UMA ABORDAGEM DE GERENCIAMENTO DE REDES BASEADO NO MONITORAMENTO DE FLUXOS DE TRÁFEGO NETFLOW COM O SUPORTE DE TÉCNICAS DE BUSINESS INTELLIGENCE

ANDRÉ VALENTE DO COUTO

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

UMA ABORDAGEM DE GERENCIAMENTO DE REDES BASEADO NO MONITORAMENTO DE FLUXOS DE TRÁFEGO NETFLOW COM O SUPORTE DE TÉCNICAS DE BUSINESS INTELLIGENCE

ANDRÉ VALENTE DO COUTO

ORIENTADOR: DR. RAFAEL TIMÓTEO DE SOUSA JÚNIOR

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGENE.DM – 107/2012

BRASÍLIA / DF: MAIO/2012

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

UMA ABORDAGEM DE GERENCIAMENTO DE REDES BASEADO NO MONITORAMENTO DE FLUXOS DE TRÁFEGO NETFLOW COM O SUPORTE DE TÉCNICAS DE BUSINESS INTELLIGENCE

ANDRÉ VALENTE DO COUTO

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:

RAFAEL TIMÓTEO DE SOUSA JÚNIOR, Doutor, UnB
(ORIENTADOR)

FLAVIO ELIAS GOMES DE DEUS, Doutor, UnB
(EXAMINADOR INTERNO)

Ed'WILSON TAVARES FERREIRA, Doutor, IFMT
(EXAMINADOR EXTERNO)

DATA: BRASÍLIA/DF, 23 DE MAIO DE 2012.

FICHA CATALOGRÁFICA

COUTO, ANDRÉ VALENTE DO
Uma abordagem de Gerenciamento de Redes baseado no Monitoramento de Fluxos de Tráfego Netflow com o suporte de Técnicas de Business Intelligence [Distrito Federal] 2012.
xiv, 116 p., 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2012).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. .Gerenciamento de Redes de Computadores 2. Inteligência computacional 3. Netflow

I. ENE/FT/UnB. II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

COUTO, ANDRÉ VALENTE DO. (2012). Uma abordagem de Gerenciamento de Redes baseado no Monitoramento de Fluxos de Tráfego Netflow com o suporte de Técnicas de Business Intelligence. Dissertação de Mestrado, Publicação PPGENE.DM – 107/2012, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 116 .

CESSÃO DE DIREITOS

NOME DO AUTOR: ANDRÉ VALENTE DO COUTO
TÍTULO DA DISSERTAÇÃO: UMA ABORDAGEM DE GERENCIAMENTO DE REDES BASEADO NO MONITORAMENTO DE FLUXOS DE TRÁFEGO NETFLOW COM O SUPORTE DE TÉCNICAS DE BUSINESS INTELLIGENCE.
GRAU/ANO: Mestre/2012.

ANDRÉ VALENTE DO COUTO
Universidade de Brasília – Faculdade de Tecnologia – Departamento de Engenharia Elétrica
CEP 70910-900 – Brasília – DF – Brasil

Dedico este trabalho à minha esposa Helaine Bucair e ao meu filho Daniel Bucair Couto.

AGRADECIMENTOS

Aos professores: Ali Veggi Atala, Tony Inácio da Silva e Fabiano João Leoncio de Pádua, por não medirem esforços para a realização deste programa de Mestrado Interinstitucional aqui no IFMT - Campus Cuiabá.

À UNB e a todos os professores que compuseram o corpo docente deste programa de capacitação em nível de Mestrado, na pessoa do prof. Dr. Franklin da Costa Silva.

Ao meu orientador prof. Dr. Rafael Timóteo de Souza Júnior, por permitir meu aprimoramento profissional pelo desenvolvimento de um tema correlato ao meu perfil e por acreditar na minha capacidade.

Ao prof. Dr. Joaquim de Oliveira Barbosa pelas inestimáveis aulas de balizamento e preparação a que se dispôs, nos sábados, domingos e feriados.

Ao prof. Dr. Ed'Wilson Tavares Ferreira, pela revisão atenta e criteriosa.

Aos colegas de capacitação, na pessoa do amigo Gérson Kida, que comigo partilharam muito mais que apenas conhecimentos, mas que foram generosos e comprometidos com a família que ali se formou em busca de um objetivo único.

Aos colegas do DAI/IFMT, pela constante motivação à conclusão do trabalho na pessoa dos meus irmãos, Gastão, Marilson e Reginaldo Hugo.

Aos colegas de trabalho, na Casa Civil do Estado de Mato Grosso, em especial ao meu amigo, Gabriel Mendes Piloni.

A todos que direta ou indiretamente contribuíram para a realização deste trabalho e que hipotecaram apoio ao meu sucesso, agradeço com um triplo e fraternal abraço.

"A gratidão desbloqueia a abundância da vida. Ela torna o que temos em suficiente, e mais. Ela torna a negação em aceitação, caos em ordem, confusão em claridade. Ela pode transformar uma refeição em um banquete, uma casa em um lar, um estranho em um amigo. A gratidão dá sentido ao nosso passado, traz paz para o hoje, e cria uma visão para o amanhã."

Melody Beattie

RESUMO

UMA ABORDAGEM DE GERENCIAMENTO DE REDES BASEADO NO MONITORAMENTO DE FLUXOS DE TRÁFEGO NETFLOW COM O SUPORTE DE TÉCNICAS DE BUSINESS INTELLIGENCE.

Autor: ANDRÉ VALENTE DO COUTO

Orientador: Dr. RAFAEL TIMÓTEO DE SOUSA JÚNIOR

Programa de Pós-graduação em Engenharia Elétrica

Brasília, Maio de 2012

As redes de computadores atualmente possuem papel convergente de tecnologias que necessitam de prévio acordo de serviço com os clientes para garantia de sua disponibilidade e desempenho. O gerenciamento dos recursos computacionais passa a ter um papel vital neste processo, pois habilita ao administrador a capacidade de antever de forma proativa a manutenção dos níveis de acordo definidos. Este trabalho apresenta um estudo sobre as abordagens de monitoramento passivo de redes de computadores e as tecnologias aplicadas aos sistemas de suporte à decisão, culminando com o desenvolvimento de uma proposta de gerenciamento de tráfego em um *backbone* de rede local através do monitoramento dos fluxos *netflow* integrado em uma solução livre de *business intelligence* pela implementação de um *Datamart* para o fornecimento de consultas *OLAP*.

ABSTRACT

AN APPROACH TO NETWORK MANAGEMENT BASED ON MONITORING OF TRAFFIC FLOWS NETFLOW WITH TECHNICAL SUPPORT FOR BUSINESS INTELLIGENCE.

Author: ANDRÉ VALENTE DO COUTO

Supervisor: RAFAEL TIMÓTEO DE SOUSA JÚNIOR

Programa de Pós-graduação em Engenharia Elétrica

Brasília, May of 2012

Computer networks today have role converging technologies that require prior service agreement with customers to ensure their availability and performance. The management of computational resources is replaced by a vital role in this process because it enables an administrator the ability to proactively anticipate the maintenance of defined levels of agreement. This paper presents a study of approaches to passive monitoring of computer networks and the technologies applied to decision support systems, culminating in the development of a proposed traffic management into a backbone LAN by monitoring flows netflow integrated in a free solution for business intelligence for implementing a datamart for the supply of OLAP queries.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIAÇÕES

API	Interface de programação de aplicativos
ARP	Protocolo de resolução de endereços
ASN.1	Notação de sintaxe abstrata
BER	Regras de codificação básica
BI	Inteligência de negócios
DES	Algoritmo padrão para criptografia de dados
DM	Data mart – repositório de dados setorial
DSS	Sistemas de suporte a decisão
DW	Data warehouse – repositório de dados
HUB	Equipamento que regenera os sinais e retransmite (barramento)
IANA	Autoridade atribuidora de números de internet
IETF	Força tarefa de engenheiros da internet
IPFIX	Protocolo de exportação de fluxos IP
KDD	Descoberta de conhecimentos em banco de dados
MAC	Endereço físico de identificação de interfaces
MIB	Base de informações de gerenciamento
OID	Identificador de objetos
OLAP	Processamento analítico online
OLTP	Processamento de transações online
OSI	Padrão Internacional aberto
PDU	Protocolo unitário de dados
PSAMP	Amostragem de pacotes
RFC	Documento que descreve padrões de cada protocolo da internet
RMON	Base de dados para o monitoramento remote de redes de computadores
SCBF	Filtro de inspeção baseado em código de espaço (Space-Code Bloom Filter)
SMI	Estrutura das informações de gerenciamento
SNMP	Protocolo simples de gerenciamento de redes de computadores
UDP	Protocolo de datagrama de usuários.

1. INTRODUÇÃO

Nos tempos atuais, os recursos tecnológicos devem ser considerados ferramentas estratégicas que possibilitam um diferencial produtivo. Como tal, devem ser otimizadas para garantir o alcance dos objetivos planejados.

Neste intento, pode-se considerar toda a gama de recursos disponibilizada pela tecnologia da informação como parte deste escopo. Enquanto partícipe deste processo, a garantia da boa utilização dos recursos alinhada aos objetivos corporativos passa obrigatoriamente pelo gerenciamento do ambiente.

Corroborando com o supra exposto, William Edwards Deming, popularmente conhecido como o guru do gerenciamento da qualidade, afirma que: "Não se gerencia o que não se mede, não se mede o que não se define, não se define o que não se entende, não há sucesso no que não se gerencia".

O gerenciamento é a peça que permite o realinhamento de condutas e processos no sentido de tornar as ações eficientes e eficazes. Não obstante esse processo não é uma atividade trivial, pois perpassa diversos domínios de estudo além de poder, em função de sua implementação, interferir nos processos e rotinas de negócios.

Existem diversos modelos de gerenciamento de redes de computadores, dentre as quais, esta dissertação se propõe a abordar os modelos passivos, que não interferem nos processos e rotinas de negócios e que pode ser amplamente adotado como parte do processo de gerenciamento de redes locais.

Desses, serão elencados: o modelo clássico baseado em agentes e gerentes *SNMP*, o modelo de monitoramento baseado na captura de pacotes e o modelo baseado em fluxos de pacotes, através do uso dos diversos protocolos, dentre eles: o protocolo aberto *IPFIX*, o protocolo de amostragem *sFlow* e o protocolo proprietário da Cisco, *Netflow*. Este último sendo o protocolo adotado para a aplicabilidade neste estudo.

Ainda, para garantir a flexibilidade ao processo de gestão e tomada de decisões, foram aplicadas as técnicas de *Business Intelligence* no intuito de garantir a escalabilidade do modelo proposto bem como suportar possíveis variabilidades relativas ao protocolo de monitoramento de fluxos *NetFlow*.

Neste caso, os estudos desta dissertação também abordarão o processo de tomada de decisões através do estudo da disciplina dos Sistemas de suporte à decisão – DSS, passando pela definição da arquitetura e componentes dos sistemas de Business Intelligence e finalizando na implantação em uma plataforma livre, as técnicas de BI.

1.1. MOTIVAÇÃO

A necessidade de em uma organização pública, com a premissa da utilização de *softwares* livres, realizar a contabilização de uso dos tráfegos de dados integrados a uma plataforma que permita o gerenciamento das informações e geração de conhecimentos.

Este estudo objetiva contribuir com a implantação de uma solução livre de gerenciamento passivo de redes, baseado no monitoramento de fluxos e análises analíticas suportadas por técnicas de Business Intelligence. Como objetivos secundários, está a verificação e validade da aplicação considerando às necessidades de gerenciamento.

Como objetivos específicos, tem-se o estudo do marco teórico:

- das técnicas de gerenciamento passivo de redes, tais como: SNMP, monitoramento baseado em pacotes e o monitoramento baseado em fluxos;
- das técnicas aplicadas aos Sistemas de Suporte a Decisão – DSS;
- da arquitetura dos sistemas de Business Intelligence – BI, em especial das implementações de Data Warehouse por meio de Data Marts;
- estudo comparativo das principais plataformas livres de BI com o aprofundamento na plataforma Pentaho BI Suite Community.

1.2. JUSTIFICATIVA

As atuais ferramentas livres de gestão de fluxos Netflow contemplam a análise apenas por meio de relatórios e gráficos estáticos, baseados na origem da coleta. Não permitem, portanto a flexibilidade no processo de descoberta de conhecimentos por meio de consultas analíticas.

Este estudo pretende abordar por meio de uma solução livre, uma proposta que seja altamente aplicável ao processo de gerenciamento passivo de redes de computadores que permita a análise analítica dos dados e que possua duas características intrínsecas no seu processo de desenvolvimento: flexibilidade e escalabilidade. A flexibilidade se refere ao fato

da solução poder ser alterada e incrementada com novas funcionalidades sem que seja necessária reengenharia da solução e a escalabilidade se constitui na característica da solução estar preparada para manipular uma porção crescente de trabalho.

Considerando as inúmeras propriedades advindas dos elementos que compõem esta solução de gerenciamento, esta pesquisa de dissertação têm como premissa uma definição mais generalista em sua proposta de implementação, fato que decorre tanto no desenvolvimento da origem (agentes e coletas) quanto no destino dos fluxos de processos estudados (estrutura de análise dos dados).

Especificamente, na coleta dos dados foi definida a utilização do protocolo Netflow em sua versão 5 considerando sua expressiva utilização e sua programação fixa de campos e no suporte ao processo de tomada de decisões, as análises foram concentradas em modelos olap para consultas analíticas auxiliadas por alguns relatórios parametrizados.

1.3. METODOLOGIA DE TRABALHO

Inicialmente partiu-se para uma pesquisa exploratória sobre as técnicas de gerenciamento de redes cujas características mais se acoplavam às necessidades de gestão que motivaram o trabalho. Definido a estratégia de gerenciamento, foi realizado estudo minucioso e comparativo das ferramentas livres disponíveis para implementação da abordagem escolhida ao qual culminou com a determinação da utilização de uma abordagem flexível para disponibilização das informações.

Desenvolveu-se a uma pesquisa exploratória e comparativa sobre as plataformas livres que disponibilizam as funcionalidades de BI, com a meta de verificar as técnicas necessárias para ampliar a capacidade de produzir conhecimento, por meio do uso dos sistemas de suporte a decisão.

Foi realizada uma pesquisa aplicada com o trabalho de integração das tecnologias de gerência e suporte a decisão com a produção de uma solução livre para o monitoramento de fluxos de tráfego.

Por fim, foi verificado e validado qualitativamente, por meio das análises analíticas, os resultados gerados pela solução.

1.4. ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação possui em seu primeiro capítulo o estabelecimento dos objetivos, a definição de seu escopo, bem como a explanação da metodologia e a organização do trabalho.

No capítulo 2, Monitoramento e Gerência de Redes, será realizado o marco teórico das técnicas de gerenciamento passivo de redes de computadores, tendo como foco: o modelo clássico de gerência pelo protocolo SNMP; a coleta e o monitoramento passivo de pacotes; e o monitoramento de redes baseado em fluxos aonde serão abordadas as principais versões do protocolo Netflow, o protocolo IPFix e o protocolo de amostragem sFlow.

No capítulo 3, Sistemas e tecnologias de suporte à decisão, será tratada a fundamentação teórica acerca dos processos de tomada de decisões, bem como das técnicas utilizadas para o seu auxílio, sob o enfoque nas técnicas de BI e o processo de descoberta de conhecimento em bases de dados (mineração de dados). Ainda, neste capítulo, serão explanadas as principais características das ferramentas livres BI, com ênfase na ferramenta Pentaho Community, que é a ferramenta adotada nos trabalhos desta dissertação.

No capítulo 4, Monitoramento de fluxos suportados por técnicas de BI, é apresentado e desenvolvido todo o modelo da proposta deste trabalho de dissertação.

No capítulo 5, são feitas as conclusões sobre o estudo realizado bem como apontamentos de trabalhos futuros.

2. MONITORAMENTO E GERÊNCIA DE REDE

Este capítulo 2 tem como objetivo apresentar o marco teórico sobre as alternativas para o gerenciamento passivo de redes abordando as principais metodologias e protocolos, dentre as quais, destaca-se o protocolo SNMP, o monitoramento baseado em captura de pacotes por meio das ferramentas *tcpdump* e *wireshark* e o monitoramento baseado em fluxos de tráfego, objeto base deste estudo, aonde será abordado o conceitos que permeiam sua arquitetura de gerenciamento bem como seus principais protocolos: *NetFlow, SFlow e IPFIX*.

2.1. GERÊNCIA DE REDES

Com o surgimento da intercomunicação digital, o esforço para a produção de um software *middleware* entre os diversos fornecedores se tornou um processo muito difícil, considerando os diferentes protocolos de intercâmbio e formatos de dados.

Reconhecendo esta situação, a Organização Internacional de Normalização, desenvolveu uma arquitetura de comunicações denominada (OSI, Information Technology, 1989), para a definição das interconexões, bem como o modelo de gerenciamento de redes que define padrões para as trocas de informações e coordenação entre os recursos, provendo mecanismos para o monitoramento e controle.

Figuram neste cenário de padronização, cinco áreas funcionais relativas à gerência de redes, sendo: Gerências de Falhas, Gerência de Contabilização, Gerência de Configuração, Gerência de Desempenho e Gerência de Segurança.

- Gestão de Falhas: tem como características a determinação da ocorrência da falha; garantir a continuidade da disponibilidade pelo isolamento da ocorrência da falha; reconfiguração das operações de modo a minimizar o impacto da falha no sistema como um todo; e reparar e substituir em caso de necessidade a fim de repor o estado original da rede. Uma falha é uma condição anormal que requer atenção e/ou intervenção, enquanto um erro é apenas um evento isolado, no entanto uma falha pode ser indicada por erros excessivos.
- Gestão de Contabilização: visa como elemento primário à contabilização dos recursos da rede para o controle do uso por parte dos usuários e grupos de usuários. Este domínio atua da detecção de limiares de usos para garantia e

controle dos privilégios através da contabilização que habilitando também a taxação e o consecutivo planejamento da infraestrutura.

- Gestão de Configuração: é a área que permite o controle do comportamento de cada dispositivo de rede através de sua configuração. É também a área que cuida da manutenção, adição e atualização das relações entre os componentes e do estado dos próprios componentes durante a operação da rede.
- Gestão do Desempenho: consiste na monitoração das atividades da rede e no controle dos recursos através de ajustes e trocas. Seu trabalho está associado à composição das métricas e valores apropriados aos recursos de rede para o fornecimento de indicadores dos diferentes níveis de desempenho que serão os insumos no planejamento, administração e manutenção das redes.
- Gestão de Segurança: é a área aonde são tratados os processos de autenticação, autorização e definição das políticas de segurança para a garantia da proteção aos recursos da rede. Seus processos e métodos se envolvem nas outras áreas de gestão integrando os domínios na obtenção de da garantia da confidencialidade, integridade e disponibilidade dos recursos.

Segundo (GOMES, 2008), além das áreas expostas acima, a gerência de redes também se distingue em duas categorias funcionais: monitoramento e controle. O monitoramento é a função que faz o acompanhamento de todas as atividades da rede no sentido de contabilizá-los, e o controle é a função que permite que os ajustes sejam feitos visando melhorar o desempenho da rede.

Segundo (LOGOCKI, 2008), o monitoramento enquanto ação de gerenciamento de rede pode ser classificada como: ativo e passivo.

O monitoramento passivo analisa os pacotes que estão trafegando na rede sem interferir no fluxo de pacotes e consequentemente no desempenho da rede, produzindo um substancial volume de dados coletados que será a base pelo qual a análise se procederá. O grande desafio dessa abordagem é conseguir restringir a quantidade de dados coletados conseguindo armazenar somente as informações necessárias.

O monitoramento ativo é executado pela inserção e análise de pacotes de teste na rede. Nesta técnica a quantidade de dados armazenados não é considerável, em contrapartida, o desafio é adequar o volume de pacotes inseridos para sucessão dos testes e atingimento das métricas desejadas sem que se interfira no desempenho e alocação excessiva de recursos.

A monitoração da rede consiste em observar as informações relevantes ao gerenciamento. Estas informações podem ser classificadas em três categorias (SPECIALSKI, 2000):

- Estática: caracteriza a configuração atual e os elementos na atual configuração, tais como o número e identificação de portas em um roteador.
- Dinâmica: relacionada com os eventos na rede, tais como a transmissão de um pacote na rede.
- Estatística: pode ser derivada de informações dinâmicas; ex. média de pacotes transmitidos por unidade de tempo em um determinado sistema.

Para obter as informações de gerenciamento, os administradores de redes são auxiliados por um sistema de gerência de redes, que por sua vez, pode ser definido como uma coleção de ferramentas integradas (STALLINGS, 1998).

Segundo (LOPES, 2003), a arquitetura geral dos sistemas de gerência de redes apresentam quatro componentes básicos:

- Elementos Gerenciados: são denominados agentes e são implementados por um software que permite que o monitoramento e controle do equipamento.
- Estações de Gerência: interage diretamente com os agentes para monitorá-los e gerenciá-los.
- Protocolo de Gerência: é a interface entre a estação de gerência e o agente por meio de uma normatização das operações de monitoramento (leitura) e controle (escrita).
- Informações de Gerência: são os dados que podem ser referenciados em operações do protocolo de gerência, que podem ser: estáticos, dinâmicos e estatísticos, conforme descrito anteriormente.

A informação de gerenciamento é coletada e armazenada por agentes e repassada para um ou mais gerentes. Duas técnicas podem ser utilizadas na comunicação entre agentes e gerentes: *polling* e event-reporting (SPECIALSKI, 2000).

A técnica de *polling* consiste em uma interação do tipo solicitação/respostas entre um gerente e um agente. O gerente pode solicitar a um agente (para o qual ele tenha autorização),

o envio de valores de diversos elementos de informação. O agente responde com os valores constantes em sua base de dados de informações *MIB.*

Na técnica de *event-reporting*, a iniciativa é do agente. O gerente fica na escuta, esperando pela chegada de informações. Um agente pode gerar um relatório periodicamente para fornecer ao gerente o seu estado atual. A periodicidade do relatório pode ser configurada previamente pelo gerente. Um agente também pode enviar um relatório quando ocorre um evento significativo ou não usual.

Tanto o *polling* quanto o *event-reporting* são usados nos sistemas de gerenciamento, porém a ênfase dada a cada um dos métodos difere muito entre os sistemas. Segundo (CALYAM, 2005), enquanto o modelo SNMP se baseia em processos de polling entre agentes e gerentes para acesso às informações o gerenciamento por fluxo de tráfego de dados se baseia inteiramente no conceito de *event-reporting*, equanto que o modelo *OSI* fica entre estes dois extremos.

2.2. O MODELO SNMP

O IETF através das RFC normatiza os protocolos padrões que governam a internet, dentre estes o SNMP, que possui suas definições em (ROSE, 1990).

Segundo (MILLER, 1999), o núcleo do SNMP é um simples conjunto de operações que permitem ao administrador a capacidade de monitorar e controlar o estado de equipamentos, softwares e redes de computadores.

2.2.1. Arquitetura SNMP

A arquitetura do SNMP é composta pela interface entre o agente e o gerente por meio de um protocolo para acesso à base de dados de informações.

Segundo (ZELTSERMAN, 1999), os elementos gerenciados possuem um agente SNMP e uma base de dados denominada MIB (*Management Information Base*), que contém informações de gerenciamento que refletem sua configuração e o seu comportamento, bem como parâmetros que podem ser usados para controlar suas operações. Portanto, um agente SNMP pode receber solicitações de leitura e gravação de dados na MIB, bem como pode gerar um alerta para o gerente SNMP no caso de algum parâmetro alcançar algum limiar determinado.

As MIB's, segundo (PERKINS, 1997), possuem tratamento e modelagem de informação de gerenciamento normalmente baseada em objetos. Essa abstração permite a modelagem de objetos que representarão a realidade de um acontecimento. Uma instância individual de um objeto gerenciado é uma variável da MIB.

A SMI é a arquitetura que especifica como são definidos os módulos da MIB e o formato dos identificadores de objetos na árvore MIB. A nova versão do SMIv2 definida em (MCCLOGHRIE K. e., 1999), incorporou as características da versão anterior e trouxe consigo a definição de novos tipo de dados: *Counter32*, *Counter 64*, *Bit String*, dentre outros.

Segundo (DING & HU, 2011) a composição das especificações SMI faz uso de um subconjunto da notação de sintaxe de abstração ASN.1 usado no SNMP. Enquanto que a ASN.1 fornece a sintaxe que compõem a estrutura SMI que define os objetos de dados gerenciados MIB's, o BER define as especificações de encapsulamento para transmissão das informações independente da plataforma (OSI, Information Technology, 2008).

Portanto as definições dos objetos gerenciados podem ser divididas em três atributos: Nome ou OID (*identificador de objeto*); Tipo ou Sintaxe, que é definição ASN.1 que especifica o tipo do dado; e as definição BER que definem como o objeto será codificado para a transmissão.

Na figura 2.1, pode-se perceber a árvore da estrutura da SMIv2 que dá acesso à MIB-2, *iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1)* ou simplesmente *1.3.6.1.2.1*. Os dados na última camada representam os grupos de objetos tais como: *system(1)* ou interfaces*(2)*.

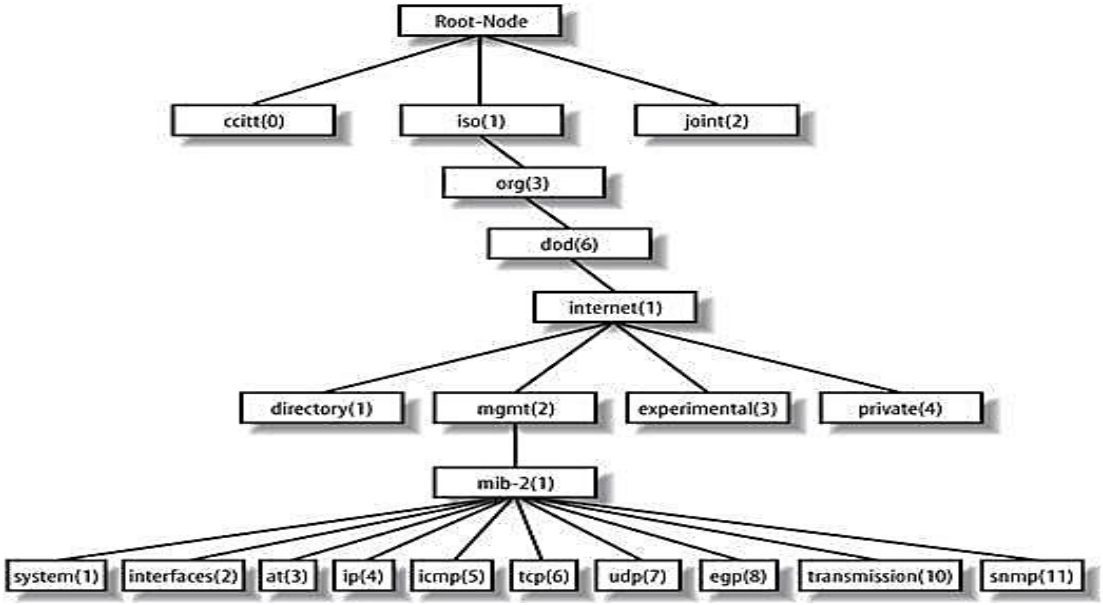


Figura 2.1 – Estrutura de objetos da MIB-2 (DOUGLAS, 2005).

A tabela 2.1, apresenta a identificação e descrição dos principais objetos da MIB-2 utilizados para o gerenciamento de ativos.

Tabela 2.1 – Descrição dos grupos de objetos da MIB-2 (MCCLOGHRIE K. , 1991).

Grupo	Qtde objetos	OID	Descrição
SYSTEM	7	1.3.6.1.2.1.1	Define a lista de objetos que perfazem a operação do sistema: *system uptime, system contact.*
INTERFACES	23	1.3.6.1.2.1.2	Monitora a atividades das interfaces, contabiliza octetos enviados e recebidos, erros e descartes.
AT	3	1.3.6.1.2.1.3	Mapeamento de endereços físicos/rede.
IP	42	1.3.6.1.2.1.4	Tabela de roteamento e contadores.
ICMP	26	1.3.6.1.2.1.5	Contadores icmp.
TCP	19	1.3.6.1.2.1.6	Tabela de conexões TCP e contadores.
UDP	7	1.3.6.1.2.1.7	Tabela UDP e contadores.
EGP	18	1.3.6.1.2.1.8	Tabela de vizinhos EGP e contadores.
SNMP	39	1.3.6.1.2.1.9	Registros estatísticos das mensagens SNMP.

Como a MIB-2 suporta somente informações locais dos dispositivos, foi necessária a implementação de uma base de dados para informações estatísticas dos segmentos de rede. A MIB RMON foi então definida com a finalidade de monitorar no nível da camada de enlace (Camada MAC) do modelo OSI (IEEE Computer Society, 2001).

A evolução da RMON resultou da necessidade de se gerenciar os protocolos de camada superior e se consolidou com a definição das novas extensões que ficaram denominadas RMON-2. Com as novas extensões tornou-se possível a geração de estatísticas de tráfego de origem e destino no nível da camada de rede e aplicação, o mapeamento de endereços de rede e portas em endereços físicos, dentre outras diversas funcionalidades conforme ilustrado pela figura 2.2.

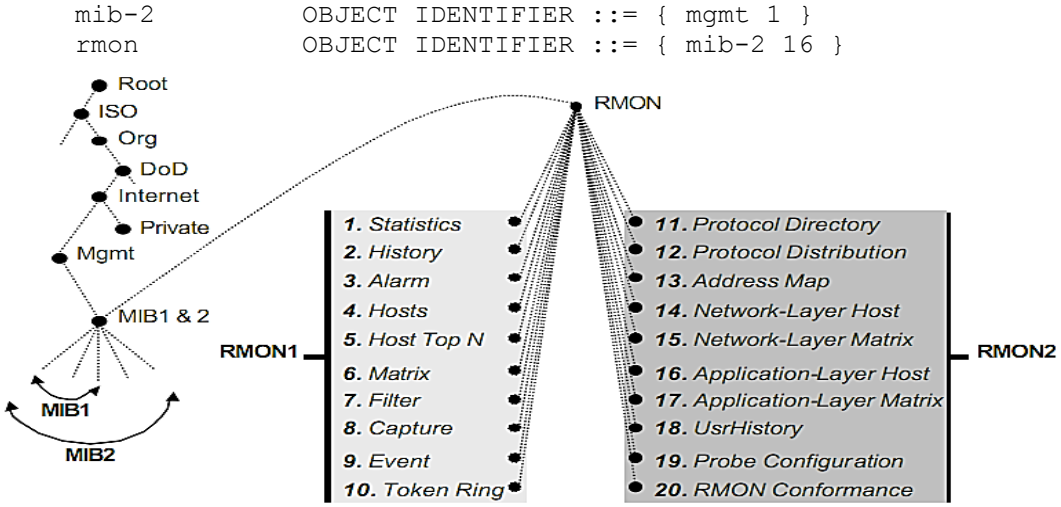


Figura 2.2 – Objetos das MIB's RMON e RMON-2 (STALLINGS, 1998).

2.2.2. Protocolo SNMP

A primeira versão do protocolo SNMP, considerada pelo IETF como histórica por ser a primeira especificação amplamente suportada pelos fornecedores, era baseada em comunidades (*read-only, read-write e trap*). O acesso aos dispositivos ocorre pela definição destas comunidades e é transmitido via UDP, portas 161/162 (envio/recepção) sem nenhum processo criptográfico para garantia da confidencialidade (CASE, 1990).

Considerando as limitações da primeira versão, tais como: impossibilidade de configuração remota dos agentes, não ser adequado a grandes redes e volumes de dados, não permitir comunicação entre os gerentes, dentre outras limitações, houve a necessidade de evolução para uma nova versão, conhecida como SNMPv2c (SANTOS, 2008).

Surge então o SNMPv2 definido pela RFC 3416, com aspectos positivos como a criação de extensões da linguagem (que facilitam a declaração de novos objetos) e o incremento do desempenho do protocolo na troca de informações através do melhoramento no tratamento de erros (PRESUHN, 2002).

No SNMPv2, além do aprimoramento na troca de mensagens, foram adicionadas duas outras PDU's: *GetBulkRequest* para o tratamento de grandes quantidades de dados e *InformRequest* permitindo a comunicação entre gerentes, conforme tabela 2.2.

Tabela 2.2 – PDU (Protocol Data Unit) em diferentes versões do SNMP (MORRIS, 2003).

SNMPv1	**SNMPv2c**	**SNMPv3**	**Response PDU**
GetRequest	GetRequest	GetRequest	GetResponse
GetNextRequest	GetNextRequest	GetNextRequest	GetResponse
SetRequest	SetRequest	SetRequest	GetResponse
Trap	Trap	Trap	None
	GetBulkRequest	GetBulkRequest	GetResponse
	InformRequest	InformRequest	GetResponse

Muito embora tenha evoluído, as questões relativas à segurança, a configuração remota e infraestrutura administrativa não foram tratadas nesta segunda versão.

Em sua terceira versão, o SNMPv3 evoluiu principalmente na questão da segurança e na definição de convenções textuais, conceitos e terminologias. A mudança mais importante é que a versão 3 abandona a noção de gerentes e agentes. Gerentes e agentes agora são chamados de entidades SNMP. Cada entidade consiste de um motor SNMP e uma ou mais aplicações SNMP (HARRINGTON & all, 2002).

Estes novos conceitos são importantes porque definem uma arquitetura mais do que simplesmente um conjunto de mensagens. A nova definição da arquitetura ajuda a segregar partes funcionais diferentes do sistema SNMP tornando o conjunto intrinsecamente mais seguro. Nesta nova nomenclatura, o motor do SNMPv3 segundo a RFC 3411 (HARRINGTON & all, 2002), é composto por quatro peças, conforme ilustrado pela figura 2.3: o *Dispatcher*, o Subsistema de Processamento de Mensagens, o Subsistema de Segurança, e do Subsistema de Controle de Acesso.

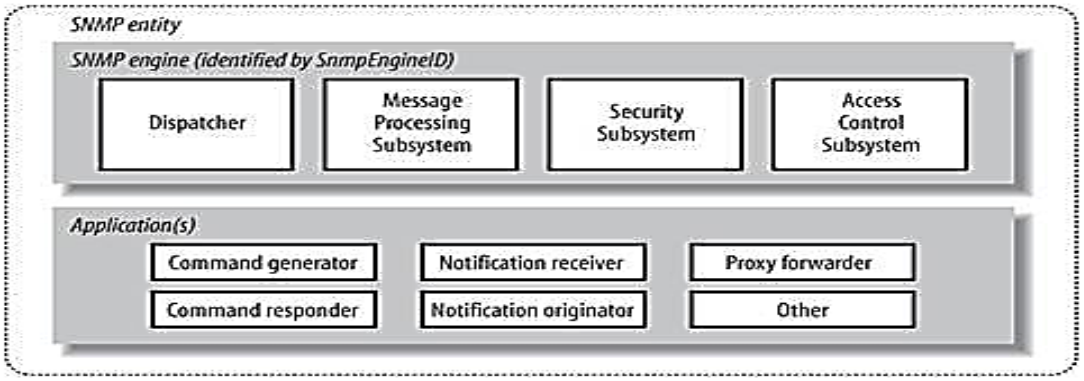


Figura 2.3 – Representação dos elementos das entidades no SNMPv3 (DOUGLAS, 2005).

O trabalho do *Dispatcher* é enviar e receber mensagens. Ele tenta determinar a versão de cada mensagem recebida (v1, v2, ou v3) e, se a versão é suportada, encaminha a solicitação para o Subsistema de Processamento de Mensagens. O *Dispatcher* também envia mensagens SNMP a outras entidades.

O Subsistema de Processamento de Mensagens prepara mensagens a serem enviadas e extrai dados de mensagens recebidas. Um Subsistema de Processamento de mensagens podem conter vários módulos de processamento de múltiplas mensagens.

O Subsistema de Segurança fornece serviços de autenticação e privacidade. A autenticação usa o conceito de comunidades para o SNMP (v1 e v2) e autenticação baseada em usuários com a utilização do HMAC (*Hash Message Authentication Code*) no SNMPv3. A garantia da confidencialidade nas comunicações SNMPv3, deriva da utilização de algoritmo de criptografia simétrica DES para cifrar as mensagens enviadas e decifrar as mensagens recebidas (STALLINGS, W., 1998).

O Subsistema de Controle de Acesso é responsável por controlar o acesso a objetos MIB. Podem ser definidos quais objetos um usuário tem acesso e quais operações que ele terá permissão para executar nesses objetos.

Os *Application(s)* SNMPv3, são a modularização das antigas e novas funcionalidades com a contextualização das novas terminologias, conforme a tabela 2.3:

Tabela 2.3 – Descrição das funcionalidades dos Applications SNMPv3 (PRESUHN, 2002).

Application(s)	**Descrição**
COMMAND GENERATOR	é implementado em um NMS para que sejam emitidos consultas (G*etnext/GetBulk) e processos (setRequest) às entidades.*
COMMAND RESPONDER	responde às solicitações (*GetResponse).*
NOTIFICATION ORIGINATOR	gera as traps SNMP (*Traps*) e notificações (*InformRequest*).
NOTIFICATION RECEIVER	é implementado em um NMS para receber as traps e informar mensagens (*informRequest*).
PROXY FORWARDER	facilitar a passagem de mensagens entre entidades.
OTHER	extensível, permite a implantação de novos aplicativos ao longo do tempo.

2.3. MONITORAMENTO BASEADO EM ANÁLISE DE PACOTES

O monitoramento baseado em análise de pacotes é definido como um aplicativo analisador de pacotes que frequentemente pode ser denominado como coletor de pacotes (*packet sniffing*) ou analisador de protocolo (*protocol analysis*) (SANDERS, 2007).

Um analisador de pacotes é tipicamente implementado por uma ferramenta coletor de pacotes, que captura os sinais que saem e chegam pela interface de rede e os armazena e analisa, conforme a interpretação baseada nos protocolos suportados pela ferramenta.

O monitoramento baseado em análise de pacotes possui várias utilidades, dentre elas: permite um melhor entendimento das características da rede; verificação de quem está ativo na rede; determinar quem ou o que está utilizando a largura de banda disponível; identificar os horários de pico de uso da rede; identificar possíveis atividades maliciosas; detectar aplicações inseguras e infectadas.

Os processos de uma ferramenta de coleta de pacotes podem ser divididos em três passos: coleta, conversão e análise.

