como *backup* na rede. Um dos componentes será eleito mestre durante o processo inicial de configuração e carga do sistema operacional. A figura 37 ilustra o digrama do esquema proposto.

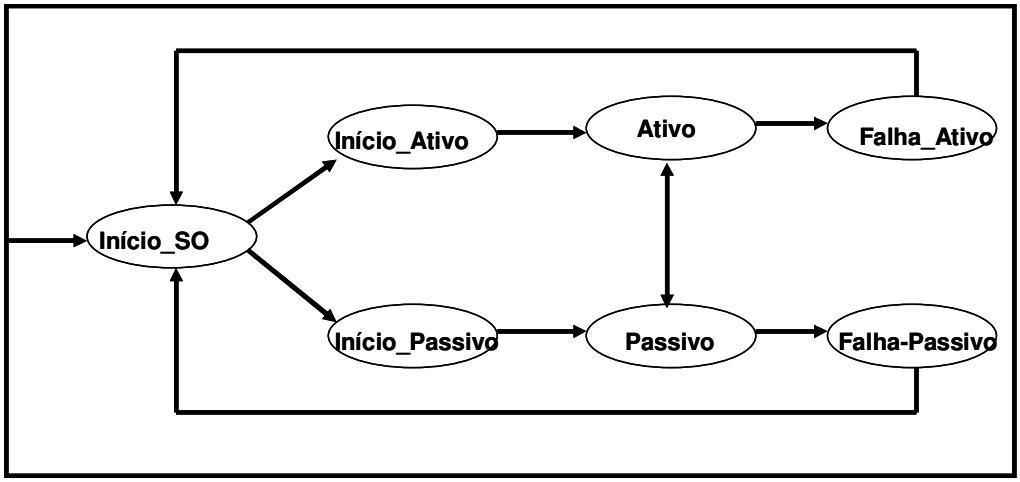

Figura 37 – Esquema HA

O processo é iniciado a partir do estado Início_SO e, dependendo da prioridade preestabelecida, o equipamento é eleito mestre (Início_Ativo) ou *Backup* (Início_Passivo). Um equipamento no estado passivo (*Backup*) pode comutar para mestre através da mudança de estado de Passivo para Ativo. Uma falha pode ocorrer resultando nos estados Falha_Ativo ou Falha_Passivo.

A figura 38 ilustra a topologia que será considerada como base para a definição e detalhamento da proposição, contendo dois *firewalls*/IPS com duas interfaces HA.

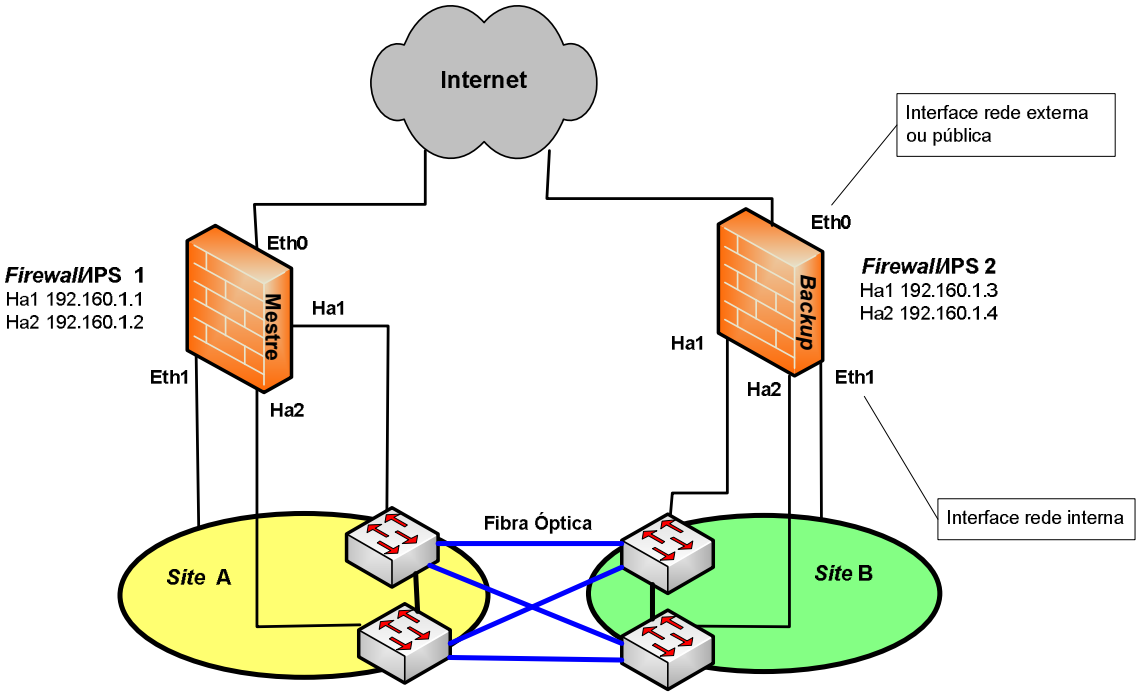

Figura 38 – Topologia associada à proposta, *firewalls*/IPS com duas interfaces HA

A proposta considera a possibilidade de conexão das interfaces HA em LAN e a utilização de mais de uma interface para este propósito. Entretanto, a funcionalidade da proposta é afetada se as interfaces HA forem conectadas diretamente, como será explicado na seção 5.4.2.

5.4.1 Metalinguagem referente ao esquema

A metalinguagem abaixo resume o esquema de funcionamento da proposta.

Início_Ativo (Mestre)

Se existe Ativo(passivo) *Então* Ativo (Libera-Passivo)	# O mestre está iniciando com *backup* operando # na rede como mestre
Senão Ativo(Mestre)	# Início normal do mestre
Fim-Se	

Início_Passivo ( )

Passivo ( )	# Passivo em operação, esperando # falha do mestre

```
Ativo ( )
  Se Libera-Passivo Então            # Backup operando como mestre e o
     Libera_IP (Backup)              # controle retornará ao mestre.
     Comuta_IP (Mestre)
     Passivo (Backup)
     Ativo (Mestre)
  Senão Se  Mestre Então             # Mestre operando normalmente
          ARP-Gratuíto (IP-Virtual)
          Envia pacotes de heartbeat e tabela de estado para o Backup (passivo)
        Senão
          Se Passivo Então           # Backup assumirá como mestre
                Para  I = 1 até 3 segundos Faça
                   Envia mensagens de  heartbeat para o mestre
                   Se  resposta = ok  Então                # O mestre retornou
                      Passivo (Backup)
                      Ativo (Mestre)
                   Fim-Se
                Fim-Faça
                Comuta_IP (Backup)   # O IP virtual é migrado para o Backup
                Variável-Passivo = "Nó ativo como mestre é o backup"
                Ativo (Backup)
          Senão # "Nó ativo como mestre é o Backup"
                                     # Não é necessário tomar o IP virtual novamente
             Armazena tabela de estado
             Ativo (Variável-Passivo)
           Fim-Se
      Fim-Se
  Fim-Se

Passivo (  )
  Se existe Ativo (Mestre) Então
    Responda as mensagens de heartbeat e tabela de estado recebidas
    Passivo (Backup)
  Senão        # Não existe nenhum mestre operando
    Ativo (Backup)
```

Fim-Se

As funções Libera_IP e Comuta_IP estão associadas ao controle de endereçamento das interfaces não HA, ou seja, às interfaces internas e externas. Para controle do endereçamento IP destas interfaces os seguintes mecanismos podem ser utilizados:

- Endereço IP virtual utilizado como *gateway* das redes, assim como o utilizado pelo VRRP, onde exista um controle de verificação de estado operacional do equipamento através de *multicast*;
- Endereço IP físico das interfaces não HA do Mestre utilizados como *gateways* das redes, onde é necessário um mecanismo baseado em protocolos como o TCP e UDP para verificar o estado funcional das interfaces (*tracking*).

