# COLANA – Uma ferramenta para recolha e análise de grandes volumes de tráfego

Pedro Vale Pinheiro[1], Mário Bernardes[1], Fernando Boavida[1,2]

[1] {vapi,mjb,boavida}@ci.uc.pt
Centro de Informática da Universidade de Coimbra
Apartado 3080, 3001-401 Coimbra
PORTUGAL

[2] boavida@dei.uc.pt
Departamento de Engenharia Informática
Pólo II, Pinhal de Marrocos
3030-290 Coimbra
PORTUGAL

## Resumo

A recolha e análise de tráfego em instituições com grande volume de comunicação coloca diversos desafios em termos de armazenamento, tratamento e acesso à informação de caracterização de tráfego. Para além destes problemas, questões de dimensionamento e desempenho podem, também, assumir importância determinante. A presente comunicação apresenta uma ferramenta de recolha e análise de tráfego que endereça os desafios e questões acima enunciados, desenvolvida e em produção no Centro de Informática da Universidade de Coimbra.

## 1. Introdução

O Centro de Informática da Universidade de Coimbra (CIUC) é a entidade responsável pela gestão da infra-estrutura de rede da Universidade de Coimbra e pela gestão do ponto de acesso à RCTS. Actualmente, este ponto de acesso consiste numa porta ATM num Cisco 3640, sendo a largura de banda contratada de 10 Mbps. Todo o tráfego da rede da Universidade de Coimbra, que inclui cerca de 35 redes classe C, é suportado por esta ligação que, normalmente, tem uma elevada taxa de utilização. Apesar desta situação ter sido significativamente melhorada com a introdução de equipamento de gestão de largura de banda, o CIUC sempre teve necessidade de produzir estatísticas referentes quer à utilização da linha pelos utilizadores departamentais quer aos protocolos utilizados. Neste contexto, foi desenvolvida pelo CIUC uma ferramenta que, para além da produção de relatórios de análise de tráfego, permite igualmente detectar casos de utilização anormal de largura de banda e analisar possíveis tentativas de intrusão.

Este artigo descreve a ferramenta desenvolvida e a experiência de utilização após os primeiros seis meses de exploração. Na secção 2 é apresentado o enquadramento e motivação para o trabalho desenvolvido, sendo identificados projectos semelhantes. A secção 3 é dedicada à apresentação da ferramenta desenvolvida, designada Colana (*Traffic **Col**lector/**Ana**lyser*), sendo abordados o desenho da base de dados subjacente, o acesso à informação recolhida, e várias questões relacionadas com o dimensionamento e desempenho da ferramenta. A secção 4 apresenta as conclusões e linhas para trabalho futuro.

## 2. Enquadramento e motivação

O trabalho que deu origem à ferramenta apresentada no presente artigo iniciou-se há cerca de dois anos com a decisão de utilizar o serviço NetFlow[1] do Cisco IOS, que permite exportar a informação de tráfego IP que atravessa um *router*, para contabilização, análise, planeamento ou outras actividades de gestão de tráfego. Esta informação parte do conceito de fluxo, uma sequência unidireccional de pacotes entre uma origem e um destino. A sua identificação é feita com base no endereço IP de origem, endereço IP de destino, protocolo de transporte e porto de aplicação. Para identificação única dos fluxos é, ainda, utilizado o tipo de serviço (ToS) e o identificador da interface de entrada.

Inicialmente, quando a Universidade de Coimbra (UC) dispunha de uma ligação a 2 Mbps à RCTS, o CIUC utilizou o Cisco NetFlow FlowCollector[2] para recolha de informação sobre fluxos, que inclui o Cisco NetFlow Analyzer para análise da informação recebida do *router*. Rapidamente se verificou que esta solução não se adaptava às necessidades do CIUC. Uma das principais limitações residia no armazenamento da informação em ficheiros ASCII, com deficiente gestão do espaço em disco. Por outro lado, o desempenho e pouca flexibilidade do Analyzer determinaram o abandono desta solução.

Decidiu-se, então, criar uma solução alternativa, suportada por uma base de dados relacional para armazenamento da informação recebida dos *routers*. As principais dúvidas relativas a esta alternativa residiam na recepção dos fluxos que são exportados pelo *router* utilizando UDP, no volume de informação gerado e no desempenho do sistema de gestão de base de dados.