2.3.1. Coleta e Conversão

As redes padrão Ethernet possuem o compartilhamento do barramento como fundamento básico de comunicação. Desta forma, usa a transmissão no sentido todos-para-todos *broadcast* como princípio, onde qualquer quadro Ethernet emitido é percebido por todos os ativos no segmento. Para evitar uma alta taxa de colisões e normatizar os

procedimentos de acesso ao meio compartilhado, o padrão CSMA/CD define a forma como um dispositivo: percebe o segmento, acessa o segmento para o envio de quadros sem que ocorram colisões e detecta as colisões, caso ocorram, para reajustamento das transmissões entre todos os dispositivos (IEEE Computer Society, 2008).

Sob essa ótica, seria possível um sistema de monitoramento de mensagens no barramento, coletar todos os quadros encaminhados e por este aspecto considerar que o princípio do *broadcast* é o ideal.

Sob o aspecto funcional, esse comportamento sobrecarrega e afronta diretamente a característica de escalabilidade da rede. Portanto, as redes padrão Ethernet normatizaram mecanismos que garantam a escalabilidade pela delimitação dos domínios de colisão *(barramento)* através dos chaveadores (*switchs*) e domínios de difusão *(broadcast)* através dos roteadores. Para garantir maior eficiência na análise dos quadros recebidos, definiu-se modos de operação para as interfaces Ethernet, que por padrão, descarta todos os quadros recebidos que não são endereçados ao próprio dispositivo.

Em primazia, no processo de coleta, o coletor de pacotes deve modificar sua interface para o modo de operação promíscuo para interface cabeada ou modo de operação monitor para interface sem fio. Com esta modificação torna-se capaz de coletar todos os pacotes que se percebe na interface, independente do destino.

Segundo (SEONG-YEE & all, 2008) um coletor de pacotes para modificar o modo de operação de uma interface pode ter sua arquitetura de coleta implementada através de uma biblioteca API *middleware* ou por *conexões brutas* (*raw sockets API*). A vantagem do uso do *middleware* é que a interface de software ou código, é portátil e independente da máquina enquanto que a ligação em conexões *raw sockets*, é diretamente no *driver* da interface de rede.

Outra diferença nas características da arquitetura por *middleware e raw socket*, ilustrada pela figura 2.4, é que a primeira pode efetuar o armazenamento temporário e a filtragem no processo da coleta a fim de otimizar a entrega dos pacotes enquanto que a segunda abordagem apenas recebe e entrega os pacotes para a camada de aplicação.

Uma das bibliotecas de código fonte aberto, que garante portabilidade, amplamente difundido para a coleta de pacotes é a interface de programação de aplicativos *API Libpcap* ou *Winpcap* (XIAOFAN & all, 2010).

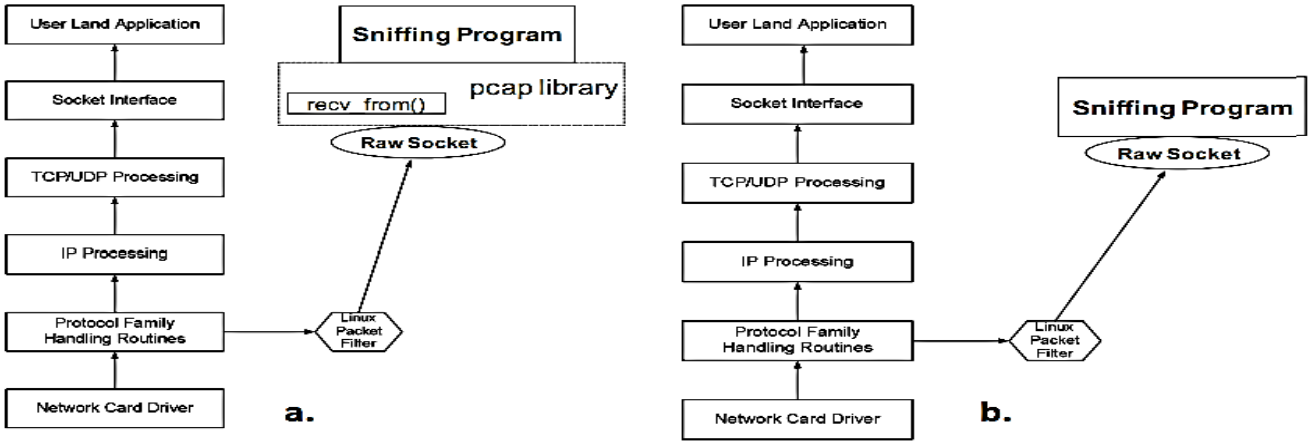


Figura 2.4 – Coletor: a. com libpcap, b. com Raw Socket. (SEONG-YEE & all, 2008).

Segundo (CARUSO, 2005), não obstante ao uso destas bibliotecas que fornece portabilidade aos aplicativos de análise dos pacotes, um aspecto a considerar, está no fato de que esses *middlewares* são *mono-threads*, não permitem processamento paralelo pela implementação de tarefas concorrentes *multi-thread*, do que decorre diretamente na limitação do desempenho do sistema, consequentemente na capacidade de coleta dos pacotes que ingressam na interface em enlaces de alto desempenho.

Além de a interface estar no modo promíscuo, em redes cabeadas os pacotes não fluem naturalmente para a interface do coletor de pacotes, fazendo com que as técnicas de: *ARP Cache Poisoning*, *Port mirroring* ou *Port Spanning* e *Hubbing Out*, devam estar implementadas neste intuito.

A técnica de *ARP Cache Poisoning* ou também conhecida como *ARP spoofing* utiliza a ausência de algum mecanismo de autenticação do protocolo *ARP* que é o responsável por preencher a tabela cache dos dispositivos com o mapeamento dos endereços físicos *MAC* e endereços de rede *IP*, conforme figura 2.5 (PUANGPRONPITAG & MASURAI, 2009).

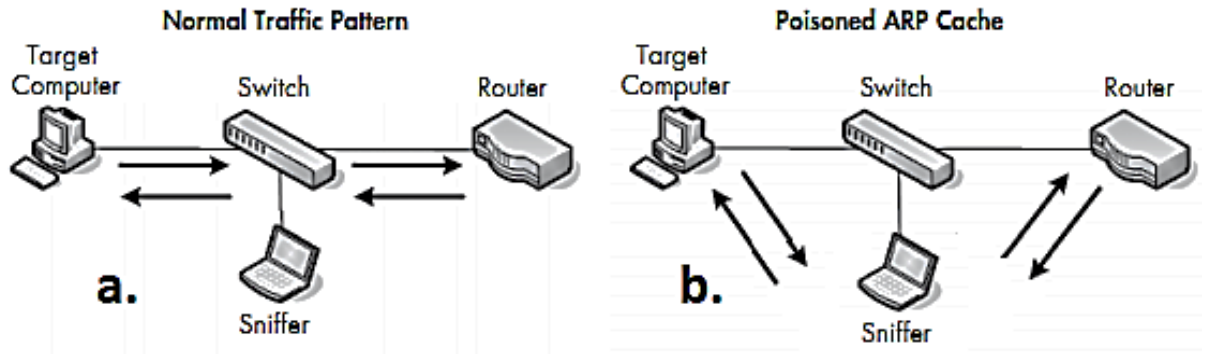


Figura 2.5 - Tráfego normal (a); Cache ARP envenenado (b) (SANDERS, 2007).

Nesta técnica, o chaveador é inundado com associações de endereço IP e endereço MAC que apontam para o dispositivo que deseja monitorar os pacotes. Existem diversas implementações de software que utilizam-se desta vulnerabilidade, dentre as quais, a ferramenta de segurança *Cain & Abel* (MONTORO, 2012), a aplica para intermediar as comunicações e com isso inspecionar os pacotes para descobrir senhas, dentre outras investidas conforme descritas em (QADEER & ZAHID, 2010).

Segundo (ZHANG & MOORE, 2007), atualmente grande parte dos chaveadores gerenciáveis tem a implementação do espelhamento de porta (*port mirroring*), que consiste no encaminhamento de uma cópia dos pacotes que entram e saem de uma determinada porta que se deseja monitorar ou de um conjunto de portas (*port spanning*) para outra porta aonde se encontra conectado o coletor de pacotes, conforme ilustrado pela figura 2.6.

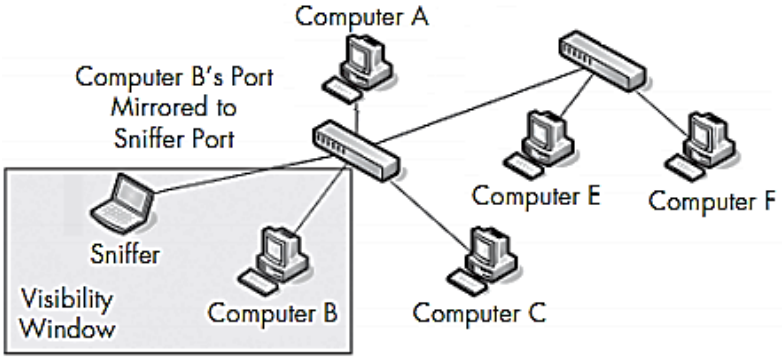


Figura 2.6 – Espelhamento de porta (SANDERS, 2007).

Nos casos em que o *switch* não possuir a funcionalidade de espelhamento de porta, a técnica de *Hubbing Out*, embora não seja um método mais recomendado, pelo fato de reduzir a duplexação do dispositivo alvo para *half duplex*, em alguns casos é o único meio de se realizar o monitoramento de captura de pacotes, conforme figura 2.7.

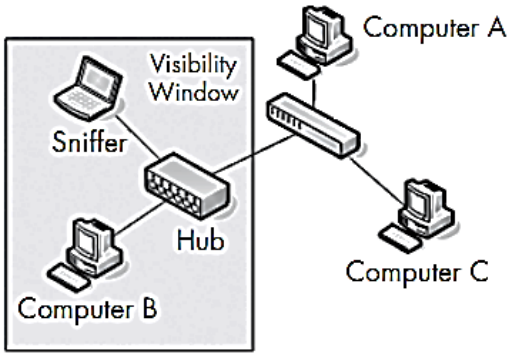


Figura 2.7 – Encaminhamento de pacotes no barramento. (SANDERS, 2007).

Esta técnica consiste na composição de um único domínio de colisão entre o coletor de pacotes e o dispositivo a ser analisado pela adição de um *HUB* no segmento que se deseja coletar. Uma vez capturado os pacotes, o processo de conversão, consiste na transformação dos dados em formato binário para um formato legível aonde os agrupamentos binários são convertidos em pacotes e armazenados em disco, figura 2.8.

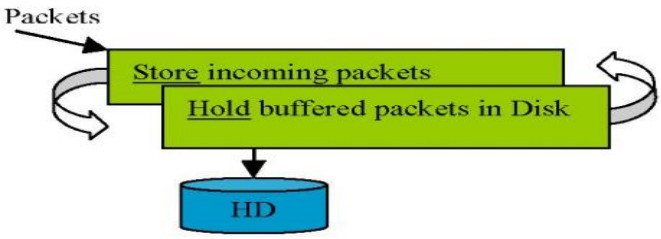


Figura 2.8 – Otimização da gravação nos discos (SANDERS, 2007).

Segundo (DABIR & MATRAWY, 2008), o gargalo do processo de coleta e conversão pode estar em grande parte associado ao processo de armazenamento. Esse aspecto deve ser analisado principalmente em relação às características de capacidade e desempenho. A capacidade se refere ao volume de dados que deverá ser provisionado enquanto que o desempenho é a eficácia do subsistema de discos no processo de armazenamento físico dos pacotes no disco.

2.3.2. Análise

O processo de análise consiste na última etapa do monitoramento baseado em pacotes e na finalidade do projeto deste modelo de gerenciamento, pois é onde são extraídos os dados, que relacionados, permitem a composição de informações e comportamentos dos protocolos.

Essencialmente, as ferramentas que analisam os pacotes ou *protocol analysis*, podem ser consideradas em função de cinco características: Suporte a sistemas operacionais; Suporte ao aplicativo; Custo; Interface amigável de gerenciamento; e Suporte a protocolos.

Dessas cinco características as duas últimas, definem a qualidade intrínseca da ferramenta de análise (*protocol analysis*), pois a capacidade de análise dos fatores coletados e convertidos se encontra no suporte aos protocolos que a ferramenta dispõem e a facilidade de se efetuar as consultas e relacionamentos entre os pacotes capturados a fim de obter respostas aos questionamentos que advém da interface amigável de gerenciamento.

Entre as ferramentas *open source* amplamente utilizadas estão o *tcpdump* e o *wireshark*. Enquanto o *tcpdump* suporta os sistemas operacionais Linux, e *FreeBSD*, o

wireshark além desses, suporta o *Windows* da Microsoft através do uso da biblioteca *Winpcap*. Ambas as ferramentas possuem um suporte ativo da ferramenta através de fóruns de colaboração, com destaque para o *wireshark* que possui até programa de treinamento e certificação. Como são disponíveis para uso pela comunidade, não possuem custo para o uso da solução.

Com relação à facilidade de uso, a interface do *wireshark* é extremamente amigável e possui diversas funcionalidades atraentes, como a visualização colorida dos pacotes, enquanto que o *tcpdump* possui interface baseado linhas de comando.

Quanto ao suporte aos protocolos e suas características associadas, ambas as ferramentas possuem um bom conjunto de funcionalidades que a colocam no patamar das ferramentas livres para monitoramento baseado em pacotes, mais amplamente utilizado.

2.4. MONITORAMENTO BASEADO EM FLUXOS DE TRÁFEGO

O monitoramento do tráfego de rede baseado no conceito de fluxos IP, segundo (KREJCÍ, 2009), foi originalmente desenvolvido pela empresa *CISCO Systems*, através do protocolo proprietário *NetFlow* que se tornou um *padrão de fato* devido a sua ampla adoção pelos dispositivos de diversos fornecedores. Em sua última versão o *NetFlow* v9 trouxe o conceito de definição de campos flexíveis baseado em modelos definidos diretamente pelos usuários. Este conceito, bem como todas as definições do protocolo *NetFlow*, foi seguido e aprimorado pelo protocolo aberto IPFIX que propõe ser o novo *padrão comum* para o monitoramento de fluxos de tráfego IP, definido pelo IETF (CLAISE, 2008).

Paralelo ao *NetFlow* e ao IPFIX, o protocolo *sFlow* surge como alternativa para o monitoramento baseado em ambientes de volume de tráfego extremamente altos pois diferentemente do método de agregação do fluxo de tráfego utilizado pelo *NetFlow* e consequentemente pelo IPFIX, o protocolo *sFlow* utiliza um método flexível de amostragem que garante alta performance na elaboração dos fluxos visto não necessitar analisar todo o tráfego para gerar as informações estatísticas conforme descrito em (PHAAL, PANCHEN, & MCKEE, 2001).

Estas técnicas de monitoramento possuem como premissa o agrupamento do tráfego de rede consolidado em sessões de transmissão unidirecional, denominado como fluxos. Esses fluxos são gerados em sensores que os encaminham de forma agrupada aos coletores aonde são armazenados e em último plano, habilitam a análise por meio de relatórios.

Segundo (LUCAS, 2010), os fluxos, também conhecidos como fluxos de tráfego ou fluxo de dados, são conjuntos de tráfego de redes (aplicativos, protocolos e informações de controle) que possuem atributos comuns, como endereços de origem/destino, tipos de informações, sentido, ou informações fim-a-fim.

As informações dentro de um fluxo são transmitidas em uma simples sessão de uma aplicação. Fluxos fim-a-fim, entre origem e destino de aplicativos, dispositivos e usuários. Desde que eles sejam identificados pelas suas informações fim-a-fim, eles podem ser diretamente relacionadas a uma aplicação, dispositivo, rede ou associadas com um usuário final. Pode-se também examinar fluxos que transitam por um enlace ou fluxos rede-a-rede básica (MCCABE, 2007).

Conforme ilustrado na figura 2.9, pode haver diversas informações a serem consolidadas em fluxos, tais como: o tráfego entre os clientes e os aplicativos fornecidos no *Data Center* (*Application flows*)*;* as trocas de informações entre os protocolos ou identificação de tráfego malicioso (*Multicast e Security flows*); o tráfego de dados entre os roteadores de borda BGP (*Peering flows*) e o tráfego de dados entre a matriz e filiais e matriz e escritórios remotos (*IP flows*).

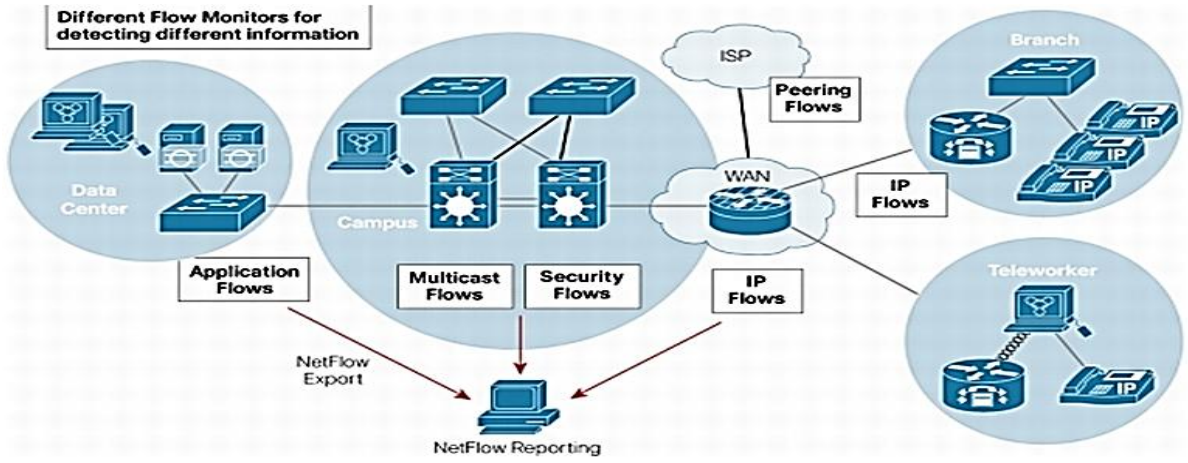


Figura 2.9 – Estrutura e tipos de fluxos (CISCO, 2004)

A arquitetura para o monitoramento destes diversos tipos de fluxos, descritos na figura 2.9, é composta por: sensores, coletores e geradores de relatórios.

Os sensores são dispositivos que têm a funcionalidade de capturar os pacotes dos tráfegos do segmento da rede, identificar os seus fluxos através do rastreamento das conexões e exportar um agrupamento de fluxos a outro dispositivo denominado coletor. O sensor normalmente é um equipamento da infraestrutura da rede (*switch, roteador e ou firewall*) ou

um dispositivo com software capaz de operar as funcionalidades descritas por meio dos protocolos de gerência de fluxos.

Segundo (MCCABE, 2007), uma das questões cruciais em relação ao projeto do gerenciamento baseado em fluxos é a definição de, quais tráfegos será necessário analisar, e partindo dessa premissa, é que a localização dos sensores deve ser definida. Sempre que possível essa definição deve ser no *gateway* do fluxo do tráfego desejado.

Nos fluxos de tráfego das redes de triagem (servidores de aplicativos internos) ou nas zonas desmilitarizadas (serviços públicos) a definição de um sensor no *gateway* de cada uma dessas redes internas, habilitará a gerência nos *hosts* e serviços internos e externos. Para analisar o fluxo de tráfego em uma rede local, a definição do sensor deve ser no *switch* que executa a função de núcleo ou distribuição. Para análise dos fluxos entre a rede local (matriz) e as redes remotas (filiais e/ou escritórios) basta a definição dos sensores no *gateway* de cada uma das redes remotas, conforme ilustra a figura 2.10.

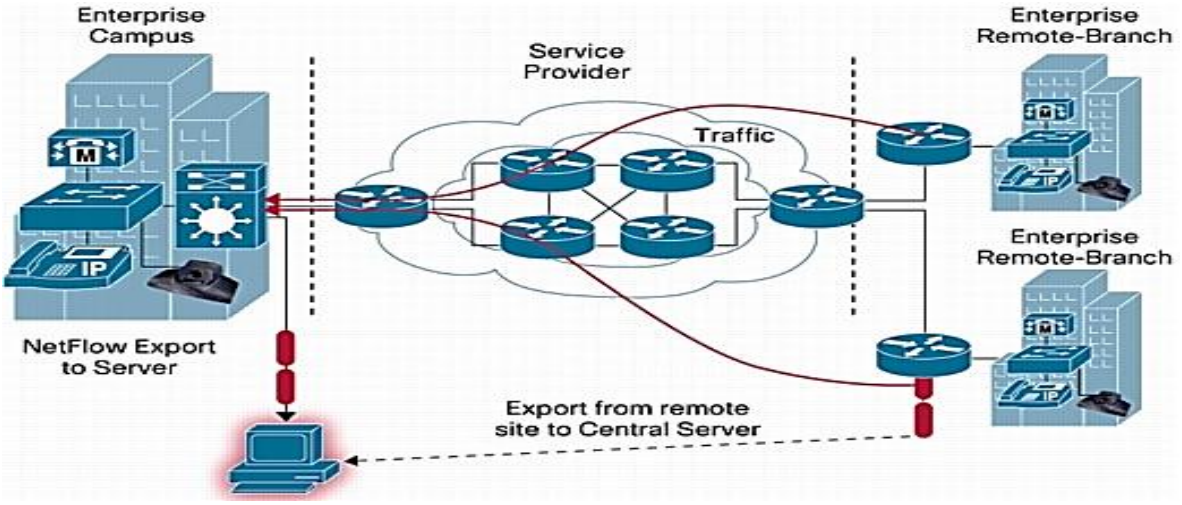


Figura 2.10 – Exportação de fluxos (CISCO, 2004).

Com a premissa da existência de um ou mais sensores que irão encaminhar os registros dos fluxos, os coletores por sua vez, possuem a funcionalidade de receber os encaminhamentos dos sensores e armazenar de forma persistente esses dados, comumente em discos físicos.

Finalmente, o sistema de relatórios deve ser capaz de entender o formato dos arquivos gerados pelo coletor e acessá-los como insumo para a produção de relatórios amigáveis que auxiliem a gestão dos recursos da rede.

Segundo (CLAISE, B.; WOLTER, R., 2007), a abordagem de gerenciamento por fluxo de tráfego traz consigo as seguintes possibilidades:

- **Monitoramento da rede:** as técnicas baseadas em fluxos podem ser utilizadas para visualização de padrões de tráfego associados a dispositivos individuais da rede. Os padrões podem ser analisados de forma agregada para toda a rede ou ainda de forma específica para cada aplicação permitindo diferentes visões para a detecção proativa de problemas, bem como, auxilia na resolução rápida e eficiente de incidentes.
- **Monitoramento de aplicativos e perfis:** é possível ter uma visão detalhada e baseada no tempo de uso de banda por aplicação na rede. Essas informações podem ser usadas para planejar e entender os requisitos de novos serviços e alocar recursos de rede e de aplicativos para atender às demandas do cliente.
- **Monitoramento de usuários e perfis:** é possível obter uma compreensão detalhada de como os usuários consomem recursos de rede. Essas informações podem ser usadas para planejar e distribuir de forma eficiente o acesso, a banda de *backbone* e recursos de aplicativos, bem como detectar e resolver potenciais violações de políticas de segurança.
- **Planejamento da Rede:** a captura de dados por um longo período, oportuniza o acompanhamento do crescimento de uso da rede bem como permite atualizar o planejamento de ações para antecipar suas necessidades minimizando o custo total das operações da rede e ao mesmo tempo que maximiza seu desempenho, capacidade e confiabilidade. Ainda, permite a detecção de tráfego indesejado, valida a largura de banda e qualidade de serviço (QoS), permitindo a análise de novas aplicações na rede.
- **Análise de Segurança:** possibilita a identificação e classificação de ataques distribuídos de negação de serviço (DDoS), vírus e worms, em um limiar próximo ao tempo real. Visualiza alterações nas mudanças no tráfego de rede indicando anomalias em potencial, podendo também ser utilizada como uma valiosa ferramenta de análise forense para auxiliar na compreensão e analise das sequências de incidentes de segurança passados.
- **Registros para contabilização e/ou Faturamento:** fornece granularidade na medição, pois o fluxo de dados inclui detalhes como: endereços IP dos pacotes, a contagem de bytes, *timestamps*, tipo de serviço, e as portas dos aplicativos, dentre outros, permitindo a contabilização flexível e detalhada da utilização de recursos. Essas informações podem ser utilizadas para um sistema de faturamento ou contabilização de alocação de custos para utilização de recursos.

- **Armazenamento de dados:** os registros armazenados podem ser utilizados como insumos em um processo de mineração de dados para a descoberta de padrões de classificação para obtenção de conhecimento que permitam encontrar quais as aplicações e serviços estão sendo usados por usuários e como orientá-los na melhoria dos serviços.

2.4.1. NetFlow

Um fluxo armazenado no *cache* NetFlow é definido por um conjunto de pacotes unidirecional identificados pela combinação de sete campos chave: IP de origem, IP de destino, Porta de origem, Porta de destino, Protocolo L3, Byte ToS e Interface de entrada, conforme ilustrado pela figura 2.11.

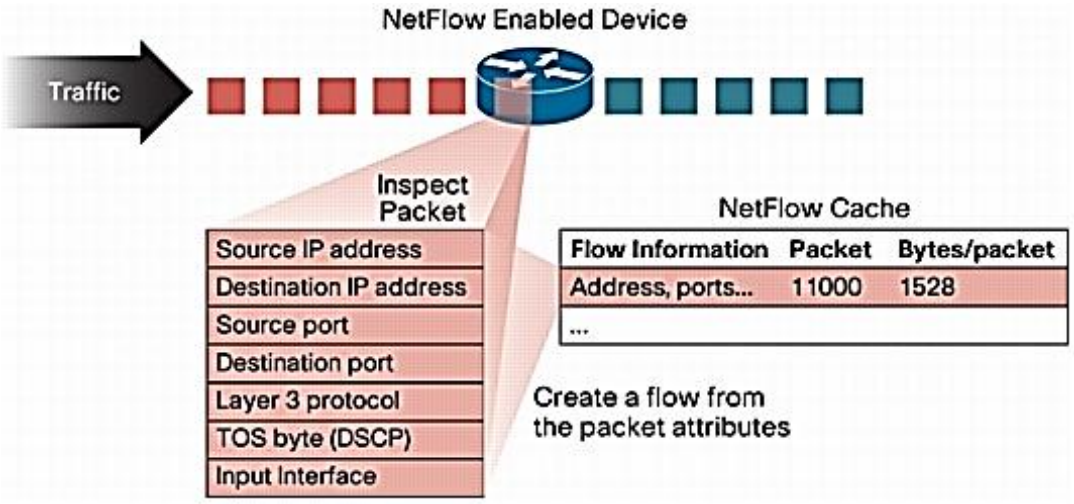


Figura 2.11 – Criação de fluxos em cache NetFlow. (CISCO, 2004)

Portanto, a combinação destes sete campos define um fluxo único. Caso um fluxo de pacote, possua um campo diferente dos fluxos ativos no cache NetFlow, então este pacote é considerado um novo fluxo, sendo acrescido um novo registro de controle no *cache* NetFlow. Os fluxos podem conter outros campos dependendo da versão do formato de registro configurado para exportação.

A interpretação para composição dos fluxos está intrinsecamente vinculada ao protocolo utilizado na comunicação fim-a-fim e resulta da utilização de sofisticados algoritmos de controle para determinação se o pacote está contido ou não em um fluxo existente no *cache* NetFlow.

2.4.1.1. Arquitetura de Serviços NetFlow

Segundo (CLAISE, B.; WOLTER, R., 2007), a arquitetura dos serviços NetFlow pode ser classificada em três categorias: pré-processamento; funções e serviços e pós-processamento, conforme ilustrado pela figura 2.12.

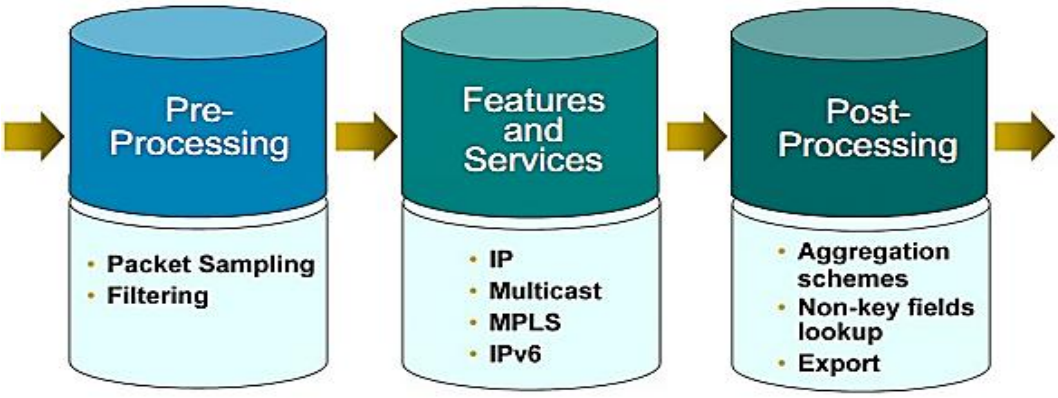


Figura 2.12 – Arquitetura da ordem de processamento NetFlow. (CISCO, 2004)

A categoria pré-processamento, permite a aplicação de filtros ou configuração de análise amostral para o conjunto do tráfego de rede coletado. A categoria de funções e serviços baseados na última versão disponível habilita a medição baseada nos diversos tipos de tráfego. A categoria de pós-processamento permite o controle de como os fluxos serão exportados.

O primeiro passo é a criação e classificação dos fluxos no *cache* NetFlow. O processo de expiração dos fluxos do *cache* NetFlow é o segundo passo. No terceiro passo é checado se está aplicado algum esquema de agregação. Caso a agregação seja executada, os registros de fluxos serão combinados no *cache* de agregação. Caso a agregação não esteja ativa, os registros de fluxos não agregados serão exportados.

Finalmente, quando os registros de fluxos estiverem prontos para serem exportados o protocolo NetFlow verifica a versão de exportação (5, 7, 8, 9) para a definição do protocolo de transporte (UDP ou SCTP) que será utilizado para a transmissão com o coletor (CLAISE, B.; WOLTER, R., 2007).

2.4.1.2. Composição dos Fluxos (ICMP, UDP E TCP)

O protocolo ICMP é comumente associado ao aplicativo *ping* que realiza, principalmente, as operações de *request e reply*, mas também pode encaminhar informações básicas de gerenciamento e roteamento da Internet.

Este protocolo, diferentemente dos protocolos TCP e UDP que indicam um serviço na camada de aplicação pelo uso de portas e mais especificamente o TCP que marca o seu fluxo pelas *flags TCP-style*, sinaliza seus pacotes pela utilização de *ICMP type* e *ICMP codes*, conforme descrito na tabela 2.4.

Tabela 2.4 – Tipos e códigos ICMP associados em decimal e hexadecimal (IANA).

Tipo	**Code**	**Decimal**	**Descrição**
0	0	0	*Resposta à soliticação Echo (echo reply)*
3			*Destino inalcançável*
	0	300	*Rede inalcançável.*
	1	301	*Host inalcançável.*
	2	302	*Protocolo inalcançável.*
	3	303	*Porta inalcançável.*
	4	304	*Necessária fragmentação, mas ainda não definida.*
	6	306	*Rede de destino desconhecida.*
	7	307	*Host de destino desconhecido.*
	9	309	*Comunicação com rede de destino administrativamente proibida.*
	10(a)	310	*Comunicação com host de destino administrativamente proibida*
	13(d)	313	*Comunicação administrativamente proibida*
5			*Redirecionado*
	0	500	*Redirecionado para subrede.*
	1	501	*Redirecionado para o Host.*
8	0	800	*Solicitação de resposta (echo request)*
11(b)			*Tempo excedido.*
	0	2816	*Tempo de vida excedido em trânsito.*
	1	2817	*Tempo excedido na remontagem do fragmento*
12(c)		3072	*Problema de parâmetro.*
13(d)		3328	*Solicita data e hora (timestamp request)*
14(e)		3584	*Resposta de data e hora (timestamp reply)*

Os sensores codificam o *ICMP type* e *code* no campo porta de destino dos fluxos, sendo que o primeiro byte codifica o *tipo* e o segundo byte codifica os *códigos* devido ao fato do ICMP não conceber originalmente o conceito de portas (LUCAS, 2010).

Conforme pode ser visualizado na figura 2.13, no primeiro fluxo a porta de origem possui o valor zero, pois não é utilizada, enquanto que o valor em hexadecimal da porta de destino é 800, que normalizado representa 0800, sendo 08 identificando o tipo, e 00 o código. Relacionando na tabela 4, temos que se trata de um fluxo ICMP que solicita retorno (*echo request*).

Sif	SrcIPaddress	Dif	DstIPaddress	Pr	SrcP	DstP	Pkts	Octets
0000	80.95.220.173	0000	36.85.32.153	01	0	❶800	2	122
0000	189.163.178.51	0000	36.85.32.130	01	0	❷b00	1	56
0000	64.142.0.205	0000	36.85.32.5	01	0	❸300	1	56
0000	201.144.13.170	0000	36.85.32.130	01	0	❹303	1	144
0000	36.85.32.9	0000	194.125.246.213	01	0	❺0	5	420

Figura 2.13 – Exemplos de fluxos NetFlow para o protocolo ICMP (LUCAS, 2010).

O segundo fluxo evidencia-se como fluxo ICMP do tipo *b* (tempo excedido) e código 00 (TTL excedido em trânsito), que denota que não foi possível contatar o destino, pois se encontra desconectado. O terceiro fluxo tem o tipo 03 (destino inalcançável) e código 00 (rede inalcançável), normalmente um problema de roteamento. O quarto fluxo tem o tipo 03 (destino inalcançável) e código 03 (porta inalcançável) normalmente uma resposta à uma solicitação UDP a uma porta no *host* que não está aberta. Finalmente o quinto fluxo, com tipo 0 e código 0 é uma resposta *echo reply* à uma solicitação *echo request* anteriormente feita.

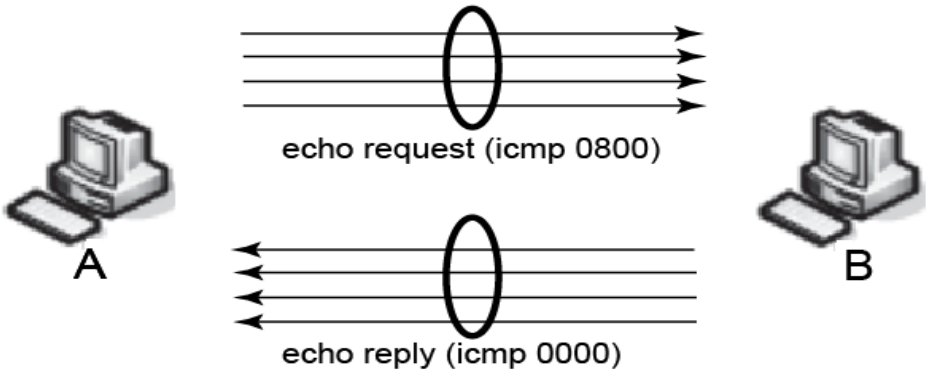


Figura 2.14 – Agrupamento de solicitações/respostas icmp em fluxos.

Quando um cliente A, faz solicitações de requisição (*type/code:* 0800) a um cliente B, que procede a resposta (*type/code:0000*), são formados 2 fluxos no *cache*, sendo um fluxo de A para B que contabiliza os pacotes de solicitação e outro fluxo de B para A que contabiliza os pacotes de resposta. Pode haver inúmeras solicitações e respostas, mas dentro de um período de vencimento do *cache* NetFlow todos os pacotes em cada sentido unidirecional serão agrupados em apenas dois fluxos.

Como podem ser visualizados na figura 2.14, os fluxos que representam os pacotes com mensagens ICMP são frequentemente respostas a outros tipos de solicitações da rede. Eles frequentemente exibem exatamente qual o tipo de erro ocorrido como resultado de uma tentativa de conexão, e a análise de seus fluxos pode ser correlacionado através de filtros com outros fluxos para identificação de suas origens (LUCAS, 2010).

Segundo (KUROSE, 2010), o protocolo UDP, por possuir a concepção do melhor esforço para a entrega de mensagens, enquadra-se na categoria de protocolo de entrega não orientado a conexões. Como consequência dessa concepção, não possui tipos/códigos ou sinalizações que os protocolos ICMP e TCP possuem para o estabelecimento de conexões através de sessões ou transações.

Concatenando com sua concepção básica, o estabelecimento de correlações para a definição de fluxos se torna um processo ligeiramente mais complexo em virtude da necessidade de se estabelecer sessões em um protocolo não orientado a sessões. Como o tráfego de rede UDP, não identifica o final de uma transação, o sensor não tem como saber exatamente quando um fluxo está completo. O sensor portanto, mantém os fluxos na memória até que o tempo de vencimento do *cache* NetFlow se expire e os fluxos sejam marcados como finalizados e a contabilidade de seus atributos fechado para serem transmitidos para o coletor, liberando o cache para a criação de novos fluxos.

O protocolo TCP, tem seus fluxos definidos e registrados pela combinação dos: endereços IP's de origem e destino (camada 3), das portas de origem e destino (assim como no protocolo UDP), e pelos diversos sinalizadores de controles, conforme a figura 2.15.

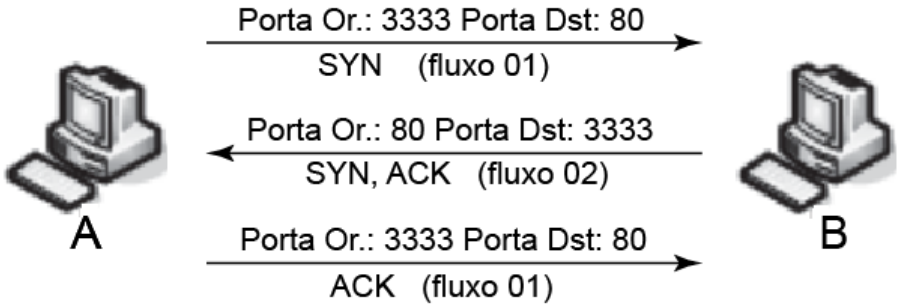


Figura 2.15 – Criação de fluxos no estabelecimento de conexões TCP.

Como o protocolo TCP possui a concepção de entrega baseado no estabelecimento de uma conexão entre emissor e receptor, o mecanismo adicionado no protocolo são os seis *bit's* sinalizadores, que garante, entre outras operações, os processos de estabelecimento e finalização das conexões.