Neste mecanismo, os endereços MAC das interfaces são divulgados. Assim, no caso de falha de uma destas interfaces no Mestre, o processo de comutação consiste em o *Backup* assumir o endereço IP atribuído às interfaces do Mestre. No processo de início do Mestre, caso o *Backup* esteja operando como Mestre, ele não poderá realizar a divulgação ARP dos endereços associados às interfaces internas e externas até que o *Backup* libere estes endereços (Libera_IP).

Avaliando as duas alternativas, o mecanismo utilizado pelo VRRP apresenta-se como uma opção de implementação mais eficiente e flexível, pois utilizará um IP virtual e a divulgação ARP não será associada a nenhum endereço MAC real das interfaces de rede internas e externas.

A arquitetura proposta abstrai-se deste problema e concentra os esforços no controle de envio de mensagens entre as interfaces HA. Entretanto, qualquer uma das abordagens citadas pode ser empregada e como sugestão de futuros trabalhos, o SCTP pode ser considerado como o protocolo para realizar o controle de estado operacional das interfaces

internas e externas na alternativa de utilização do próprio endereço IP (MAC real) das interfaces não HA como *gateways* da rede.

Entretanto, não é possível utilizar o SCTP para realizar o controle de estado operacional das interfaces internas e externas se o endereçamento MAC associado ao *Multicast* for utilizado, pois o SCTP não suporta *Multicast*.

5.4.2 Fluxo de mensagens *heartbeat* da arquitetura

A associação estabelecida entre os equipamentos ativo e passivo, considera um fluxo de dados (fatia DATA) e outra fatia do tipo HEARTBEAT para situações onde existam mais de uma interface HA. A fatia de Dados é utilizada para o transporte das informações de estado do equipamento, bem como para realizar a verificação de saúde da máquina *Backup*. A fatia HEARTBEAT será utilizada para verificar a saúde das interfaces alternativas da máquina *Backup*.

Assim, considerando a topologia da figura 38, as fatias ilustradas na figura 39 são transportadas em uma única associação com o seguinte formato:

Associação IP-Mestre-Ha1:Porta-Origem = { [IP-Backup-Ha1, IP-Backup-Ha2: Porta-HA]}

ou, utilizando os respectivos endereços IP:

192.160.1.1:2008 = {[192.160.1.3,192.160.1.4:20005]}

Outra possibilidade considera que as faixas de endereçamento IP atribuídas às interfaces HA sejam distintas. Logo, é necessária a presença de algum mecanismo de *routing* na rede LAN para garantir a existência de comunicação entre as redes. Porém neste caso o emprego de mecanismos de qualidade de serviço QoS [Armitage 2000] é expressamente recomendado para ampliar o nível de confiabilidade da arquitetura.

O intervalo preestabelecido no SCTP para envio de fatias HEARTBEAT é de 30s e como a maioria dos mecanismos de HA para *firewall* e IPS operam na faixa de 2s, este valor será considerado na arquitetura proposta. Além disso, é atribuído um único endereço IP a cada interface HA e que podem estar em faixas de endereçamento diferentes, desde que exista mecanismo de *routing* na rede.

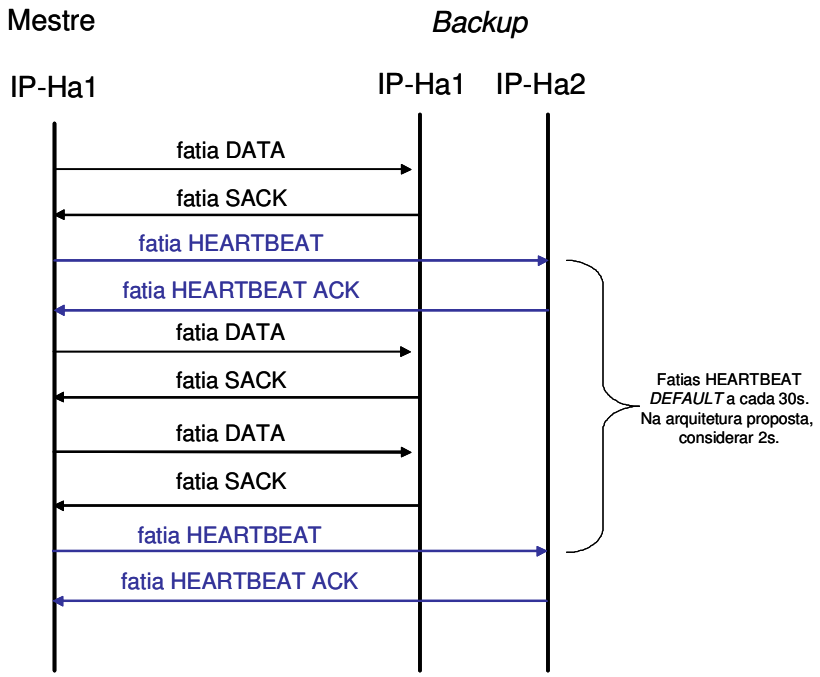

Figura 39 – Fluxo de fatias associado à proposição

Para identificar falha de enlace envolvendo as interfaces HA, a arquitetura utiliza dois métodos: fatias *heartbeat* e condição limite de retransmissão de dados. No início da associação contendo múltiplos endereços IP, um deles é eleito como primário e o mecanismo de monitoramento de *heartbeat* funcionará da seguinte forma:

- A fatia HEARTBEAT é enviada para qualquer endereço de destino que não tenha contabilizado resposta por um período (*timeout*) maior que o do contador de tempo do *heartbeat*, igual a 2s;
- A contabilização de resposta utiliza fatias que atualizam o RTT, as fatias DATA e HEARTBEAT;

- O contador de tempo do *heartbeat* é iniciado novamente toda vez que uma fatia de DATA ou HEARTBEAT é enviada;
- O receptor responde com uma fatia HEARTBEAT-ACK;
- Toda vez que uma fatia HEARTBEAT é enviada, a variável de registro de erro para o destino específico é incrementada;
- Toda vez que a fatia HEARTBEAT-ACK é recebida, o contador de erros é zerado;
- Toda vez que uma fatia de dados (DATA) que foi enviada para o destinatário é confirmada (SACK), o contador de erros é zerado;
- Toda vez que o contador de erros exceder o limite preestabelecido 5 (cinco), o destino é declarado como não acessível;
- Se o endereço de destino primário é marcado como não acessível e se existir endereço alternativo, este será escolhido e utilizado;
- Fatias HEARTBEAT continuarão a ser enviadas para os endereços não acessíveis. Se ocorrer resposta o contador de erros é zerado e o destino é marcado como acessível.

  Se este destino (endereço não acessível) é o endereço IP eleito como primário no início da associação e não ocorreu nenhuma intervenção do usuário, a comunicação é restaurada para este endereço.

A arquitetura proposta considera que as interfaces HA estão conectadas à LAN e não diretamente entre si como nos modelos tradicionais de HA, pois a conexão através da rede permitirá que a funcionalidade de *multihoming* do SCTP seja utilizada para prover caminhos alternativos.

Se as interfaces HA forem conectadas diretamente uma às outras, no denominado modelo *dual-homed* [Jungmaier 2002], é necessário que a camada de controle do HA detecte a falha de um dos caminhos e interfira se a falha ocorrer no caminho principal, restabelecendo

a associação através do caminho alternativo. Contudo, esta abordagem insere complexidade na aplicação de controle e anula toda a flexibilidade e transparência provida pelo SCTP.