Encorajados pela experiência do Argonne National Laboratory[3], decidiu-se avançar para esta alternativa, que tem por base a junção de três tecnologias: Netflow, base de dados relacional e WWW para acesso à informação pretendida.

## 3. COLANA – Sistema de recolha e análise de tráfego

A Figura 1 representa o ponto de acesso da Universidade de Coimbra à RCTS que, actualmente, consiste num *router* Cisco 3640 com três portas: uma porta FastEthernet que liga ao *backbone* da UC, uma porta ATM NM-1A-OC3MM para a ligação à RCTS, e uma porta Ethernet à qual se encontra ligado o equipamento de análise de tráfego.

O facto de a universidade ter um único ponto de acesso ao exterior constitui um aspecto que simplifica o sistema, na medida em que se evita agregação adicional de informação.

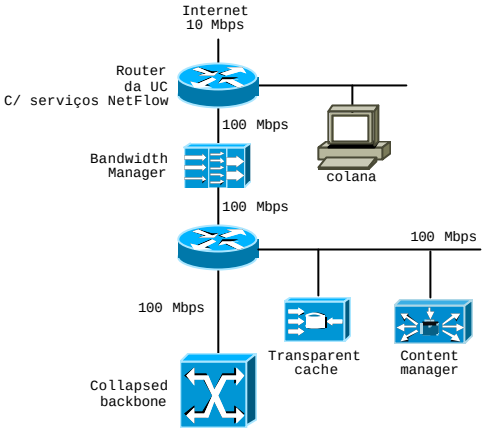


**Figura 1** – Porta de acesso da Universidade de Coimbra à RCTS

O desenvolvimento foi iniciado num pequeno sistema onde foi testado o funcionamento de um programa de recepção de fluxos e a respectiva inserção na base de dados. Os resultados obtidos permitiram concluir que um equipamento bem dimensionado e um desenho adequado da base de dados poderia satisfazer os objectivos propostos.

### 3.1 Desenho da base de dados

A base de dados, construída sobre MySQL [4], tem um desenho bastante simples que se representa na Figura 2, onde se destacam, fundamentalmente, duas tabelas: TRAFEGO_DIARIO e TRAFEGO_MENSAL. As restantes – REDES, APLICACOES e ORGANIZACOES – são tabelas auxiliares.

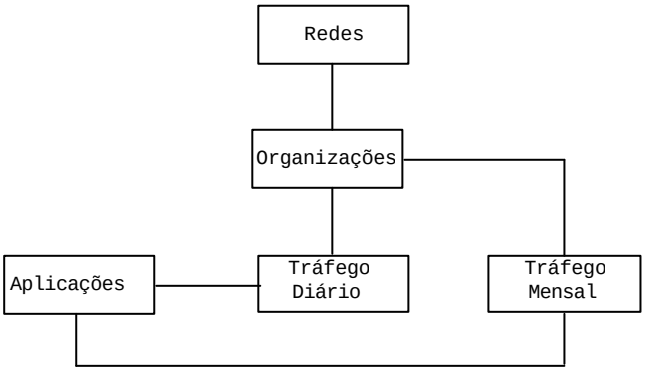


**Figura 2** – Modelo E-R da base de dados de suporte do COLANA

Na tabela TRAFEGO_DIARIO, representada na Figura 3, são inseridos os fluxos enviados pelo *router* durante o dia. A estrutura desta tabela reflecte uma das principais diferenças de implementação relativamente a [3].

```
CREATE TABLE (
        id_trafego       int                   NOT NULL auto_increment unique,
        id_org           int                   NOT NULL,
        id_aplicacao     int                   NOT NULL,
        data             timestamp(8)          NOT NULL,
        hora             time                  NOT NULL,
        io               char(1)               NOT NULL,
        knownaddr        BIGINT                NOT NULL,
        otheraddr        BIGINT                NOT NULL,
        knownport        SMALLINT UNSIGNED     NOT NULL,
        otherport        SMALLINT UNSIGNED     NOT NULL,
        protocol         TINYINT UNSIGNED      NOT NULL,
        pkts             SMALLINT UNSIGNED     NOT NULL,
        octets           BIGINT UNSIGNED       NOT NULL,
        flows            SMALLINT UNSIGNED     NOT NULL,
        starttime        TIMESTAMP             NOT NULL,
        endtime          TIMESTAMP             NOT NULL,
        input_interface  TINYINT               NOT NULL,
        output_interface TINYINT               NOT NULL,
        PRIMARY KEY (id_trafego)
);
```

**Figura 3** – Tabela TRAFEGO_DIARIO

Num pacote NetFlow, representado na Figura 4, os campos *Source Address*, *Source Port*, *Destination Address* e *Destination Port* contêm os endereços/portos de origem e destino de uma ligação.