Os fluxos em transações TCP são criados quando se localiza a emissão de uma solicitação de conexão *SYN* e finalizados quando da transmissão dos sinalizadores *FIN e RST*. Como pode ser identificado na figura 2.15, existem dois fluxos, um no sentido de A para B e outro no sentido inverso, sendo que ambos foram criados pela indicação da flag *SYN*. Nota-se que a terceira transmissão faz parte do primeiro fluxo e não de um fluxo inédito.

Tabela 2.5 – Bit's de controles TCP. (TANENBAUM, 2011)

Flags	**Hexa Decimal**	**Descrição**
FIN	0x01	*É utilizada para finalizar uma conexão, indicando que o transmissor não possui mais dados para transmitir. Entretanto, um processo pode continuar a receber dados dados indefinidamente, mesmo depois de encerrada. Este segmento possui um número de sequência que indica a ordem de processamento.*
SYN	0x02	*É utilizada para estabelecer conexões, ou sincronizar origem e destino para a transmissão de dados em um segmento. Igual ao flag FIN, possui numeração sequencial para o seu processamento.*
RST	0x04	*É utilizado para reiniciar uma conexão que tenha ficado confusa devido a uma falha no host ou por qualquer outra razão. Também pode ser utilizado para rejeitar um segmento ou uma tentativa de conexão. Em geral, é um sintoma de problemas.*
PSH	0x08	*O receptor é solicitado a entregar os dados à aplicação mediante sua chegada, em vez de armazená-los até que um buffer completo tenha sido recebido.*
ACK	0x10	*Indica que o número de número de confirmação é válido. Isso acontece para quase todos os pacotes dentro de uma transmissão.*
URG	0x20	*Indica que os dados precisam ser interpretados, ou seja, que o pacote contém a indicação que o receptor precisa para saber processar os pacotes que pertencem a um fluxo*

Um ciclo de vida normal de uma operação unidirecional TCP, envolve a solicitação de sincronização (*SYN* 0x02) as transmissões de dados (*ACK* 0x10), eventualmente uma sinalização de controle de entrega à aplicação (*PSH 0x08*) e a sinalização de finalização (*FIN 0x01*). Esses *flags* transmitidos nos pacotes que formam o fluxo são armazenados de forma cumulativa. Neste exemplo anteriormente citado, o valor armazenado no fluxo seria *0x1b*, que representa a soma dos valores hexadecimais dos *flags* (0x01+0x02+0x08+0x10).

Na exportação dos fluxos NetFlow, o campo *tcp_flags* é o responsável pelo armazenamento de todas as sinalizações TCP realizadas durante a concepção do fluxo. Caso não haja finalização da transação por meio de um sinalizador *FIN* ou *RST*, o fluxo segue os mesmos parâmetros aplicados aos protocolos UDP e ICMP, que aguardam pela expiração de validade do fluxo ou extinção do *cache* NetFlow.

2.4.1.3. Gerenciamento do Cache NetFlow e Exportação de Dados

O tamanho do *cache* NetFlow pode ser definido por plataforma de softwares e varia de 1.024 à 524.288 mil entradas de fluxos. Cada entrada (*registros de fluxos*) consome no

mínimo 64 *bytes* de memória, sendo que o número máximo de registros está diretamente associado à quantidade de memória física do dispositivo sensor.

Os registros de fluxos, em um dispositivo sensor que gerencia o *cache* NetFlow, são expirados baseados nas seguintes regras (CISCO, 2004):

- **Tempo de inatividade:** os fluxos inativos por um período específico de tempo são expirados e removidos do *cache* para exportação. O tempo padrão de inatividade é de 15 segundos podendo ser configurável para o período de 10 à 600 segundos.
- **Tempo de atividade:** por padrão não é permitido que os fluxos se mantenham ativos por mais de 30 minutos no *cache,* e ocorrendo este caso, o fluxo considerado de longa duração será expirado e removido do *cache* para exportação. Se a transação entre a origem e destino (fato que gerou o fluxo) ainda estiver em atividade, será criado um novo fluxo para contabilizar essa transmissão.
- **Limite de tamanho:** caso a capacidade limite do cache seja atingido, uma função heurística atua, no sentido de liberar espaço de memória para novos registros de fluxos, expirando e exportando os fluxos atuais.
- **Sinalização TCP:** o término das transmissões TCP é sinalizado pelas flags FIN e RST que também são utilizadas para identificação do término do fluxo associado à sessão TCP ativa no cache NetFlow.

Conforme ilustrado na figura 2.16, as entradas de fluxo no *cache* NetFlow são expiradas com base nas regras supra citadas e exportadas para um coletor que receberá essas informações do sensor via transmissão UDP ou SCTP, para posterior análise.

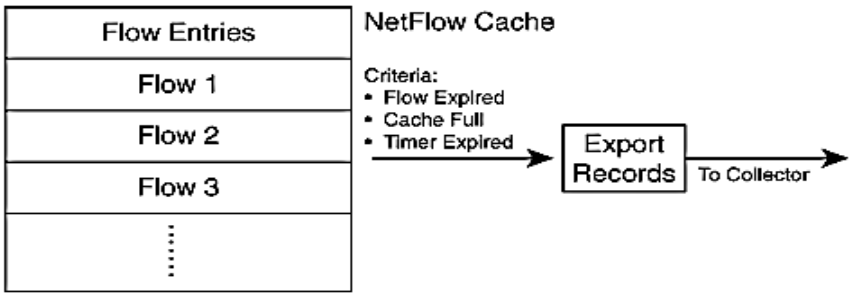


Figura 2.16 – Mecanismo de cache NetFlow (CLAISE, B.; WOLTER, R., 2007)

Conforme ilustrado na figura 2.17, a temporização de atividade (AT1) inicia quando o primeiro pacote é recebido na interface e é criada a primeira entrada de fluxo no *cache*. O temporizador (AT1) analisa no término de 30 minutos os fluxos considerados de longa duração. Como nesta ilustração o fluxo possui duração superior a 30 minutos, o temporizador

(AT1) expira e exporta o fluxo do *cache*, recriando uma nova entrada de fluxo e reiniciando nova temporização (AT2). Esse processo se repete no fim do temporizador (AT2), porém o fluxo de pacotes não ocorre mais sendo expirado pelo temporizador de fluxos inativos (IT1).

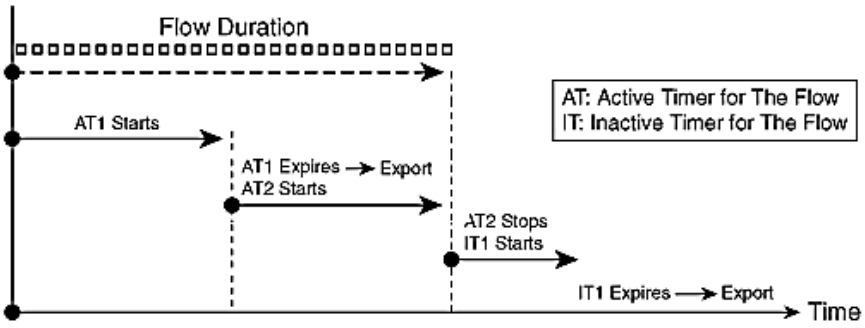


Figura 2.17 – Regras de temporização do cachê (CLAISE, B.; WOLTER, R., 2007).

Posteriormente ao processo de vencimento dos fluxos no *cache* NetFlow (*item 2. Expiration* da figura 2.18), os fluxos para serem exportados devem ser analisados quanto à agregação ou não dos campos disponibilizados.

O suporte à agregação está disponível apenas para as versões 8 e 9 do protocolo NetFlow através de *templates* de fluxos. Em caso de utilização dos dados gerados em formato nativo pelo sensor, a exportação, com base em uma definição prévia de versionamento 5 ou 9, deverá ser realizada para que os dados sejam armazenados de forma persistente em um dispositivo denominado coletor.

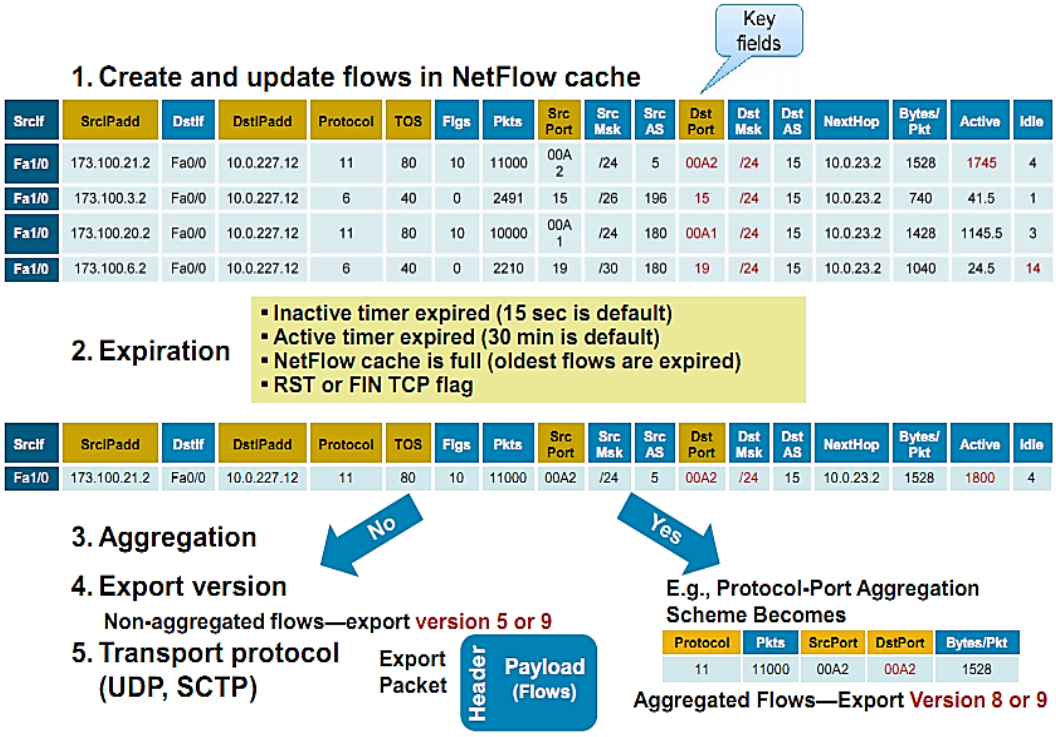


Figura 2.18 – Ciclo de vida NetFlow. (CISCO, 2004).

O processo de comunicação entre o sensor e o coletor é realizado por meio do protocolo de exportação. Historicamente o UDP é o protocolo escolhido para exportar os registros NetFlow, por minimizar o impacto aos elementos da rede. Entretanto, a ausência de confiabilidade de entrega pode ocasionar um impacto negativo no processo como um todo, visto que a perda de um pacote que compõe um conjunto de fluxos agregados coloca em descrédito todo o sistema de monitoramento baseado em fluxos.

Alternativamente, permite-se que os sensores possam exportar seus agrupamentos de fluxos para até dois coletores e recomenda-se que estes estejam dispostos em rotas alternativas para garantia de redundância do caminho do tráfego.

Paralelamente a alternativa de redundância no processo de exportação, o protocolo NetFlow em sua última versão apresenta uma opção de exportação de fluxos baseado no protocolo SCTP (*Stream Control Transpot Protocol*).

O protocolo SCTP definido pela RFC 2960, juntamente com sua extensão de implementação de um protocolo parcialmente confiável (PR-SCTP, definido pela RFC 3758) é um protocolo da camada de transporte, orientado a mensagens que permite a transmissão de dados entre dois pontos de maneira totalmente confiável, parcialmente confiável ou não confiável (STEWART, 2007).

Quando o SCTP estiver operando em modo de confiabilidade total, é usado um esquema de reconhecimento seletivo para garantir a entrega ordenada de mensagens. A pilha de protocolo SCTP armazena as mensagens em um *buffer* até que o recebimento seja confirmado pelo coletor NetFlow. Possui ainda, um mecanismo de controle de congestionamento que pode ser usado para limitar a quantidade de memória consumida pelo SCTP para o *buffer* de pacotes.

Quando o SCTP estiver operando em modo parcialmente confiável, um limite é colocado sobre a quantidade de memória dedicada ao armazenamento dos pacotes não confirmados. Se o limite for excedido e o sensor tentar armazenar outro pacote no *buffer*, o pacote mais antigo será descartado e uma mensagem denominada *forward-TSN* (número de sequência de transmissão) é encaminhada ao coletor para indicar que este pacote não deverá ser confirmado. Esse mecanismo impede o NetFlow de consumir toda a memória livre de um dispositivo em uma situação que requeira o armazenamento de muitos pacotes no *buffer*, como quando o sensor experimentar tempos longos de resposta de um coletor.

Quando o SCTP operar em modo não confiável, o pacote é simplesmente enviado sem a utilização de armazenamento temporário em *buffer* ou qualquer mecanismo de confirmação de entrega, atuando de modo similar ao protocolo UDP.

Uma sessão SCTP consiste de uma associação entre dois pontos, que podem conter um ou mais canais lógicos chamados s*treams.* O modelo de transmissão baseado em s*treams* facilita a exportação de um conjunto diferente de tipos de dados através da mesma conexão.

O protocolo Netflow versão 9, permite a definição de um modelo de campos personalizados *templates*, que são transmitidos em um canal de *stream* diferente dos fluxos de dados exportados. O número máximo de *streams* de entrada e saída suportados pelos dispositivos NetFlow é negociado durante a inicialização do processo de associação do SCTP.

Quando um sensor NetFlow v9 é configurado para usar o protocolo SCTP são criados, no mínimo, dois *streams.* O primeiro *stream*, definido para transmissão totalmente confiável, é usado para o envio dos *templates* e suas opções, bem como as opções de registros. Os demais *streams*, que podem ser definidos para transmissões totalmente confiáveis, parcialmente confiáveis ou não confiáveis, são utilizados para o transporte dos dados.

No caso de indisponibilidade do coletor, e considerando que o SCTP é um protocolo orientado a conexões, um coletor *backup* previamente configurado pode assumir a função de coleta e ativar a associação SCTP com o sensor.

O coletor SCTP *backup* pode atuar de duas maneiras: modo recuperação de falhas e modo de redundância. Os modos se diferenciam pelo momento de ativação da associação SCTP. No modo recuperação de falhas, a ativação da associação SCTP com o coletor backup, ocorre quando detectada a indisponibilidade do coletor principal enquanto que no modo redundância a ativação da associação SCTP ocorre simultaneamente ao coletor principal, ficando o encaminhamento de mensagens para o momento da detecção de indisponibilidade do coletor principal.

2.4.1.4. Versões do NetFlow

A primeira versão do formato de exportação (*NetFlow v1*) suportado inicialmente pelo sistema operacional dos roteadores cisco, não é mais utilizada atualmente em virtude da ausência de um controle do sequenciamento dos fluxos, disponível em versões posteriores.

Os campos chave para a composição dos fluxos NetFlow v1 são: endereços IP de origem e destino; Portas de origem e destino; Protocolo IP, ToS e interface de entrada. Efetua, ainda, a contabilidade de: pacotes, octetos, tempos de início e fim e interface de saída.

As tabelas 2.6 e 2.7, exibem o formato do cabeçalho e o formato dos registros dos fluxos, respectivamente, da primeira versão do protocolo NetFlow.

Tabela 2.6 – Formato do cabeçalho do NetFlow v1 (CALIGARE, 2012)

Bytes	**Campos**	**Descrição**
0-1	version	*Número da versão do formato de exportação NetFlow.*
2-3	count	*Números de fluxos exportados neste pacote (1-24).*
4-7	sys_uptime	*Tempo em milissegundos desde a reinicialização do dispositivo.*
8-11	unix_secs	*Data e hora baseados em segundos desde 000 UTC 1970.*
12-16	unix_nsecs	*Nanosegundos residuais no formato timestamp Unix 000 UTC 1970.*

Tabela 2.7 – Formato do registro dos fluxos do NetFlow v1 (CALIGARE, 2012)

Bytes	**Campos**	**Descrição**
0-3	**srcaddr**	*Endereço IP de origem.*
4-7	**dstaddr**	*Endereço IP de destino.*
8-11	nexthop	*Endereço Ip do roteador do próximo hop.*
12-13	**input**	*Índice SNMP da interface de entrada.*
14-15	output	*Índice SNMP da interface de saída.*
16-19	dPkts	*Pacote no fluxo.*
20-23	dOctets	*Numero total de bytes Layer 3 nos pacotes do fluxo.*
24-27	first	*SysUptime do início do fluxo.*
28-31	last	*SysUptime do último pacote recebido no fluxo.*
32-33	**srcport**	*Portas de origem TCP/UDP ou equivalente.*
34-35	**dstport**	*Portas de destino TCP/UDP ou equivalente.*
36-37	pad1	*Bytes não usados (zeros).*
38	prot	*Tipo do protocolo (exemplo, TCP=6; UDP = 17)*
39	**tos**	*Tipo do serviço IP.*
40	flags	*TCP flags ou cumulativo.*
41-48	pad2	*Bytes não usados (zeros)*

A versão 5 do protocolo NetFlow, foi a evolução da primeira versão e trouxe consigo duas principais extensões: as informações de origem e destino dos sistemas autônomos (AS) e a adição no cabeçalho do pacote de exportação do número de sequência, conforme pode ser identificada nas tabelas 2.8 e 2.9 e pela figura 2.19.

Tabela 2.8 – Formato do cabeçalho do NetFlow v5 (CALIGARE, 2012)

Bytes	**Campos**	**Descrição**
0-1	version	*Número da versão do formato de exportação NetFlow.*
2-3	count	*Números de fluxos exportados neste pacote (1-24).*
4-7	sys_uptime	*Tempo em milissegundos desde a reinicialização do dispositivo.*
8-11	unix_secs	*Data e hora baseados em segundos desde 000 UTC 1970.*
12-15	unix_nsecs	*Nanosegundos residuais no formato timestamp Unix 000 UTC 1970.*
16-19	flow_sequence	*Contagem sequencial do total de fluxos enviados*
20	engine_type	*Tipo do flow-switching*
21	engine_id	*Número do slot do engine flow-switching*
22-23	sampling_interval	*Os primeiro dois bits controlam o modo de amostragem enquanto que os 14 bit's definem o intervalo da amostragem.*

Tabela 2.9 – Formato do registro dos fluxos do Netflow v5. (CALIGARE, 2012)

Bytes	**Campos**	**Descrição**
0-3	**srcaddr**	*Endereço IP de origem.*
4-7	**dstaddr**	*Endereço IP de destino.*
8-11	nexthop	*Endereço Ip do roteador do próximo hop.*
12-13	**input**	*Índice SNMP da interface de entrada.*
14-15	output	*Índice SNMP da interface de saída.*
16-19	dPkts	*Pacote no fluxo.*
20-23	dOctets	*Numero total de bytes Layer 3 nos pacotes do fluxo.*
24-27	first	*SysUptime do início do fluxo.*
28-31	last	*SysUptime do último pacote recebido no fluxo.*
32-33	**srcport**	*Portas de origem TCP/UDP ou equivalente.*
34-35	**dstport**	*Portas de destino TCP/UDP ou equivalente.*
36	pad1	*Bytes não usados (zeros).*
37	tcp_flags	*TCP flags ou cumulativo.*
38	prot	*Tipo do protocolo (exemplo, TCP=6; UDP = 17)*
39	**tos**	*Tipo do serviço IP.*
40-41	src_as	*Número de origem do sistema autônomo.*
42-43	dst_as	*Número de destino do sistema autônomo.*
44	src_mask	*Bit's no prefixo da máscaras de endereço de origem.*
45	dst_mask	*Bit's no prefixo da máscara de endereço de destino.*
46-47	pad2	*Bytes não usados (zeros)*

Com a extensão das informações dos sistemas autônomos (AS), tornou-se possível a identificação agregada dos fluxos entre os *AS* o que habilitou o seu uso pelos provedores de acesso. Quanto à extensão do número de sequência, este número pode ser usado pelos aplicativos coletores para a detecção e relatório de registros de fluxos perdidos. A detecção de

sequência é verificada pela validação de igualdade entre o número de sequência entre o pacote que ingressou com a somatória do número de sequência do pacote mais a contagem de fluxo no pacote anterior.

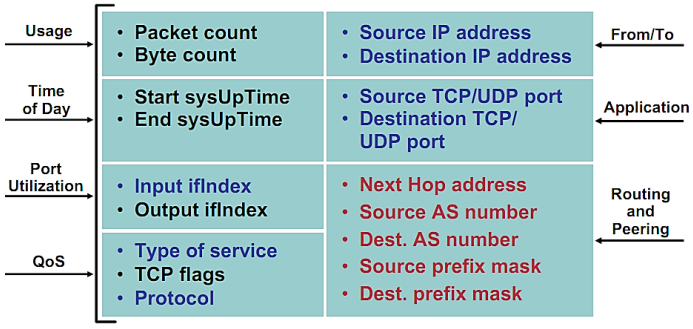


Figura 2.19 – Formato de Exportação NetFlow versão 5 (CISCO, 2004).

A versão 6 foi uma versão cujo desenvolvimento não se tornou público sendo muito similar à versão 5. Na versão 7 o formato de registros de fluxos incluiu informações de chaveamento *switching e* roteamento que não estavam disponíveis na versão 5 e permitiu o suporte à nova linha de equipamentos cisco ao NetFlow.

A versão 8, ofereceu um conjunto 13 tipos de agregação para os formatos de registros de fluxos, sendo os agrupamentos definidos por: *Router AS; Router Protocol; Router DSTPrefix; RouterSRCPrefix; Router Prefix; TOAS; TOSProtocol; PReportProtocol; ToSRCPrefix; ToSDSTPrefix; TosPrefix; DestOnly; SrcDst e FullFlow*. Como resultado, há uma menor necessidade de largura de banda e requisitos de recursos da plataforma nos coletores para a implementação do protocolo NetFlow. Como ficou suportada apenas pelos dispositivos CISCO, esta versão 8 do protocolo NetFlow não é amplamente utilizada em soluções gerenciamento de fluxo.

As atuais versões desenvolvidas até então possuíam um formato fixo e inflexível, quanto à definição dos formatos de exportação. Com base nessas características, seria necessário o desenvolvimento de novas versões para cada necessidade, bem como, a execução de reengenharia para o suporte do novo formato de exportação em casos de integração com novas versões.

Surge então a versão 9 do NetFlow, totalmente extensível e baseado em modelos *template-based*, permitindo a obtenção de informações arbitrárias para os registros NetFlow, admitindo uma maior integração com soluções de gerenciamento de terceiros, sendo ainda, a primeira versão a ter o suporte ao protocolo *IPv6*. Os pacotes de exportação Netflow v9[1],

[1] O Anexo A, contém todas as especificações do NetFlow v9.

iniciam pelo cabeçalho, conforme ilustrado pelas figuras 2.20 e 2.21, são seguidos por um ou mais conjuntos de fluxos *flowSets* diferentemente das versões anteriores que apresentavam os registros de fluxos de forma direta.

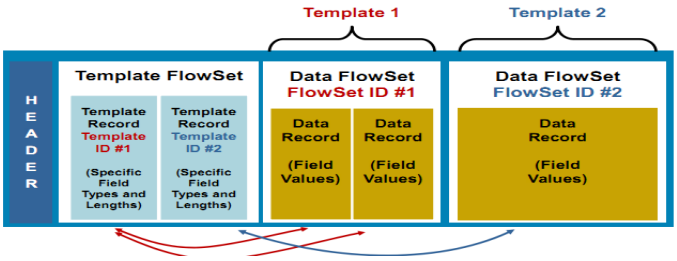


Figura 2.20 – Formato do pacote de exportação NetFlow v9 (CISCO, 2004).

Outra modificação desta versão se refere ao modo como é controlado o sequenciamento de exportação entre os sensores e coletores. Desde a versão 5 o NetFlow fazia a contagem sequencial do total de fluxos por meio do campo *flow_sequence* no cabeçalho. No Netflow v9, o controle de sequenciamento é realizado pelo campo *package*_sequence que determina a quantidade de pacotes exportados do sensor ao coletor e não mais pela quantidade de fluxos.

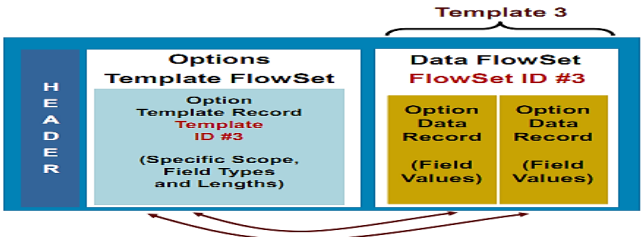


Figura 2.21 – Formato do Options Template Flowset (CISCO, 2004).

Segundo (KREJCÍ, 2009), os conjuntos de fluxos podem ser: Modelos (*Template FlowSet*); Opções de Modelos (*Options Template FlowSet*) e Dados (*Data FlowSet*).

Os modelos *templates*, contêm as informações estruturais sobre os campos dos registros de fluxos e são usados pelos coletores para determinar onde e qual informação será gravada do registro de fluxo recebido. Caso um coletor não entenda a semântica de algum campo, este é capaz de ignorá-lo e continuar com o processamento dos campos seguintes, tornando o novo protocolo resistente a incompatibilidades em conjuntos de campos suportados entre os sensores e coletores permitindo a redução do volume de dados exportados e consequentemente redução do espaço de disco pelos coletores.

A utilização do conceito de registros de dados, baseados em modelos é que trouxe a flexibilidade ao protocolo para definição dos formatos de registros de forma arbitrária, com base nas necessidades dos clientes. Os *Templates FlowSets* contém os modelos que definem a estrutura de dados dos registros, ou *Data FlowSet.* Os campos dos modelos são definidos

quanto ao seu tipo e tamanho um campo específico (*Template ID*) faz a associação dos dados com o modelo.

Um tipo especial de modelo (*Option Template FlowSet*), representa um conjunto de parâmetros que são válidos para todos o conjunto de registros de um escopo específico. Normalmente estes parâmetros descrevem algum comportamento do dispositivo de exportação, tais como: parâmetros de amostragem e modelo do equipamento *engine*. O escopo representa a quem os parâmetros se referem abarcando os seguintes elementos: Sistema, Interface, Cache e Modelos.

Finalmente, os dados de um conjunto de fluxos (*Data FlowSet*) representam um conjunto de registros de fluxos com base nas definições de um modelo e suas opções que descrevem a estrutura desses registros.

Depois de definidos pelos modelos, os fluxos de dados podem ser apresentados nos pacotes de exportação em qualquer ordem. Os pacotes de exportação podem conter: os modelos de fluxos, os conjuntos de dados ou uma combinação dos dois. Todos os modelos devem ser armazenados pelo coletor e usados pelos fluxos de dados para se associar ao modelo pelo *template ID*. A figura 2.22, ilustra o esquema de exportação do formato NetFlow v9 pela associação entre os fluxos de dados e os modelos pelos campos: *template ID= 256* e *FlowSet ID= 256*.

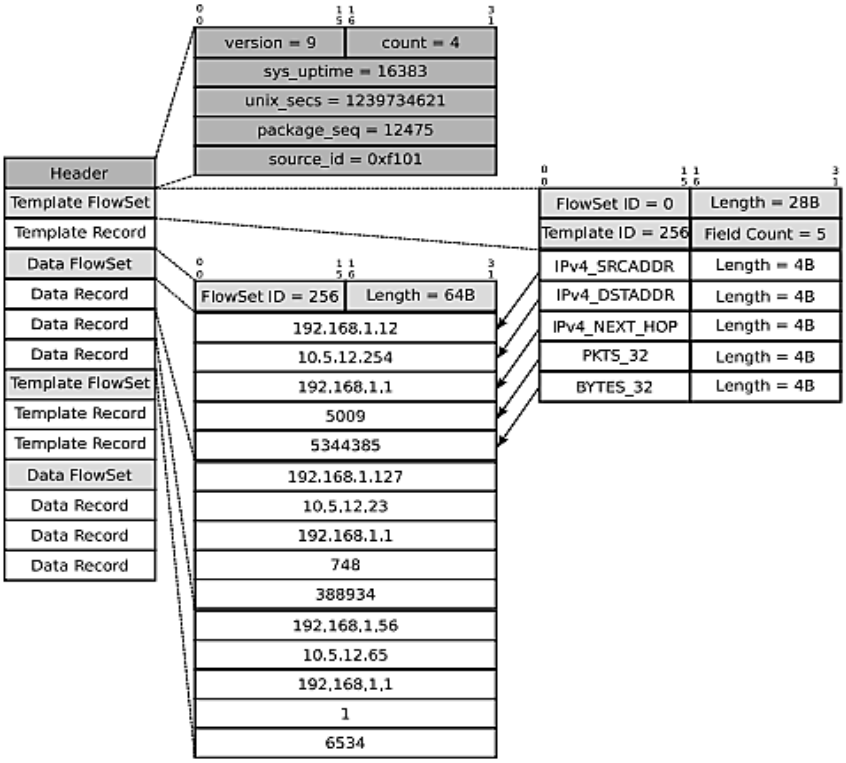


Figura 2.22 – Esquema do formato NetFlow v9 (KREJCÍ, 2009).

2.4.1.5. Ferramentas NetFlow

Muito embora o protocolo NetFlow fosse originalmente desenvolvido para gerar e coletar os fluxos em um dispositivo proprietário, existe um grande número de aplicativos livres e comerciais que atuam como sensores e coletores, permitindo a implantação flexível de um projeto de monitoramento com NetFlow. A identificação do aplicativo que se encaixe nas necessidades de monitoramento é uma das questões a se considerar para o sucesso do projeto.

Segundo (CISCO, 2004), a identificação de como serão utilizados os dados NetFlow facilita a identificação da ferramenta que se acopla de forma mais eficiente e eficaz às necessidades de monitoramento. Essa identificação pode ser facilitada por meio dos seguintes questionamentos:

- Qual será o principal uso do NetFlow? Segurança; Planejamento de capacidades; Análise de tráfego; Inclui o monitoramento de aplicativos e usuários?
- Os relatórios devem trazer informações em tempo-real ou históricos mais importantes?
- Qual é o sistema operacional preferencial utilizado no servidor?
- Qual o tamanho da implantação e o qual o grau de escalabilidade do projeto?
- Qual o valor disponível para o investimento na solução de monitoramento?
- Existem produtos atualmente disponíveis na infraestrutura que podem ser incrementados para suportar a solução de monitoramento NetFlow?

Uma vez identificadas as necessidades de gerenciamento, parte-se para a definição da solução que atenda os requisitos levantados. Como auxílio no processo de escolha, foram elencadas diversas soluções de software livres e comerciais, relacionando suas principais características, conforme descrito nas tabelas 2.10 e 2.11.

Tabela 2.10 – Produtos Comerciais NetFlow (CISCO, 2004).

Produto	**Uso Primário ***	**Tipo de Usuários ****	**Sistema Operacional**	**Custo *****
Cisco NetFlow Collector	AT	*ENT / PS*	*Linux, Solaris*	*Médio*
Cisco CS-Mars	MS	*ENT / SMB*	*Linux*	*Médio*
AdventNet	AT	*ENT / SMB*	*Windows*	*Baixo*
Apoapsis	AT	*ENT*	*Linux*	*Médio*
Arbor Networks	AT / MS	*ENT / PS*	*BSD*	*Alto*
Caligare	AT /MS	*ENT / PS*	*Linux*	*Médio*
Fluke Networks	AT	*ENT / SMB*	*Windows*	*Médio*

CA Software	AT	*ENT / PS*	*Windows*	*Alto*
Evident Software	AT / CT	*ENT*	*Linux*	*Alto*
HP	AT	*ENT / PS*	*Linux, Solaris*	*Alto*
IBM Aurora	AT /MS	*ENT / PS*	*Linux*	*Médio*
IdeaData	AT	*ENT*	*Windows, Linux*	*Médio*
InfoVista	AT	*ENT / PS*	*Windows*	*Alto*
IsarNet	AT	*ENT / PS*	*Linux*	*Médio*
Landscope	AT / MS	*ENT / PS*	*Linux*	*Alto*
Micromuse	AT	*ENT / PS*	*Solaris*	*Alto*
NetQoS	AT / MS	*ENT*	*Windows*	*Alto*
Valencia Systems	AT	*ENT*	*Windows*	*Alto*
Solarwinds	AT	*ENT / SMB*	*Windows*	*Baixo*
Wired City	AT	*ENT*	*Windows*	*Alto*

** AT (Analisador de Tráfego); MS (Monitoramento de Segurança); CT (Contabilidade.)*
*** ENT (Enterprise); SMB (Pequenos negócios); PS (Provedor de Serviços).*
**** Custo: Baixo < U$ 7.5 mil, Médio < U$ 25mil, Alto > U$ 25 mil.*

Tabela 2.11 – Produtos OpenSource NetFlow. (CISCO, 2004)

Produto	Uso Primário *	Comentários	Sistema Operacional
CFlowd	AT	*Não possui suporte.*	*Unix, Linux*
Flow-tools	**CF**	***Netflow v1, v5, v7, v9. Bom suporte da comunidade.***	***Unix, Linux***
Flowd	CF	*Netflow v1, v5, v7, v9.*	*BSD, Linux*
Fprobe / Fprobe-ulog	**AT**	***Suporta v5. Utiliza biblioteca libpcap ou diretamente no kernel***	***Linux***
IPFlow	AT	*NetFlow v9, IPv4/v6, MPLS, SCTP, etc.*	*Linux, FreeBSD, Solaris*
Nfdumd	AT	*Suporta Netflow v5, v7, v9. Bom suporte da comunidade.*	*Linux, BSD*
Nfsen	GR	*Trabalha em conjunto com nfdump.*	*Linux, BSD*
NetFlow Monitor	AT	*Netflow v9.*	*Linux*
NTOP	CF	*NetFlow v5, v9 e faz inspeção DPI.*	*Unix, Linux*
PFFlowd	AT	*NetFlow v5, Ipv4 e Ipv6*	*OpenBSD*
SiLK	CF	*Netflow v1, v5, v7, v9.Desenvolvido por CERT NetSA Security suíte.*	*Linux, Solaris, BSD, Mac OS X*
SoftFlowd	AT	*NetFlow v1, v5 e v9*	*Linux, OpenBSD*
Stager	GR	*Bom suporte da comunidade.*	*Unix, Linux*

** AT (Analisador de Tráfego); CF (Coletor de Fluxos); MS (Monitoramento de Segurança); GR (Gerador de relatórios de fluxos)*

Nos procedimentos de pesquisa desta dissertação, os dados de fluxos utilizados como insumos para a plataforma de *BI*, foram gerados pelo *fProbe* (FPROBE, 2012), utilizado como sensor NetFlow v5 e coletados para armazenamento persistente em disco pela ferramenta *flow-tools.*

O conjunto de aplicativos *flow-tools* efetua a coleta dos fluxos por meio da ferramenta *flow*-capture e a exportação para o banco de dados *MySQL* pela ferramenta *flow-export* (FULLMER, 2012).

2.4.2. IPFIX

Segundo (IETF, 2011) o protocolo IPFIX deve ser o sucessor de vários protocolos proprietários, sendo o padrão comum para o monitoramento de redes baseado em exportação de fluxos composto por dispositivos de sensores e coletores que processam e armazenam os dados. O desenvolvimento pelo IEFT do IPFIX iniciou-se em 2002 e através do padrão RFC 3917, foram definidos os requisitos que norteou o grupo de trabalho na confecção do novo protocolo.

Dentre as diversas definições, o item 4 da norma RFC 3917, aborda os requisitos de geração dos fluxos que devem ser analisados sob os seguintes aspectos: que os fluxos deveriam ser estabelecidos em pacotes que não fossem criptografados, aonde os dados de todos os campos do cabeçalho estivessem disponíveis para análise; que a identificação do sentido dos fluxos pudessem ser baseados quanto à entrada de pacotes nos sensores pelas interfaces de entrada e saída; que a geração dos fluxos fossem definidos com base em alguns campos do cabeçalho (*Versão do endereços IP; IP origem/destino; Tipo de protocolo; Portas*); que os sensores tivessem suporte ao MPLS e codificação Diffserv na geração e classificação dos fluxos.

Os requisitos abordam ainda, questões relativas ao processo de medição dos indicadores dos fluxos gerados. No item 5 da norma RFC 3917, foi abordado os seguintes pontos: que o processo de medição fosse confiável a ponto de identificar divergências entre o que foi definido para sua execução e sua efetiva capacidade de medição; que o processo de medição pudesse utilizar diversos métodos de amostragem, tais como, sistemática ou de seleção aleatória de um subconjunto de elementos (amostra) de uma população previamente definida; que o processo de medição fosse capaz de identificar um comportamento de sobrecarga reagindo de forma a alterar o processo de medição para diminuí-la; dentre outras características.

No item 6 da norma RFC 3917, os requisitos de exportação dos fluxos foram abordados. Nestes, ficou definido que: o modelo de informação para exportação dos fluxos deve ser extensível e expansível; que a transferência dos fluxos deve incluir mecanismos de

confiabilidade, controle de congestionamento e segurança; que o processo de exportação dos fluxos deve suportar os modos de solicitação e encaminhamento (*pull/push*), dentre outros.

Com base nos requisitos supracitados, o grupo de trabalho IPFIX examinando a conformidade e aplicabilidade de diversos protocolos candidatos: *CRANE, Diameter, LFAP, Netflow v9 e Streaming IPDR,* recomendou o uso do NetFlow v9 como base para as definições do novo protocolo IPFIX, através da RFC 3955.

O IPFIX, então, assume a estrutura geral do formato de exportação do protocolo NetFlow v9, ou seja, traz consigo a estrutura flexível e escalável de definições abarcadas pelos conceitos de: Modelo do conjunto de fluxos (*Template FlowSet*); Opções do modelo do conjunto de fluxos (*Options Template FlowSet*) e o conjunto de dados de fluxos (*Data FlowSet*), conforme ilustrado pela figura 2.23.

Figura 2.23 –Formato de exportação do IPFIX (KREJCÍ, 2009).

No pacote de cabeçalho do IPFIX, foi introduzido o conceito de identificação global dos sensores através do campo *Observation Domain ID*. A identificação global pode ocorrer ou não dependendo desta configuração, mas quando efetivada permite a identificação única dos sensores baseado em um ponto de observação dos fluxos. O gerenciamento dos identificadores dos fornecedores e consequentemente dos seus dispositivos é realizado de forma centralizada pela IANA.