A figura 40 ilustra os estados de transição e a troca de fatias entre o Mestre e *Backup*.

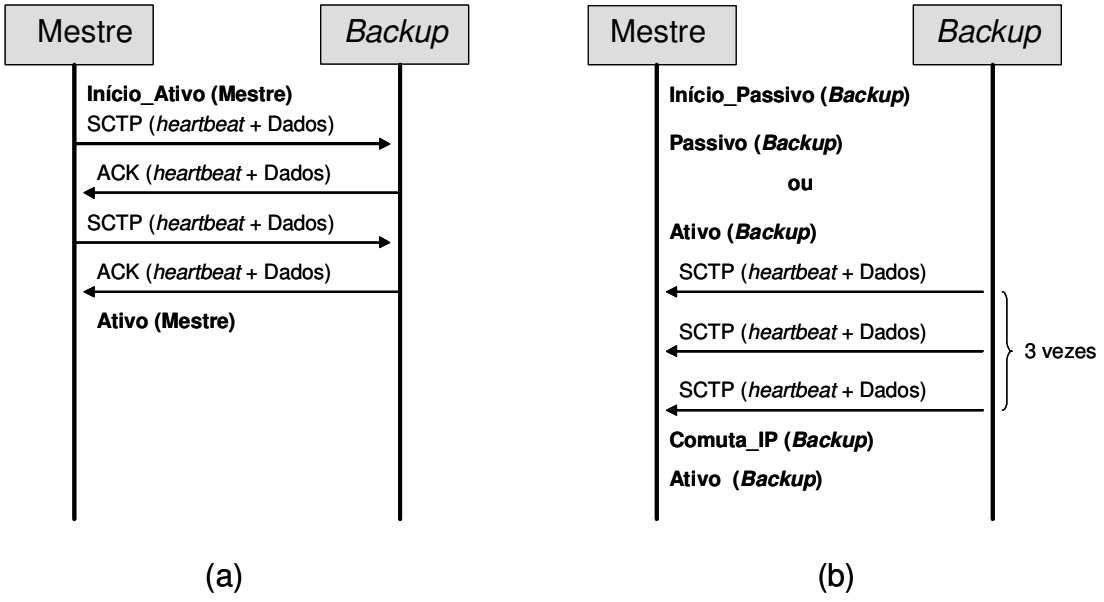

Figura 40 – Estados de transição, Mestre operando na rede e ausente

No estado Início_Ativo (Mestre) em (a) o equipamento principal assume como Mestre, passa ao estado Ativo (Mestre), opera em condições normais e envia as mensagens de *heartbeat* e dados para o *Backup*. Se o *Backup* estiver operando, responderá estas mensagens (ACK).

No processo Início_Passivo (*Backup*) em (b) se não houver nenhum Mestre em operação na rede, o *Backup* assumirá como Mestre resultando no estado Ativo (*Backup*). Para assegurar que não existe mais de um Mestre serão enviadas três mensagens de *heartbeat* para endereço IP do Mestre. Neste estado é necessário que o *Backup* assuma os endereços IPs das interfaces internas e externas do Mestre.

Caso exista um Mestre em operação, o *Backup* terá um processo de inicialização normal e resultará no estado Passivo (*Backup*).

A figura 41 ilustra os estados de transição e mensagens quando ocorre a falha do Mestre e seu retorno ao estado operacional.

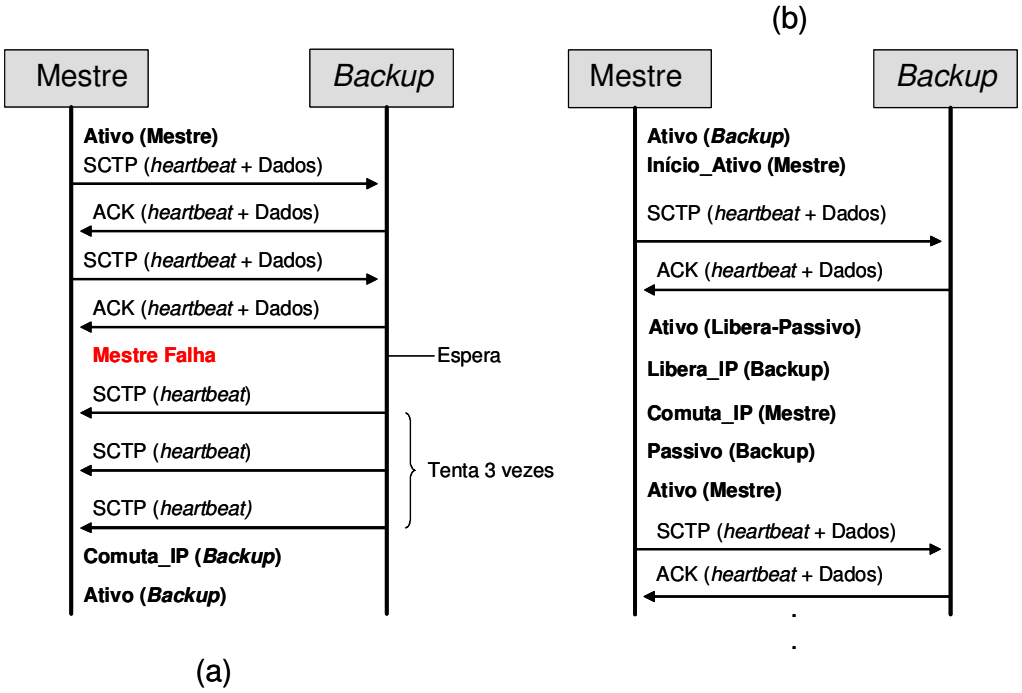

Figura 41 – Estados de transição, Mestre em estado de falha e retorno a operação

No estado Ativo (Mestre) o Mestre envia as mensagens de dados e *heartbeat* para o *Backup*. As possíveis falhas associadas ao Mestre incluem: falha de alguma das interfaces internas ou externas; falha das interfaces HA; falha parcial ou total do equipamento. Na ocorrência de uma falha parcial o Mestre deverá declarar-se como inoperante e não responderá às mensagens de *heartbeat* do *Backup*.

Se houver uma falha no Mestre, figura 41 (a), o *Backup* verificará após alguns instantes que não está mais recebendo fatias de Dados e, então, enviará três mensagens de *heartbeat* a cada segundo. Caso não haja resposta o *Backup* assumirá os endereços IP das interfaces internas e externas (Comuta_IP) e passará ao estado Ativo (*Backup*).

Para evitar tráfego desnecessário na rede e conseqüentes variações de desempenho, enquanto o *Backup* for o Mestre, ele não enviará mensagens de *heartbeat* para o Mestre em

estado de falha. Ou seja, o *Backup* só mudará de estado se o Mestre retornar a rede e iniciar a transmissão de mensagens de dados e *heartbeat*, figura 41 (b).

Esta fase do processo é a mais complexa, pois quando o Mestre retornar à rede e verificar que existe *Backup* operando como Mestre deverá esperar pelo envio das informações de estado das conexões ativas através do *firewall*/IPS *Backup* e pela liberação dos endereços IP associados às interfaces internas e externas do *Backup*. Assim, existe a possibilidade de uma condição de "acéfalo" na arquitetura, caso ocorra uma falha do *Backup* neste momento.