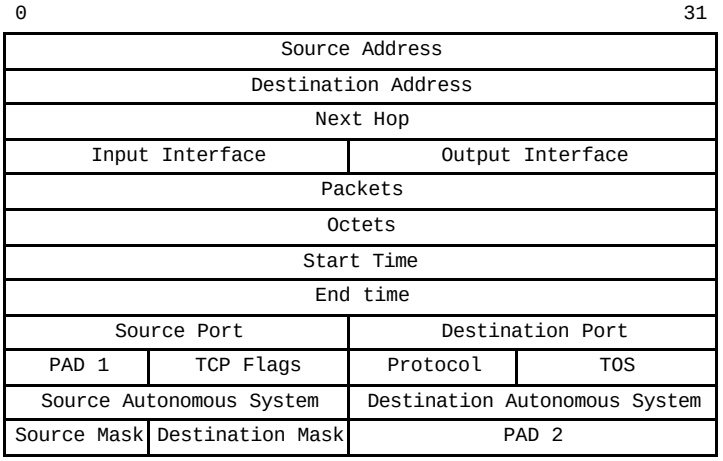


**Figura 4** – Formato dos pacotes NetFlow versão 5

O programa de recepção/inserção de fluxos verifica se as redes correspondentes a estes endereços existem na tabela REDES. Quando um deles existe, é armazenado no atributo KNOWNADDRESS, que o identifica como um endereço interno à UC. O outro é armazenado no atributo OTHERADDRESS. Em casos excepcionais, em que ambos os campos contêm endereços existentes na tabela REDES (ou ambos não existentes), o endereço de origem é mapeado no KNOWNADDRESS e o endereço de destino no OTHERADDRESS, sendo o fluxo classificado como pertencendo a uma organização inválida. Este mapeamento faz com que se perca a informação sobre o fluxo (entrada ou saída). Assim, foi adicionado à tabela o atributo IO que, analisando os campos NetFlow de interface, permite classificar o fluxo como de entrada ou de saída. Desta forma, torna-se viável indexar os atributos KNOWNADDRESS e IO.

Esta abordagem permitiu simplificar significativamente os *queries*. Por exemplo, com um mapeamento directo, determinar todas as ligações efectuadas para um IP interno seria uma tarefa bastante complexa, dado que o endereço IP interno, poderia ocorrer quer no campo *Source Address*, quer no campo *Destination Address*.

Com a alternativa de mapeamento descrita, este mesmo *query* é, apenas, o que se apresenta na Figura 5.

```
SELECT otheraddress, io, sum(octects) AS consumo
   FROM trafego_diario
   WHERE knownaddress = <ip interno>
   GROUP BY otheraddress, io
   ORDER BY consumo;
```

**Figura 5** – *Query* para determinar todas as ligações efectuadas para um IP interno

Todas as análises de detalhe (ao nível de ligação) são feitas sobre esta tabela. Como se pode imaginar, esta tabela tem um ritmo de crescimento bastante elevado, da ordem dos 6 a 11 milhões de fluxos por dia. Um dia com 10 milhões de fluxos dá origem a uma tabela de TRAFEGO_DIARIO com cerca de 1,2GB. Diariamente, é criada uma tabela idêntica com o tráfego do dia. O número de tabelas de tráfego diário mantidas depende do espaço em disco disponível.

Após a inserção de fluxos na tabela de TRAFEGO_DIARIO, é imediatamente agregada esta informação na tabela TRAFEGO_MENSAL. A agregação de dados nesta tabela permite obter informação frequentemente solicitada de forma efectiva, minimizando o tempo de processamento. Os atributos da tabela TRAFEGO_MENSAL são apresentados na Figura 6.

Como o nome sugere, a tabela APLICACAO, permite criar um descritivo, em função do protocolo e do porto utilizados, para associar um flow com uma aplicação.

A tabela REDES associa uma gama de endereçamento a uma organização da Universidade.