Muito embora o IPFIX possua uma definição de estruturação dos dados a serem exportados pela RFC 6313, a implementação de novos elementos de informação, pelos fornecedores, é permitida pela flexibilidade na modelagem de dados através do campo *enterprise number*, conforme ilustrado na figura 2.24.

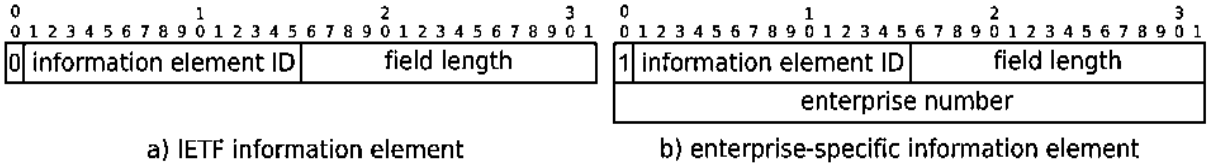


Figura 2.24 – Esquema de informações dos elementos IPFIX (KREJCÍ, 2009).

No IPFIX, assim como o NetFlow v9, o transporte dos dados exportados pode ser realizado de forma não segura, parcialmente segura ou totalmente segura por meio dos protocolos UDP, PR-SCTP e SCTP, respectivamente. Adicionalmente o IPFIX também

permite o uso do TCP para transmissões seguras. Como requisito, os sensores devem ser capazes de exportar os dados a múltiplos coletores usando diferentes protocolos de transporte.

A análise dos pacotes que entram nos sensores pode ser realizada de forma ordenada um-a-um, porém pode ser necessária a análise amostral do conjunto de pacotes que passam pelos sensores. Essa necessidade pode ser oriunda de uma sobrecarga nos subsistemas do sensor que realiza a criação, manutenção e exportação dos fluxos aos coletores ou simplesmente baseado em predeterminação arbitrária.

Considerando que o tratamento da sobrecarga é um dos requisitos basilares do IPFIX, a incorporação da análise amostral se faz parte integrante de seus aspectos funcionais, pois permite a continuidade da análise reduzindo a carga aplicada ao sensor. Diferentemente do Netflow v9, que programa o suporte a análise amostral diretamente no protocolo, através dos campos 34 e 35 (*sampling_interval, sampling_algorithm*) o IPFIX faz o uso do protocolo PSAMP para extensão do seu modelo de informação relacionado aos processos de amostragem.

O protocolo PSAMP, define os elementos de informações que descrevem os diversos processos de amostragem, tais como: Amostragem sistemática baseado em contagem; Amostragem sistemática baseada em temporização; Amostragem aleatória n-para-N; Amostragem uniforme probabilística; Amostragem baseada em regras de filtros; Amostragem baseada em filtros de Hash.

A amostragem sistemática baseado em contagem coleta os pacotes em uma ordem sequencial pré-definida. Conforme ilustrado na figura 2.25, a análise amostral ocorre a cada 5 pacotes, independente do tamanho ou tipo do protocolo do pacote.

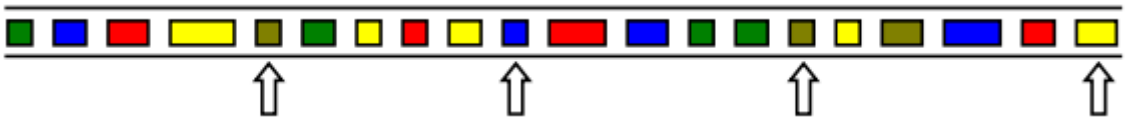


Figura 2.25 – Amostragem sistemática baseada em contagem (KREJCÍ, 2009).

A análise sistemática baseada em temporização, coleta independente da quantidade de pacotes, tipo ou tamanho e sim baseado na temporização pré-determinada, conforme ilustrado pela figura 2.26.

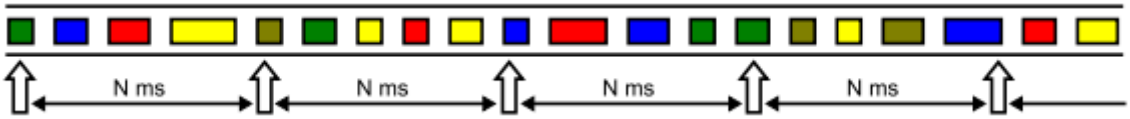


Figura 2.26 – Amostragem sistemática baseada em temporização (KREJCÍ, 2009).

A amostragem aleatória n-para-N ilustrada pela figura 2.27, determina um espaço amostral sequencial de pacotes e nestas alocações, a coleta aleatória. No caso ilustrado os valores definidos são: n=1 para N=5, ou seja, coleta 1 pacote aleatório do conjunto sequencial composto por blocos de 5 pacotes.

Figura 2.27 – Amostragem aleatória n-para-N (KREJCÍ, 2009).

A amostragem probabilística uniforme determina o cálculo de probabilidade para cada pacote e o seleciona, no caso de sua probabilidade se situar em um valor acima de um valor previamente determinado. A amostragem baseada em regras de filtros coleta todos os pacotes caso o valor de um campo do pacote seja igual ao valor previamente definido. A amostragem baseada em filtros de hash coleta todos os pacotes cujo hash do cabeçalho IP usados como entrada estiver dentro de um intervalo selecionado.

A análise por amostragem é, segundo (CHENG, 2008), uma das contramedidas possíveis quando o processo de agregação de fluxos consumir uma quantidade excessiva de recursos do dispositivo, a custa de menor precisão na contabilidade dos fluxos. Esse comportamento é comum em redes de alta velocidade, como descrito em (LEE, 2010), devido ao alto volume de tráfego que demandam alto poder de processamento ou técnicas de adaptativas de coleta.

O grupo de trabalho PSAMP do IETF é o grupo que padroniza o conjunto de funcionalidades relativas às técnicas de amostragem para um subconjunto estatístico de pacotes. Em (CHAUDURI, MOTWANI, & NARASAYYA, 1998), foi provado que a amostragem pode ser compensada pela estimação de tráfego em pacotes ou bytes. Ainda, conforme proposto em (KUMAR, XU, LI, & WANG, 2003), o SCBF pode executar ações baseados em contagens de fluxos sem necessariamente manter o controle do estado em um algoritmo de estimação de distribuição de fluxos (KUMAR & all, 2004). Esta última abordagem também foi seguida por (DUFFIELD, N. G.; LUND, C.; THORUP, M., 2003), que em seu trabalho inferiu através de amostragens estatísticas a distribuição de fluxos.

As propriedades estatísticas para amostragem no nível de pacotes usando rastros do tráfego de internet real foram estudadas por (DUFFIELD, N. G.; LUND, C.; THORUP, M., 2002), que desenvolveu um modelo para predizer tanto a taxa exportada de fluxos em uma estatística de fluxos de uma amostragem de pacotes quanto o número de fluxos ativos. A medição do número de fluxos tanto quanto sua distribuição de tamanhos foi analisada em (FELDMANN & all, 1999), para avaliar melhoramentos no desenvolvimento de *proxys web* e na determinação de limites de configuração para conexões em redes baseadas em fluxos.

Em suas pesquisas (RIBEIRO, TOWNSLEY, YE, & BOLOT, 2006) realizou uma abordagem sistemática para entender as contribuições que diferentes tipos de informação em ambiente de amostragem de pacotes possuem na qualidade do nível de fluxo estimado e constatou que existem soluções de medida que melhoram a precisão estimada dos fluxos observados.

Conforme pode se concluir, baseado nas pesquisas supracitadas, é que a amostragem é um método que otimizado para uma determinada situação específica pode atender demandas de gerenciamento também específicas com a vantagem de não consumir excessivamente os recursos disponíveis nos dispositivos.

No NetFlow v9, as definições de data e hora são relativas ao tempo em que o dispositivo foi inicializado, portanto a determinação da data e hora absoluta obedecem ao cálculo: *absoluteTime = bootRelativeTime +(UNIXSecs – sysUpTime)*.

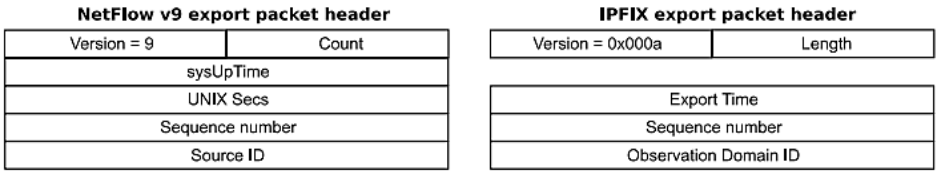


Figura 2.28 – Cabeçalho dos pacotes de exportação NetFlow v9 e IPFIX (KREJCÍ, 2009).

No protocolo IPFIX, o campo *sysUpTime* foi excluído do cabeçalho, permitindo maior liberdade para as definições de data e hora entre os dispositivos sensores e coletores, pois os mesmos podem contratar um acordo de formato absoluto (*que não necessita de conversão*) ou relativo como acontece no NetFlow, sendo ainda possível a definição da precisão com o qual serão armazenados.

2.4.3 sFlow

O protocolo sFlow, segundo (PHAAL, PANCHEN, & MCKEE, 2001), é uma tecnologia de amostragem que embutido nos chaveadores (*switches*) e roteadores, que

fornece-lhes a habilidade de monitorar continuamente o tráfego de rede, que flui em todas as interfaces simultaneamente.

O projeto de monitoramento passivo baseado em amostragem foi abordado pela RFC 3176, que em seu escopo abordou três características basilares no seu desenvolvimento: **precisão** ao monitoramento de redes de alta performance, através do ajuste dinâmico nos parâmetros das técnicas de amostragem; **escalabilidade**, que se reflete na capacidade do sistema gerenciar inúmeros sensores de um ponto centralizado; e que fosse de **baixo custo** de implementação.

Os dispositivos que programam a solução sFlow, consistem em sensores e coletores. Os sensores, denominados *agentes* nesta solução, possuem a função de monitorar o tráfego de rede e gerar os dados sFlow que será recebido pelo coletor. O coletor consiste em uma aplicação de softwares que analisará o tráfego sFlow recebido, gerando as métricas necessárias para o gerenciamento, conforme figura 2.29 (sFlow, 2001).

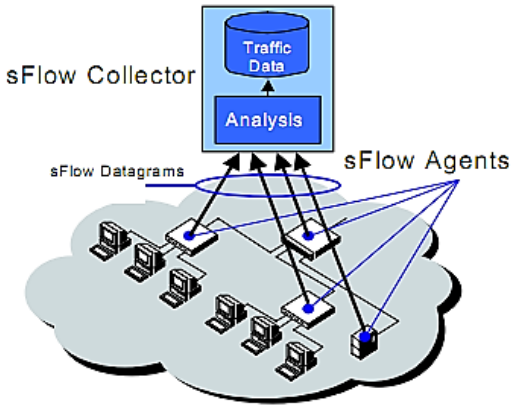


Figura 2.29 – Agentes de sensores e coletores sFlow (sFlow, 2001).

Os dispositivos que atuam como sensores, na geração dos dados sFlow possuem dois componentes funcionais principais, sendo: a implementação de *hardware* especializado *ASIC* (*Application-Specific Integrated Circuit*) para a execução do processo de amostragem do tráfego de rede; e a implementação do agente que faz o encaminhamento dos *Datagramas sFlow* aos coletores, conforme ilustrado pela figura 2.30.

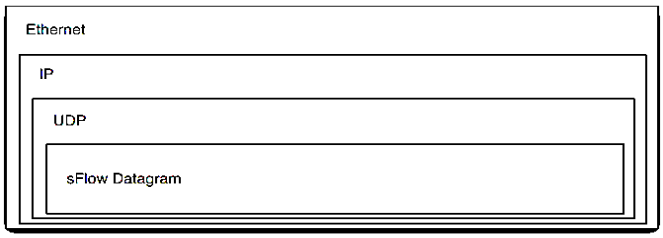


Figura 2.30 – Diagrama do Datagrama sFlow (sFlow, 2001).

Segundo (sFlow, 2001), o sFlow pode utilizar-se de dois mecanismos de amostragem de tráfego para obtenção de suas informações, sendo: amostragem estatística baseada em pacotes; ou amostragem de contadores baseada no tempo. A atuação dos dois mecanismos pode ser visualizada nas figuras 2.25 e 2.26 respectivamente.

Diferentemente do protocolo *NetFlow* e *IPFIX*, que mantém o controle de estado das conexões a fim de obter estatísticas, por amostragem ou não, da quantidade de pacotes e octetos transmitidos no contexto que denominam fluxos, o protocolo sFlow aborda a confecção de informações gerenciais pelo monitoramento totalmente baseado em amostragem sem o controle de estado das conexões.

Por ter seu mecanismo de amostragem de pacotes, ilustrado pela figura 2.31, implementado em *hardware* especializado, o procedimento de coleta de dados no sensor não afeta o desempenho no encaminhamento de pacotes, sendo considerada uma opção de monitoramento passivo de baixo custo operacional para o ambiente de alto desempenho como em redes gigabits (sFlow, 2001).

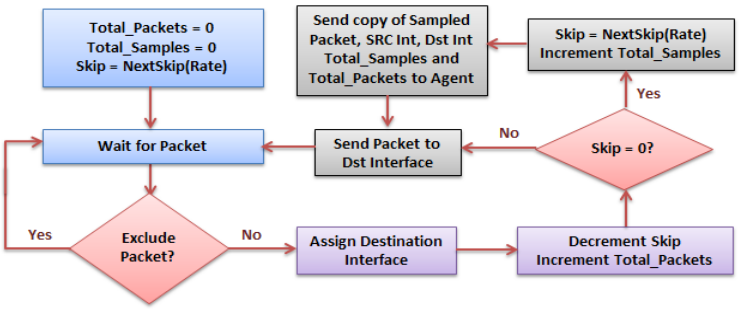


Figura 2.31 – Fluxograma da amostragem. Adaptado de (sFlow, 2001).

A precisão no monitoramento, baseado em amostragem, implementado pela solução sFlow se reflete na definição dos parâmetros do algoritmo de amostragem que segundo (PHAAL, PANCHEN, & MCKEE, 2001), ocorre mediante geração de números aleatórios que permite a convergência para a melhor taxa de amostragem. Essa convergência resulta em valores que possuem uma variação da taxa de erro da ordem de 10%, sendo considerado um valor adequado.

Em uma análise comparativa de precisão dos protocolos NetFlow e sFlow, realizado por (REESE, 2010), ficou demonstrado que neste estudo pontual a precisão de estimação do tráfego baseado na amostragem do protocolo sFlow em relação ao protocolo NetFlow foi da ordem de 25%, ou seja, enquanto o Netflow, reportou um volume de tráfego *SSH* (*Secure Shell*) do IPv4 de origem 66.186.184.219 em 225,44 Mb o sFlow estimou um volume de apenas 50,46Mb.

Muito embora a análise comparativa acima descredencie o sFlow para análise de precisão, esta conclusão não pode ser considerada indistintamente pelo fato de que não foi desvelado neste estudo os parâmetros de amostragem utilizados no sFlow, bem como a realização de uma análise temporal mais ampla para identificação do processo de convergência da solução.

2.5. SÍNTESE DO CAPÍTULO 2

Este capítulo procurou abordar os aspectos relativos ao gerenciamento de redes de computadores, passando pela sua conceituação e classificações para focar nos aspectos referentes ao gerenciamento passivo.

Inicialmente foi abordada a fundamentação teórica acerca do SNMP, protocolo clássico do gerenciamento passivo de redes de computadores. Em sequencia, os tópicos de suporte ao gerenciamento baseado no monitoramento por captura de pacotes, finalizando pela fundamentação dos protocolos de monitoramento de fluxos: NetFlow, IPFIX e sFlow.

O foco principal do estudo deste capítulo aborda as questões relacionadas aos protocolos de monitoramento de fluxos, em seus aspectos de: projeto e arquitetura de um sistema de coleta de fluxos, formas de comunicação entre coletores e agentes, características do protocolo em seus diversos versionamentos e capacidades de amostragens.

A abordagem de gerenciamento por SNMP permite a identificação estatística do tráfego de rede em suas bases de monitoramento remotas RMON e RMON-2, bem como algumas informações estáticas dos dispositivos coletores, enquanto que na abordagem de coleta de pacotes a análise pode ser dada em qualquer granularidade visto a disponibilidade do tráfego.

Muito embora estas duas abordagens tenham seu escopo de aplicação, não é possível pelo emprego do SNMP e muito dispendioso por meio da captura de pacotes a análise do tráfego em formato de registros de conexões (fluxos).

Conforme as diversas aplicabilidades definidas em (CLAISE, B.; WOLTER, R., 2007), em especial a capacidade de contabilização do tráfego, com a devida granularidade de informações geradas pela abordagem de monitoramento passivo por fluxos que habilita a gestão do tráfego de rede pela contabilização das conexões realizadas permitindo o incremento de um sistema escalável, esta abordagem foi considerada a mais adequada ao estudo adotado nesta dissertação.

Dentre os diversos protocolos de monitoramento passivo de fluxos o adotado neste trabalho é o Netflow v5, em virtude de que os dados necessários para o gerenciamento e contabilização dos tráfegos de um *backbone* de rede local já estão disponíveis nesta versão do protocolo e ainda devido a sua ampla utilização no mercado o que o torna uma referencia em aplicabilidade.

3. SISTEMAS E TECNOLOGIAS DE SUPORTE À DECISÃO

Em complexas organizações, publicas ou privadas, as decisões são realizadas basicamente de forma continuadas. As decisões podem ser mais ou menos críticas, terem impactos de longo e curto prazo e envolverem pessoas e papéis em vários níveis hierárquicos. A habilidade e conhecimento do negócio pelos colaboradores para a tomada de decisão, tanto individual quanto coletiva é um dos fatores que influenciam o desempenho e a força competitiva de uma organização.

Muitas decisões são tomadas baseadas em metodologias simples e intuitivas, que levam em conta elementos específicos, tais como: experiência, conhecimento dos domínios de uma aplicação e de informações disponíveis (VERCELLIS, 2008).

Esta abordagem conduz a um estilo estagnado no processo de tomada de decisões que é inapropriado em condições de instabilidade determinadas por rápidas e frequentes alterações no comportamento do ambiente.

Ainda, os processos de decisões em determinados ambientes são demasiado complexos e dinâmicos para serem tratados com métodos intuitivos, necessitando de uma abordagem mais rigorosa baseada em metodologias analíticas e modelos matemáticos, que evidenciem os valores estratégicos para o processo de decisão.

Neste intuito de rigor, os sistemas de *Business Intelligence* vêm justamente fornecer o conhecimento do negócio, por meio de ferramentas e metodologias que habilitam o processo de tomada de decisão de forma efetiva e oportuna.

A aplicação rigorosa de métodos analíticos no suporte ao processo de tomada de decisão força a explicitação da descrição dos critérios de avaliação de alternativas e dos mecanismos que regulam o problema a ser investigado. Permite a revelação, revisão e adequação da lógica do processo de decisão que se consolida em planos de ação baseados nas melhores decisões permitindo que os objetivos sejam alcançados de forma mais efetiva.

Segundo (VERCELLIS, 2008), em ambientes de flutuação com alto grau de competitividade e dinamismo a capacidade reagir rapidamente é um fator crítico de sucesso. Considerando que em um ambiente confiável de *business intelligence* o processo de tomada de decisões vá com o decorrer do tempo se tornando mais eficiente é possível a utilização de modelos matemáticos e algoritmos, para análise de um grande número de alternativas,

possibilitando a prospecção de conclusões de futuro mais precisas, auxiliando no processo de decisões oportunas.

Portanto, pode-se concluir que a principal vantagem decorrente da adoção de um sistema de BI é encontrada na maior eficácia do processo de tomada de decisão (*decisões efetivas e oportunas*).

Segundo (PONNIAH, 2010), os sistemas de suporte a decisões baseados em tecnologias da informação, muito embora variem em tamanho e natureza dos negócios, apresentam o seguinte histórico:

- ***Relatórios Ad hoc****:* neste estágio os usuários solicitavam relatórios específicos para se basear nas tomadas de decisões. Em contrapartida a atuação do departamento de TI era o de elaborar aplicativos para produzir estes relatórios específicos.

- ***Aplicativos Especiais de Extração:*** neste estágio o departamento de TI tentava antever as necessidades de relatórios que eram requisitados de tempos em tempos, pela programação de aplicativos que de forma automatizada extraíam os dados dos vários aplicativos.

- ***Pequenas Aplicações:*** neste estágio os processos de extração foram normalizados, permitindo a criação de pequenos aplicativos baseados nos dados extraídos.

- ***Centros de Informações:*** consistia tipicamente em um local aonde os usuários poderiam requisitar relatórios ou verificar informações no monitor.

- ***Sistemas de Suporte a Decisão:*** início da construção de sistemas sofisticados para o fornecimento de informações estratégicas, muito embora ainda baseados no processo de extração de dados, conforme nos modelos anteriores.

- ***Sistemas de Informações Executivas:*** é um modelo que se baseia na proximidade das informações estratégicas e seus tomadores de decisões, cujo principal critério é o da simplicidade e facilidade de uso. Estes sistemas se baseiam na exibição de indicadores diversos, vinculados às informações agregadas.

3.1. SISTEMAS DE SUPORTE A DECISÃO – DSS

Um sistema, de modo geral, é descrito como uma fronteira de atuação aonde percebe um conjunto de entradas e retorna um conjunto de saídas mediante transformação por processos regulados por condições internas e externas.

Um sistema frequentemente pode incorporar um mecanismo de retorno *feedback.* O *feedback* ocorre quando um componente do sistema gera um fluxo de saída que é usado como entrada em outro componente dentro do sistema, possibilitando uma transformação do resultado. Sistemas que são capazes de modificar seus próprios fluxos baseados em *feedback* são denominados de: *sistemas de ciclo fechados*.

Em conexão com o processo de tomada de decisões, é frequentemente necessário avaliar o desempenho dos sistemas sendo apropriada sua categorização em duas principais classes: efetividade e eficiência.

Segundo (VERCELLIS, 2008), em termos gerais, no estudo dos sistemas de suporte a decisão, a efetividade é uma métrica que indica o grau de acerto das decisões tomadas para o caminho dos objetivos enquanto que a eficiência é uma métrica que verifica se as decisões tomadas estão conduzindo para o melhor caminho ou não. Constitui-se que a métrica eficiência pondera de maior granulação e especialização que a métrica efetividade e ambas são complementares.

3.1.1. Representação do Processo de Tomada de Decisão

Uma decisão é uma escolha entre várias alternativas, usualmente realizadas com baixo grau de racionalidade. O processo de tomada de decisão é parte de um tema mais amplo, referenciado como: *resolução de problemas*, que se refere ao processo através do qual os indivíduos tentam preencher a lacuna entre as condições operacionais atuais de um sistema (*como é*) e as condições supostamente melhores a serem alcançados no futuro (*como será*).

Em geral, a transição de um sistema para o estado desejado implica a superação de certos obstáculos sendo que esta não é uma tarefa trivial. Isto força os tomadores de decisão da necessidade de elaborar um conjunto de opções alternativas viáveis para alcançar o objetivo desejado, e depois escolher uma decisão baseadas numa comparação entre as vantagens e desvantagens de cada opção. Assim, a decisão selecionada deve ser colocada em prática e, em seguida, verificada para determinar se ele permitiu que os objetivos planejados fossem alcançados. Quando isto não acontecer, o problema volta a ser reconsiderado, de acordo com uma lógica recursiva.

Conforme ilustrado na figura 3.1, as *Alternativas* representam as possíveis ações que visam resolver o problema dado e ajudar a alcançar o objetivo planejado. Em alguns casos, o

número de alternativas a ser considerado pode ser pequeno enquanto que outros casos a quantidade pode ser de infinitas possibilidades.

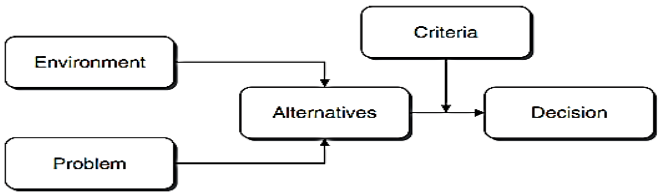


Figura 3.1 –Fluxo lógico do processo de resolução de problemas (VERCELLIS, 2008).

Os *Critérios* são as medidas de efetividade das várias alternativas e correspondem a diferentes tipos de desempenho do sistema, conforme exposto na figura 3.2. Uma abordagem racional no processo de decisão implica na seleção e escolha das alternativas baseando-as no melhor desempenho dos critérios, sendo estes selecionados em detrimento das alternativas com baixo desempenho de avaliação nos critérios.

Existem diversos critérios de restrição para avaliação das alternativas em um processo de resolução de problemas, dentre eles podemos destacar: econômicos, técnicos, legais, éticos, processual e político.

O processo de avaliação das alternativas pode ainda ser dividido em dois estágios principais, conforme ilustrado na figura 3.2, sendo: exclusão e avaliação. Durante o estágio de exclusão, as regras de compatibilidade e restrição são aplicadas a todas as ações alternativas identificadas. Enquanto que algumas alternativas são descartadas outras seguem como alternativas possíveis (*feasible alternatives*) e são comparadas uma a uma com os critérios básicos de desempenho, para que se defina a decisão preferencial como sendo a melhor oportunidade.

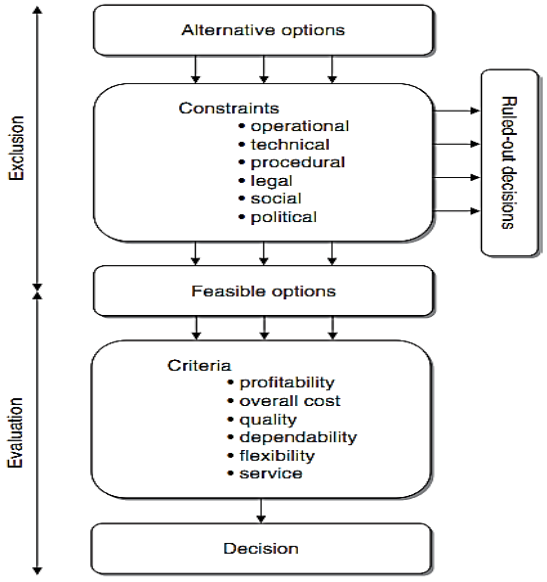


Figura 3.2 –Estrutura lógica do processo de tomada de decisão (VERCELLIS, 2008).

3.1.2. O Processo de Tomada de Decisão

Segundo (HOLSAPPLE, 1996), o processo de tomada de decisão é fundamentalmente um conjunto de processos de reconhecimento para resolução de problemas focando o alcance de um objetivo por meio da produção de decisões.

A representação do processo de decisão proposta por (SIMON, 1960), composto por cinco fases: inteligência, projeto, escolha, implementação e controle, conforme ilustrado pela figura 3.3, continua mantido nos tempos atuais como a metodologia referencial.

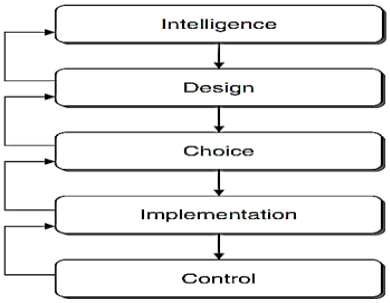


Figura 3.3 – Fases do processo de decisão (SIMON, 1960).

Na fase *Inteligência* a tarefa do tomador de decisão é a identificar, circunscrever e definir explicitamente o problema que emerge no sistema em estudo. A análise do contexto e todas as informações disponíveis podem permitir que os tomadores de decisão captem rapidamente os sinais e sintomas que apontam para uma ação corretiva para melhorar o desempenho do sistema.

A fase *Projeto* é destinada a resolver se o problema identificado deve ser desenvolvido e planejado. Neste nível, a experiência e criatividade dos tomadores de decisão desempenham um papel fundamental, como eles são convidados a conceber soluções viáveis que permitam alcançar a finalidade pretendida. Sempre que o número de ações disponíveis é pequeno, os tomadores de decisão podem fazer uma enumeração explícita das alternativas para identificar a melhor solução. Se, por outro lado, o número de alternativas é muito grande, ou mesmo ilimitado, a sua identificação ocorre de forma implícita, através de uma descrição de regras de ações factíveis de satisfazer.

A fase *Escolha* aplica os critérios de desempenho considerados significativos na avaliação das alternativas identificadas pela fase inicial. Modelos matemáticos e métodos analíticos correspondentes à solução, geralmente são aplicados e desempenham valioso papel preponderante nos resultados. Por exemplo, os modelos e métodos de otimização permitem que a melhor solução seja encontrada em situações complexas que envolvem infinitas

soluções viáveis. Por outro lado, as árvores de decisão podem ser usados para lidar om processos de decisão influenciados por eventos estocástivos.

A fase *Implementação* traduz em ações através de um plano de execução a melhor alternativa definida pela fase anterior, atribuindo responsabilidades e papéis todos os envolvidos no plano de ação.

Uma vez que a ação tenha sido implementada, a fase de *Controle* faz a conciliação entre as expectativas iniciais e o resultado do plano de ação analisando o grau de coincidência dos fatores. Em um *DSS* bem estruturado, o resultado do grau de coincidência obtido é reutilizado como o embasamento de experiência que subsidiará os novos processos de decisão.

Muito embora o modelo de processo de tomada de decisão proposto por (SIMON, 1960), ainda seja considerado como referência, existem outros modelos a serem considerados, conforme tabela 3.1.

Tabela 3.1 – Comparação dos modelos de processo de decisão (HOLSAPPLE, 1996)

Simon *	Boyd	Turban and Aronson	Mora et al.
Inteligência	**Observar**	**Inteligência**	**Inteligência**
	Observação de ciscunstâncias desdobradas. Obter informações externas.	Objetivos organizacionais. Coleta de dados. Identificação dos problemas, responsáveis, classificação e estado.	Detectar o problema. Obter dados e formular o problema.
Projeto	**Orientar**	**Projeto**	**Projeto**
	Perceber ameaças e oportunidades. Focar em direção particular.	Formular o modelo. Definir os critérios de escolha. Pesquisar alternativas. Predizer ou medir as saídas.	Classificar, construir e validar o modelo.
Escolha	**Decidir**	**Escolha**	**Escolha**
	Fazer a decisão.	Solucionar problemas. Análises minuciosas. Selecionar a melhor alternativa. Planejar a implementação	Avaliar. Analisar minuciosamente. Selecionar as alternativas.
Implementação	**Agir**	**Implementação**	**Implementação**
	Agir conforme a decisão.	Implementação da decisão.	Apresentar os resultados. Planejar as tarefas. Monitorar os resultados.
Controle			**Aprender**
			Analisar as saídas. Sintetizar o processo de tomada de decisões. Determinar o que deve ser mudado.

** Os processos de decisões definidos por Simon foram explicitados no item 3.1.2 desta dissertação.*

3.1.3. Tipos de decisões

A definição da taxonomia das decisões é muito importante para um projeto de DSS, pois a identificação de um processo de tomada de decisões com características similares permite o uso do mesmo conjunto de metodologias.

As decisões podem ser classificadas em duas principais dimensões, de acordo com sua *natureza* e *escopo*. Com base na dimensão natureza, as decisões podem ser classificadas como: estruturada, semi-estruturada e não estruturada e segundo a dimensão escopo, em: estratégica, tática ou operacional.

Segundo (TAYLOR, 2012), uma decisão estruturada é aquela cujo processo de tomada de decisões está claramente definido, ou seja, os fluxos de entrada, processamento e saída estão bem identificados podendo ser visualizados como algoritmos, por exemplo. Quando os processos não estão definidos são considerados *não-estruturados* e quando possuem apenas um dos processos não claramente definidos são considerados *semi-estruturados*.

A dimensão de escopo refere-se a abrangência de atuação que determinada decisão afeta. Quando sua atuação se abrange por toda a organização, denomina-se como *estratégica*, quando abrange apenas um ou alguns departamentos, denomina-se *tática* e quando abrange o melhoramento de processos de um departamento ou área são denominados *operacionais*. Todas essas dimensões podem ser inter-relacionadas conforme figura 3.4.

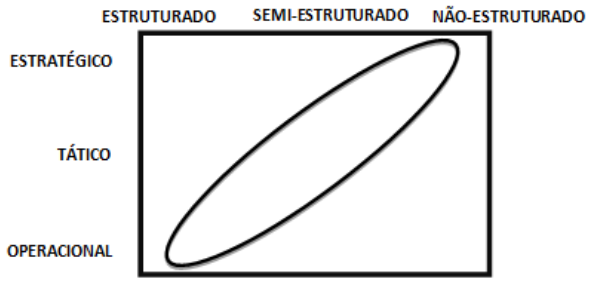


Figura 3.4 – Taxonomia das decisões (VERCELLIS, 2008).

3.1.4. Definição dos Sistemas de Suporte a Decisões

Desde a década de 1980, os *DSS* são definidos como um sistema ativo computacional que auxilia os tomadores de decisões combinando os dados e modelos para resolver problemas *não estruturados* ou *semi-estruturados*. Esta definição engloba três principais elementos de um *DSS* exibidos na figura 3.5: um banco de dados, um repositório de modelos matemáticos e um módulo para controlar o diálogo entre o sistema e os usuários.

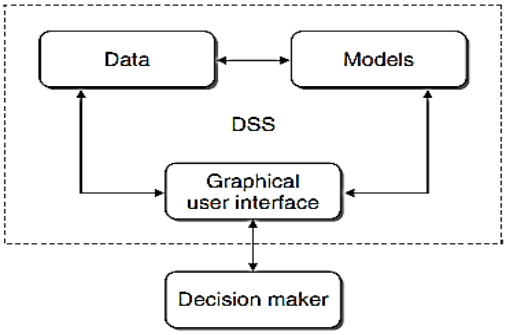


Figura 3.5 –Estrutura de um DSS (VERCELLIS, 2008).

A evolução nos DSS encontra-se focado em dois aspectos de áreas distintas. De um lado o processamento dos dados através das tecnologias da informação e do outro, as disciplinas que estudam e aplicam os modelos e métodos matemáticos.

Pela análise da estrutura estendida de um DSS, ilustrada na figura 3.6, verifica-se a identificação de novos componentes, tais como: Gerenciamento dos dados, Gerenciamento de modelos, Interações e Gerenciamento de conhecimento.

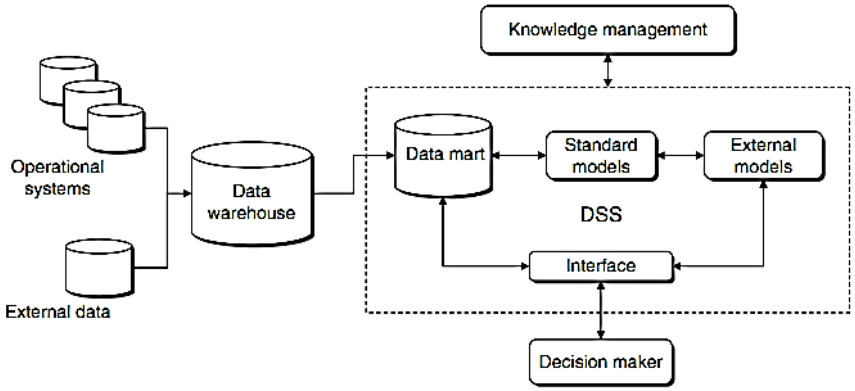


Figura 3.6 – Estrutura estendida de um DSS (VERCELLIS, 2008).

O componente Gerenciamento de Dados inclui o ciclo de vida das informações que subsidiarão todo o processo, desde o projeto do banco de dados, sua concepção em *Data marts* departamentais que comporão os *Data Warehouses*, até a conexão deste à arquitetura de BI para que se façam as análises no repositório de dados.

O componente de Gerenciamento de Modelos provê a coleção de modelos matemáticos derivados de operações de pesquisas, estatísticas e análises financeiras. Em certas aplicações podem-se fazer necessários módulos externos especializados para suprir e ampliar a capacidade do modelo de recursos padrão.

O componente de Interação é o que permite a utilização simplificada e intuitiva por parte dos usuários do *DSS*, que por meio de interface gráfica possam extrair as informações e conhecimentos gerados.

Finalmente o componente de Gerenciamento do Conhecimento, permite aos tomadores de decisão projetar as várias formas de conhecimento coletivo, normalmente não estruturado, para representar a cultura organizacional, podendo estar integrado ao sistema de gestão de conhecimento da empresa.

Segundo (SIMON, 1960) as principais vantagens derivadas da adoção de um DSS são: aumento no número de alternativas e opções consideradas; um aumento no número de decisões efetivas; uma maior compreensão dos domínios analisados e dos problemas investigados; a aplicação de cenários baseados em hipóteses e parâmetros nos modelos matemáticos de predição; aumento na capacidade de reagir a eventos e situações não previstas; melhoria nas comunicações e coordenação entre os indivíduos e os departamentos organizacionais; maior confiabilidade nos mecanismos de controle e nos processos de decisão.

3.2. BUSINESS INTELLIGENCE – BI

O termo *business intelligence* é conceituado por (GARTNET, 2009), como um termo guarda-chuva que abrange pessoas, processos e aplicações / ferramentas para organizar as informações, permitir o acesso a ela e analisá-lo para melhorar as decisões e gerenciar o desempenho. Ainda, segundo (WITHEE, 2010) o termo *business intelligence* pode ser qualificado como qualquer atividade, ferramenta ou processo usado para obter a melhor informação para apoiar o processo de tomada de decisões, definição esta que corrobora com a supracitada. A figura 3.7 apresenta uma visão geral dos componentes de um sistema analítico.

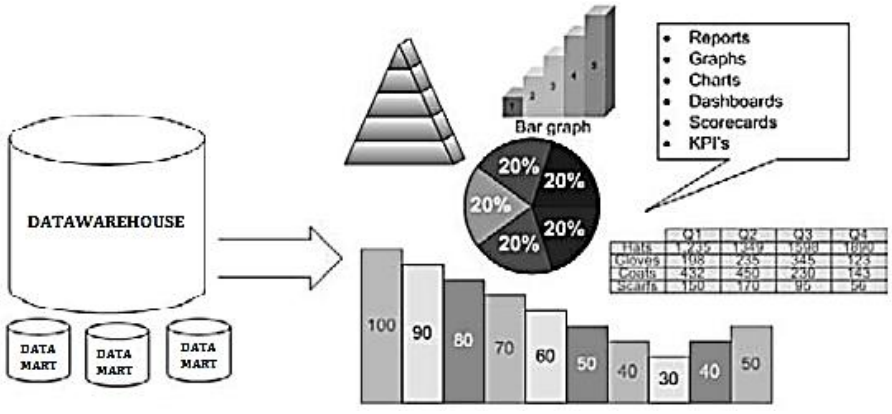


Figura 3.7 – Sistema analítico de BI (WITHEE, 2010).