Para resolver esta situação é recomendável a definição de um tempo de espera para o recebimento das mensagens HA de no máximo 2s. Somente após o recebimento das mensagens de estado das conexões ou do término do tempo de espera (*timeout*) o Mestre passará ao estado Ativo (Libera_Passivo). Em seguida o *Backup* libera os endereços IPs das interfaces internas e externas executando a função Libera_IP(*Backup*) e o Mestre assume estes endereços através da função Comuta_IP (*Mestre*) culminando com o estado Ativo (Mestre).

5.4.3 Especificação da arquitetura de HA

O objetivo desta seção é especificar a arquitetura de HA para *firewall*/IPS através das camadas e planos conforme figura 42.

A figura 42 ilustra a interação entre as camadas de administração de segurança, de gerência, a camada de controle e serviços de HA. As camadas de administração de segurança e de gerência são resumidamente descritas, pois não constituem o foco deste trabalho. Assim, abstrair-se-á das complexidades internas das mesmas.

A camada de administração de segurança é constituída pelos módulos de controle de fluxo de pacotes e módulo de configuração de segurança.

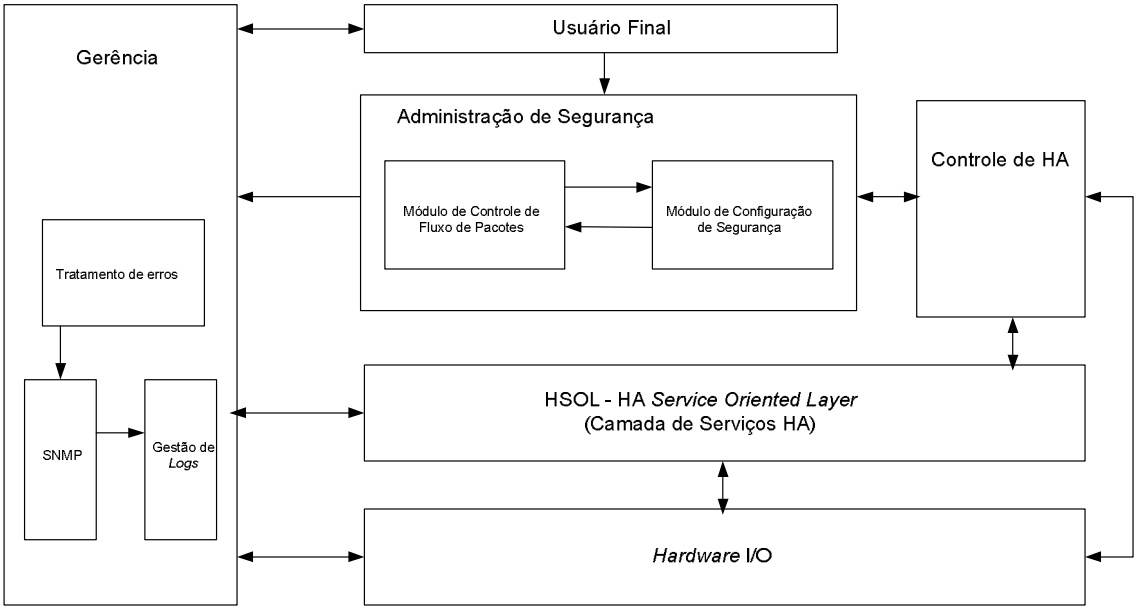

Figura 42 – Camadas da arquitetura HA proposta

5.4.3.1 Camada de administração de segurança

Esta camada é responsável pelo controle do fluxo de pacotes entrante e saínte do *firewall*. É composta pelos módulos de configuração de segurança e módulo de controle de fluxo de pacotes. Comunica-se com as camadas de gerência e controle de HA toda vez que uma nova sessão for estabelecida ou encerrada. Opcionalmente, pode comunicar-se com a camada de gerência toda vez que a entrada ou saída de um pacote for negada. Esta comunicação depende de habilitação prévia através de configuração e pode consumir recursos de armazenamento (volume de mensagens em *logs*).

- Módulo de controle de fluxo de pacotes – recebe os pacotes das interfaces de rede (I/O) através da camada HSOL (*Hardware Oriented Service Layer*), verifica o cabeçalho do pacote, se já existe sessão estabelecida e se existe regra permitindo o pacote. Para garantir desempenho satisfatório, as verificações podem ser realizadas diretamente em ASIC como nos modelos de *firewall* atuais. Quando não existe sessão estabelecida, este módulo cria uma nova entrada na tabela de sessões caso exista regra permitindo o fluxo do pacote. Se não existir regra permissiva, o pacote

é descartado e uma mensagem poderá ser enviada a camada de gerência. Toda vez que uma nova sessão é criada ou encerrada este módulo informa a camada HSOL sobre a sessão enviando as informações de início ou término. Estas informações incluem endereços IP de origem e de destino, informações da tradução NAT, regras de controle associadas, *timestamp* de início ou fim da sessão, entre outras.

- Módulo de configuração de segurança – recebe as configurações de segurança definidas pelo administrador, como as regras de controle de entrada e saída de pacotes, configurações de tradução de endereços IP (NAT), quando gerar ou não uma mensagem para ser enviada a camada de gerência e configurações específicas de controle para fluxos de pacotes. As regras de controle são utilizadas toda vez que existir a necessidade de criação de uma nova sessão. Assim, toda vez que uma configuração é criada ou removida, a camada de controle de HA é informada.

5.4.3.2 Camada de gerência

Esta camada é responsável por armazenar informações de sessões estabelecidas, rejeitadas, informações de controle de *hardware* e *software*. Também gerencia os erros que podem ocorrer no equipamento. No caso de detecção de problemas de *hardware* ou *software*, dependendo da severidade, esta camada informa a camada HSOL para tornar o *firewall* inoperante, iniciando o processo de comutação para o *firewall* secundário (*failover*). Esta camada armazena informações em *logs*, envia mensagens SNMP (*Simple Network Management Protocol*) [Harrington 1999] e mensagens para o console do *firewall*.

5.4.3.3 Camada de controle de HA

Esta camada é responsável pelo início das funções de HA Início_Ativo( ) e Início_Passivo( ), ou seja, o equipamento deverá se tornar mestre ou *backup* conforme esquema da figura 37, dependendo da prioridade preestabelecida. Comutação entre ativo e passivo e armazenamento das informações de estado do *firewall* também são tarefas desta

camada. Envia para a camada de serviços HA (HSOL) as informações de estado e recebe desta as informações de saúde do equipamento componente do conjunto. Dependendo das informações recebidas da camada de gerência ou do próprio controle de *heartbeat* (Verificador de Saúde) da camada HSOL, poderá acionar os mecanismos de liberação de endereçamento IP (Comuta_IP) e outros recursos.

5.4.3.4 Camada de serviços HA (HSOL)

Esta camada é responsável pela verificação de saúde dos componentes do conjunto, bem como a atualização de informações de estado operacional entre os equipamentos ativo e passivo. As funções Ativo( ) e Passivo( ) são executadas nesta camada. A figura 43 ilustra a proposição de módulos para a arquitetura da camada de serviços HA utilizando o SCTP.

A arquitetura utiliza os mecanismos de controle de IP virtual definidos pelo VRRP, para controlar os *gateways* das redes internas e externas. São necessárias tantas instâncias de VRRP quanto a quantidade de interfaces internas e externas.

Um mecanismo para notificação de erros e mensagens informativas é estruturado sobre o protocolo SNMP. Este mecanismo é responsável pelo envio de mensagens *traps* para o controle de *log* interno do *firewall*/IPS e/ou para um servidor de *log* na rede, através da camada de gerência.

Os componentes mais importantes da estrutura são: Verificador de Saúde; e Controlador de IP Virtual.