Finalmente, a tabela ORGANIZACAO, inclui as organizações da UC.

```
CREATE TABLE trafego_mensal (
        id_trafego_mensal       int         NOT NULL auto_increment unique,
        id_org          int                 NOT NULL,
        id_aplicacao    int                 NOT NULL,
        data            timestamp(8)        NOT NULL,
        io              char(1)             NOT NULL,
        bytes           BIGINT UNSIGNED     NOT NULL,
        PRIMARY KEY (id_trafego_mensal)
        );
```

**Figura 6** – Tabela TRAFEGO_MENSAL

## 3.2 Acesso à informação

A simplicidade de acesso à informação é determinante na utilidade de uma aplicação deste tipo. Por exemplo, obter resposta sobre o tráfego gerado por uma organização num dado dia, sobre as aplicações mais utilizadas no mesmo departamento ou mesmo saber o endereço IP de todas as máquinas externas que contactaram um servidor interno, devem constituir operações simples que não exijam escrever um comando SQL complexo.

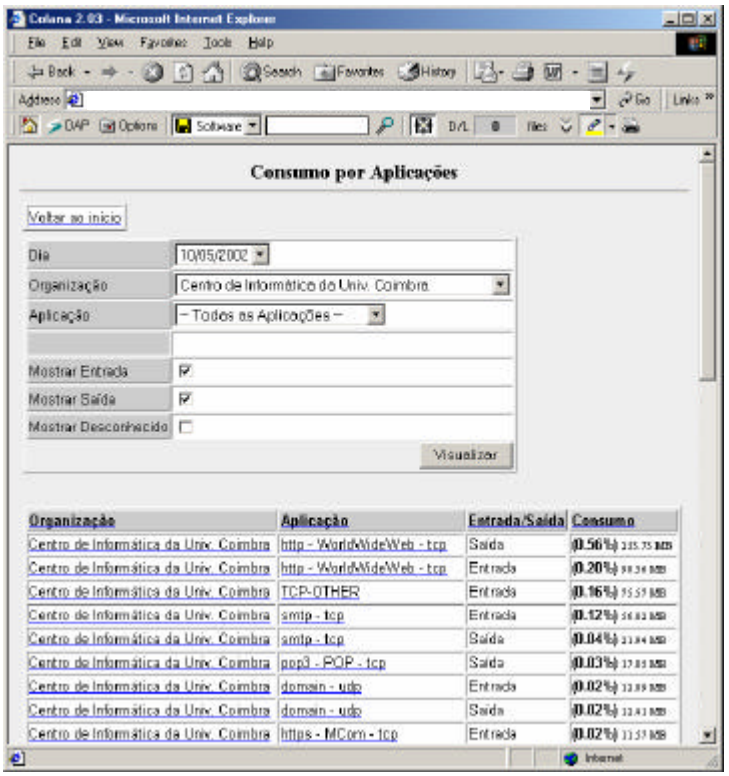


**Figura 7** – Consumo por aplicações de um departamento

Com este objectivo foi criado uma interface *Web* que reúne as interrogações que mais frequentemente se fazem sobre a base de dados na forma de relatórios ou de gráficos. O acesso a esta interface deverá estar restrito ao gestor de rede, dado o impacto que alguns *queries* colocam sobre o sistema.

A Figura 7 inclui um ecrã com uma página que exemplifica a análise da utilização de aplicações TCP/IP num departamento ao longo de um dia.

O acesso à informação via SQL está naturalmente contemplado através do interface de linha de comando do MySQL.

Para além dos aspectos de contabilização de consumo de tráfego, a aplicação pode, igualmente, ser utilizada para análise de possíveis tentativas de intrusão. Assim, um IP externo com acessos a um elevado número de IPs internos, num reduzido intervalo de tempo (um dia, por exemplo), pode indiciar uma tentativa de intrusão. Uma análise mais detalhada das ligações efectuadas por aquele IP externo pode confirmar a suspeita.