As plataformas de BI permitem aos usuários criar aplicativos que ajudam as organizações a aprender e entender o seu negócio. As plataformas de BI são plataformas de software que oferecem certas capacidades de integração (infra-estrutura de BI, gerenciamento

de metadados, ferramentas de desenvolvimento, colaboração), a entrega de informações (relatórios, dashboards, consulta ad hoc, com base em pesquisa BI), e a análise (OLAP, visualização interativa, modelagem preditiva de dados e mineração, scorecards).

Segundo (MOSS & ATRE, 2003), dentre as diversas atividades que utilizam as facilidades das aplicações de suporte à decisão, estão: as Análises Multidimensionais de negócios; Mineração de dados; Previsão de dados e comportamentos (*Forecast*)*;* Monitoramento baseado em indicadores (*Balance Scorecard*); Sistemas diversos de consulta, relatórios e gráficos; Análises Geoespaciais e o Gerenciamento do conhecimento.

Destas diversas atividades, ainda cita os possíveis exemplos de banco de dados que suportam essas aplicações de BI, sendo: DataWarehouse empresarial; Data Marts (funcionais e departamentais); Exploração de datawarehouse (estatísticas); Bases de dados de mineração de dados; Web warehouses e Armazenamento de dados operacionais *ODS's*.

De acordo com a profundidade da análise e nível de complexidade, é possível agrupar as atividades de *business intelligence* em três categorias: Sistemas de consultas e relatórios; Sistemas para processamento analítico on-line (OLAP) e Sistemas de mineração de dados. Podem ainda ser adicionados a três novas categorias de aplicações, sendo: Aplicações de *dashBoard*; Aplicações de Alerta; e Portais.

Os sistemas de consultas e relatórios possuem a premissa de recuperação dos dados armazenados em *data warehouse* para apresentação em diversas mídias aos usuários que possuem a devida permissão, podendo ser por meio automatizado ou manual.

Os sistemas para processamento analítico on-line (OLAP) são atividades que analisam interativamente os dados das transações de negócios armazenadas nos modelos dimensionais dos *data warehouses* para auxiliar na tomada de decisões estratégicas e táticas.

Os sistemas de mineração de dados são compostos por processos que exploram os dados a fim de encontrar padrões e relacionamentos que descrevam os dados e possam predizer comportamentos até então desconhecidos ou predizer um comportamento futuro baseado em projeções.

As aplicações de *dashBoard* exibem os indicadores de desempenho de negócio, por meio de gráficos coloridos permitindo o agrupamento *roll-up* ou detalhamento *drill-down* das informações que compõem os indicadores pelos usuários autorizados.

As aplicações de Alerta, realizam a ação de notificação a usuários, baseado na ocorrência de determinados eventos ou condições. Os sistema de alerta baseados em *data warehouses* possuem sistema de notificação em nível de agregação enquanto que os sistemas de Alerta transacionais somente identificam atividades operacionais.

As aplicações de Portal possuem a função de *gateway* para o acesso e gerenciamento dos diversos aplicativos de BI, tendo como principal benefício a centralização da apresentação das soluções de BI em e como benefício secundário o gerenciamento centralizado da segurança.

Muito embora as atividades de BI não estejam limitadas aos dados armazenados em *data warehouses,* pois podem realizar consultar em ODS ou ERP ou outro sistema empresarial que efetue as análises necessárias para o fornecimento do desempenho dos negócios, esta dissertação pelo fato de usar o desenvolvimento baseado em *data warehouse* concentrará sua fundamentação dos tópicos relacionados a esta abordagem.

3.2.1. Data Warehouse

O crescente poder de processamento e a sofisticação das ferramentas e técnicas analíticas resultaram no desenvolvimento do que são conhecidos como *data warehouse,* oferecendo armazenamento, funcionalidade e responsividade às consultas além das capacidades oferecidas aos bancos de dados orientados a transação (ELMASRI, 2011).

Segundo (INMON, 2005), *data warehouse* é uma coleção de dados orientada a assunto, integrada, não volátil e variável no tempo para suporte às decisões de gerência, sendo ainda considerado, o núcleo da arquitetura do ambiente de BI.

A orientação a assunto se refere à modelagem dimensional que caracteriza a arquitetura que define a armazenagem dos dados e as suas relações de temporalidade dentro do *DW,* no sentido da supressão de dados com detalhes operacionais inertes de informações de suporte a decisão.

Para os propósitos de tomada de decisão, faz necessária a obtenção de todos os dados relevantes das várias aplicações. A origem dos dados reside em diferentes bancos de dados, arquivos ou segmentos de dados que por sua vez podem ser obtidos de diversos aplicativos (*sistemas transacionais*). Considerando que cada sistema pode ser construído de forma personalista, presume-se que pode haver inconsistências nos dados entre esses diversos aplicativos.

Manter o *DW* integrado significa remover todas as inconsistências das fontes externas antes da carga de dados, presumindo uma ação prévia de planejamento e análise dos dados externos no sentido de padroniza-los quanto: à convenção de nomes, codificação de caracteres, atributos e medidas dos dados.

A não volatilidade se refere exatamente à característica intrínseca do DW de manter-se como uma fotografia do estado dos dados no momento da sua captura. Portanto não deve sofrer alterações ou atualizações nos dados já armazenados. Portanto, diferentemente de uma base OLTP que possuem característica de se manter atualizado pelas diversas transações de modificações, uma base de DW deve ser acesso para obtenção dos dados dispostos apenas pela transação de leitura, conforme ilustra a figura 3.8.

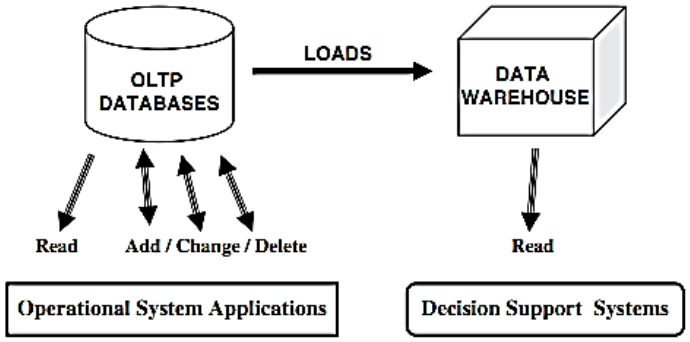


Figura 3.8 – O data warehouse é não-volátil (PONNIAH, 2010).

Os dados em um DW são utilizados essencialmente para análises e tomadas de decisões que são consubstanciados por dados históricos que permitem: a análise do passado, relacionar informações com o presente e permitir a análise de séries temporais para prever características futuras baseados nos dados atuais.

A característica de variabilidade no tempo é garantida pela representação dos dados sobre um horizonte de tempo distante ou pela utilização de um relacionamento de chaves que perfaz um relacionamento temporal dimensional com o DW (INMON, 2005).

3.2.2. Processamento Transacional e Processamento Analítico

O conceito armazenamento de DW para o suporte ao processamento analítico (OLAP) requer funções e desempenho que diferem dos sistemas de processamento de transações (OLTP). Enquanto que os sistemas OLTP baseiam-se no suporte da *execução* dos processos de negócios concentrando-se nos registros desses eventos, os sistemas OLAP suportam a *avaliação* destes processos (CHAUDHURI & DAYAL, 1997).

Um sistema transacional suporta diretamente a execução dos processos de negócios pela captura dos detalhes significativos de eventos ou transações e pela construção dos registros das atividades, frequentemente acaba se tornando parte deste processo.

Os sistemas transacionais têm seu projeto de banco de dados relacional, definidos no esquema organizacional pela terceira forma normal que permite suportar os processos de inserção, atualização e remoção de dados atômicos em alto desempenho de maneira consistente para manutenção dos dados sempre atualizados.

Considerando que os sistemas de processamento analítico são baseados em *DW* e têm o foco na avaliação dos processos, as interações nas bases de dados serão exclusivamente para fins de recuperação de dados por meio de consultas não ocorrendo interações de criação e atualização dos dados nesta base.

O projeto da base de um *DW* ocorre pelos princípios da modelagem dimensional que identifica os requisitos definidos por (INMON, 2005). O projeto do modelo dimensional em *Estrela* é otimizado para consultas de grande volume de dados e suporta a manutenção do histórico dos dados independentes das manipulações nas fontes externas.

Como pode ser visualizado na figura 3.9, o padrão de utilização dos hardwares é diferenciado nas duas soluções em virtude de suas características intrínsecas. Nos sistemas transacionais (*Operational*) a utilização é constante devido ao suporte dos processos de negócio por meio diversas transações (*inserção, atualização e remoção de dados*) enquanto que no sistema de processamento analítico (*Data Warehouse*) o uso ocorre somente no momento das consultas, configurando-se em picos de utilização altas e baixas.

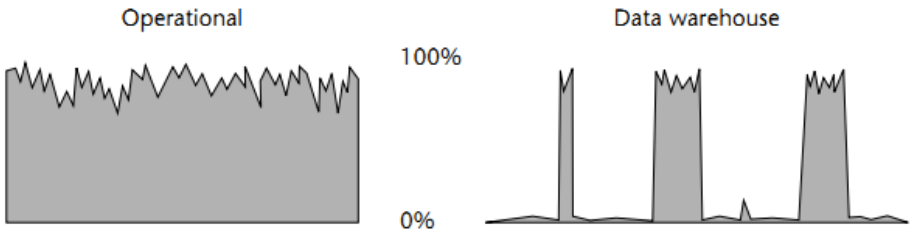


Figura 3.9 – Padrão da utilização do hardware em diferentes ambientes (INMON, 2005).

Essencialmente os sistemas transacionais e analíticos podem ser analisados e comparados, em função de suas propriedades conforme descrito na tabela 3.2.

Tabela 3.2 – Comparação entre os Sistemas Transacional e Analítico (ADAMSON, 2010).

Características	***Sistemas Transacionais***	***Sistemas Analíticos***
Propósito	*Execução de um processo de negócios*	*Medição de um processo de negócios.*
Estilo de interação primária	*Inserção, Atualização, Consulta e Remoção.*	*Consulta*
Escopo de interação	*Transação individuais.*	*Transações agregadas.*
Padrões de consulta	*Estável e previsível.*	*Mutável e imprevisível*
Foco temporal	*Presente.*	*Presente e passado.*
Otimização de projeto	*Atualização concorrente.*	*Alto desempenho em consultas.*
Princípio do projeto	*Entidades relacionamentos projetados na 3ª forma normal.*	*Modelagem dimensional (esquema estrela ou cubo)*
Denominações	*OLTP (On Line Transaction Processing System)* *Sistemas de Origem.*	*OLAP (On Line Analytical Process)* *Data Warehouse, Data Mart*

3.2.3. Abordagens Top-Down e Bottom-Up

Segundo (PONNIAH, 2010), antes de qualquer decisão sobre o desenvolvimento de um projeto de *data warehouse* em uma organização, devem ser questionados alguns pontos fundamentais que definem a abordagem pelo qual se dará o seu desenvolvimento. Dentre estes pontos fundamentais se encontram os seguintes questionamentos:

1. Qual será a abordagem adotada (top-down ou bottom-up)?
2. O *data warehouse* será empresarial ou departamental?
3. O que será desenvolvido primeiro (*data warehouse ou um data mart)*?
4. Será desenvolvido um piloto inicial ou o desenvolvimento será direto?
5. Os *data marts* serão independentes ou dependentes?

A identificação destes questionamentos permite o desenvolvimento do projeto baseado em abordagens com características e requisitos ligeiramente peculiares.

O pesquisador (INMON, 2005), proponente da abordagem top-down, define o *data warehouse* como um repositório centralizado para toda a organização. Nesta abordagem os dados no *data warehouse* são armazenados com baixo grau de granularidade baseado no modelo de dados normalizados.

Em sua visão, o *data warehouse* se constitui em um centro fornecedor de informações (*Corporate Information Factory - CIF*) que dispõem de uma arquitetura lógica para fornecimento dos serviços de BI. Neste caso, o *data warehouse* centralizado alimenta os *data*

marts dependentes que por sua vez são projetados baseados em um modelo de dados dimensional.

A abordagem *top-down* de Inmon possui as seguintes vantagens: visão corporativa dos dados; Arquitetura herdada em camadas, diferente de uma união de *data marts* diferentes; armazenamento centralizado dos dados; controle e regras centralizadas; e visualização rápida de resultados caso implementado com iterações.

No entanto, possui as seguintes desvantagens: seu desenvolvimento tende a ser mais demorado; alta exposição a risco de falhas; Necessita de profissionais com alto nível de habilidades multifuncionais; e alto investimento financeiro sem uma prova de conceito.

Alternativamente a abordagem *top-down* de (KIMBALL, 2002), propõe a abordagem *bottom-up* que se constitui de uma visão aonde o *data warehouse* é uma coleção de diversos *data marts* separados. Nesta abordagem, os *data marts* são desenvolvidos com base no modelo dimensional para o atendimento das demandas departamentais.

O armazenamento dentro de um *data mart* é feito no mais baixo grau de granularidade mas também pode apresentar os dados sumarizados. Estes *data marts* podem ser unidos ou combinados pelas suas dimensões e esta união perfaz a conceituação de *data warehouse* corporativa.

A abordagem *bottom-up* apresenta as seguintes vantagens: desenvolvimento e gerenciamento rápido e fácil; favorável ao retorno do investimento através de provas de conceitos; menos risco de falhas; crescimento incremental pela definição do desenvolvimento dos *data marts* prioritários; permite a evolução do conhecimento no desenvolvimento pela equipe do projeto.

No entanto apresenta as seguintes desvantagens: cada *data mart* possui sua própria visão estreita dos dados; permeia dados redundantes em cada *data mart*; perpetuam dados inconsistentes e irreconciliáveis; e prolifera interfaces incontroláveis.

Não obstante às características das duas abordagens da tabela 3.3, é citado por (PONNIAH, 2010) uma abordagem prática que se resume analisar minuciosamente as necessidades corporativas pelos aspectos no nível estratégico e posteriormente táticos, realizando a concepção do *data warehouse,* determinando o conteúdo de cada supermart que nada mais é que a combinação estratégica de vários *data marts*.

Tabela 3.3 – Comparação entre *Data Warehouse* e *Data Mart* (PONNIAH, 2010).

DATA WAREHOUSE	***DATA MART***
Visão empresarial corporativa.	*Visão departamental.*
União de todos os data marts.	*Um simples processo de negócio.*
Recebimento dos dados da área de preparação (Stagging área).	*União no esquema estrela pelas tabelas: fatos e dimensões.*
Consultas em recursos de apresentação.	*Tecnologia otimizada para acesso e análise*
Estrutura corporativa para visualizar os dados	*Estrututura departamental para visualização dos dados.*
Organizado no modelo de Entidades-Relacionamentos.	*Organizado no modelo dimensional (estrela ou flocos de neve) com estrutura desnormalizada.*

Nesta abordagem prática proposta são elencados quatro passos: 1. Planejar e definir os requisitos gerais da corporação; 2. Criar uma arquitetura que compreenda uma solução completa do *data warehouse*; 3. Padronizar e integrar o conteúdos de todos os dados; e 4. Implementar o *data warehouse* como uma série de *supermarts*.

3.2.4 Arquitetura dos Data Warehouse

Os tipos de arquitetura de um *data warehouse* permitem identificar como os dados são armazenados e as relações entre o *data warehouse* e os *data marts.* Os cinco tipos descritos estão ilustrados pela figura 3.11.

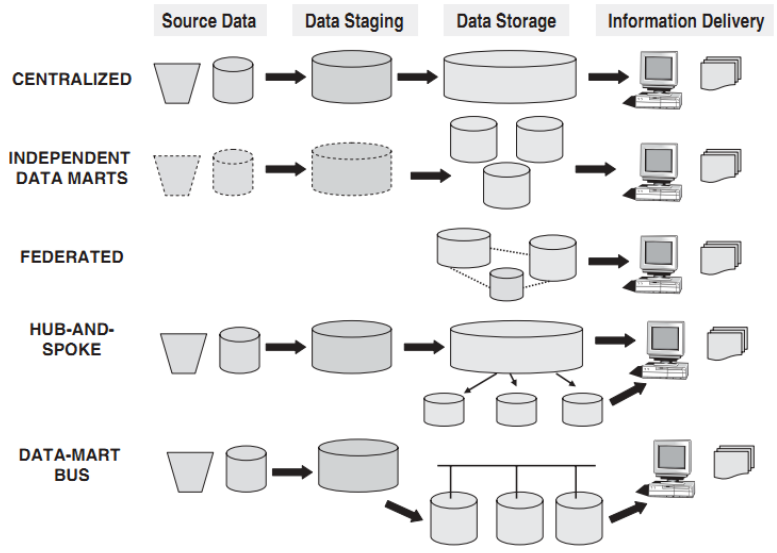


Figura 3.10 – Arquitetura de data warehouse (PONNIAH, 2010).

Na arquitetura centralizada os requisitos corporativos do nível estratégico devem estar bem definidos em sua concepção inicial. O nível atômico dos dados normalizados é possui baixo grau de granularidade e obedece a terceira forma normal. As consultas e aplicações

acessam os dados normalizados no *data warehouse*, que não é separada em *data marts* (WATSON & ARIYACHANDRA, 2005).

Na arquitetura de *Data Marts* independentes, cada unidade organizacional desenvolve seus próprios *data marts* conforme seus próprios propósitos. Podem apresentar inconsistência entre os dados de diferentes unidades no que concerne às definições de dados e padrões quando se necessita de uma integração dos dados do *data marts*.

A arquitetura federativa é um tipo onde os dados podem estar fisicamente ou logicamente integrados, para consultas distribuídas ou outros métodos. Nesta arquitetura não existe um *data warehouse* global e sim uma integração de soluções entre as corporações.

A arquitetura *Hub-and-Spoke* é a abordagem de (INMON, 2005) que idealiza o CIF (*Corporate Information Factory*). Os dados, modelados em sua terceira forma normal são depositados em um repositório central denominado *data warehouse* que são usados como insumos aos *data marts* que são uma representação dos dados em modelagem dimensional com a finalidade de fornecer as consultas como suporte à decisões. Esta arquitetura resulta da adoção da abordagem *top-down* no desenvolvimento do *data warehouse*.

Finalmente na arquitetura *Data-Mart Bus* (barramento de *data marts*) a modelagem dimensional dos *data marts* é orientado ao atendimento das necessidades setoriais e quando integrados em *supermarts* lógicos fornece a visão empresarial dos dados. Utiliza-se da abordagem bottom-up no desenvolvimento do *data warehouse* conforme definido por (KIMBALL, 2002).

3.2.5 Componentes de Data Warehouse

A arquitetura de um *data warehouse* define como o arranjo de seus componentes será estabelecido para o atendimento das necessidades de negócios. Logo a identificação de todos os componentes de *software* ou *hadware* que integram uma solução de *DW* deve ser pormenorizada para o melhor entendimento de suas funções. Com base na figura 3.12, pode ser identificado os seis componentes principais, sendo: *Source Data, Data Stagging, Data Storage, Metadata, Information Delivery, e Management & Control.*

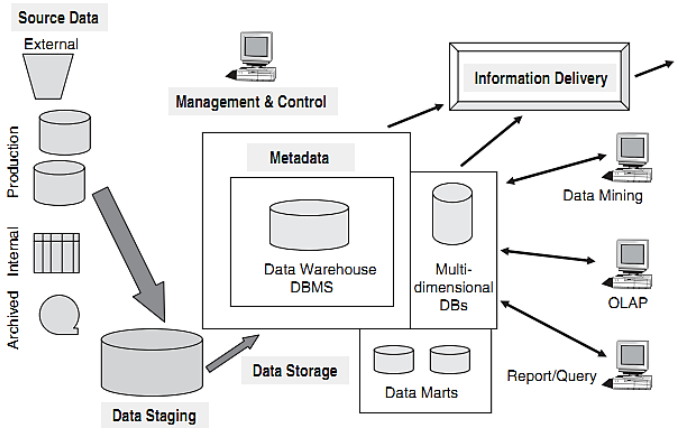


Figura 3.11 - Componentes de construção dos Data Warehouses (PONNIAH, 2010).

3.2.5.1 Componente Source Data

O componente *source data*, ou origem dos dados é o insumo de todo o processo pois é o repositório dos dados a serem coletados no início do projeto. Os dados com base em sua origem podem ser agrupados em 4 tipos, sendo: dados de produção, dados internos, dados externos e dados arquivados.

Os dados de produção são obtidos de sistemas transacionais das corporações. Os dados internos são os dados departamentais considerados privados que podem ser disponibilizados, tais como documentos e planilhas. Os dados externos são dados obtidos de outras fontes que não são da corporação, mas que são recuperados para composição do *data* warehouse. Os dados arquivados, são os dados oriundos de sistemas transacionais inoperantes ou de dados que estão armazenados para fins de histórico.

Esta etapa, ainda possui alguns requisitos a serem considerados quando de sua análise, tais como: a disponibilidade das fontes de dados, a análise das estruturas das fontes de dados; a localização das fontes de dados; os sistemas operacionais, redes, protocolos e arquiteturas clientes disponíveis; os procedimentos de extração e a disponibilidade dos registros de logs.

3.2.5.2 Componente Data Stagging

O componente *Data Stagging* tem a função de armazenar temporariamente os dados extraídos das fontes de dados antes destes serem exportados para persistência no *data warehouse*, pois constitui-se em um local mais apropriado para a aplicação das diversas operações de preparação dos dados, tais como: limpeza dos dados, troca, combinação, conversão, deduplicação e preparação da origem dos dados para serem armazenados e usados em um *data warehouse.*

Este componente pode ainda ser separado funcionalmente em três grupos: extração, transformação e carga, que o nomina comumente como o *processo de ETL*.

A função de extração se constitui pela capacidade de se conectar a diferentes fontes de dados de diversas origens em diversos formatos com a finalidade de se obter os dados e torna-los disponíveis aos processos do componente *Data Stagging.*

A função de transformação é o estágio onde se concentra a "inteligência" deste componente, pois todos os dados alimentados pela função de extração, possuem características próprias dos sistemas que os geraram que podem ser dispares dos requisitos da modelagem do *data warehouse*, cabendo a essa função equalizar e padronizar os dados em função dos requisitos de modelagem.

A função de carga refere-se ao processo de persistência dos dados, já extraídos e transformados pelos processos anteriores no banco de dados da *data warehouse.* Essa função ocorre em dois momentos cruciais, sendo: o momento da carga inicial dos dados e a carga agendada.

Na carga inicial é exigido um alto volume de tráfego de transferência a ser persistido no *data warehouse* e posteriormente, os processos de extração dos dados na bases de dados, os processos de transformação das revisões dos dados e a alimentação da base ocorrerá de forma incremental.

3.2.5.3 Componente Data Storage

O componente *Data Storage* para o *data warehouse* deve ser considerado um repositório separado do sistema transacional, projetado para a obtenção de grande volume de dados obtidas pelas consultas aos dados persistidos.

Este repositório deve ser projetado quanto às características de escalabilidade considerando os aspectos de performance das consultas e performance de entrega dos dados através do dimensionamento da banda de acesso. Pode ser estruturado no modelo ROLAP (relacional) ou MOLAP (multidimensional).

Independentemente da estrutura são dados que devem representar instantes de períodos específicos e se mantendo estável e sempre disponível somente para leitura. A estabilidade se refere ao fato de que essa base deve manter o volume de dados de históricos sempre preservados.

3.2.5.4 Componente Information Delivery

Para fornecer informações das mais diversas formas aos usuários dos sistemas de *data warehouse*, este componente inclui diferentes métodos e entrega das informações sendo os mais comuns: as consultas online e os relatórios.

As consultas por meio de relatórios predefinidos Ad hoc são consumidas principalmente pelos usuários com pouca experiência ou usuários casuais do sistema de *data warehouse*. O provisionamento de consultas complexas, análises multidimensinonais e análises estatísticas são disponibilizados aos analistas de negócios ou usuários com vasto conhecimento. As informações que alimentam os sistemas EIS (*executive information systems*) destinam-se aos executivos e gerentes de alto escalão. Finalmente, muitos *data warehouses* fornecem dados a aplicações de mineração de dados que pela aplicação de algoritmos auxiliam na descoberta de características e padrões de uso nos dados.

3.2.5.5 Componente Metadata

O Metadado em um *data* warehouse é similar a um dicionário de dados em um sistema de gerenciamento de banco de dados. Em um dicionário de dados, são mantidas informações sobre a estrutura lógica dos dados, informações sobre os arquivos e endereços, informações sobre os índices, ou seja, contém dados sobre a estrutura dos dados. O Metadado em um *data warehouse* pode ser categorizado em três grupos, sendo: metadados operacionais, metadados de extração e transformação, e metadados dos usuários finais.

Os metadados operacionais, se referem aos dados no *data warehouse* que são oriundos dos diversos sistemas transacionais corporativos, ou seja, contém todas as informações sobre as fontes de dados operacionais.

Os metadados de extração e transformação contém dados sobre a extração dos dados nos sistemas de origem, tradução de nomeações, frequências de extração, métodos de extração e regras de negócios utilizada no processo de extração. Possui também informações sobre todas as transformações realizadas pelo componente de *Data Stagging*.

Os metadados de usuários finais refere-se ao mapa de navegação do *data warehouse*, que permite aos usuários finais localizar as informações do *data warehouse* usando uma terminologia própria no modo de entender o negócio corporativo.

3.2.5.6 Componente Management & Control

Este componente se situa sobre todos os outros componentes, coordenando os serviços e atividades do *data warehouse.* A função de controle refere-se às transformações e transferência dos dados do *storage* do *data warehouse* moderando a entrega de informações aos usuários enquanto que a função de gerenciamento refere-se ao monitoramento dos fluxos de dados na área de *stagging* na área do *storage* do *data warehouse.*

Este componente interage com o componente de metadados para executar as funções de gerenciamento e controle, pelo fato do metadados possuir os dados sobre o *data warehouse* em si e portanto ser a fonte de informações para o módulo de gerenciamento.

3.2.6 Modelagem Multidimensional

A modelagem dimensional é uma técnica de projeto lógico que procura apresentar os dados em um modelo padrão intuitivos com alto desempenho de acesso. Seu projeto deve ter como premissa a otimização do desempenho das consultas com alto volume de dados, mantendo o suporte aos dados históricos para a manutenção dos instantâneos transacionais.

Um método para o projeto do modelo dimensional, foi proposto por (KIMBALL, 2002) que se consubstancia em quatro passos:

1. A escolha do processo de negócio que será coberto pelo modelo, ou seja, a identificação das métricas e medidas quantitativas para análise (fatos);
2. Definição do nível de detalhe (grão) da tabela fato;
3. Definição das dimensões;
4. Definição dos fatos.

A modelagem dimensional de um processo de negócio é realizada por dois componentes: medidas quantitativas, e seus contextos; também conhecidos como fatos e dimensões. Estes componentes são organizados em um projeto de banco de dados para facilitar a ampla variedade de consultas analíticas. Quando implementados em um banco de dados relacional, o modelo dimensional é denominado *esquema estrela ou esquema floco de neve*, porém quando implementado em um banco de dados dimensional é denominado *cubo*. As principais características dessas abordagens podem ser identificadas pela tabela 3.4.

Tabela 3.4 – Diferenças entre SGBDM e SGBDR (INMON, 2005).

Cubos – SGBDM	***Relacional – SGBDR***
Não suporta muitos dados. Possui restrição quanto ao número de dimensões.	Suporta um grande volume de dados.
Tecnologia começa a ser empregada.	Tecnologia comprovada.
Junção dinâmica questionável.	Apresenta junção dinâmica de dados.
Não suporta processamento de atualização de uso geral.	Apresenta bom processamento de atualização.
Desempenho otimizado para o processamento de apoio à decisão.	Desempenho não chega a ser excelente.
Estrutura de dados pode ser otimizada para um padrão de acesso conhecido.	Não pode ser otimizada exclusivamente para processamento de acesso.
Não apresenta estrutura flexível para acessar dados por caminhão não preparado.	Fácil acesso a dados.

Considerando que a implementação mais comum do modelo dimensional ocorre em uma base de dados relacional, a sua estrutura segue o padrão definido por (KIMBALL, 2002) que o define como *esquema estrela* ou um esquema estendido denominado *esquema Snowflake*. Este esquema é composto por uma tabela central denominada *tabela fato* que possui as medidas quantitativas dos processos de negócios que se liga às outras tabelas denominadas *tabelas dimensões* que representam os seus contextos de aplicação.

Cada tabela dimensão possui uma única chave primária, e o conjunto dessas chaves primárias forma a chave composta da tabela fato que permite identificar as ocorrências quantitativas com as devidas correlações destas com os diversos contextos (*dimensões*) analisados do modelo dimensional.

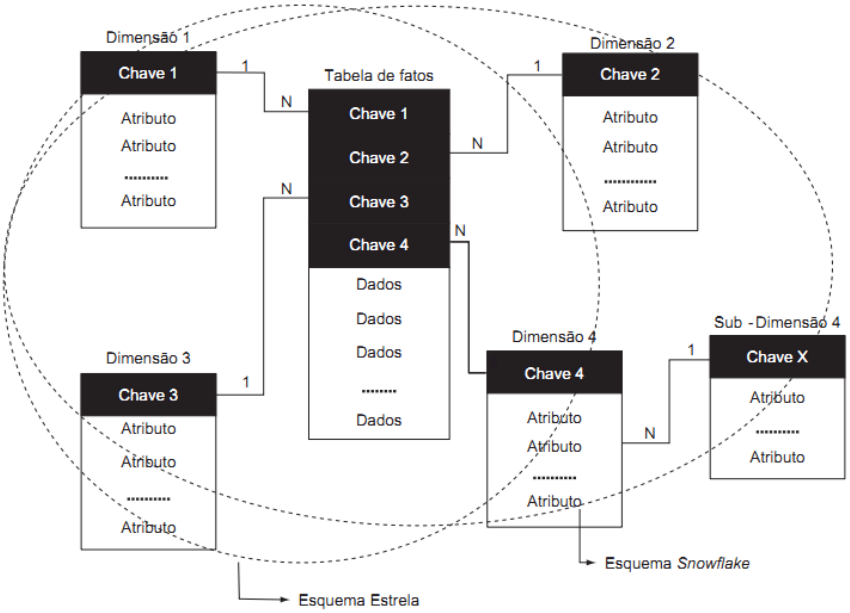


Figura 3.12 – Modelo Multidimensional - Estrela e Snowflake (FORTULAN, 2005)

Uma variação do esquema estrela é o S*nowflake*, que consiste, na realidade, de uma normalização do primeiro. No esquema *SnowFlake*, as tabelas *dimensão* são estruturadas de modo que atendam à terceira forma normal, mantendo as tabelas Fato em sua estrutura inicial.

Segundo (FORTULAN, 2005), o uso do esquema *Snowflake* traz como desvantagens o aumento da complexidade da estrutura de dados, dificultando a compreensão do modelo por parte de usuários que trabalham diretamente com a estrutura física das tabelas. No entanto, o uso do *Snowflake* pode ser indispensável em alguns casos em que, por exemplo, o modelo desnormalizado (*estrela*) requeira muito espaço em disco ou suas tabelas dimensionais sejam muito grandes, prejudicando o desempenho do sistema.

A modelagem dimensional permite a realização de operações analítica tais como: Drill-Down, RollUp, Slice-Dice, Pivot, além das operações como "ranking" (sort), seleções e definição de atributos computados, como média e soma (SOARES, 1998).

A figura 3.15 ilustra o cubo, ao centro, formado pelas dimensões: Tempo, Protocolo e Endereço_IP e as operações analíticas de *Slice-Dice* para análises específicas com ênfase em cada uma das dimensões.

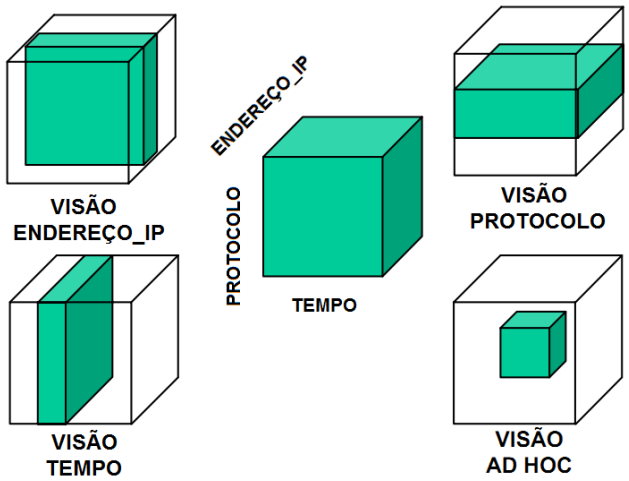


Figura 3.13 – Visões do Cubo Multidimensional.

3.3. DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS - KDD

Uma das definições mais populares para o termo KDD foi proposta por (FAYYAD, HAUSSLER, & STOLORZ, 1996) que afirma que: "KDD é um processo, de várias etapas, não trivial, interativo e iterativo, para identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de grandes conjuntos de dados".

Na definição formal de KDD, o termo interativo indica a necessidade de atuação do homem como responsável pelo controle do processo. O termo iterativo, sugere a possibilidade de repetições integrais ou parciais do processo de KDD, nas busca de resultados satisfatórios, por meio de refinamentos sucessivos. A expressão não trivial, alerta para a complexidade normalmente presente na execução de processos KDD (VIGLIONI, 2007).

Segundo (ELMASRI, 2011), as principais fases do processo de descoberta de conhecimento (KDD) ilustradas pela figura 3.16, pode ser definidas por:

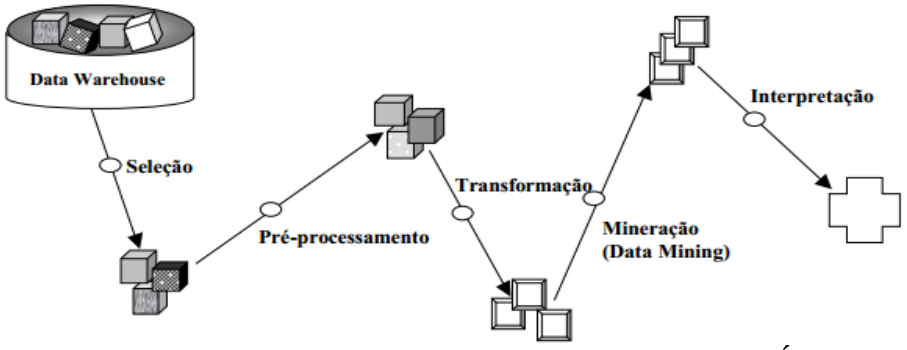


Figura 3.14 – Principais fases do processo de KDD (THOMÉ, 2007).

- **Seleção:** é a etapa que consiste na análise dos dados existentes e na seleção daqueles a serem utilizados na busca por padrões e na geração de conhecimento novo.
- **Pré-Processamento:** consiste no tratamento e na preparação dos dados para uso pelos algoritmos. Nesta etapa devemos identificar e retirar valores inválidos, inconsistentes ou redundantes;
- **Transformação:** consiste em aplicar, quando necessário, alguma transformação linear ou mesmo não linear nos dados, de forma a encontrar aqueles mais relevantes para o problema em estudo. Nesta etapa geralmente são aplicadas técnicas de redução de dimensionalidade e de projeção dos dados.
- **Mineração:** consiste na busca por padrões através da aplicação de algoritmos e técnicas computacionais específicas.
- **Interpretação:** consiste na análise dos resultados da mineração e na geração de conhecimento pela interpretação e utilização dos resultados em benefícios do negócio.

3.3.1 Mineração de Dados

A mineração de dados é um dos processos da *descoberta de conhecimento nos bancos de dados*, conhecida como KDD (*Knowledge Discovery in Databases*). Portanto, segundo (TAYLOR, 2012), o processo de mineração de dados auxilia na extração de novos padrões

significativos que podem não ser necessariamente encontrados apenas ao consultar ou processar dados ou metadados no *data warehouse.*

Segundo (THOMÉ, 2007), *Data Mining* é a concepção de modelos computacionais capazes de identificar e revelar padrões desconhecidos mas existentes entre dados pertencentes a uma ou mais bases de dados distintas – um *Data Warehouse*.

O termo *conhecimento*, é interpretado de forma livre como algo que envolve algum grau de inteligência. Normalmente é classificado como indutivo *versus* dedutivo.

O conhecimento dedutivo deduz novas informações com base na aplicação de regras lógicas *previamente especificadas* de dedução sobre o dado indicado. A mineração de dados enfoca o conhecimento indutivo, que descobre novas regras e padrões com base nos dados fornecidos.

As principais estratégias empregadas nesta tarefa incluem a classificação, a agregação, a associação, a regressão e a predição. Em todas as estratégias, o objetivo maior é o de poder generalizar o conhecimento adquirido para novas ocorrências do fenômeno ou para outros contextos ou situações parecidas com a utilizada na construção do modelo computacional.

A estratégia de *Classificação* consiste na busca por uma função que consiga mapear (classificar) uma determinada ocorrência em uma dentre um conjunto finito e pré-definido de classes. A construção do modelo segundo esta estratégia, pressupõe o conhecimento prévio das possíveis classes e a correta classificação dos exemplos usados na modelagem.

A estratégia de *Agregação* (ou *Clustering*) consiste na busca de similaridades entre os dados, tal que permita definir um conjunto finito de classes ou categorias que os contenha e os descreva. A principal diferença entre esta abordagem e a classificação é que em agregação não se tem conhecimento prévio sobre o número de classes possíveis nem a possível pertinência dos exemplos usados na modelagem.

A estratégia de *Associação* consiste em identificar fatos que possam ser direta ou indiretamente associados. Esta estratégia é geralmente usada em aplicações onde se busca identificar itens que possam ser colocados juntos em um mesmo pacote de negociação, sendo também utilizada para avaliar a existência de algum tipo de relação temporal entre os itens constantes de uma base de dados.

A estratégia de *Regressão* consiste na busca por uma função que represente, de forma aproximada, o comportamento apresentado pelo fenômeno em estudo. A forma mais

conhecida é a regressão linear, por exemplo, uma reta que minimiza o erro médio entre todos os valores considerados, mas também pode ser não linear.

A estratégia de *Predição* envolve uma componente temporal, isto é, aquela classe de problemas nos quais estamos interessados em prever o comportamento ou valor futuro de uma determinada variável com base em valores anteriores desta mesma variável (monovariável) ou em valores anteriores da variável de interesse de outras variáveis (multivariável).