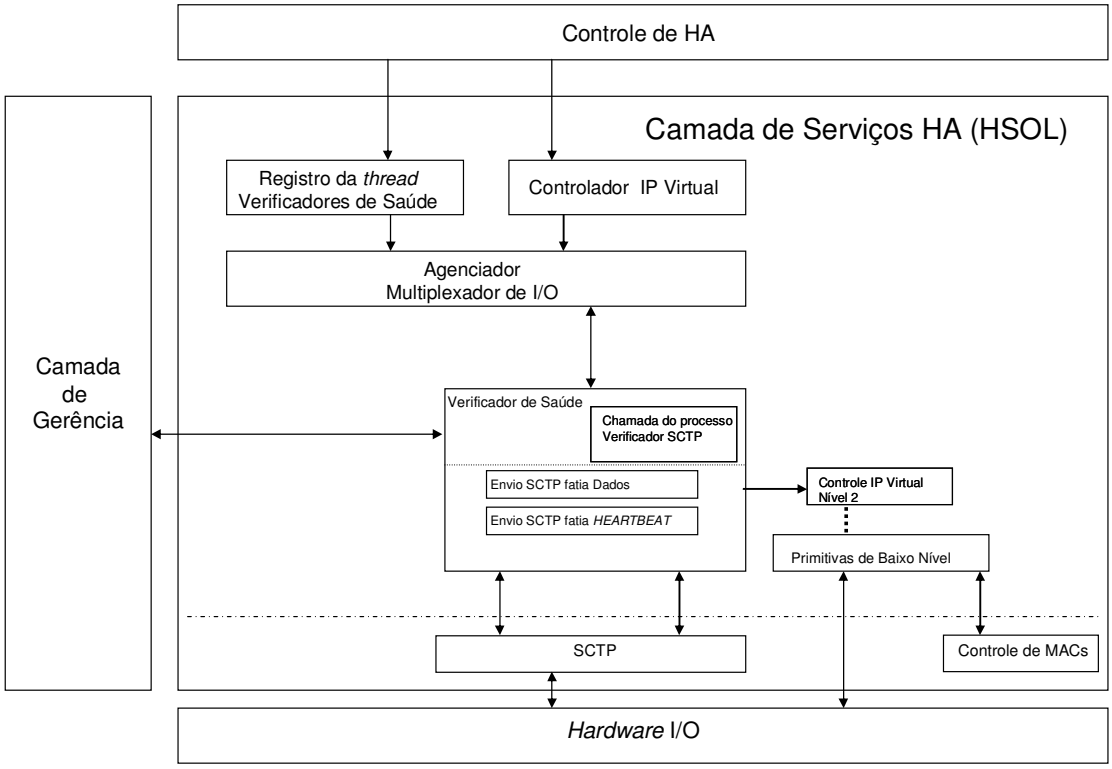

Figura 43 – Detalhamento da camada HSOL

Verificador de saúde – é responsável pela chamada interna dos processos associados ao protocolo SCTP, pelo controle de envio e recebimento de respostas das mensagens de Dados e *heartbeat* para cada interface HA, detalhados na seção 5.4.2. Comunica-se com os processos de notificação SNMP e controle de IP Virtual. Esta camada apresenta-se como o grande diferencial em relação à estrutura apresentada no *Keepalived* seção 2.5, pois utilizará um único protocolo para a verificação de estado operacional da estrutura HA.

Controlador de IP virtual – Para cada interface de rede interna e externa é necessário um mecanismo para controlar o endereço IP virtual. Este mecanismo é estruturado sobre o protocolo VRRP sendo necessária uma instância de controle de IP virtual para cada interface de rede não HA. Este mecanismo não utiliza o protocolo IGMP para verificar o estado operacional (saúde) das interfaces não HA. Como o SCTP não suporta IP *Multicast*, é realizado um controle através do envio de mensagens de Dados para os endereços IP primários de cada equipamento, ou seja, são utilizados os endereços MAC reais das

interfaces. Assim, se há resposta para as mensagens é possível afirmar que o IP virtual está ativo. Este módulo está diretamente relacionado ao processo de início do conjunto HA, divulgando o endereço virtual e determinando a qual equipamento este estará diretamente relacionado.

Controle de IP Virtual Nível 2 – Este módulo funciona basicamente do mesmo modo que o módulo Controlador de IP virtual, porém está diretamente relacionado à verificação de *heartbeat* e alteração de IPs no caso de detecção de problemas.

SCTP – Este módulo recebe os pacotes de Dados e *heartbeat* oriundos do Verificador de Saúde e os encaminham para a camada IP e vice-versa. Também realiza o processo de *multihoming* de forma transparente para as outras camadas.

A arquitetura proposta utiliza um combinado do SCTP e VRRP para o controle de saúde das interfaces HA, internas e externas, respectivamente. A utilização do SCTP para controle evita as vulnerabilidades associadas com implementações estruturadas sobre IGMP e outros protocolos IP *Multicast*.

5.5 Avaliação da Proposta

Esta seção apresenta a avaliação da proposta. O objetivo é apresentar informações sobre experimentos realizados em outras pesquisas que atestem as funcionalidades requeridas do protocolo SCTP para o modelo proposto.

Uma avaliação mais precisa e real somente poderá ser realizada após implementação desta proposta. A implementação desta proposta não está contemplada no escopo deste trabalho, porém vale destacar que a alteração do código fonte do *Keepalived* para operar com o SCTP em substituição aos verificadores TCP, HTTP e SSL é factível, pois o código fonte é aberto e o sistema operacional Linux já suporta nativamente o SCTP, sendo objeto de trabalhos futuros.

As vantagens do SCTP sobre o TCP basicamente consideram o fato de que o SCTP entrega dados em fatias com fluxos independentes na mesma associação, eliminando problemas associados com o bloqueio *head-of-line*. Contraditoriamente, o TCP entrega as mensagens através de fluxos de *bytes*.

Entretanto esta vantagem não é substancial no transporte de mensagens HA conforme discutido na referência [Grinnemo 2005] e a proposta da arquitetura utilizar fatias de DATA e HEARTBEAT, ou seja, uma única associação com fluxos independentes e ordenados.

Este estudo realizou um experimento detalhado no impacto do bloqueio *head-of-line* na entrega de mensagens ordenadas utilizando o SCTP. Concluindo que apesar do *head-of-line* introduzir atrasos significativos em uma pequena fração de mensagens SCTP, ele tem impacto marginal no atraso médio de entrega de transmissões fim-a-fim e que este impacto não varia quando o tráfego aumenta. O impacto do *head-of-line* pode ser substancial se a

janela de transmissão do emissor for relativamente grande e existirem eventos ainda ativos de bloqueio *head-of-line*.

Uma importante vantagem do SCTP para o transporte de mensagens HA é o *multihoming*. Em uma associação SCTP, mais de um endereço IP pode ser utilizado para o estabelecimento desta associação. Assim, se o endereço primário falhar o fluxo é direcionado para o outro endereço.

Em [Rane 2003] é realizado um experimento de avaliação de *multihoming* considerando características de desempenho com altas taxas de transferência e transparência para a aplicação considerando balanceamento de carga. O experimento conclui que é possível atingir altas taxas de desempenho na transferência de dados utilizando *multihoming*, que a implementação é transparente para a camada de aplicação e que pequenas alterações nas bibliotecas *socket* de implementação são necessárias para este objetivo. O experimento considerou uso de protocolos de *routing* para garantir a diferenciação entre os caminhos.

Em [Jungmaier 2002] é realizado um experimento para verificar a funcionalidade de *multihoming* no transporte de mensagens SS7 e em condições de falha dos caminhos. Dois cenários são avaliados: uma associação *dual-homed*; e duas associações, conforme figura 44 (a) e (b) respectivamente. Ambos os cenários podem ser utilizados para permitir a recuperação rápida em caso de falha.