O exemplo ilustrado na Figura 8 inclui um *query* que mostra o número de endereços internos (`t_knownaddr`) acedidos a partir de um endereço externo (`t_otheraddr`), o tráfego gerado (`consumo`) e o número de aplicações usadas (`t_id_aplicacao`). Este *query* ilustra, de facto, possíveis tentativas de intrusão, pois são estabelecidas inúmeras ligações para endereços internos.

```
mysql> select t.data as t_data,
          count(distinct t.id_aplicacao) as t_id_aplicacao,
          count(distinct t.knownaddr) as t_knownaddr,
          sum(t.octets) as consumo,
          t.otheraddr as t_otheraddr
       from C20020509 t
       group by t.otheraddr
       having t_knownaddr > 20 and t_id_aplicacao < 10
       order by t_knownaddr Desc
       limit 10;

+----------+----------------+-------------+----------+--------------+
| t_data   | t_id_aplicacao | t_knownaddr | consumo  | t_otheraddr  |
+----------+----------------+-------------+----------+--------------+
| 20020509 |              6 |        9397 |  1208451 | 212030090008 |
| 20020509 |              9 |        8634 |  1071968 | 210178225001 |
| 20020509 |              3 |        8460 |  1956991 | 217225091077 |
| 20020509 |              7 |        4845 | 19577099 | 195251209006 |
| 20020509 |              5 |        4826 |  1241907 |  80013045153 |
| 20020509 |              8 |        4576 |   585542 | 217097087035 |
| 20020509 |              5 |        4508 |   335897 | 213022104052 |
| 20020509 |              6 |        4426 |   522104 | 203165131215 |
| 20020509 |              7 |        4318 |   810770 | 217083246079 |
| 20020509 |              3 |        4053 |   636991 | 217083251103 |
+----------+----------------+-------------+----------+--------------+
10 rows in set (4 min 28.68 sec)
```

**Figura 8** – *Query* e dados que ilustram possíveis tentativas de intrusão

A listagem permite concluir que o endereço IP 212.30.90.8 tentou contactar 9397 IPs internos, gerando um tráfego de cerca de 1MB, utilizando 6 aplicações distintas.

Ainda no capítulo da segurança, é importante registar que as tabelas de TRAFEGO_DIARIO (detalhe dos fluxos) que são removidas por esgotamento de espaço, são armazenadas em banda magnética, sendo possível a sua posterior recuperação para análise em caso de necessidade.

## 3.3 Dimensionamento e desempenho

A base de dados foi implementada em MySQL sobre um sistema com dois processadores 1GHz, 1GB de RAM e três discos de 30GB, explorado sob Linux 7.2. Para além do MySQL, foi ainda instalado neste sistema um servidor Apache com suporte PHP4. O sistema encontra-se em utilização deste Outubro de 2001.

### Controlo de fluxos recebidos

Uma das questões que foi inicialmente colocada diz respeito à recepção de fluxos do *router*, sendo necessário avaliar se o *script* de recepção de *flows* não perde informação. Este controlo é feito através de um número sequencial que o *script* de recepção adiciona à informação de cada fluxo recebido e que é comparado com a saída do comando `sh ip flow export` executado no *router*. Deste modo, consegue-se ter, com uma aproximação razoável, uma ideia sobre se, num dado momento, está ou não a ocorrer perda de pacotes. Os testes não indiciaram perdas.

### Dimensão das tabelas

A tabela de TRAFEGO_DIARIO é a que, naturalmente, está sujeita a um maior crescimento, dado que armazena todos os fluxos recebidos do *router*. O número de fluxos recebidos depende não só do tráfego na linha, mas também do tipo de tráfego. De facto, o número de fluxos produzido por uma sessão *telnet* é, normalmente, bastante inferior ao número de fluxos que resulta de um acesso a uma página WWW com muitos objectos. A análise dos dados disponíveis permite concluir que, com a actual (elevada) taxa de utilização da linha de 10 Mbps da UC, a tabela de TRAFEGO_DIARIO armazena entre 6 e 11 milhões de registos, a que corresponde uma espaço em disco entre os 600MB e 1,4GB.

A tabela TRAFEGO_MENSAL apresenta um crescimento mensal de, aproximadamente, 2MB.

### Gestão do espaço em disco

A gestão do espaço em disco é feita periodicamente (a cada 3 minutos), posteriormente à inserção na tabela TRAFEGO_DIARIO de um novo conjunto de fluxos e respectiva actualização da tabela TRAFEGO_MENSAL. O número de dias a manter em disco é lido de um ficheiro de configuração e, se o espaço disponível em disco assim o exige, a tabela de TRAFEGO_DIARIO mais antiga é apagada. Este programa, pode aumentar ou diminuir o número de dias do ficheiro de configuração de modo a optimizar o espaço em disco e a garantir a operacionalidade do sistema.