Em cada uma destas estratégias diferentes técnicas e algoritmos podem ser aplicados, conforme descrito na tabela 3.5.

Tabela 3.5 – Estratégias de Mineração de Dados (THOMÉ, 2007).

Estratégias	**Algoritmos**
Classificação	Árvores de decisão e redes neurais.
Agregação	Métodos estatísticos e redes neurais.
Associação	Métodos estatísticos e teoria dos conjuntos.
Regressão	Métodos de regressão e redes neurais.
Predição	Métodos estatísticos e redes neurais.

Considerando que o conhecimento indutivo não é gerado por técnicas triviais, a aplicabilidade das técnicas de mineração de dados, possibilita a utilização de novas estratégias de negócio que por se diferenciarem das demais estratégias permite um fator inovador para os negócios. Portanto, conforme descrito em (THOMÉ, 2007), a aplicabilidade destas técnicas podem se dar em diversos contextos, tais como: *marketing, vendas, finanças, saúde, energia,* dentre outros, que necessitem usar os dados e os conhecimentos gerados em *commodities*.

3.4. FERRAMENTAS DE BI OPEN-SOURCE

A atual oferta de plataformas de BI está efetivamente segmentada em duas abordagens de modelos de negócios: na primeira abordagem a plataforma de BI é suportada por empresas como produto comercial e sua comercialização se baseia no licenciamento de uso e fornecimento de serviços profissionais, enquanto que na segunda abordagem a plataforma é suportada por poucos indivíduos que mantém uma pequena comunidade que usa e distribui livremente a plataforma, sem intenções iniciais financeiras (BITTERER, 2008).

As plataformas de BI, indistintamente da abordagem de desenvolvimento, devem permitir aos usuários construir aplicações que auxiliam as corporações a aprender e entender melhor os seus negócios.

Segundo o (SALLAM, RICHARDSON, HAGERLY, & HOATMANN, 2011), uma plataforma de BI é definida como uma plataforma de software que possui treze funcionalidades dispostas em três categorias: integração, entrega de informações e análise.

Na categoria *Integração* estão contidas as funcionalidades: Infraestrutura de BI, Gerenciamento de Metadados, Ferramentas de desenvolvimento, Colaboração.

A *Infraestrutura de BI* refere-se ao uso compartilhado das ferramentas para o fornecimento da mesma sensação de uniformidade quanto à segurança, uso dos metadados, administração, integração no portal, modelo de objetos e motor de pesquisa.

O *Gerenciamento do Metadados*, deve fornecer um meio robusto para pesquisas, captura, armazenamento, reuso e publicação dos objetos de metadados como dimensões, hierarquias, medidas, métricas de desempenho, dentre outros.

A funcionalidade referente às *Ferramentas de Desenvolvimento* é composta pelos conjuntos de softwares que fornecem um ambiente de desenvolvimento visual acompanhado com um kit de desenvolvimento para criação de aplicações de BI.

A *Colaboração* refere-se a capacidade da plataforma de BI de permitir que os usuários compartilhem informações e gerenciem as hierarquias e métricas via fórum de discussão, bate papo ou anotações que podem estar embutidas na própria plataforma de BI ou integrada com outros softwares de colaboração ou redes sociais.

Na categoria de *Entrega de Informações* estão elencadas as seguintes funcionalidades: Relatórios, Dashboards, Consultas sob demanda e integração com Microsoft Office.

Os *Relatórios* referem-se a capacidade de criar relatórios formatados ou interativos (parametrizados) com possibilidade de distribuição e agendamento em alta escala permitindo ainda o acesso totalmente interativo ao seu conteúdo pelos diversos dispositivos móveis.

Os *Dashboards* são um subconjunto de relatórios que incluem a habilidade formal de publicar os dados em relatórios baseados na *Web* por meio de interfaces interativas que exibem as informações por meio de um painel de instrumentos todas as informações em formato de gráficos.

As *Consultas Sob Demanda* permitem que os próprios usuários manipulem as informações com o intuito de gerar seus relatórios por conta própria. As ferramentas que implementam essa funcionalidade, devem possuir uma robusta camada de semântica que permite a navegação nas fontes de dados disponíveis.

Em muitos casos a plataforma de BI atua em uma camada intermediária para gerenciar e garantir a segurança das tarefas de BI, delegando a um cliente de BI (especificamente o excel) a tarefa de auxiliar na interpretação dos dados.

Por fim, na categoria de *Análise* estão as funcionalidades: OLAP, Visualização interativa, Modelagem preditiva e Mineração de dados, e Scorecards.

A função OLAP (*Processamento Analítico on-line*) permite que os usuários realizem consultas e obtenha métricas calculadas com base nos dados em alta performance. Habilita uma análise baseada no fatiamento de cubos (*slice and dicing*) pela navegação multidimensional.

A *Visualização Interativa* permite a visualização de vários aspectos dos dados de forma mais eficiente pelo uso de imagens e gráficos interativos que representam os dados em linhas e colunas.

A *Modelagem Preditiva e Mineração de Dados* são realizados pela integração de modelos matemáticos que permitem a categorização e associação de variáveis para estimar uma previsão futura baseado nos dados do passado. Ainda, devem permitir as diversas atividades de descoberta de conhecimento sob o silo de dados.

As técnicas de *Scorecard* implicam no uso da metodologia de gerenciamento de desempenho pela exibição das métricas em um *dashboard* como suporte na integração dos mapas de indicadores chaves de negócios com os objetivos estratégicos.

Com o maior interesse e visibilidade nos softwares livres, muitas organizações decididas a diminuir os custos das implementações de soluções de BI, procuram inserir e embutir as funcionalidades de BI como parte dos seus processos de negócios.

Segundo (BITTERER, 2008), as plataformas de BI open source que atendem às necessidades corporativas tendo em vista que puderam ser avaliados às plataformas comerciais com base nos mesmos critérios de avaliação, são: BIRT, SpagoBI, Jasper e Pentaho.

Atualmente as plataformas de BI open source compartilham vários softwares para desenvolverem suas funcionalidades, conforme pode ser identificado na tabela 3.6.

Tabela 3.6 – Componentes das soluções livres de BI (PAOLANTONIO, 2010).

Capacidade	**Spago BI**	**JasperSoft**	**Pentaho**
DBMS incluído			HSQLDB / MySQL
ETL	Talend	Talend / JasperETL	KETTLE / PDI
Relatórios	BIRT JasperReports	JasperReports iReports	jFreeReports
Analisador	JPivot PaloPivot	JasperServer JasperAnalysis	jPivot PAT
OLAP	Mondrian	Mondrian	Mondrian
Mineração de Dados	Weka		Weka

Segundo (TERESO & BERNARDINO, 2011), com base em sua pesquisa sobre o estado da arte das plataformas livres de BI, ficaram elencadas quatro plataformas de BI consideradas robustas, conforme análise comparativa de suas funcionalidades dispostas na tabela 3.7.

Tabela 3.7 – Plataformas livres de BI. (TERESO & BERNARDINO, 2011)

Capacidade	**Spago BI**	**JasperSoft**	**Pentaho**	**Vanilla**
Relatórios	✓	✓	✓	✓
Gráficos	✓	✓	✓	✓
Dashboards	✓	✓	✓	✓
OLAP	✓	✓	✓	✓
ETL	✓	✓	✓	✓
Data Mining	✓	✘	✓	✓
KPIs	✓	✘	✓	✓
Exportação de dados	✓	✓	✓	✓
GEO/GIS	✓	✓	✘	✘
Consultas ad-hoc	✓	✓	✓	✓
Linguagem Java	✓	✓	✓	✘
Linguagem Perl	✘	✓	✘	✘
Linguagem PHP	✘	✓	✘	✘
Licença GNU GPL	✓	✓	✓	✓
Linux	✓	✓	✓	✓
Windows	✓	✓	✓	✓
Unix	✓	✘	✓	✓
Versão Community	✓	✓	✓	✓
Versão Enterprise	✘	✓	✓	✘

Das plataformas supracitadas, a que possui a comunidade mais ativa e maior atuação no mercado brasileiro é a plataforma Pentaho. Muito embora esta ferramenta não dê suporte

aos dados georeferenciados como a *SpagoBI* oferece, esta funcionalidade não é primordial ao objeto desta dissertação que é o monitoramento dos fluxos. Com base nessas considerações, a Pentaho BI Suite é a plataforma livre de BI adotada nesta dissertação.

3.4.1 Pentaho Open BI Suite

Segundo (Pentaho CE, 2004), esta plataforma é a ferramenta livre de BI mais popular, sendo disponibilizada pela licença de uso comercial em sua versão *Enterprise* e pela licença GNU GPL em sua versão *Community*. O pacote *Pentaho Open BI Suite* é diferente dos fornecedores tradicionais de plataformas de BI, pois sua plataforma é baseada em processos e orientada a soluções por componentes integrados.

O pacote Pentaho Open BI, consiste dos seguintes componentes macros: plataforma de BI; Funcionalidades de BI para usuários finais; e Pentaho Design Studio.

A plataforma de BI da Pentaho é o núcleo de sua arquitetura, sendo baseada em processos, pois se consubstancia no controlador central que utiliza das definições dos processos como insumo para a definição dos processos de BI na plataforma. Ainda, é considerada orientada a solução devido ao fato de que as operações na plataforma são especificadas pelos processos e operações de forma coletiva para a definição das soluções de um problema de BI.

A plataforma de BI fornece uma estrutura de execução e serviços que incluem: registros (*logs*), auditoria, segurança agendamento, processos de ETL, *web services*, auditoria, repositório de atributos e motor de regras.

As funcionalidades de BI para usuário finais incluem as capacidades de: relatórios, análises, *workflow*, painel de instrumentos e mineração de dados.

O Pentaho Design Studio é um conjunto de ferramentas de projeto e administração que integradas, permitem aos desenvolvedores a criação de relatórios, painel de instrumentos, modelos de análise, regras de negócios e processos de BI.

A plataforma de BI e as funcionalidades de BI para usuário final compõem no que se denomina Pentaho Server, que tem sua atuação como um coordenador das comunicações entre todos os componentes do BI.

A plataforma de BI do Pentaho incorpora diversos módulos, que executados como parte de um processo habilitam à solução, procedimentos de BI. Os principais módulos são:

- Pentaho BI Platform and Server (*Pentaho User Console e Pentaho Administrator Console*);
- Pentaho Report (*Pentaho Reporter Designer*);
- Pentaho Data Integration Community Edition – PDI CE (*Kettle*);
- Pentaho Analysis Services Community Edition – PAS CE (*Mondrian*);
- Pentaho Data Mining (*Weka*).

No Pentaho BI Server, o PAC (*Pentaho Administrator Console*) fornece uma console central para simplificar a administração de tarefas administrativas, tais como: gerenciamento de usuários e grupos, agendamento de tarefas, e gerenciamento de serviços. Enquanto isso o PUC (*Pentaho User Console*), é a interface que facilita o gerenciamento de relatórios e visões de análises.

O *Pentaho Report* é o conjunto de ferramentas livres que permitem a geração de relatórios relacionais ou analíticos obtidos a partir de uma extensa lista de origem de dados. Estes relatórios podem ter diversos formatos de saída, tais como: pdf, Excel, HTML, texto puro, XML, dentre outros, conforme tabela 3.15.

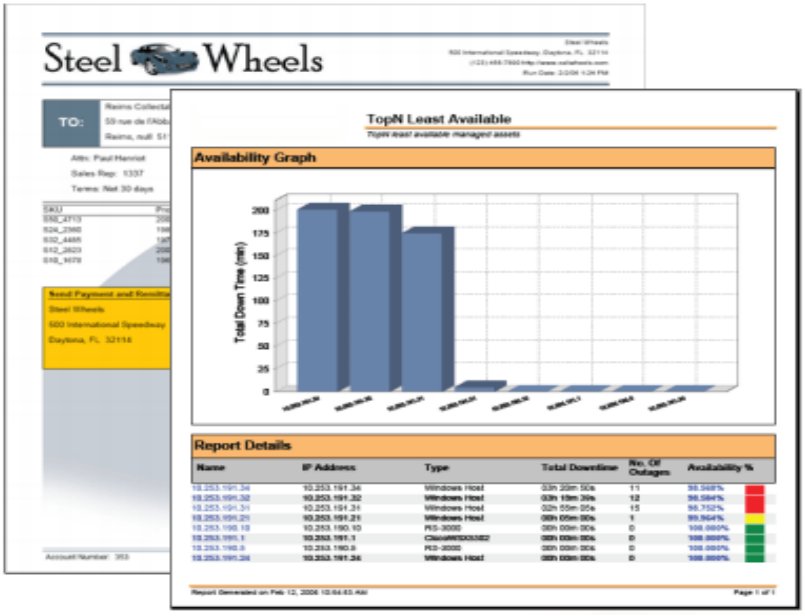
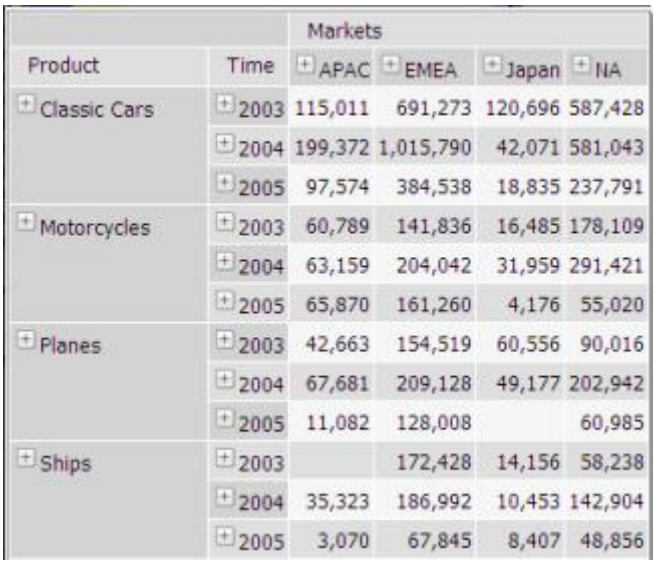


Figura 3.15 – Pentaho Report Designer (CASTERS, 2010).

O *Pentaho Data Integration*, executa os processos de extração, transformação e carga, por meio de uma interface gráfica intuitiva, em um ambiente escalável baseado em uma arquitetura baseada em padrões.

O *Pentaho Analysis Services*, também conhecido como Mondrian, é um servidor OLAP que habilita a análise de grandes quantidades de dados em tempo real. Os dados dos usuários podem ser analisados em uma tabela que permite aumentar ou diminuir as granulações das dimensões exibindo as informações com tempos de respostas rápidas às consultas analíticas complexas, conforme ilustrado pela figura 3.16.



		Markets			
Product	Time	APAC	EMEA	Japan	NA
Classic Cars	2003	115,011	691,273	120,696	587,428
	2004	199,372	1,015,790	42,071	581,043
	2005	97,574	384,538	18,835	237,791
Motorcycles	2003	60,789	141,836	16,485	178,109
	2004	63,159	204,042	31,959	291,421
	2005	65,870	161,260	4,176	55,020
Planes	2003	42,663	154,519	60,556	90,016
	2004	67,681	209,128	49,177	202,942
	2005	11,082	128,008		60,985
Ships	2003		172,428	14,156	58,238
	2004	35,323	186,992	10,453	142,904
	2005	3,070	67,845	8,407	48,856

Figura 3.16 – Pentaho Analysis Services - *Mondrian* (CASTERS, 2010).

O *Pentaho Data Mining (Weka)* permite a aplicação das técnicas de mineração de dados por meio da implementação dos seus algoritmos conforme descrito na tabela 16, em uma interface amigável. Em sendo uma solução de mineração de dados habilita a exploração dos dados a fim de se obter um melhor entendimento dos negócios por meio da geração de conhecimentos.

3.5 Síntese do Capítulo 3

Este capítulo aborda de um modo geral a fundamentação teórica acerca do domínio de estudo relacionado ao processo de tomada de decisões e os sistemas que auxiliam este processo. Inicialmente fundamenta a conceituação do processo de tomada de decisões e a definição dos tipos de decisões para rematar uma esquematização que caracteriza os sistemas de suporte a decisões (DSS) e o termo *Business Intelligence* (BI).

Em sequência, aborda-se a caracterização do termo *Data Warehouse* pela fundamentação de seus processos, tais como: entendimento das características do processamento analítico e sua consequente modelagem dimensional, suas abordagens de implementação (*top-down e botton-up*), seus componentes e arquiteturas de implementação passando pelo processo de descoberta de conhecimentos em banco de dados (KDD).

Finalizando pela análise comparativa das características das ferramentas livres de BI, com aprofundamento na plataforma *Pentaho Open BI Suite Community Edition*, que é a plataforma utilizada para o desenvolvimento do sistema de suporte a decisões desta dissertação.

4. MONITORAMENTO DE FLUXOS SUPORTADO POR BI

O monitoramento baseado em fluxos, conforme fundamentado no capítulo 02, pode ser realizado por diversos protocolos em diversos arranjos de seus componentes.

Originalmente foi proposto para ser implementado como uma funcionalidade adicional de gerenciamento nos equipamentos que já são utilizados para a composição da infraestrutura ds redes de computadores, tais como, chaveadores (*switch's*) e roteadores, mas que também podem ter sua implantação baseada em uma solução *stand*-alone pelo emprego de softwares que efetuam essa função, conforme listados nas tabelas 2.10 e 2.11.

Portanto, a funcionalidade de gerenciamento baseado no monitoramento de fluxos depende da disponibilização deste recurso pelos equipamentos disponíveis no ambiente, ou pela implementação via *software* dessas funcionalidades em um equipamento dedicado a este fim.

Considerando este aspecto, o protocolo sFlow é a implementação que dispõe de maior quantidade de fornecedores comparado ao protocolo NetFlow que é exclusivo da Cisco e ao protocolo IPFIX que é ainda uma especificação relativamente recente.

Entretanto a fatia de mercado não determina a relevância do protocolo no processo de definição da implementação de uma solução e sim as características e requisitos de contabilização das informações de fluxo. Conforme descrito no item 2.4.3 desta dissertação, o protocolo sFlow faz uma contabilização amostral do fluxo, enquanto que o protocolo NetFlow e o IPFIX realizam a contabilização dos fluxos de comunicações unilaterais em formato transacional.

Isso denota que, o uso do protocolo sFlow está atrelado ao consentimento do monitoramento do fluxo por meio amostral com relativa taxa de erro, que se situa entre 10 a 25 por cento em relação ao que efetivamente trafegou conforme relatado pelos protocolos de registro transacionais de fluxo NetFlow e IPFIX, em (REESE, 2010).

Considerando ainda que esse tipo de solução de monitoramento é suportada por dispositivos de capacidade mais elevada, pois exige recursos que não estão disponíveis em equipamentos de baixo custo, grande parte das redes de computadores de pequeno e médio porte não possuem dispositivos habilitados a esta funcionalidade de forma nativa de modo

que a utilização de uma solução baseada em *host* é uma estratégia interessante a ser considerada.

Não obstante, as soluções baseadas em *software livre* disponíveis, permitem a implantação flexível e escalável de uma solução de monitoramento baseado em fluxos, porém com capacidade relativamente limitada para gerar conhecimentos, pois a análise está abalizada em relatórios pré-concebidos ou à combinação de centenas de parâmetros para a confecção de relatórios mais específicos.

Neste ponto, as técnicas de *business intelligence*, peculiarmente o processo de análise analítica por meio de um modelo dimensional construído em um *data warehouse* adere facilmente ao propósito de disponibilizar o conhecimento por meio da correlação dos dados e análise das informações nesta dissertação.

4.1. MODELO PROPOSTO

Esta pesquisa de dissertação se propõe a desenvolver uma solução de monitoramento dos fluxos do tráfego de um backbone de uma rede local para o gerenciamento da contabilização dos fluxos por meio da análise analítica das informações geradas pelo acoplamento de um sistema de BI.

Esta solução é aplicada em um caso real em um órgão do governo do Estado do Mato Grosso, conforme figura 4.1, que possui a necessidade de gerenciamento da contabilização dos tráfegos de sua rede local, em relação à rede metropolitana (infovia) e à internet.

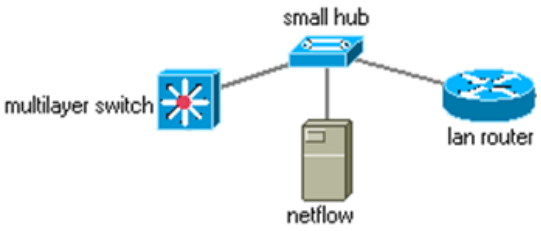


Figura 4.1 – Estrutura de contabilização netflow na rede lan.

Devido ao fato de que esta solução não poder alterar as configurações dos ativos atualmente definidos e para garantir que a estrutura funcional da rede não fosse afetada, foi adicionado na saída da rede local um *HUB* para realizar o espelhamento (*técnica de Hubbing-Out*) de todo o tráfego que entra e saí para análise do sensor Netflow.

Para a geração e coleta de forma persistente dos fluxos netflow v5 foram implantados softwares para essa finalidade que rastreará todo o tráfego de comunicação no *backbone* da rede local do órgão.

Associado ao processo transacional de captura dos registros dos fluxos pela solução NetFlow, tem-se o uso das técnicas de BI, especificamente da plataforma livre *Pentaho CE*.

A solução de BI vem acoplar suas características de análise aos dados coletados pelo computador *NetFlow* através de seus processo de extração, transformação e carga, bem como, nas funcionalidades disponibilização de dados por meio de análises analíticas.

Esta solução será capaz de apresentar os dados da mesma maneira que outras diversas ferramentas de análise de fluxos por meio de relatórios pré-concebidos ou relatórios ad-hoc, porém com o diferencial de habilitar as análises analíticas por meio das comparações analíticas do modelo multidimensional baseado nas métricas do protocolo.

4.2. DESENVOLVIMENTO DO MODELO PROPOSTO

A solução de gerenciamento baseado no monitoramento dos fluxos Netflow versão 5, se consubstancia essencialmente em duas partes distintas. A primeira trata da estrutura necessária para a implantação dos componentes NetFlow v5 dentro de uma infra-estrutura de tecnologia da informação já constituída sem interferir nos processos vigentes. A segunda trata da estrutura de BI necessária para disponibilizar informações úteis que habilitam o pleno gerenciamento do ambiente de TI.

4.2.1. Ambiente de Implantação do monitoramento NetFlow v5

O primeiro passo no processo de instalação da plataforma é a adequação do dispositivo que irá implementar os softwares sensores e coletores dos fluxos, para tal foi disponibilizado o computador, AMD Phenom II X2 com 2Gb de memória RAM com 80 Gb de disco e uma interface de rede gigabit ethernet, conforme apêndice A.

O sistema operacional Linux, foi essencial neste processo inicial pois todos os softwares livre que implementam o protocolo *NetFlow* possuem suporte apenas a esta plataforma operacional, que depois de devidamente configurada e atualizada permitiu a instalação dos aplicativos *fprobe* (FPROBE, 2012) e *flow-tools* (FLOW-TOOLS, 2012).

O aplicativo *fprobe* implementa o sensor Netflow v5 que gera os fluxos do tráfego que percebe pela interface de rede e os encaminha ao coletor pela porta 5678. O aplicativo *flow-tools* por meio da ferramenta *flow-capture* recebe os fluxos e armazena em arquivo no em disco com periodicidade de 10 minutos.

Com os aplicativos configurados e instanciados, basta iniciar o coletor (*flow-tools*) e posteriormente instanciar o sensor (*fprobe*), pois como toda aplicação cliente servidor, o servidor deve estar instanciado antes da chamada do cliente, conforme figura 4.2.

```
root@NetFlow:/etc/default# ps -ax | grep -E '(fprobe|flow-capture)'
 1046 ?        Ssl  789:52 /usr/sbin/fprobe -ieth0 -fip localhost:5678
 1767 ?        Ss    16:49 flow-capture -n 143 -w /home/andre/flows -S 5 0/0/5678
```

Figura 4.2 – Confirmação da execução do fprobe e flow-tools.

O dispositivo NetFlow v5, está pronto para gerar as transações de fluxos na função de sensor e habilitado a receber estes fluxos para o devido armazenamento em disco. Porém o fluxo deve ser encaminhado para a interface do dispositivo a fim de que ele possa realizar o seu trabalho.

A técnica preferida para o espelhamento do tráfego é: o *port mirroring* ou o *port spanning*, mas para a aplicação desta técnica, o switch deve ter suas configurações alteradas para habilitar essa funcionalidade. Considerando que a técnica de *Hubbing Out*, é a mais generalista e simples possível, não impondo nenhuma alteração de configuração dos ativos da infra-estrutura, esta foi escolhida para a implementação do redirecionamento do tráfego para o dispositivo *NetFlow*.

A ilustração 4.3, exibe o dispositivo *NetFlow* conectado diretamente a um *hub* que está situado entre o tráfego da rede local e as redes remotas e entre a rede local e a rede de servidores, permitindo o monitoramento dos fluxos entre as redes lan (*rede local e rede de servidores*), man (*rede corporativa*) e wan (*rede externa – internet*).

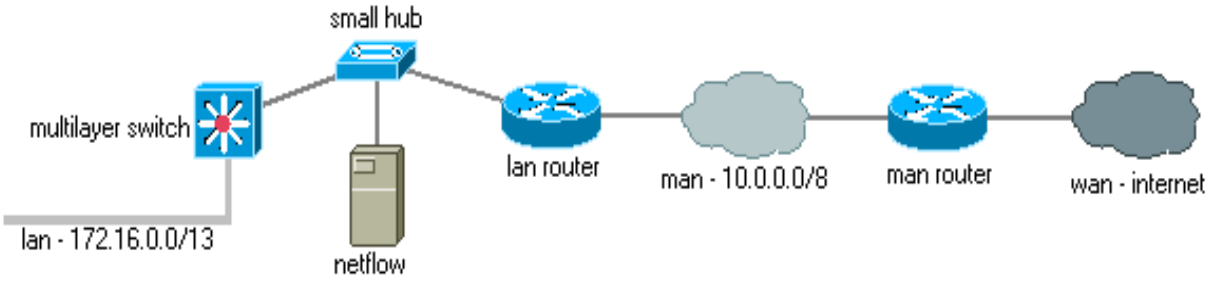


Figura 4.3 – Diagrama da infraestrutura de implantação.

A verificação e validação do funcionamento da plataforma de monitoramento NetFlow v5, por meio dos softwares fprobe e flow-tools, ocorre pela verificação da existência dos arquivos persistidos em uma periodicidade de 10 minutos conforme, figura 4.4.

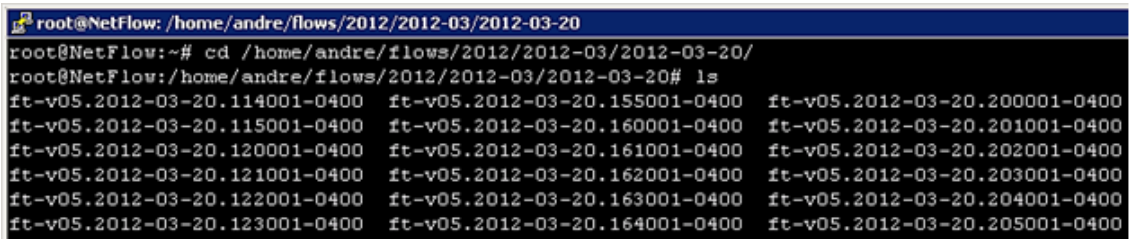


Figura 4.4 – Arquivos de fluxos NetFlow v5 do dia 20.03.2012.

4.2.2. Modelo Dimensional NetFlow v5

A modelagem dimensional objetiva o projeto de uma base de dados orientada a consultas que seja constituída por medidas quantitativas relacionadas aos seus contextos que por sua vez possuem a característica de serem inter-relacionados.

Na proposta do modelo dimensional do protocolo *NetFlow v5*, três são os campos da tabela fato (*tbl_ft_netflow*), considerados como medidas quantitativas: *fluxos*, *pacotes e octetos*, pois representam as medidas mensuráveis para a quantidade de fluxos trafegado, a quantidade de pacotes encaminhados e o volume de bytes trocados, respectivamente.

Todos os contextos, ou também denominadas dimensões, são interligadas à tabela das medidas quantitativas, denominada tabela fato, pelos seus campos chave (*PK*) às respectivas chaves estrangeiras (*PFK*), dessa forma efetivam a relação direta entre a dimensão e os fatos e a relação indireta entre as dimensões através da correlação das chaves estrangeiras na tabela fato.

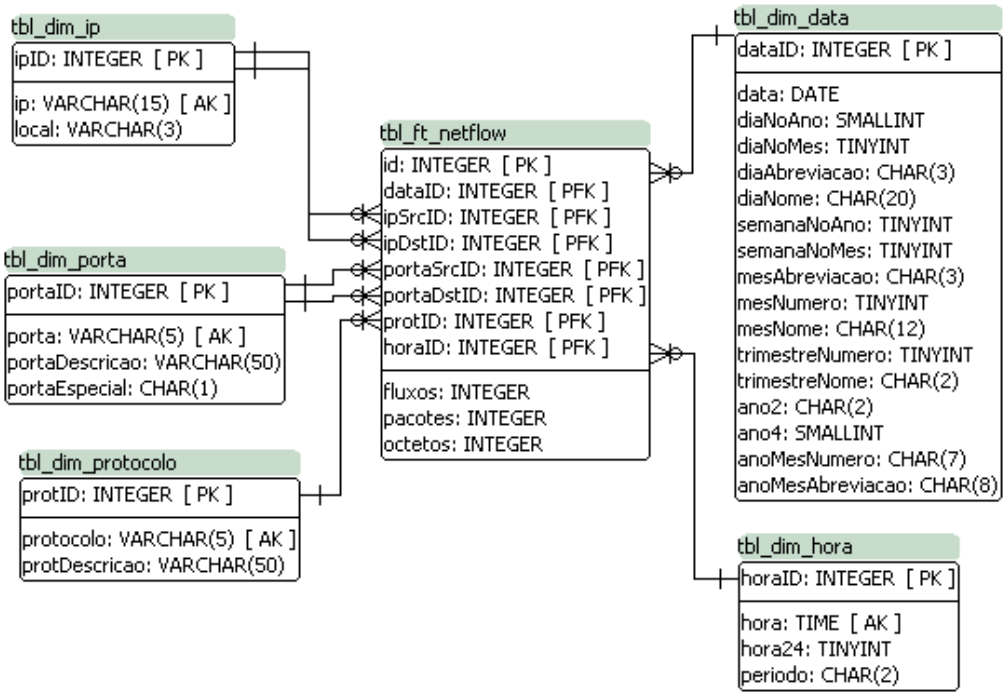


Figura 4.5 – Modelo dimensional (netflowstar).

As dimensões, para efeito da composição dos cubos OLAP, podem apresentar duas classificações: dimensões temporais ou dimensões padrão. As dimensões temporais deste modelo são compostas pelas tabelas *tbl_dim_data* e *tbl_dim_hora* que representam as dimensões *Data* e *Hora*, respectivamente. As outras dimensões consideradas padrão, são compostas pelas tabelas: *tbl_dim_ip*, *tbl_dim_protocolo* e *tbl_dim_porta* que são denominadas dimensões: IP, Protocolo e Porta, respectivamente.

As dimensões Data e Hora destinam-se às correlações temporais entre as medidas quantitativas e as dimensões padrões, sendo que a dimensão Data possui granulação máxima em dias e permite uma hierarquização em cinco níveis (*ano, trimestre, mês, semana e dia*), enquanto que a dimensão Hora está composta para a classificação das medidas quantitativas baseadas no horário, permitindo a granulação máxima em horas e com hierarquização em dois níveis (*período e horas*) sendo os períodos o agrupamento de seis horas (0-6, 6-12, 12-18 e 18-24).

A dimensão IP, está interligada à tabela fato por duas relações, pois todos os fluxos *NetFlow* registram os endereços do emissor (*ipSrcID*) e do receptor (*ipDstID*). Sua função é a identificação dos endereços IP que geraram e receberam comunicação e no processo de ETL foram classificados no campo local, em três categorias: lan, man e wan, conforme determinado pela figura 56. Portanto, esta dimensão pode ser hierarquizada em dois níveis (*local e endereço ip*).

A dimensão Protocolo, representam os protocolos da camada de transporte dos fluxos produzidos pelo NetFlow, não possuindo hierarquização.

A dimensão Porta, da mesma forma que a dimensão IP, também está interligada à tabela fato por duas relações, pois todas as comunicações registradas nos fluxos devem indicar assim como o endereço IP (da dimensão IP) a porta de comunicação pelas quais receberão os pacotes tanto a origem quanto o destino. Considerando que este campo possui muitos valores irrelevantes para a análise, foi implementado o campo *portaEspecial* para o agrupamento das portas consideradas reservadas a serviços e relevantes para análise.

4.2.3. Processo de Extração Transformação e Carga

O processo de ETL, conforme fundamentado no item 3.2.5.2 desta dissertação, objetiva: a obtenção dos dados de fluxos capturados, a preparação dos dados a serem

importados, a importação e o armazenamento temporário dos dados, a transformação e validações dos dados e o armazenamento na base de dados do modelo dimensional.

4.2.3.1. Preparação dos dados de fluxos

Os fluxos NetFlow obtidos pelo sensor *fprobe* são enviados ao coletor *flow-capture* que procede ao trabalho de organizar os fluxos recebidos em agrupamentos e realizar o armazenamento dos dados no disco rígido.

Como a configuração de armazenamento do *flow-capture* foi definida para persistir os dados em períodos de 10 minutos, a composição dos fluxos de somente um dia gera 144 arquivos (6 x 24 horas) armazenados em formato binário. Portanto, o primeiro desafio de preparação dos dados passa pelo agrupamento diário de todos os arquivos binários de fluxos em um arquivo em formato de texto puro, conforme operações ilustradas pela figura 4.6

```
root@NetFlow: # cd /home/andre/flows/2012/2012-04/2012-04-29/
root@NetFlow: # flow-cat * | flow-print -f 5 > fluxos20120429.txt
```

Figura 4.6 – Preparação os dados de fluxos diários para importação.

A figura 4.7 ilustra o acesso a pasta que contém os arquivos de fluxos binários e a execução concatenada dos comandos *flow-cat* (que faz a leitura de todos os arquivos binários) e do comando *flow-print* que transforma formata os dados para a versão 5 do protocolo NetFlow e persiste os dados no arquivo com nomenclatura que identifica o período de coleta dos dados (fluxosANOMESDIA.txt), conforme figura 4.7.

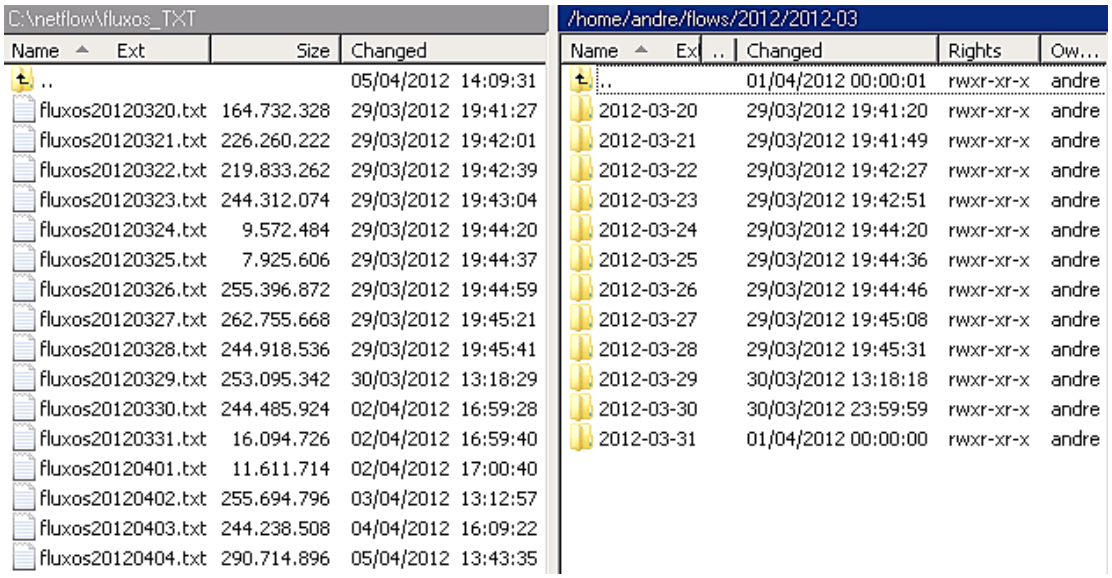


Figura 4.7 – Arquivo de fluxos pronto para a importação.

Com os arquivos de fluxos prontos para a importação, as próximas atividades seguem essencialmente compostas por 3 fases, sendo: a carga dos fluxos nos bancos de dados; a carga das dimensões temporais Data e Hora; e a carga das dimensões padrões IP, Porta e Protocolo e da tabela fato NetFlow.

Todos os processos de ETL a seguir, foram desenvolvidos pela ferramenta *Pentaho Data Integration CE* e todas as bases de dados citadas possuem suas referências estruturais no apêndice B.

4.2.3.2. Importação dos dados para a base de dados *netflowstg*

Esta fase é constituída por uma tarefa denominada *#0_CargaFluxosDB* que contém duas transformações: *#0_carga_fluxos*_OLTP e *#0_carga_fluxos_StgNetFlow*, que efetivamente realizam todo o processo de ETL, conforme figura 4.8.

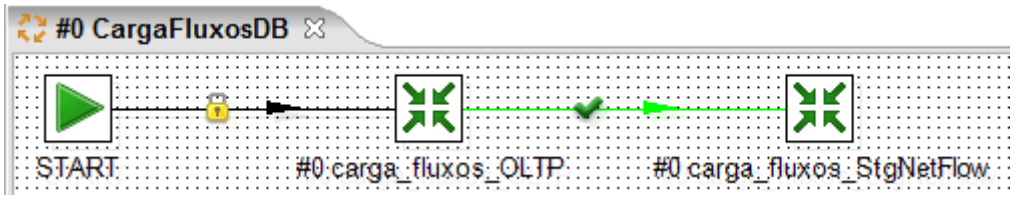


Figura 4.8 – Tarefa que faz a carga dos fluxos nas bases de dados.

O processo de importação dos dados consiste na leitura dos arquivos de texto, ilustrados pela figura 4.7, e a consequente persistência destes fluxos na base de dados *mysql netflowoltp*, conforme figura 4.9.