Entretanto, existem algumas diferenças entre os dois: o cenário da figura 44 (a) permite uma transição uniforme mantendo o atraso médio por fatia menor que o cenário (b) durante o processo de comutação de interfaces (*failover*). O cenário (a) possibilita a comutação rápida no tempo obrigatório de 800ms, portanto é mais recomendado que o cenário (b). Em uma falha no cenário (a) todas as fatias são automaticamente transmitidas pelo caminho secundário. Os mecanismos do SCTP podem levar a um reconhecimento um pouco tardio de falhas no cenário (a).

No cenário (b) nenhuma fatia é recebida até que a falha do caminho seja detectada e anunciada, para que a aplicação envie as mensagens através da associação alternativa. Neste cenário é necessário agregar complexidade à aplicação de controle de HA, pois está será responsável por manter o fluxo de mensagens através da associação alternativa.

O experimento compara os dois mecanismos de suporte a *multihoming*, conclui a viabilidade de utilização dos mesmos considerando o SCTP como protocolo de transporte de mensagens de sinalização e demonstra o grau de importância desta característica do SCTP.

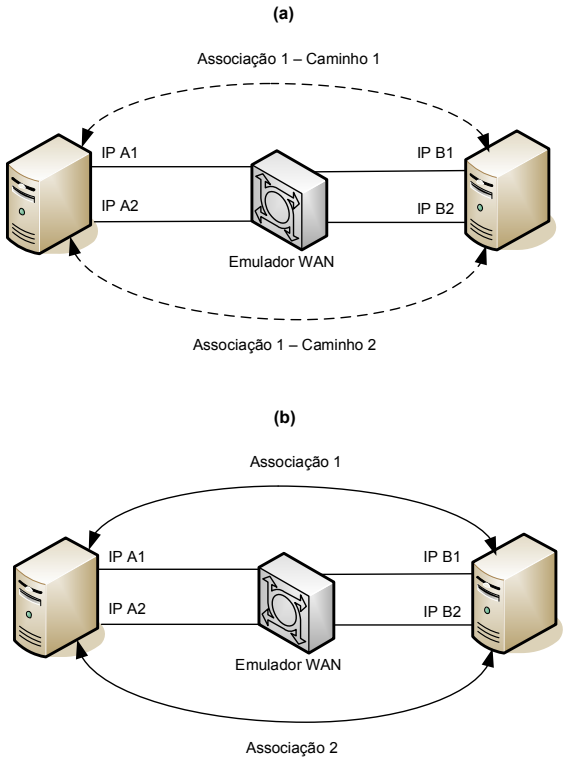

Figura 44 – Cenários *dual-homed* e duas associações

Uma grande vantagem do emprego do SCTP na arquitetura de HA é sua imunidade a ataques DDoS do tipo *SYN Flood*. Este tipo de ataque utiliza geralmente endereços IP de origem falsificados (IP *spoofing*) e o mecanismo de autenticação do SCTP só estabelece a associação após confirmar o endereço de origem (*SYN COOKIES*). Além disso, este mecanismo de autenticação é mais eficiente do que o mecanismo utilizado pelo VRRP,

evitando que ataques contra VRID e de sobreposição de endereços *multicasting* tenham efeito contra o SCTP. Assim, com a utilização da autenticação provida pelo SCTP, não são necessárias camadas de filtro para conter estes tipos de ataques.

A confiabilidade do SCTP é avaliada em [Kiesel 2006] que consiste em um estudo para transportar o protocolo SIMCO (*Simple Middlebox Configuration Protocol*) mensagens de sinalização sobre *multstreaming* através de *firewalls* e em [Camarillo 2003] que compara a transmissão de mensagens de sinalização SIP (*Session Initiation Protocol*) sobre TCP, UDP e SCTP. Este último descreve que a implementação de retransmissão na camada de aplicação através do protocolo UDP pode ser eficaz para pequeno volume de mensagens de sinalização. Entretanto, para grandes volumes de tráfego, por causa dos mecanismos de retransmissão rápida e controle de congestionamento do TCP e SCTP, estes dois caracterizam-se como escolhas melhores do que o UDP.

Em [Pfützenreuter 2007] o desempenho do SCTP é comparado como o dos protocolos UDP, TCP e PR-SCTP. Os testes mostram que o *Linux Kernel* SCTP tem desempenho aceitável na transmissão de dados, embora de forma consistente menor que o desempenho do TCP no mesmo sistema operacional. Este artigo sugere ajustes de configurações no RTO para patamares mais baixos, visando a melhoria de desempenho do SCTP.

Na arquitetura HA proposta, o fatores chave a serem considerados são: confiabilidade; e garantia de entregada das informações. Portanto as evidências dos experimentos listados asseguram esta confiabilidade e desempenho suficiente para a utilização do SCTP como o protocolo de transporte das mensagens HA na arquitetura. Outros aspectos positivos da utilização do SCTP incluem a ampliação do nível de segurança da arquitetura HA e minimização de problemas relacionados com desempenho da rede, uma vez que não há a necessidade de envio desnecessário de cópias de mensagens como no IGMP.

Além disso, a característica de *multihoming* permite que mais de uma interface HA seja utilizada e que o processo de comutação em caso de falha seja transparente para a camada que controla o estado operacional das interfaces HA. A utilização de mais de uma interface HA resolve de forma implícita e transparente o problema do "cérebro partido", pois haverá um caminho alternativo através das interfaces HA alternativas ou em implementações que utilizem as interfaces externas ou internas para a verificação de estado.

A tabela 10 apresenta uma comparação entre a arquitetura proposta e os mecanismos analisados em ambiente de teste.

Tabela 10 – Comparativo HA sobre SCTP com NSRP, SCPS e *SunScreen*

	HA sobre SCTP	**NSRP**	**SCPS**	***SunScreen***
Conectividade a LAN	Alta flexibilidade	Média flexibilidade – com possíveis problemas de desempenho (*broadcast*)	Baixa flexibilidade	Não suporta
Suporte a múltiplas interfaces HA	Suporta	Suporta – através do conceito de *Secondary Path*	Não suporta	Não suporta
Confiabilidade da estrutura HA sobre LAN	Alta	Alta – com emprego do conceito de *Secondary Path*; Baixa – sem o emprego de *Secondary Path*.	Baixa – possibilidade de problema acéfalo com ruptura da conexão HA	Não suporta
Multihoming	Suporta	Não suporta	Não suporta	Não suporta

Portanto, a arquitetura proposta apresenta-se com maior flexibilidade e confiabilidade do que as demais.

5.6 Incrementando a Disponibilidade da LAN

Entre outras considerações, para evitar problemas de desempenho e disponibilidade da própria rede, é recomendável aplicar as seguintes técnicas:

- Separar o tráfego de HA em VLANs específicas, para evitar que haja contaminação de tráfego de *broadcast* de outras redes;
- Assegurar que existe *routing* entre as VLANs caso sejam utilizadas redes diferentes para as interfaces HA e que este mecanismo também possui redundância;
- Aplicar técnicas de priorização de tráfego, políticas de descarte de pacotes e reserva de banda (QoS) para o tráfego de HA nas camadas 2 e 3 da rede, assegurando nível de serviço adequado para este tráfego. Os seguintes modelos de QoS podem ser aplicados:
  - *IntServ* (*Integrated Services*) [Armitage 2000] – neste modelo a aplicação informa a rede suas necessidades de garantia de serviço e, então, ocorre a reserva de banda. Este modelo possui limitações de escala e o principal protocolo utilizado é o RSVP (*Resource Reservation Protocol*) [Armitage 2000, Braden 1997];
  - *DiffServ* (*Differentiated Services*) [Armitage 2000, Nichols 1998] – neste modelo flexível e sem limitações de escala, os equipamentos de rede são configurados para reconhecer as diferentes classes de tráfego e aplicar políticas de QoS previamente definidas;

- Assegurar que a topologia de rede possua caminhos alternativos e equipamentos redundantes para garantir o nível de disponibilidade requerido pelo tráfego HA;
- Aplicar as configurações de segurança relacionadas às melhores práticas em todos os elementos da rede, para controle de acesso e minimizar ameaças relacionadas às vulnerabilidades inerentes aos sistemas operacionais dos equipamentos; e
- Realizar cópia de segurança das configurações (*backup*), armazenando-as em local seguro.