A tabela de TRAFEGO_DIARIO de um determinado dia é copiada para banda magnética à 01:00 do dia seguinte pelo que, quando houver necessidade de ser apagada, já se encontra segura.

### Utilização de CPU

Um dado igualmente significativo é a utilização do processador, pelo que se incluem alguns indicadores recolhidos durante intervalos de 10 minutos com o *script* de recepção de fluxos a recolher informação do *router* e a introduzir os respectivos registos na base de dados.

Nesta situação, e sem qualquer *query* do utilizador em execução, registou-se uma disponibilidade superior a 85% dos dois processadores:

```
Media consumo CPU0:  8.8 user; 1.5 system; 0.3 nice; 88.2 idle
Media consumo CPU1: 10.7 user; 1.8 system; 0.1 nice; 85.9 idle
```

Diversos *queries* mais usuais, executados através da interface *Web*, durante um intervalo de 7 minutos, introduzem um esforço pouco significativo aos processadores:

```
Media consumo CPU0: 23.3 user; 3.2 system; 0.0 nice; 71.7 idle
Media consumo CPU1: 15.7 user; 2.9 system; 0.1 nice; 80.1 idle
```

Com um *query* pesado em execução (listar todas as ligações efectuadas num dia por todas as máquinas, agrupadas por endereço de destino e protocolo de aplicação) durante um intervalo de 30 minutos, verificou-se, ainda, uma elevada disponibilidade dos dois processadores:

```
Media consumo CPU0: 35.7 user; 8.9 system; 0.1 nice; 53.7 idle
Media consumo CPU1: 34.4 user; 9.0 system; 0.1 nice; 54.9 idle
```

Deste modo, podemos concluir que o sistema se encontra adequadamente dimensionado para o actual ambiente da UC.

### Tempos de *queries*

Finalmente, são apresentados alguns tempos de execução de *queries* dirigidos sobre a tabela TRAFEGO_DIARIO que, como referido, tem uma cardinalidade da ordem dos 11 milhões de registos:

```
Os 10 IPs internos que mais consumiram no dia 2002/05/09

mysql> select knownaddr, sum(octets) consumo from C20020509 group by
knownaddr order by consumo desc limit 10;
10 rows in set (36.10 sec)

Qual o número de IPs externos contactados por um servidor de mail?

mysql> select count(distinct(otheraddr)) from C20020509 where
knownaddr=192168000001;
1 row in set (1 min 3.89 sec)
```

# 4. Conclusão

A necessidade de contabilizar a utilização de um recurso dispendioso como a ligação a um fornecedor de serviços Internet é um problema comum a diversas organizações. Esta necessidade é extensiva à análise dessa mesma utilização numa perspectiva de segurança.

A utilização de tecnologia de bases de dados relacionais permitiu encontrar uma resposta satisfatória para o problema, com tempo de desenvolvimento reduzido. Todos os produtos utilizados são distribuídos em *open license* e o seu actual estado de desenvolvimento não colocou qualquer problema de estabilidade ao sistema.

Uma limitação da actual solução resulta do compromisso de separar o tráfego detalhado em tabelas diárias, o que dificulta a análise para intervalos superior a um dia. Trata-se, contudo, de um compromisso de desempenho face à dimensão das tabelas. Por outro lado, a interface *Web* desenvolvida encontra-se, ainda, numa forma um pouco primitiva.

Esta situação constitui um incentivo para desenvolvimentos futuros orientados para a responder a ambientes com maior largura de banda e/ou com conectividade através de mais do que um ponto de acesso. A possibilidade de análise dos dados recolhidos com recurso a ferramentas mais sofisticadas (OLAP, *Data Mining*) poderá, igualmente, constituir um outro aspecto de desenvolvimento futuro. Finalmente, mas não menos importante, será desenvolvido trabalho no sentido de preparar o sistema com vista a permitir a sua simples distribuição.

# Referências

[1] http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm

[2] http://www.cisco.com/warp/public/732/Tech/netflow/

[3] Navarro, John-Paul, et. al, "Combining Cisco NetFlow Exports with Relational Database Techonlogy for Usage Statistics, Intrusion Detection and Network Forensics", 14th System Administration Conference (LISA 2000), New Orleans, Louisiana, 2000.

[4] http://www.mysql.com