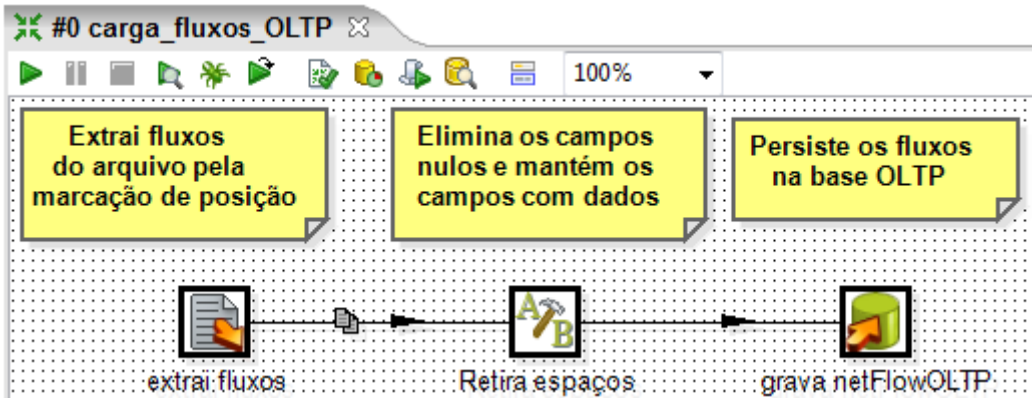


Figura 4.9 – Extração para persistência na base OLTP.

Na sequência, somente os fluxos com endereços IP até a classe C e que são originados ou destinados aos endereços de IP da rede LAN, conforme ilustrado pela figura 56, são recuperados para a persistência na tabela *stg*_netflow da base de dados *netflowstg*, conforme ilustrado pela figura 4.10.

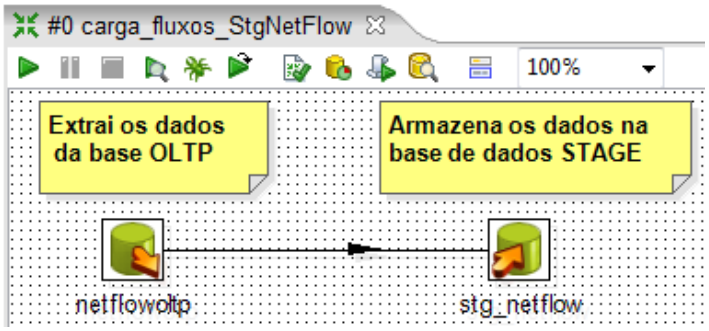


Figura 4.10 – Extração e normalização dos fluxos da rede local.

4.2.3.3. Carga das dimensões temporais

A carga das dimensões que representam a temporalidade dos diversos contextos e das métricas quantitativas, são combinadas pela tarefa #0_CargaDimDataHora, conforme figura 4.11, que engloba as transformações #1_carga_tbl_dim_Data e #1_carga_tbl_dim_Hora.

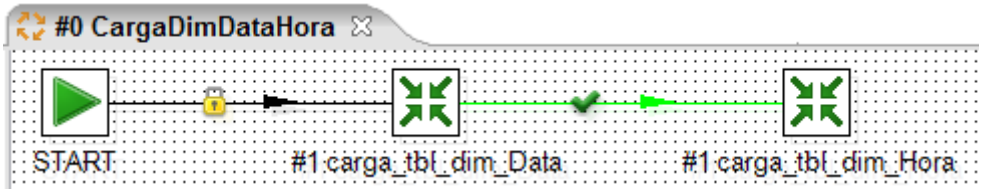


Figura 4.11 – Tarefa para carga das dimensões Data e Hora.

A transformação #1_carga_tbl_dim_Data popula a dimensão Data, através de parâmetros iniciais que definem a quantidade de dias a serem gerados e a data inicial a partir do qual serão gerados todos os dados que posteriormente serão persistidos na tabela *tbl_dim_data* da base de dados *netflowstar*, conforme ilustrado pela figura 4.12.

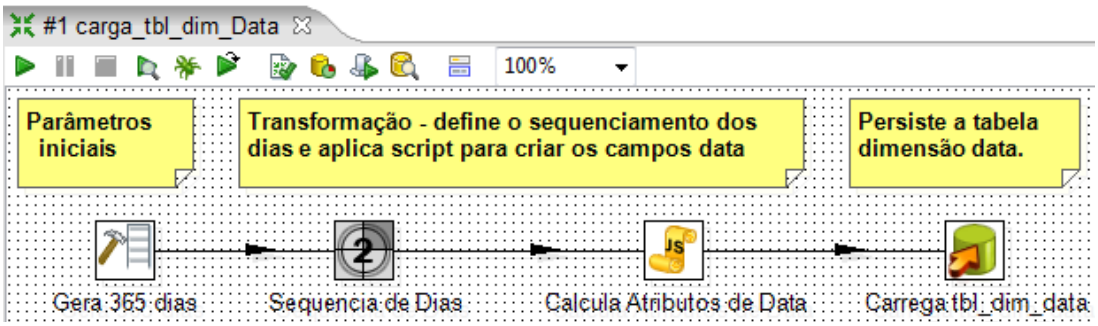


Figura 4.12 – Transformação que popula a dimensão Data.

A transformação #1_carga_tbl_dim_Hora popula a dimensão Hora, através de parâmetros iniciais que geram os elementos que o definem os seus termos, tais como: hora, minuto e segundos. Neste caso como a granulação máxima definida no modelo dimensional são as horas, esta transformação foi adaptada para gerar somente as horas e realizar a

categorização dos períodos para posteriormente persistir na tabela *tbl_dim_Hora* da base de dados *netflowstar*, conforme ilustrado pela figura 4.13.

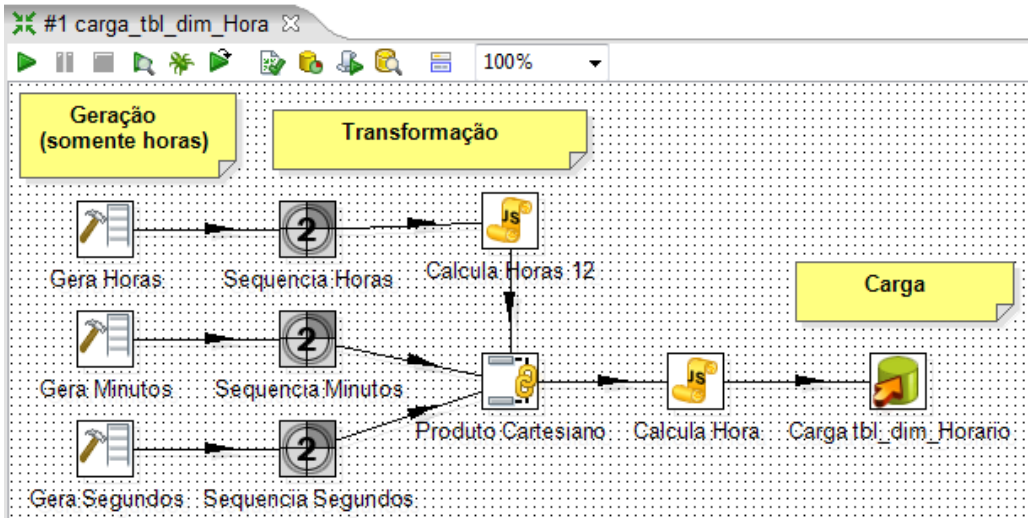


Figura 4.13 – Transformação que popula a dimensão Hora.

A dimensão Hora é uma estática, não necessitando que o processo de carga seja realizado sob nenhuma circunstância. Entretanto a dimensão data necessita de novas cargas pois possui apenas 365 dias carregados, sendo portanto necessária novas inserções a medida se que aproxime da data final armazenada na tabela desta dimensão.

4.2.3.4. Carga da tabela fato e das dimensões padrões

A carga das dimensões padrões *IP, Protocolo e Porta* e da tabela fato *netflow*, são realizadas pelas transformações da tarefa #0_CargaDimensoesFato, conforme figura 4.14,

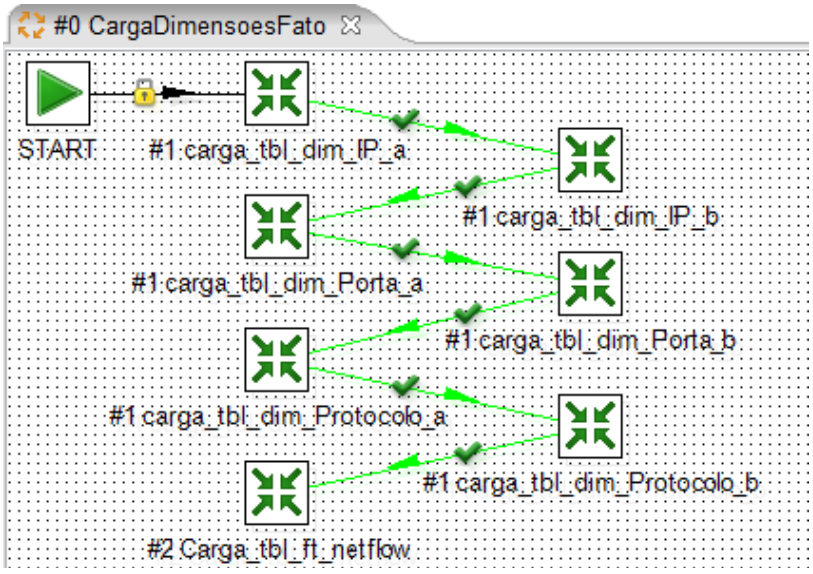


Figura 4.14 – Tarefa que popula a tabela fato e as dimensões padrões.

O processo de transformação e carga da dimensão IP #1_carga_tbl_dim_IP_a, inicia com a extração de todos os endereços IP únicos de origem e destino da tabela stg_*netflow*, unificando as origens em uma única coluna de fluxo para o armazenamento em uma tabela temporária *tmp_ip* na base de dados *netflowstar*, conforme figura 4.15.

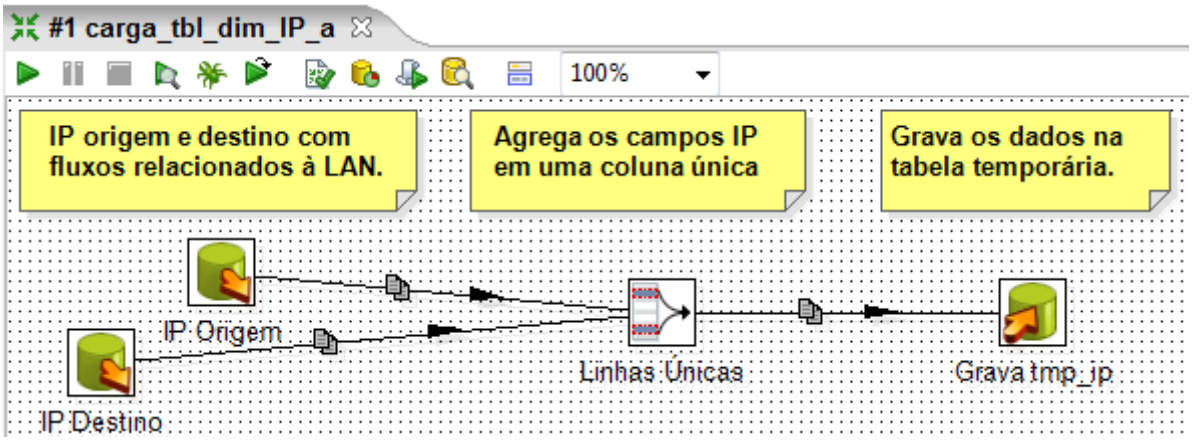


Figura 4.15 – Transformação intermediária para popular a dimensão IP.

No próximo passo, através da transformação #1_carga_tbl_dim_IP_b, somente os endereços IP que não estão cadastrados na dimensão IP (*tbl_dim_ip*) são recuperados da tabela temporária e passam pela transformação do campo local que define a origem do endereço IP, entre: lan, man e wan, pela arquitetura apresentada na figura 56. Posteriormente os dados são persistidos na tabela *tbl_dim_ip* da base de dados *netflowstar*, do modelo dimensional, conforme ilustrado pela figura 4.16.

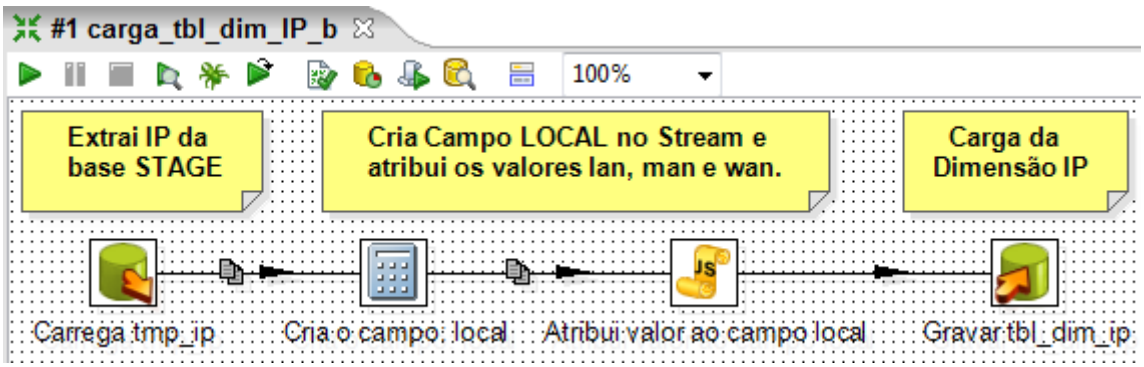


Figura 4.16 – Transformação e carga da dimensão IP.

Muito similar ao processo de transformação e carga dos endereços IP, a transformação #1_carga_tbl_dim_Porta_a, figura 4.17, realiza funcionalmente as mesmas ações com a diferença de que neste caso a busca, a unificação dos fluxos e ocorrem para os campos porta de origem e porta de destino e o armazenamento temporário na tabela *tmp_porta* na base de dados *netflowstar*.

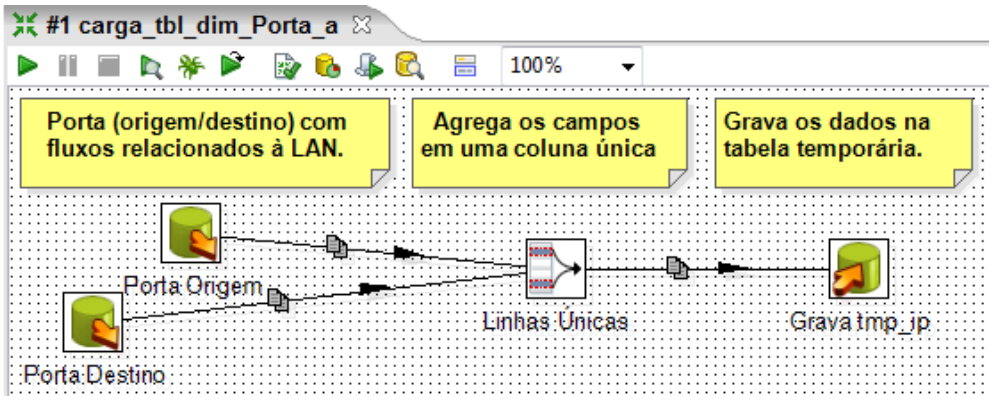


Figura 4.17 – Transformação intermediária para popular a dimensão Porta.

Da mesma forma que no passo anterior, a transformação que popula os dados na dimensão porta #1_carga_tbl_dim_Porta_b, possui as mesmas características funcionais da transformação que popula a dimensão IP quanto ao processo de filtro na extração dos dados da tabela *stg_netflow* que não estão cadastrado na tabela *tbl_dim_porta* efetuando seu ordenamento e finalizando com a persistência dos dados na dimensão Porta, conforme ilustra a figura 4.18.

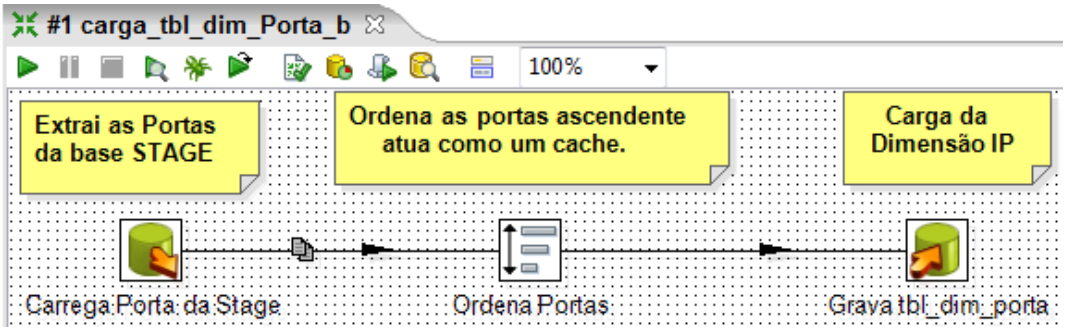


Figura 4.18 – Transformação e carga da dimensão Porta.

A carga da dimensão Protocolo inicia pela execução da transformação denominada #1_carga_tbl_dim_Protocolo_a que realiza a captura dos registros únicos do campo protocolo na tabela *stg_netflow* e persiste na tabela temporária *tmp_protocolo* da base *netflowstar*, conforme ilustrado pela figura 4.19.

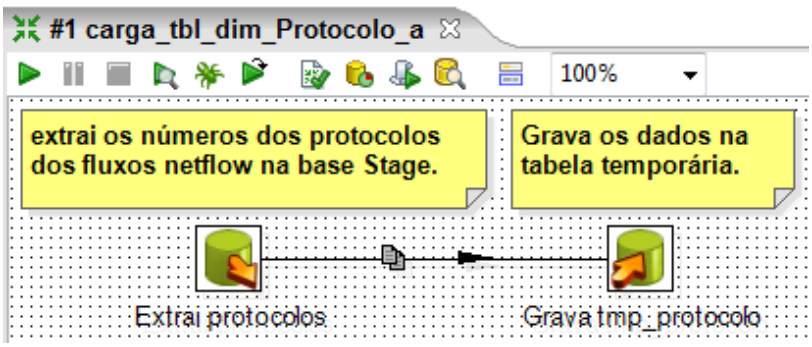


Figura 4.19 – Extrai e prepara a Transformação na tabela temporária.

Com os registros únicos de Protocolos armazenados temporariamente na tabela *tmp_protocolo*, a transformação #1_carga_tbl_dim_Protocolo_b recupera somente os registros que ainda nãos constam da tabela *tbl_dim_Protocolo* que representa a dimensão Protocolo, e depois procede à persistência dos registros novos na tabela *tbl_dim_Protocolo* da base *netflowstar*, conforme ilustrado pela figura 4.20.

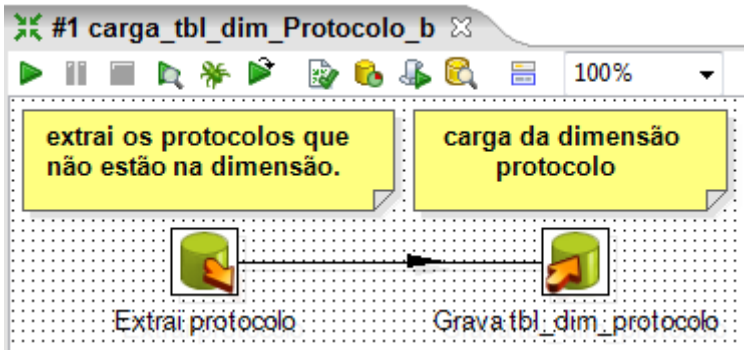
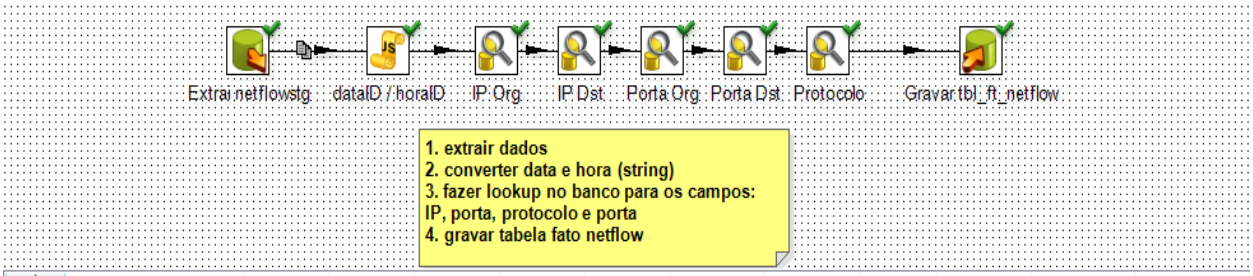


Figura 4.20 – Transformação e carga da dimensão Protocolo.

Finalmente a transformação #2_Carga_tbl_ft_netflow entra em ação, conforme figura 4.21. A extração de todos os registros da tabela *stg_netflow* da base *netflowstg* é procedida pelo ajustamento de todos os campos denominados chaves estrangeiras pela sua comparação com os registros chaves primárias das dimensões que será incorporada à tabela fato *tbl_ft_netflow* juntamente com os demais campos *fluxos, pacotes e octetos*.



#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	Extrai netflowstg	0	0	887978	887978	0	0	0	0	Finished	48mn 24s	306
2	dataID / horaID	0	887978	887978	0	0	0	0	0	Finished	48mn 57s	302
3	IP Org	0	887978	887978	887978	0	0	0	0	Finished	49mn 27s	299
4	IP Dst	0	887978	887978	887978	0	0	0	0	Finished	49mn 56s	296
5	Porta Org	0	887978	887978	887978	0	0	0	0	Finished	50mn 18s	294
6	Porta Dst	0	887978	887978	887978	0	0	0	0	Finished	50mn 36s	292
7	Protocolo	0	887978	887978	887978	0	0	0	0	Finished	50mn 46s	292
8	Gravar tbl_ft_netflow	0	887978	887978	0	887978	0	0	0	Finished	50mn 48s	291

Figura 4.21 – Transformações e carga da tabela fato.

A figura 4.21, acima, ilustra o procedimento de carga na tabela fato *tbl_ft_netflow* dos registros de fluxos relativos aos dias 20, 21, 22 e 23 do mês de março de 2012, aonde foram observados que os quase 890 mil registros consumiram aproximadamente 51 minutos para serem finalizados.

4.2.4. Modelagem dos Cubos OLAP

A modelagem do cubo OLAP, seguindo a abordagem de desenvolvimento da plataforma Pentaho, procedeu através da ferramenta *Pentaho Schema WorkBench* sendo definida sua implementação conforme ilustrado pela figura 4.22.

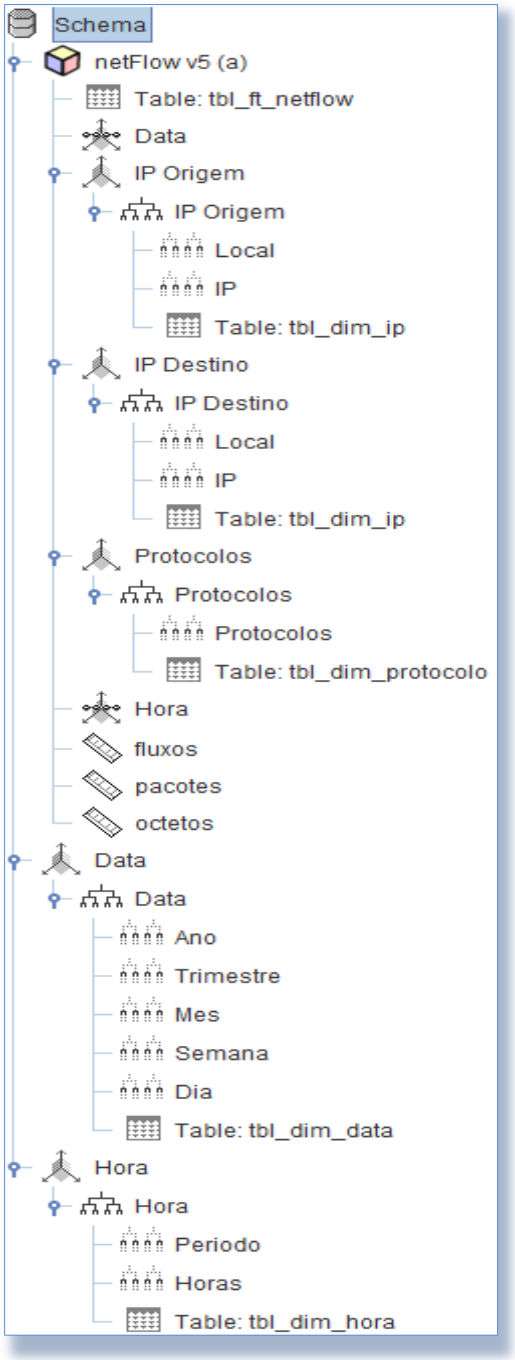


Figura 4.22 – Modelagem do cubo OLAP no PSW.

No Schema *Gerenciamento por Fluxos*, utilizado neste estudo, foram incorporadas diretamente duas dimensões temporais Data e Hora de forma que pudessem ser acessíveis a qualquer novo cubo definido neste *schema*.

A dimensão Data, baseado na tabela *tbl_dim_data* da base *netflowstar*, possui cinco níveis que definem sua granulação, começando por: Ano, Trimestre, Mês, Semana e finalizando em Dia.

A dimensão Hora, baseado na tabela *tbl_dim_hora* da base *netflowstar*, possui agrupamento em apenas dois níveis sendo constituído pelo: período e as horas. O período é o agrupamento de seis em seis horas que se inicia em zero horas até as vinte e quatro horas, conforme definido no modelo dimensional.

O cubo OLAP *NetFlow v5 (a)*, possui como requisito obrigatório para sua criação a definição das métricas quantitativas que neste caso recaiu sobre os campos: fluxos, pacotes e octetos.

As dimensões do cubo *NetFlow v5 (a)*, são: as duas temporais Data e Hora que foram importadas do *schema* para o cubo; e três dimensões regulares: IP Origem, IP Destino e Protocolos.

As dimensões IP Origem e IP Destino, são semelhantes em funcionalidades e configurações pois se baseiam na mesma tabela *tbl_dim_ip*, excetuando-se a questão do campo a que se relacionam na tabela fato, sendo que a dimensão IP Origem se relaciona com o campo *ipsrcID* enquanto que a dimensão IP Destino se relaciona com o campo *ipdstID*.

As duas dimensões são agrupadas de forma semelhante, tendo granulação baseada em dois níveis, sendo: Local e IP, onde Local representa a origem dos fluxos no âmbito da infraestrutura da rede, descrito na figura 4.3, e o IP representa os endereços IP e se consubstancia na granulação máxima destas dimensões.

A dimensão Protocolo que se baseia na tabela *tbl_dim_Protocolos* é a mais simples das dimensões tratadas neste cubo, pois não possui agregação, apenas a disponibilização do campo protocolo para análises analíticas de protocolos em nível da camada de rede do modelo OSI.

Com a modelagem do cubo OLAP *NetFlow v5* pelo *PSW - Pentaho Schema Workbench*, e seguinte publicação na plataforma *PUC - Pentaho User Console* foi possível realizar as análises analíticas conforme ilustrada pela figuras 4.23.

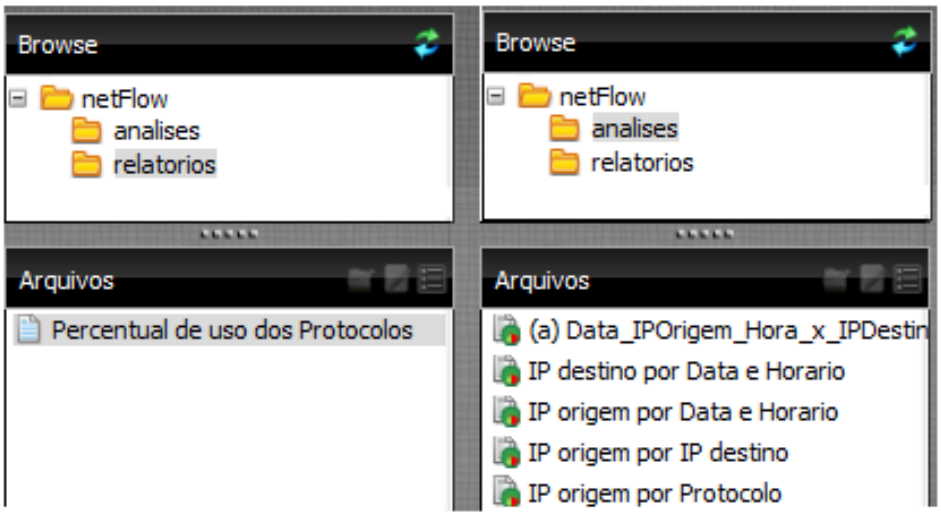


Figura 4.23 – Interface do PUC para análise analíticas e relatórios.

4.2.5. Relatórios Ad-Hoc

Os relatórios são uma ferramenta indispensável na análise das informações disponíveis no modelo dimensional, pois respondem aos questionamentos imediatos e rotineiros. No tocante à plataforma de BI, o componente *Pentaho Report Designer* foi o aplicativo utilizado para a composição dos relatórios e gráficos dos dados disponíveis.

Na figura 4.24, que ilustra um gráfico de torta gerado pelo *PRD*, fica evidenciado a maciça utilização do protocolo TCP nas transmissões ocupando cerca de 96% dos pacotes transmitidos enquanto que o protocolo UDP perfaz apenas 3% e o ICMP 1%.

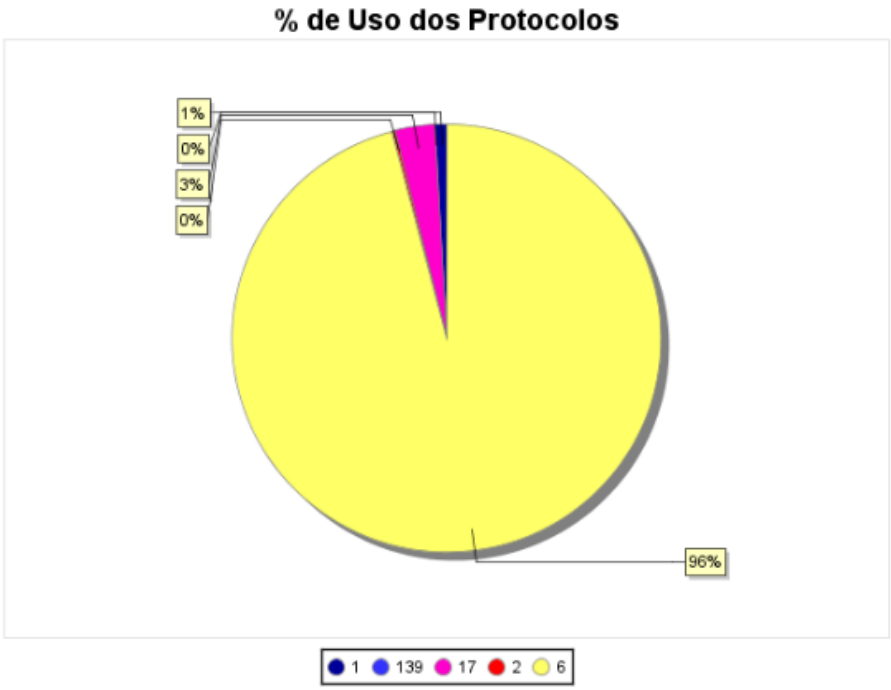


Figura 4.24 – Gráfico gerado pelo Pentaho Report Designer.

4.3. RESULTADOS OBTIDOS

O processo de obtenção dos fluxos, oriundo da definição do ambiente de implantação e da arquitetura baseada no monitoramento do *backbone* de saída de uma rede local, se mostrou estável. Os aplicativos de sensoriamento e coleta, *fprobe* e *flow-tools* respectivamente, se mostraram adequados quando comparado com outras soluções livres disponíveis.

A persistência dos dados pelo coletor, em períodos de 10 minutos pode ser uma boa estratégia para garantir que o uso de memória não seja excedido pelo consumo excessivo dos fluxos encaminhados pelo sensor, o que garante maior disponibilidade dos dados em caso de indisponibilidade dos serviços sensores ou coletores.

As capacidades de análise analíticas possibilitadas pelas técnicas de BI aplicadas aos dados de gerência de fluxos remeteram novas alternativas de análises considerando a construção de um conhecimento mais apurado do comportamento do tráfego de rede, pois habilita o cruzamento imediato de informações, o que se amplia na ótica das probabilidades gerenciais.

Corroborando com a afirmação supracitada, pode ser analisada abaixo com maior granularidade de informações, a visão analítica das informações dispostas no gráfico exibido pela figura 4.25.

		Measures		
IP Origem	Protocolos	fluxos	pacotes	octetos
Todos IP Origem	Todos os Protocolos	29.018.050	166.875.815	144.669.867.888
	6	28.999.218	160.032.651	143.725.851.988
	17	15.016	5.000.288	663.093.261
	1	3.816	1.498.879	193.328.487
	2	0	343.995	87.594.008
	139	0	2	144
wan	Todos os Protocolos	11.531.230	85.608.385	122.336.471.514
	6	11.529.282	84.810.482	122.217.335.775
	17	196	779.655	110.351.170
	1	1.752	18.248	8.784.569
lan	Todos os Protocolos	17.485.626	80.755.652	22.195.391.789
	6	17.469.909	75.221.076	21.508.103.531
	17	13.653	3.710.388	415.187.788
	1	2.064	1.480.191	184.506.318
	2	0	343.995	87.594.008
	139	0	2	144
man	Todos os Protocolos	1.194	511.778	138.004.585
	17	1.167	510.245	137.554.303
	6	27	1.093	412.682
	1	0	440	37.600

Figura 4.25 – Análise dos fluxos nas dimensões IP de origem e Protocolo.

A visão analítica demonstrada na figura 78 evidencia as métricas pela superposição dos Protocolos em relação aos endereços IP de Origem dos fluxos. Neste caso o gráfico da figura 77 aborda os aspectos gerais do uso dos protocolos enquanto que a visão analítica permite o desmembramento de sua análise para as situações da infraestrutura apresentada. Destes, evidencia-se que principal protocolo utilizado em todas as métricas é o TCP, seguido dos protocolos UDP e ICMP.

Pode ser identificado claramente que os fluxos oriundos das redes: lan e wan fazem uso abundante das comunicações orientadas a conexão pelo protocolo *TCP* (6), enquanto que a rede metropolitana (*man*) possui comportamento diferenciado pois faz uso intenso do protocolo de comunicação não orientado a comunicação, UDP.

Na visão do cubo, *IP Origem por IP Destino*, ilustrada pela figura 4.26, as dimensões *Data e IP Origem* estão aninhadas o que garante a análise temporal dos dados em relação ao encaminhamento dos fluxos pela origem. Estas informações ainda podem ser cruzadas pela granulação da dimensão *IP Destino*.

		Measures											
		fluxos				pacotes				octetos			
		IP Destino				IP Destino				IP Destino			
Data	IP Origem	Todos IP Destino	lan	man	wan	Todos IP Destino	lan	man	wan	Todos IP Destino	lan	man	wan
Todas as datas	Todos IP Origem	29.018.050	11.633.976	17.014	17.367.060	166.875.815	89.537.436	587.562	76.750.817	144.669.867.888	122.888.313.609	43.976.164	21.737.578.115
2012	Todos IP Origem	29.018.050	11.633.976	17.014	17.367.060	166.875.815	89.537.436	587.562	76.750.817	144.669.867.888	122.888.313.609	43.976.164	21.737.578.115
3	Todos IP Origem	29.018.050	11.633.976	17.014	17.367.060	166.875.815	89.537.436	587.562	76.750.817	144.669.867.888	122.888.313.609	43.976.164	21.737.578.115
	lan	17.485.626	101.552	17.014	17.367.060	80.755.652	3.417.273	587.562	76.750.817	22.195.391.789	413.837.510	43.976.164	21.737.578.115
	man	1.194	1.194			511.778	511.778			138.004.585	138.004.585		
	wan	11.531.230	11.531.230			85.608.385	85.608.385			122.336.471.514	122.336.471.514		

Figura 4.26 – Análise dos fluxos nas dimensões Data e IP de origem e destino.

A primeira informação visível desta tabela analítica é a ausência de dados nos cruzamentos entre as linhas e colunas man e wan, que se explica pela própria definição do problema a ser tratado nesta dissertação que é a análise dos fluxos de um *backbone* de rede local, que retira essa abrangência do escopo do estudo proposto.

Essa visão permite a verificação do comportamento geral da infraestrutura bem como dos fluxos enviados e recebidos pela rede *lan* e que pela sua análise evidencia-se que a principal confluência ocorre no sentido *wan* para *lan* como respostas às solicitações clientes oriundas da rede *lan*.

Esta visão do cubo apresenta ainda uma particularidade analítica de ordenação aonde a dimensão *IP Destino* está apresentada de forma agrupada pelas métricas quando tradicionalmente o que se comumente usa é o inverso.

A visão do cubo apresentada pela figura 4.27, é composta pelas dimensões Data e Hora, intercalada com a dimensão IP Origem. Esta visão permite a identificação pelas métricas do comportamento das solicitações de fluxos de forma agrupada ou desagrupada pelos locais de origem.

Analisando as solicitações de forma agrupadas, pode ser verificado que os horários de pico de volume tráfego ocorreram as 08 e 15 horas, enquanto que o pico de transações de pacotes ocorreu as 08 e 11 horas e os horários em que mais solicitações simultâneas ocorreram foi as 11 e 08 horas.