6 CONCLUSÃO

Uma falha afeta a disponibilidade quando esta resulta em interrupções de serviço não planejadas que têm tempo de duração suficiente para criar problemas para os usuários do sistema.

Atualmente, suporte a alta disponibilidade é mais do que um requisito, pois fatores como o crescimento da quantidade de usuários de conexões banda larga no mundo, larga utilização de dispositivos móveis como celulares e PDAs (*Personal Digital Assistants*) para acesso à LANs; e acesso a partir de qualquer lugar (*anywhere*) através de VPNs possibilitam maior interação dos clientes com as aplicações, tornando portanto, a alta disponibilidade fator chave de sucesso para o negócio.

Firewall e IPS são considerados essenciais na proteção do negócio e como elementos constituintes da rede têm que ser considerados no processo de análise de riscos do negócio e, conseqüentemente a adoção de implementações considerando topologias em HA é essencial.

O estudo realizado sobre os protocolos IP Multicast, VRRP, coleta e análise de tráfego HA de *firewalls*, permitiu ao autor um bom entendimento do funcionamento e das vulnerabilidades associadas à utilização destes mecanismos diretamente em LANs.

A análise e entendimento do protocolo SCTP indicou a possibilidade de utilização do mesmo como uma opção de substituição de outros protocolos e conseqüente melhoria dos mecanismos de HA. Assim, foi produzida uma proposta que utiliza o protocolo SCTP como componente principal de uma arquitetura HA para *firewall* e IPS.

A proposta considera as camadas de controle de acesso e aplicação de regras do *firewall*/IPS independentes da camada de controle HA. Também considera o SCTP como protocolo de transporte para a as mensagens de *heartbeat* e informações necessárias ao controle da tabela de estado dos *firewalls*/IPS. Como observado nas análises e testes nas referências citadas, O SCTP se comparado a outros protocolos como TCP e UDP, pode ser considerado como a escolha ideal para prover confiabilidade e conseqüentemente suportar HA.

A proposta, embora não implementada, foi avaliada de forma teórica. Tal avaliação se baseia no estudo de vários artigos e experimentos realizados por outros pesquisadores utilizando o protocolo SCTP.

Certamente, esta proposta pode auxiliar no desenvolvimento futuro de novas soluções para suportar HA entre *firewall*/IPS, outros elementos de rede e no suporte a balanceamento de carga.

Como sugestão para futuros trabalhos é necessário investigar a aplicação da arquitetura proposta para topologias onde existam mais de um *firewall*/IPS operando como *Backups*. Também é necessário investigar a aplicação de criptografia em conjunto com SCTP [Bellovin 2003] para proteger as informações de HA transferidas na rede, para evitar quebra da confidencialidade.

Finalmente, outra linha de trabalho, já em desenvolvimento e com a participação do autor desta dissertação consiste na implementação da arquitetura proposta utilizando a estrutura da arquitetura do *Keepalived daemon* como base, porém utilizando o protocolo SCTP como verificador de estado funcional dos componentes da topologia HA.

REFERÊNCIAS BIBLIOGRÁFICAS

Allman, M. (2003), "TCP Congestion Control with Appropriate Byte Counting (ABC)", RFC 3465.

Armitage, G. (2000), "Quality of Service in IP Networks: Foundations for a Multi-Service Internet", Macmillan Technical Publishing.

Bellovin, S., Ionnides, J., Keromytis, A., and Stewart, R., (2003), "On the Use of Stream Control Transmission Protocol (SCTP) with IPSec", RFC 3554.

Branden, R., Zhang, L., Berson, S., Herzog, S., and Jamin, S. (1997), "Resource Reservation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205.

British Standards Institution-BSI (2006), "Business Continuity Planning (BCP)", BS 25999.

Cain, B., Deering, S., Kouvelas, I., Fenner, B., and Thyagarajan, A. (2002), "Internet Group Management Protocol, Version 3", RFC 3376.

Camarillo, G., Kantola, R., and Schulzrinne, H. (2003), "Evaluation of transport protocols for the session initiation protocol", IEEE Network 17 (5) pp. 40-46.

Cameron, R., Woodberg, B., Madwachar, K., M., Swarm, M., Wyler, N. R., Albers, M., and Bonnell, R. (2007), "Configuring Juniper Networks NetScreen & SSG Firewalls", Syngress Publishing, Inc.

Cartlidge, A., Hanna, A., Rudd, C., Macfarlane I., Windbanb J., and Rance, S. (2007), "An Introductory Overview of ITIL® V3", The UK Chapter of the itSMF, acesso em 20/01/2008 20:00 (-03GMT), http://www.itsmf.com/upload/bookstore/itSMF_ITILV3_Intro_Overview.pdf.

Cisco Systems (2008), "Catalyst Switched Port Analyzer (SPAN) Configuration Example", acesso em 20/01/2008 18:25 (-03GMT), http://www.cisco.com/en/US/products/hw/switches/ps708/products_tech_note09186a008015c612.shtml.

DARPA (1981), "Transmission Control Protocol DARPA Internet Program Protocol Specification", RFC 793.

DARPA (1981), "Internet Protocol – IP", RFC 791.

Deering, S. (1989), "Host Extensions for IP Multicasting", RFC 1112.

Doraswamy, N., and Harkins, D. (1999), "IPSec – The New Security Standard for Internet, Intranets and Virtual Private Networks", Prentice Hall.

Eastlake, D., and Jones, P. (2001), "US Secure Hash Algothm 1 – SHA1", RFC 3174.

Egevang, K., and Francis, P. (1994) "The IP Network Address Translator (NAT)", RFC 1631.

Eletronic Industries alliance – EIA (1969), "RS-232 standard", RS-232.

Fenner, B., Handley, M., Holbrook, H., and Kouvelas, I. (2006), "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601.

Ferguson, P., and Senie, D. (2000), "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", RFC 2827.

Fielding, J., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. (1999), "Hypertext Transfer Protocol HTTP/1.1", RFC 2616.

Floyd, S. (2000), "Congestion Control Principles", RFC 2914.

Grinnemo, K.-J., Anderson, T., and Brunstrom, A. (2005), "Performance benefits of avoiding head-of-line blocking in SCTP", Autonomic and Autonomous Systems and International

Conference on Networking and Services, ICAS-ICNS, Volume, Issue, 23-28 Oct. 2005 pp. 44-44.

Hansche, S., Berti, J., and Hare, C. (2004), "Official (ISC)² Guide to the CISSP Exam", AUERBACH PUBLICATIONS, pp. 617-627.

Harrington, D., Presuhn, R., and Wijnen, B. (1999), "An Architecture for Describing SNMP Management Frameworks", RFC 2571.

Hawkinson, J., and Bates, T. (1996), "Guidelines for creation, selection, and registration of an Autonomous System (AS)", RFC 1930.