		IP Origem											
		Todos IP Origem			lan			man			wan		
		Measures			Measures			Measures			Measures		
Data	Hora	fluxos	pacotes	octetos	fluxos	pacotes	octetos	fluxos	pacotes	octetos	fluxos	pacotes	octetos
Todas as datas	all	29.018.050	166.875.815	144.669.867.888	17.485.626	80.755.652	22.195.391.789	1.194	511.778	138.004.585	11.531.230	85.608.385	122.336.471.514
	0	66.215	671.822	195.949.456	36.425	378.762	54.068.117	0	7.983	2.405.677	29.790	285.077	139.475.662
	1	6.944	34.852	15.283.082	3.825	18.067	2.306.063				3.119	16.785	12.977.019
	2	2.794	15.793	10.036.170	1.611	7.091	1.178.301				1.183	8.702	8.857.869
	3	2.189	8.937	6.091.208	1.217	4.111	1.066.393				972	4.826	5.024.815
	4	1.811	15.196	11.592.026	1.053	6.392	657.072				758	8.804	10.934.954
	5	1.912	10.522	6.336.426	1.027	4.748	734.160				885	5.774	5.602.266
	6	114.990	460.516	342.837.382	91.611	252.728	43.116.667	0	523	181.482	23.379	207.265	299.539.233
	7	1.505.772	7.460.090	5.298.165.064	927.824	3.701.503	711.576.652	177	30.664	9.860.863	577.771	3.727.923	4.576.727.549
	8	4.123.120	22.793.475	17.865.048.134	2.438.325	10.885.683	1.960.111.713	363	143.428	37.915.715	1.684.432	11.764.364	15.867.020.706
	9	2.766.767	15.141.372	13.186.246.310	1.650.363	7.210.886	1.457.986.991	54	40.029	13.377.356	1.116.350	7.890.457	11.714.881.963
	10	1.828.327	10.928.858	10.890.272.099	1.083.332	4.978.023	960.476.561	192	3.564	1.197.248	744.803	5.947.271	9.928.598.290
	11	4.149.975	20.745.037	15.452.657.365	2.637.605	10.831.720	2.011.426.204	102	142.632	32.911.457	1.512.268	9.770.685	13.408.319.704
	12	2.353.125	11.950.683	11.758.498.598	1.448.391	5.750.169	1.218.246.817	108	37.722	12.160.865	904.626	6.162.792	10.528.090.916
	13	2.196.094	12.964.054	11.210.069.513	1.272.841	6.047.134	1.301.116.307	90	35.225	11.158.386	923.163	6.881.695	9.897.794.820
	14	3.420.451	19.241.004	16.111.048.547	2.031.454	9.158.280	3.090.431.098	60	28.658	9.990.780	1.388.937	10.054.066	13.010.626.669
	15	2.567.002	18.727.734	17.117.917.938	1.499.229	9.356.568	6.086.522.114	48	10.880	3.597.858	1.067.725	9.360.286	11.027.797.966
	16	1.957.181	12.644.878	11.819.565.712	1.176.056	5.939.836	2.183.856.439	0	2.736	738.048	781.125	6.702.306	9.634.971.225
	17	1.466.671	9.234.526	10.171.274.658	886.738	4.119.137	803.401.605	0	1.794	655.742	579.933	5.113.595	9.367.217.311
	18	279.754	1.713.935	1.692.968.086	167.443	770.397	128.413.827	0	5	1.557	112.311	943.533	1.564.552.702
	19	97.234	533.120	594.800.508	58.062	250.204	55.253.731				39.172	282.916	539.546.777
	20	28.209	135.955	141.062.748	17.288	62.510	10.312.010				10.921	73.445	130.750.738
	21	12.451	74.815	93.908.294	7.691	35.687	5.992.339				4.760	39.128	87.915.955
	22	60.190	324.324	487.929.804	41.026	144.462	29.062.541	0	70	15.136	19.164	179.792	458.852.127
	23	8.872	1.044.317	190.308.760	5.189	841.554	78.078.067	0	25.865	1.836.415	3.683	176.898	110.394.278

Figura 4.27 – Análise dos fluxos nas dimensões Data, Hora e IP origem.

A análise desagrupada demonstra que o grande volume dos dados é originado pela rede *wan* tendo seus picos de volume de tráfego as 08 e 11 horas, fato que se repete para a rede *man*. Em contrapartida, a rede *lan* tem seu pico de tráfego nos horários das 15, 14 e 16 horas, ou seja, no período vespertino.

Mesmo em horários diferentes o volume de tráfego da rede *lan* comparada com a rede *wan* gira em torno de 20% a desfavor para a primeira, tendo como comportamento anormal

exatamente o horário das 15 horas quando o volume da rede *lan* situa-se acima dos 50% do total da rede *wan.*

No indicador de quantidade de fluxos e quantidade de pacotes todas as redes possuem o pico no horário das 08 horas, exceto pela rede *lan* que possui a maior quantidade de fluxos às 11 horas.

Muito embora as visões analíticas tenham grande poder de análise, os relatórios disponibilizados permitem a análise pontual e circunstancial de qualquer fato que se deseje obter ciência, como pode ser verificado pela figura 77, e neste aspecto, as técnicas de BI também se mostraram como um fator diferencial, pois a possibilidade de geração de novos relatórios é facilitada pela arquitetura dos aplicativos de BI.

4.4. SÍNTESE DO CAPÍTULO 4

Neste capítulo foi abordado todo o desenvolvimento do modelo proposto de gerenciamento passivo baseado em fluxos suportado pelas técnicas de BI por meio do conjunto de ferramentas da *Pentaho*.

A arquitetura e infraestrutura utilizada para a captura, composição e armazenamento dos fluxos pelos sensores e coletores foi dissecada, bem como os procedimentos necessários para a disponibilização dos dados nos arquivos de fluxos. Ainda, todos os processos de extração e transformação empregados para a persistência temporária na base de Stagging e a consequente carga dos dados transformados na base de dados do modelo dimensional foi objeto de análise e descrição.

A composição da base de dados utilizada na modelagem dimensional do protocolo NetFlow v5, bem como a definição das agregações utilizadas na composição da modelagem do cubo OLAP que capacita a análise analítica dos dados, foi amplamente alicerçado.

Finalizando, foi realizada a análise do comportamento do tráfego na rede local com base nas diversas visões do cubo *netflow*, demonstrando o poder analítico da integração das técnicas de BI com as informações gerenciais de fluxos.

5. CONCLUSÃO

O gerenciamento baseado no monitoramento por fluxos vem ocupar uma lacuna relativa ao aspecto gerencial de contabilização, pela definição e conceituação de fluxos de comunicação, fato inexistente nas duas abordagens de gerenciamento: protocolo SNMP e monitoramento por captura de pacotes, tratadas inicialmente nesta dissertação.

Essa conceituação habilita o registro das transações de fluxos de comunicação sem a carga do pacote o que viabiliza seu sensoriamento e coleta, atividades que são amplamente disponibilizadas por diversos protocolos e ferramentas livres.

Dentro deste contexto, a escolha pelo protocolo *netflow* em sua quinta versão, decorre de sua ampla utilização mercadológica fato que herda a este trabalho de pesquisa uma maior relevância sob o ponto de vista prático da solução proposta.

Ainda, considerando a característica flexível da abordagem da modelagem dimensional, aplicada ao desenvolvimento do projeto de *Business Intelligence* este trabalho de pesquisa, ainda possui sua aplicabilidade amplificada, pois pelo próprio conceito da elaboração dos *Data Marts*, toda a solução pode ser interoperável com novos requisitos que por ventura surjam.

Analisando a capacidade de análise, todo o poder disponibilizado pelas ferramentas de BI, em especial à plataforma *Pentaho* que foi utilizada nesta dissertação, traduz-se em um diferencial perante as ferramentas livres disponíveis, que se concentram na disponibilização das informações por meio de relatórios e consultas analíticas.

Porém sob o ponto de vista operacional, ficou evidente que esta solução necessita de recursos computacionais considerados para que os tempos de carga dos bancos de dados e respostas às consultas OLAP ocorram de forma habilidosa.

A grande contribuição desta pesquisa de dissertação está na validação da proposta de integração das técnicas de BI aplicadas ao gerenciamento passivo baseado no monitoramento por fluxos, pela capacidade, flexibilidade e escalabidade providas nesta solução.

5.1. TRABALHOS FUTUROS

A continuação das pesquisas, com base nos estudos apresentados por este trabalho de dissertação pode ter vários enfoques, tais como:

- Incorporar ao modelo apresentado, informações adicionais de outros sistemas de infraestrutura para aumentar a granulação das informações disponíveis pelo BI, tais como: inserção de uma nova dimensão denominada Usuários para identificação da origem dos fluxos ou nova propositura da arquitetura de infraestrutura para identificação de mais locais de confluência de fluxos.
- Análise comparativa da implementação desta solução proposta em outras plataformas de BI livres.
- Realizar um estudo de capacidade e desempenho para o suporte pelas plataformas de BI, dos grandes volumes de dados obtidos.
- Realizar um estudo de séries temporais com os dados obtidos, para realizar provisionamento antecipado de fluxos de tráfego.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAMSON, C. (2010). *Star Scheme – The Complete Reference.* McGraw-Hill.

BITTERER, A. (2008). *Who´s Who in Open-Source Business Intelligence.* Gartner RAS Core Research Note G00156326.

CALIGARE, s. (Janeiro de 2012). *NetFlow Portal*. Fonte: http://netflow.caligare.com/index.htm

CALYAM, P. e. (2005). Active and Passive Measurements on Campus regional and National Network Backbone Plath*. IEEE.

CARUSO, L. C. (2005). Proposta de Arquitetura para NIDS acelerado por Hardware. *Dissertação de Mestrado* . PUC-RS.

CASE, J. (Maio de 1990). *RFC 1157. Simple Network Management Protocol (SNMP).* Acesso em Fevereiro de 2012, disponível em http://datatracker.ietf.org/doc/rfc1157

CASTERS, M. (Abril de 2010). *Pentaho Data Integration 4 and MySql.* Acesso em Abril de 2012, disponível em http://www.ibridge.be/files/Pentaho%20Data%20Integration%204.0%20and%20MySQL.pdf.

CHAUDHURI, S., & DAYAL, U. (1997). An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record.*

CHAUDURI, S., MOTWANI, R., & NARASAYYA, V. (1998). Random Sampling for histogram construction: How much is enough? *In Proceedings of the ACM SIGMOD.*

CHENG, G. G. (2008). Adaptive Aggregation Flow Measurement on high speed Links. IEEE.

CISCO. (2004). *Cisco Netflow.* Acesso em Janeiro de 2012, disponível em Cisco Systems, Inc: http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html

CLAISE, B. (Janeiro de 2008). *RFC 5101. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow InformationMAC: Keyed-Hashing for Message Authentication.* Fonte: IETF: http://tools.ietf.org/html/rfc5101

CLAISE, B.; WOLTER, R. (2007). *Network management: Accounting and Performance Strategies.* EUA: Cisco Press.

DABIR, A., & MATRAWY, A. (2008). Bottleneck Analysis of Traffic Monitoring using Wireshark. *IEEE* .

DING, Y., & HU, Z. (2011). A Study and realization on Searching the SNMP Agent based on BER. *IEEE* .

DOUGLAS, M. e. (2005). *Essential SNMP, 2nd Edition.* O'reilly.

DUFFIELD, N. G.; LUND, C.; THORUP, M. (2003). Estimation Flow Distributions from Sampled Flow Statistics. *ACM SIGCOMM.*

DUFFIELD, N. G.; LUND, C.; THORUP, M. (2002). Properties and Predictions of Flow Statistics from Sampled Packet Streams. *ACM SIGCOMM IMW.*

ELMASRI, R. (2011). *Sistemas de Banco de Dados. 6ª Ed.* Pearson Addison Wesley.

FAYYAD, U., HAUSSLER, D., & STOLORZ, P. (1996). KDD for Science Data Analysis: Issues and Examples. . *Proceedings of the Second International Conference on Knowledge*

Discovery and Data Mining (KDD-96), (pp. 55-56). Evangelos Simoudis and Jia Wei Han en Usama Fayyad.

FELDMANN, A., & all, e. (1999). Performance of Web Proxy Caching in Heterogeneous Bandwidth Environments. *IEEE INFOCOM.*

FLOW-TOOLS. (2012). Acesso em Janeiro de 2012, disponível em http://www.splintered.net/sw/flow-tools

FORTULAN, M. R. (2005). Uma proposta de Aplicação de Business Intelligence no Chão-de-Fábrica. *Revista Gestão & Produção* , v. 12, nº 1, p. 55-66.

FPROBE. (2012). Acesso em Janeiro de 2012, disponível em http://fprobe.sourceforge.net

FULLMER, M. (2012). *FLOW-TOOLS*. Acesso em Janeiro de 2012, disponível em http://www.splintered.net/sw/flow-tools

GARTNET, G. (2009). *IT Glossary - Business Inteligence.* Acesso em Fevereiro de 2012, disponível em http://www.gartner.com/ technology/ it-glossary/business-intelligence.jsp

GOMES, C. L. (2008). Análise de Tráfego de Backbones baseada em Sistemas Gerenciadores de Streams de Dados. *Dissertação de Mestrado* . UFPR.

HARRINGTON, D., & all, e. (Dezembro de 2002). *STD62. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks.* Acesso em Fevereiro de 2012, disponível em http://tools.ietf.org/html/std62.

HOLSAPPLE, C. W. (1996). *Decision Support Systems.* St.Paul,MN: West Publishing Company.

IANA. (s.d.). *Internet Control Message Protocol (ICMP) Parameters.* Acesso em Janeiro de 2012, disponível em http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xml

IEEE Computer Society. (2001). *IEEE 802. Standard for Local and Metropolitan Area Networks: Overview and Architecture.* Acesso em Fevereiro de 2012, disponível em http://standards.ieee.org/getieee802/download/802-2001.pdf

IEEE Computer Society. (2008). *IEEE Standard for Information Technology-Specific requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.* Fonte: 802.3-2008, IEEE: http://standards.ieee.org/about/get/802/802.3.html

IETF. (2011). *IP FLow Information Export Working Group.* Acesso em Janeiro de 2012, disponível em http://datatracker.ietf.org/wg/ipfix/charter/

INMON, W. H. (2005). *Building a Data Warehouse, 4ª Ed.* Wiley Publishing.

KIMBALL, R. (2002). *The Data WareHouse Toolkit – The Complete Guide to Dimensional Modeling. 2ª ed.* New York: NY: Wiley Publishing.

KREJCÍ, R. (2009). Network Traffic Collection with IPFIX Protocol. *Master Thesis* . Masarykova Univerzita.

KUMAR, A., & all, e. (2004). Data streaming algorithms for efficient and accurate estimations of flow size distribution. *ACM SIGMERICS.*

KUMAR, A., XU, J., LI, L., & WANG, J. (2003). Space Code Bloom Filter for Efficient Traffic Flow Measurement. *ACM/USENIX IMC.* Miami.

KUROSE, J. F. (2010). *Redes de Computadores e a Internet: uma abordagem top-down.* São Paulo: Pretince Hall.

LEE, Y. K. (2010). An Internet Traffic Analysis Method with MapReduce. IEEE.

LOGOCKI, N. P. (2008). Uma Ferramenta de Monitoramento de Redes usando Sistemas Gerenciadores de Streams de Dados. *Dissertação de Mestrado* . UFPR.

LOPES, R. V. (2003). *Melhores Práticas para Gerência de Redes de Computadores.* Rio de Janeiro: Campus.

LUCAS, M. (2010). *Network Flow Analysis.* San Francisco: No Starch Press.

MCCABE, J. D. (2007). *Network Analysis Architecture, and Design 3ª Ed.* EUA: Morgan Kaufmann.

MCCLOGHRIE, K. e. (Abril de 1999). *RFC 2578. Structure of Management Information Version 2 (SMIv2).* Acesso em Dezembro de 2011, disponível em http://datatracker.ietf.org/doc/rfc2578

MCCLOGHRIE, K. (Março de 1991). *RFC 1213. Management Information Base for Network Management of TCP/IP-based internets: MIB-II.* Acesso em Fevereiro de 2011, disponível em http://datatracker.ietf. org/ doc/rfc1213.

MILLER, M. A. (1999). *Managing Internetworks with SNMP.* IDG Books Worldwide.

MONTORO, M. (2012). *Cain & Abel*. Acesso em Dezembro de 2011, disponível em http://www.oxid.it

MORRIS, S. B. (2003). *Network Management, MIBs and MPLS: Principles, Design and Implementation.* Prentice Hall.

MOSS, L. T., & ATRE, S. (2003). *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications.* Addison Wesley.

OSI, Information Technology. (1989). ISO/IEC 7498-4 - Basic Reference Model Part 4: Management Framework.

OSI, Information Technology. (2008). ISO/IEC 8825-1 - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

PAOLANTONIO, J. A. (Dezembro de 2010). *Technology for the OSS DSS Study Guide.* Acesso em Abril de 2012, disponível em http:// press.teleinteractive.net/oss

Pentaho CE. (2004). Acesso em Abril de 2012, disponível em http://community.pentaho.com

PERKINS, D. (1997). *Understanding SNMP MIBs.* Prentice-Hall.

PHAAL, P., PANCHEN, S., & MCKEE, N. (Setembro de 2001). *RFC 3176. InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks.* Fonte: IETF: http://tools.ietf.org/html/rfc3176

PONNIAH, P. (2010). *Data Warehouse Fundamentals For IP Professionals - Second Edition.* Wiley & Sons.

PRESUHN, R. (Dezembro de 2002). *RFC 3416. Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP).* Acesso em Fevereiro de 2012, disponível em http://datatracker.ietf.org/ doc/rfc3416.

PUANGPRONPITAG, S., & MASURAI, N. (2009). An efficient and feasible solution do ARP Spoof Problem. *IEEE* .

QADEER, M. A., & ZAHID, M. (2010). Network Traffic Analysis and Intrusion Detection using Packet Sniffer. *2a. International Conference on Communication Software and Networks.* IEEE.

REESE, B. (2010). *Closer look: sFlow better than NetFlow?* Acesso em Fevereiro de 2012, disponível em http://www.networkworld.com/community/node/29117

RIBEIRO, B., TOWNSLEY, D., YE, T., & BOLOT, J. F. (2006). Information of Sampled Packets: an Application to Flow Size Estimation. *IMC 06.* Rio de Janeiro.

ROSE, M. (Maio de 1990). *IETF STD 16 - Structure and Identification of Management Information for TCP/IP-based Internets.* Acesso em Dezembro de 2011, disponível em http://tools.ietf.org/html/std16

SALLAM, R., RICHARDSON, J., HAGERLY, J., & HOATMANN, B. (2011). *Magic Quadrant for Business Intelligence Platforms.* Gartner RAS Core Research Note G00210036:.

SANDERS, C. (2007). *Practical Packet Analysis.* EUA: No Starch Press, Inc.

SANTOS, M. T. (2008). *Administração e Projetos de Rede – Gerência de Redes de Computadores.* Rio de Janeiro: RNP.

SEONG-YEE, P., & all, e. (2008). Design and Implementation of V6SNIFF: an efficient IPv6 Packet Sniffer. *3a. International Conference on Convergence and Hybrid Information Technology.*

sFlow. (2001). *sFlow Specifications.* Acesso em Fevereiro de 2012, disponível em http://www.sflow.org

SIMON, H. A. (1960). *The New Science of Management Decision.* New York, NY: HarperandRow.

SOARES, V. J. (1998). Modelagem Incremental no Ambiente de Data Warehouse. *Dissertação de Mestrado* . UFRJ.

SPECIALSKI, E. S. (2000). Gerência de Redes de Computadores e de Telecomunicações. *Material de estudo – Apostila* . Florianópolis: UFSC.

STALLINGS, W. (1998). *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2 Third Edition.* Addison-Wesley Professional.

STALLINGS, W. (1998). SNMPv3: A Security Enhancement for SNMP. *IEEE, Communications Surveys* .

STEWART, R. (Setembro de 2007). *RFC 4960. Stream Control Transmission Protocol.* Acesso em Fevereiro de 2012, disponível em IETF: http://datatracker.ietf.org/doc/rfc4960.

TANENBAUM, A. S. (2011). *Redes de Computadores - 5ª edição.* São Paulo: Pretince Hall.

TAYLOR, J. (2012). *Decision Management Systems - A Practical Guide to Using Business Rules and Predictive Analytics.* IBM Press.

TERESO, M., & BERNARDINO, J. (2011). Ferramentas Open Source de Business Intelligence para PME´s. *6ª Conferência Ibérica de Sistemas de Tecnologias de Informação.*

THOMÉ, A. C. (2007). *Redes Neurais – Uma Ferramenta para KDD e DataMining.* Acesso em Abril de 2012, disponível em Inteligência Computacional - Notas de aulas: http://equipe.nce.ufrj.br/thome/grad/nn/mat_didatico/apostila_kdd_mbi.pdf

VERCELLIS, C. (2008). *Business Intelligence: Data Mining and Optimization for Decision Making.* Wiley Publishing.

VIGLIONI, G. M. (2007). Metodologia para previsão de demanda ferroviária utilizando DataMining. *Dissertação de Mestrado* . IME.

WATSON, H. J., & ARIYACHANDRA, T. (Julho de 2005). *Data Warehouse Atchitectures: Factos in the Selection Decision and the Success of the Architecture.* Acesso em Fevereiro de 2012, disponível em http://www.terry. uga. edu/~hwatson/DW_Architecture_Report.pdf

WITHEE, K. (2010). *Microsoft Business Intelligence for Dummies.* Wiley Publishing.

XIAOFAN, L., & all, e. (2010). Design and Research based on WinPcap Network Protocol Analysis System. IEEE.

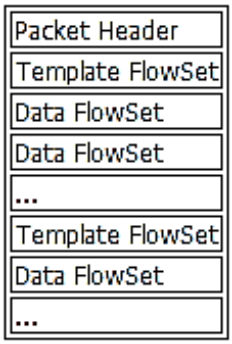
ZELTSERMAN, D. (1999). *A Practical Guide to SNMPv3 and Network Management.* Prentice-Hall.

ZHANG, J., & MOORE, A. (2007). Traffic trace artifacts due to Monitoring via Port Mirroring. IEEE.

ANEXOS

ANEXO A - PROTOCOLO NETFLOW V9

Tabela A.1 – Pacote de exportação NetFlow v9. Fonte:[62].

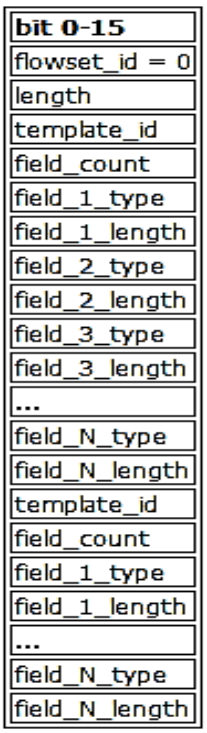


Campos	*Descrição*
Packet Header	*Cabeçalho do conjunto de Fluxos*
Template FlowSet	*Modelo do conjunto de Fluxos*
Data FlowSet	*Dados de Fluxos*

Tabela A.2 – Formato do cabeçalho (*Packet Header*). Fonte:[62].

Bytes	Campos	Descrição
0-1	version	*Número da versão do formato de exportação NetFlow.*
2-3	count	*Números de fluxos exportados neste pacote, tanto modelo quanto dados (1-30).*
4-7	sys_uptime	*Tempo em milissegundos desde a reinicialização do dispositivo.*
8-11	unix_secs	*Data e hora baseados em segundos desde 000 UTC 1970.*
12-15	package_sequence	*Contador sequencial de todos os pacotes enviados pelo sensor.*
16-19	source_id	*Um valor de 32 bit's usado para identificação única para todos os fluxos exportados de um dispositivo sensor.*

Tabela A.3 – Modelo do conjunto de fluxos (*Template FlowSet)*. Fonte:[62].



Bit 0-15	*Descrição*
flowset_id = 0	*É usado para distinguir o modelo de registros dos registros de dados. Seu valor sempre se situa entre 0 e 255. Um registro de dados (data record) sempre tem um valor para flowset_id maior que 255.*
length	*É o tamanho total do conjunto de fluxos. Pelo fato de um modelo de conjunto de fluxos conter múltiplos identificadores de modelos (template ID), este valor pode ser usado para determinar a posição do próximo registro de fluxo.*
template_id	*Identificador único do modelo de conjunto de fluxos, e pode ter valores entre 0 e 255.*
field_count	*Quantidade de registros dentro do conjunto de registros do modelo.*
field_type	*Especificação do tipo do campo.*
field_length	*Tamanho do campo.*

APÊNDICE A – RECURSOS TECNOLÓGICOS PARA IMPLANTAÇÃO

B.1 - Processador

```
root@NetFlow:/etc/default# cat /proc/cpuinfo
      cpu cores       : 2
      vendor_id       : AuthenticAMD
      model name      : AMD Phenom(tm) II X2 B55 Processor
      cpu MHz         : 800.000
      cache size      : 512 KB
```

B.2 - Memória

```
root@NetFlow:/etc/default# cat /proc/meminfo
      MemTotal:        1794320 kB
      MemFree:          255152 kB
```

B.3 – Disco Rígido

```
root@NetFlow:/etc/default# df
Sist. Arq.           1K-blocos      Usad Dispon.   Uso% Montado em
/dev/sda5             85982624   8845604  72769304  11% /
none                    889452       656    888796   1% /dev
none                    897160       196    896964   1% /dev/shm
none                    897160       100    897060   1% /var/run
none                    897160         0    897160   0% /var/lock
```

B.4 – Interfaces de Rede

```
root@NetFlow:/etc/default# ifconfig
eth0      Link encap: Ethernet  Endereço de HW 64:31:50:48:bb:0d
          inet end.: 172.20.22.120  Bcast:172.20.23.255  Masc:255.255.248.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
lo        Link encap: Loopback Local
          inet end.: 127.0.0.1  Masc:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Métrica:1
```

B.5 – Kernel do Linux

```
root@NetFlow:/etc/default# cat /proc/version
Linux version 2.6.38-8-generic (buildd@allspice) (gcc version 4.5.2
(Ubuntu/Linaro 4.5.2-8ubuntu3) ) #42-Ubuntu SMP Mon Apr 11 03:31:24 UTC 2011
```

B.6 – Versão do Sistema Operacional

```
root@NetFlow:/etc/default# cat /etc/issue
Ubuntu 11.04 \n \l
```

APÊNDICE B – CÓDIGO SQL DAS BASES DE DADOS

netflowoltp.sql

```
# Dumping database structure for netflowoltp
CREATE DATABASE IF NOT EXISTS `netflowoltp`
USE `netflowoltp`;
# Dumping structure for table netflowoltp.netflow
CREATE TABLE IF NOT EXISTS `netflow` (
  `inicio` varchar(17) DEFAULT NULL,
  `fim` varchar(17) DEFAULT NULL,
  `srcIf` varchar(1) DEFAULT NULL,
  `srcIP` varchar(15) DEFAULT NULL,
  `srcPorta` varchar(5) DEFAULT NULL,
  `dstIf` varchar(1) DEFAULT NULL,
  `dstIP` varchar(15) DEFAULT NULL,
  `dstPorta` varchar(5) DEFAULT NULL,
  `protocolo` varchar(4) DEFAULT NULL,
  `fluxos` varchar(5) DEFAULT NULL,
  `pacotes` varchar(50) DEFAULT NULL,
  `octetos` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

netflowoltp.sql

```
# Dumping database structure for netflowstg
CREATE DATABASE IF NOT EXISTS `netflowstg
USE `netflowstg`;
# Dumping structure for table netflowstg.stg_dim_ip
CREATE TABLE IF NOT EXISTS `stg_dim_ip` (
  `IP` varchar(15) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
# Data exporting was unselected.
# Dumping structure for table netflowstg.stg_netflow
CREATE TABLE IF NOT EXISTS `stg_netflow` (
  `data` varchar(4) DEFAULT NULL,
  `hora` varchar(5) DEFAULT NULL,
  `srcIP` varchar(15) DEFAULT NULL,
  `srcPorta` varchar(5) DEFAULT NULL,
  `dstIP` varchar(15) DEFAULT NULL,
  `dstPorta` varchar(5) DEFAULT NULL,
  `protocolo` varchar(5) DEFAULT NULL,
  `fluxos` mediumint(8) unsigned DEFAULT NULL,
  `pacotes` mediumint(8) unsigned DEFAULT NULL,
  `octetos` int(10) unsigned DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

netflowstar.sql

```
# Dumping database structure for netflowstar
CREATE DATABASE IF NOT EXISTS `netflowstar`
USE `netflowstar`;
# Dumping structure for table netflowstar.tbl_dim_data
CREATE TABLE IF NOT EXISTS `tbl_dim_data` (
  `dataID` int(10) NOT NULL,
  `data` date NOT NULL,
  `dataCurta` char(12) NOT NULL,
  `dataMedia` char(16) NOT NULL,
  `dataLonga` char(24) NOT NULL,
  `dataCompleta` char(50) NOT NULL,
```

```
  `diaNoAno` smallint(5) NOT NULL,
  `diaNoMes` tinyint(3) NOT NULL,
  `ePrimeiroDiaMes` char(10) NOT NULL,
  `eUltimoDiaMes` char(10) NOT NULL,
  `diaAbreviacao` char(3) NOT NULL,
  `diaNome` char(20) NOT NULL,
  `semanaNoAno` tinyint(3) NOT NULL,
  `semanaNoMes` tinyint(3) NOT NULL,
  `ePrimeiroDiaSemana` char(10) NOT NULL,
  `eUltimoDiaSemana` char(10) NOT NULL,
  `mesNumero` tinyint(3) NOT NULL,
  `mesAbreviacao` char(3) NOT NULL,
  `mesNome` char(12) NOT NULL,
  `ano2` char(2) NOT NULL,
  `ano4` smallint(5) NOT NULL,
  `trimestreNome` char(2) NOT NULL,
  `trimestreNumero` tinyint(3) NOT NULL,
  `anoQuadrimestre` char(7) NOT NULL,
  `anoMesNumero` char(7) NOT NULL,
  `anoMesAbreviacao` char(8) NOT NULL,
  PRIMARY KEY (`dataID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
# Data exporting was unselected.
# Dumping structure for table netflowstar.tbl_dim_hora
CREATE TABLE IF NOT EXISTS `tbl_dim_hora` (
  `horaID` int(8) NOT NULL,
  `hora` time NOT NULL,
  `hora24` tinyint(3) NOT NULL,
  `hora12` tinyint(3) DEFAULT NULL,
  `minutos` tinyint(3) DEFAULT NULL,
  `segundos` tinyint(3) DEFAULT NULL,
  `am_pm` char(3) DEFAULT NULL,
  `periodo` char(2) DEFAULT NULL,
  PRIMARY KEY (`horaID`),
  UNIQUE KEY `time_value` (`hora`),
  FULLTEXT KEY `am_pm` (`am_pm`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
# Data exporting was unselected.
# Dumping structure for table netflowstar.tbl_dim_ip
CREATE TABLE IF NOT EXISTS `tbl_dim_ip` (
  `ipID` int(10) NOT NULL AUTO_INCREMENT COMMENT 'Surrogate key dimensao IP.',
  `ip` varchar(15) DEFAULT NULL COMMENT 'endereco ip dos dispositivos.',
  `local` varchar(3) DEFAULT NULL COMMENT 'identifica se pertence a rede lan = 0,
man=1 ou wan = 2.',
  PRIMARY KEY (`ipID`),
  UNIQUE KEY `ip` (`ip`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Tabela dimensão que armazenará as
origens e destinos.';
# Data exporting was unselected.
# Dumping structure for table netflowstar.tbl_dim_porta
CREATE TABLE IF NOT EXISTS `tbl_dim_porta` (
  `portaID` int(11) NOT NULL AUTO_INCREMENT COMMENT 'Surrogate Key dimensao
aplicacao.',
  `porta` varchar(5) NOT NULL COMMENT 'porta que identifica a aplicacao.',
  `portaDescricao` varchar(50) DEFAULT NULL,
  `portaEspecial` char(1) DEFAULT NULL,
  PRIMARY KEY (`portaID`),
  UNIQUE KEY `idxPortaNumero` (`porta`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Dimensao que armazena os
protocolos da camada de aplicacao.';
# Data exporting was unselected.
```

```
# Dumping structure for table netflowstar.tbl_dim_protocolo
CREATE TABLE IF NOT EXISTS `tbl_dim_protocolo` (
  `protID` int(10) NOT NULL AUTO_INCREMENT COMMENT 'Key olap surrogate key.',
  `protocolo` varchar(5) NOT NULL COMMENT 'numero que identifica o protocolo.',
  `protDescricao` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`protID`),
  UNIQUE KEY `Index 2` (`protocolo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Dimensao que armazena os
protocolos da camada de transporte.';
# Data exporting was unselected.
# Dumping structure for table netflowstar.tbl_ft_netflow
CREATE TABLE IF NOT EXISTS `tbl_ft_netflow` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `dataID` int(10) NOT NULL,
  `horaID` int(10) NOT NULL,
  `ipSrcID` int(10) NOT NULL,
  `portaSrcID` int(10) NOT NULL,
  `ipDstID` int(10) NOT NULL,
  `portaDstID` int(10) NOT NULL,
  `protID` int(10) NOT NULL,
  `fluxos` int(11) NOT NULL,
  `pacotes` int(11) NOT NULL,
  `octetos` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `dataID` (`dataID`),
  KEY `horaID` (`horaID`),
  KEY `ipSrcID` (`ipSrcID`),
  KEY `ipDstID` (`ipDstID`),
  KEY `protID` (`protID`),
  KEY `portaDstID` (`portaDstID`),
  KEY `portaSrcID` (`portaSrcID`),
  KEY `dataID_horaID_ipSrcID` (`dataID`,`horaID`,`ipSrcID`),
  KEY `dataID_horaID_ipDstID` (`dataID`,`horaID`,`ipDstID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='tabela armazena as métricas: qtde
fluxos, pacotes e octetos.';
# Data exporting was unselected.
# Dumping structure for table netflowstar.tmp_ip
CREATE TABLE IF NOT EXISTS `tmp_ip` (
  `IP` varchar(15) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
# Data exporting was unselected.
# Dumping structure for table netflowstar.tmp_porta
CREATE TABLE IF NOT EXISTS `tmp_porta` (
  `PORTA` varchar(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
# Data exporting was unselected.
# Dumping structure for table netflowstar.tmp_protocolo
CREATE TABLE IF NOT EXISTS `tmp_protocolo` (
  `protocolo` varchar(5) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

APÊNDICE C – ARQUIVOS ETL (PDI - SPOON)

Todos os arquivos de transformação com extensão *ktr* utilizados no *Pentaho Data Integration* para o desenvolvimento desta dissertação poderão ser obtidos através de solicitação ao email **andre.valente@cba.ifmt.edu.br**.

Nome	Data de modificaç...	Tipo	Tamanho
#0 carga_fluxos_OLTP.ktr	13/04/2012 03:52	Arquivo KTR	27 KB
#0 carga_fluxos_StgNetFlow.ktr	13/04/2012 10:11	Arquivo KTR	15 KB
#1 carga_tbl_dim_Data.ktr	13/04/2012 11:51	Arquivo KTR	30 KB
#1 carga_tbl_dim_Hora.ktr	13/04/2012 11:41	Arquivo KTR	22 KB
#1 carga_tbl_dim_IP_a.ktr	13/04/2012 15:12	Arquivo KTR	17 KB
#1 carga_tbl_dim_IP_b.ktr	13/04/2012 23:16	Arquivo KTR	18 KB
#1 carga_tbl_dim_Porta_a.ktr	13/04/2012 18:41	Arquivo KTR	17 KB
#1 carga_tbl_dim_Porta_b.ktr	13/04/2012 18:15	Arquivo KTR	16 KB
#1 carga_tbl_dim_Protocolo_a.ktr	13/04/2012 19:03	Arquivo KTR	14 KB
#1 carga_tbl_dim_Protocolo_b.ktr	13/04/2012 19:13	Arquivo KTR	12 KB
#2 Carga_tbl_ft_netflow.ktr	23/04/2012 12:26	Arquivo KTR	21 KB

APÊNDICE D – ARQUIVO MONDRIAN XML (*PSW*)

netflow_v5.xml

```
<Schema name="Gerenciamento por Fluxos">
  <Dimension type="TimeDimension" visible="true" highCardinality="false"
name="Data">
    <Hierarchy name="Data" visible="true" hasAll="true" allMemberName="Todas as
datas" primaryKey="dataID">
      <Table name="tbl_dim_data">
      </Table>
      <Level name="Ano" visible="true" column="ano4" type="Numeric"
uniqueMembers="false" levelType="TimeYears" hideMemberIf="Never">
      </Level>
      <Level name="Mes" visible="true" column="mesNumero" type="Numeric"
uniqueMembers="false" levelType="TimeMonths" hideMemberIf="Never">
      </Level>
      <Level name="Dia" visible="true" column="diaNoMes" type="Numeric"
uniqueMembers="false" levelType="TimeDays" hideMemberIf="Never">
      </Level>
    </Hierarchy>
  </Dimension>
  <Dimension type="TimeDimension" visible="true" highCardinality="false"
name="Hora">
    <Hierarchy name="Hora" visible="true" hasAll="true" allMemberName="all"
primaryKey="horaID">
      <Table name="tbl_dim_hora">
      </Table>
      <Level name="Horas" visible="true" column="hora24" type="Numeric"
uniqueMembers="false" levelType="TimeHours" hideMemberIf="Never">
      </Level>
    </Hierarchy>
  </Dimension>
  <Cube name="netFlow v5" visible="true" cache="true" enabled="true">
    <Table name="tbl_ft_netflow">
    </Table>
    <DimensionUsage source="Data" name="Data" visible="true" foreignKey="dataID"
highCardinality="false">
    </DimensionUsage>
    <Dimension type="StandardDimension" visible="true" foreignKey="ipSrcID"
highCardinality="false" name="IP Origem">
      <Hierarchy name="IP Origem" visible="true" hasAll="true"
allMemberName="Todos IP Origem" primaryKey="ipID">
        <Table name="tbl_dim_ip">
        </Table>
        <Level name="Local" visible="true" column="local" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="IP" visible="true" column="ip" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
      </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="ipDstID"
highCardinality="false" name="IP Destino">
      <Hierarchy name="IP Destino" visible="true" hasAll="true"
allMemberName="Todos IP Destino" primaryKey="ipID">
        <Table name="tbl_dim_ip">
```

```
        </Table>
        <Level name="Local" visible="true" column="local" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="IP" visible="true" column="ip" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
      </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="protID"
highCardinality="false" name="Protocolos">
      <Hierarchy name="Protocolos" visible="true" hasAll="true"
allMemberName="Todos os Protocolos" primaryKey="protID">
        <Table name="tbl_dim_protocolo">
        </Table>
        <Level name="Protocolos" visible="true" column="protocolo" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
      </Hierarchy>
    </Dimension>
    <DimensionUsage source="Hora" name="Hora" visible="true" foreignKey="horaID"
highCardinality="false">
    </DimensionUsage>
    <Measure name="fluxos" column="fluxos" datatype="Numeric" aggregator="sum"
visible="true">
    </Measure>
    <Measure name="pacotes" column="pacotes" datatype="Numeric" aggregator="sum"
visible="true">
    </Measure>
    <Measure name="octetos" column="octetos" datatype="Numeric" aggregator="sum"
visible="true">
    </Measure>
  </Cube>
</Schema>
```