Higginson, P., and Shand, M. (1997), "Development of Router Clusters to Provide Fast Failover in IP Networks", Digital Technical Journal, vol. 9 no. 3, Inverno 1997.

Hinden, R. (2004) "Virtual Router Redundancy Protocol (VRRP)", RFC 3768.

IEEE 802 Local Area Networks – LANs (2006), "802.1Q - Virtual LANs", ISO/IEC 15802-3.

International Organization for Standardization – ISO (1979), "Open Systems Interconnection Basic Reference Model – OSI", ISO 7498.

ISACA (Information Systems Audit and Control Association) (2008), "COBIT 4.1 Executive Summary and Framework", acesso em 14/01/2008 21:00 (-03GMT), http://www.isaca.org/AMTemplate.cfm?Section=Downloads&Template=/ContentManagement/ContentDisplay.cfm&ContentID=34172.

ITGI (IT Governance Institute) (2008), "About IT Governance", acesso em 14/01/2008 21:00 (-03GMT), http://www.itgi.org/template_ITGI.cfm?Section=About_IT_Governance1&Template=/ContentManagement/HTMLDisplay.cfm&ContentID=19657.

Jungmaier, A., Rathgeb, E. P., and Tüxen, M. (2002), "On the Use of SCTP in Failover-Scenarios", Proceedings of the SCI 2002, Mobile/Wireless Computing and Communication Systems II, VOL. X, Orlando, USA, pp. 363-368, July 2002.

Keepalived (2003), "HealthChecking for LVS & High Availability", acesso em 20/01/2008 18:20 (-03GMT), http://www.keepalived.org/.

Khan H., and Naseh Z. (2006), "Designing Content Switching Solutions", Cisco Press, Chapter 5: Firewall Load Balancing, acesso em 12/02/2008 18:00 (-03GMT), http://www.networkworld.com/subnets/cisco/082807-design-content-switching.html.

Kiesel, S., and Scharf, M. (2006), "Modeling and Performance Evaluation of SCTP as Transport Protocol for Firewall Control", Lecture Notes in Computer Science LCNS, Volume 3976/2006, pp. 451-462.

Klensin, J. (2001), "Simple Mail Transfer Protocol", RFC 2821 (Obsoletes: 821, 974, 1869).

Knight, S., weaver, D., Whipple, D., Hinden, R., Mitzel, D., Hunt, P., Higginson, P., Shand, M., and Lindem, A. (1998), "Virtual Router Redundancy Protocol", RFC 2338 (Obsolete).

Krawczyk, H., Bellare, M., and Canetti R. (1997), "HMAC: Keyed-Hashing for Message Authentication", RFC 2104.

Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D., and Jones, L. (1996), "SOCKS Protocol Version 5", RFC 1928.

Li, T., Cole, B., Morton, P., and Li, D. (1998), "Cisco Hot Standby Router Protocol (HSRP)", RFC 2281.

Light Reading, "Deep Packet Inspection", (December 2006), acesso em 14/05/2007 21:00 (-03GMT), http://www.lightreading.com/document.asp?doc_id=111404.

Matcalfe, R., Boggs, D., Thacker C., and Lampson, B., Xerox (1975), “Multipoint Data Communication System with Collision Detection”, U.S. patent 4,063,220.

Mathis, M., Mahdavi, J., Floyd, S., and Romanow, A. (1996), “TCP Selective Acknowledgment Options”, RFC 2018.

Meyer, D. (1998), “Administratively Scoped IP Multicast”, RFC 2365.

Meyer, D., and Lothberg, P. (2001), “GLOP Addressing in 233/8”, RFC 3180.

Mills, D. (2006), “Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI”, RFC 4330.

Mockapetris, P. (1987), “Domain Names – Implentation and Specification”, RFC 1035.

Mogul, J., and Deering, S. (1990), “Path MTU Discovery”, RFC 1191.

Moy, J. (1998), “OSPF Version 2”, RFC 2328.

Nichols, K., Blake, S., Baker, F., and Black, D. (1998), “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, RFC 2474, (Obsoletes: 1455, 1349).

Ong, L., et al. (1999), “Framework Architecture for Signaling Transport”, RFC 2719.

Ouchi, N. K., IBM (1978), "System for recovering data stored in failed memory unit.", U.S. patent 4,092,732.

Patterson, D. A., Gibson, G. A., and Katz, R. (1987), "A Case for Redundant Arrays of Inexpensive Disks (RAID)", SIGMOD conference (Junho 1988).

Pfützenreuter, E., and Friedrich, L., F. (2007), “Avaliação de Desempenho do Protocolo SCTP no Linux”, IEEE Latin America Transactions, vol. 5, no. 2, Junho 2007.

Plato, A., NetworkICE Corporation (1998), “BlackICE Guard – User Guide”.

Plummer, D. (1982), “An Ethernet Address Resolution Protocol - ARP”, RFC 826.

Postel, J. (1980), “User Datagram Protocol - UDP”, RFC 768.

Postel, J. (1985), “File Transfer Protocol - FTP”, RFC 959.

Ramakrishnan, K., Floyd, S., and Black, D. (2001), “The Addition of Explicit Congestion Notification (ECN) to IP”, RFC 3168.

Rane, J., kumbhar, N., Sovani, K., Ingle, R. and Kini, A. (2003),"Exploiting Multi homing in SCTP for High Performance", in ADCOM 2003, Coimbotor, India.

Rivest, R. (1992), “The MD5 Message-Digest Algorithm”, RFC 1321.

Romanow, A., Mogul, J., Talpey, T., and Bailey, S. (2005), “Remote Direct Memory Access (RDMA) over IP Problem Statement”, RFC 4297.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E. (2002), "SIP: Session Initiation Protocol", RFC 3261.

Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka, M., and Zeidner, E. (2004), “Internet Small Computer Systems Interface (iSCSI)”, RFC 3720.

SCPS (1997), “Space Communications Protocol Standards (SCPS)”, acesso em 14/05/2007 21:00 (-03GMT), http://www.scps.org.

Screen Play, “Deep-Packet Inspection Blossoms as Enabler of Advanded Services”, (novembro 2006), 14/05/2007 20:00 (-03GMT), http://www.screenplaysmag.com/Editor/Article/tabid/96/articleType/ArchiveView/month/11/year/2006/Default.aspx.

Semeria, C. (2001), “RFC 2547bis: BGP/MPLS VPN Fundamentals”, Juniper Networks Inc, Part Number: 200012-001 03/01, acesso em 15/01/2008 18:00 (-03GMT), http://mia.ece.uic.edu/~papers/WWW/Flexi-Tunes/segment/mpls_bgp_VPN/200012.pdf.

Stevens, W. R. (2000), "TCP/IP Illustrated Volume 1 – The Protocols", Addison-Wesley Professional Computing Series.

Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and Conrad, P. (2004), "SCTP Partial Reliability Extension", RFC 3758.

Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and Paxson, V. (2000), "SCTP: Stream Control Transmission Protocol", RFC 2960.

Stone, J., Stewart, R., and Otis, D. (2002), "Stream Control Transmission Protocol (SCTP) Checksum Change", RFC 3309.

Sun Microsystems. (2001), "SunScreen 3.2 Administration Guide", part-number 806-6346 (September 2001), acesso em 19/01/2008 17:00 (-03GMT), http://docs.sun.com/app/docs/doc/806-6348/6jfa1eop8?l=ru&a=view.

Wireshark (2008), "Wireshark Software", acesso em 20/01/2008 18:30 (-03GMT), http://www.wireshark.org/.

Ylonen, T., and Lonvick, C. (2006), "The Secure Shell (SSH) Protocol Architecture", RFC 4251.