Architettura di passive-monitoring implementata sulla rete GARR

Autori: Christian Cinetto, GARR *(christian.cinetto@garr.it)*
Michele Sciuto, GARR *(michele.sciuto@garr.it)*

Abstract:
In questo documento presentiamo lo studio svolto nel 2002/2003 presso il NOC di GARR sul traffico in transito sulla rete GARR-B.
L'analisi è stata implementata seguendo un approccio di passive monitoring, sfruttando lo strumento CISCO NetFlow, con acquisizione di flussi dai router di backbone. L'obiettivo è stato quello di ottimizzare le risorse ed il planning di rete (traffic-engineering) e di fornire al GARR-NOC uno strumento più efficace dei precedenti nel risolvere anomalie nel traffico e incidenti di sicurezza.
I risultati comprendono un'interfaccia grafica, consultabile via web, che riporta la percentuale di banda utilizzata da ogni singola applicazione o protocollo su ogni singolo circuito. Inoltre si è lavorato alla produzione real-time di report statistici che evidenziano da un lato lo sfruttamento della rete con attenzione all'utilizzo di banda da parte dei singoli host (identificando anche le applicazioni coinvolte), dall'altro indicano la presenza di anomalie nel traffico come attività di DoS (Denial of Service) e port-scanning.
Il lavoro di ricerca ha portato a risultati convincenti, tali da essere divulgati a tutta la comunità scientifica; nella parte di appendice abbiamo cercato di dare una serie di indicazioni per l'implementazione delle tecniche di monitoring che vengono presentate.
Il documento ha quindi l'ambizione di dare supporto sia a tutto il NOC della rete GARR sia agli APM, che possono implementare l'architettura sulle proprie Local Area Network.

Data creazione: giugno 2003
Data ultima modifica: giugno 2003

Livello di distribuzione: pubblico
Nome file: GARR-03-001.pdf

Indice

1 Monitoring e NetFlow

L'analisi che presentiamo è basata su un approccio di passive-monitoring (Appendice A), basata sullo strumento CISCO NetFlow. Un'architettura di passive-monitoring è fondata sull'utilizzo del traffico di produzione per affrontare in modalità quasi real-time incidenti di sicurezza, anomalie del traffico, billing, assicurazione dello SLA, traffic-engineering. Il passive-monitoring non implica quindi l'introduzione di traffico nella rete, cosa che invece viene effettuata nel caso di active monitoring, ma è generalmente basato sull'utilizzo di hardware-device, per monitorare il traffico in transito sulla rete; le informazioni sono collezionate per essere analizzate a posteriori.
Tre sono le modalitá principali con cui si effettuano analisi di passive monitoring, ovvero polling attuato secondo il protocollo snmp, port mirroring in cui il traffico totale viene replicato dal router su un'interfaccia dedicata (questa tecnica può essere implementata sia a livello software, nel caso di router Juniper, sia a livello hardware con ad esempio degli splitter ottici negli altri casi), infine NetFlow.

NetFlow (Appendice B) è una feature di CISCO IOS, viene abilitato sulle interfacce del router e permette l'esportazione dei flussi, entranti sulle interfacce, ad un calcolatore. In questo documento indichiamo con il termine flusso la coppia IpSorgente-IpDestinazione-PortaSorgente-PortaDestinazione. Alla base dello strumento risiede la *flow-cache*, una memoria dedicata, che viene allocata dal router all'atto dell'abilitazione di NetFlow; il funzionamento di NetFlow prevede la mappatura dei flussi all'interno della cache da cui gli stessi flussi possono essere esportati (Appendice B) verso una macchina dedicata alla collezione.

I flussi vengono esportati in datagram UDP in formati differenti a seconda della versione di NetFlow utilizzata. I datagram vengono detti PDU (Protocol Data Unit) e consistono di un header e di uno o più flow-record che vengono presentati nella figura 1.

La versione che abbiamo utilizzato è la 5, adatta ai router presenti sulla rete GARR e riportante, oltre ai campi fondamentali già presenti nelle versioni precedenti, anche informazioni sugli Autonomous System (origin o peer AS) ed un flow sequence number che permette un maggior debugging .

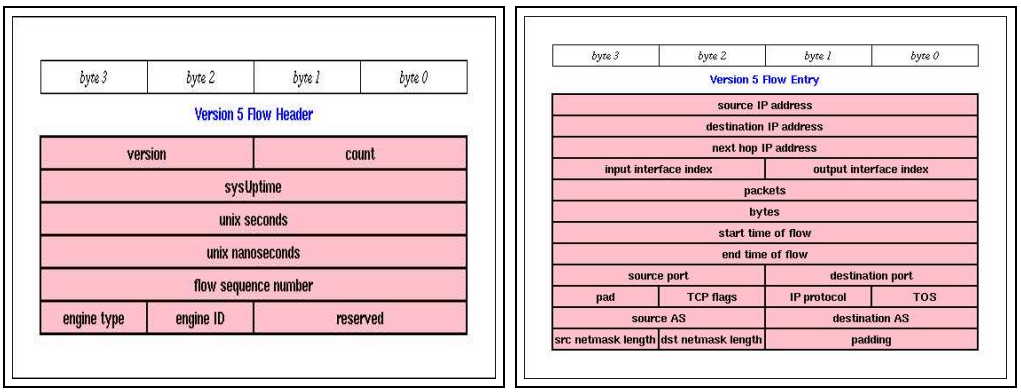


Figura 1: Header e flow entry del pacchetto NetFlow Version 5

2 Architettura

L'architettura del sistema di monitoring che abbiamo realizzato comprende le seguenti parti: esportazione, collezione, analisi e visulizzazione. In figura 2 è presentato il layout del sistema.

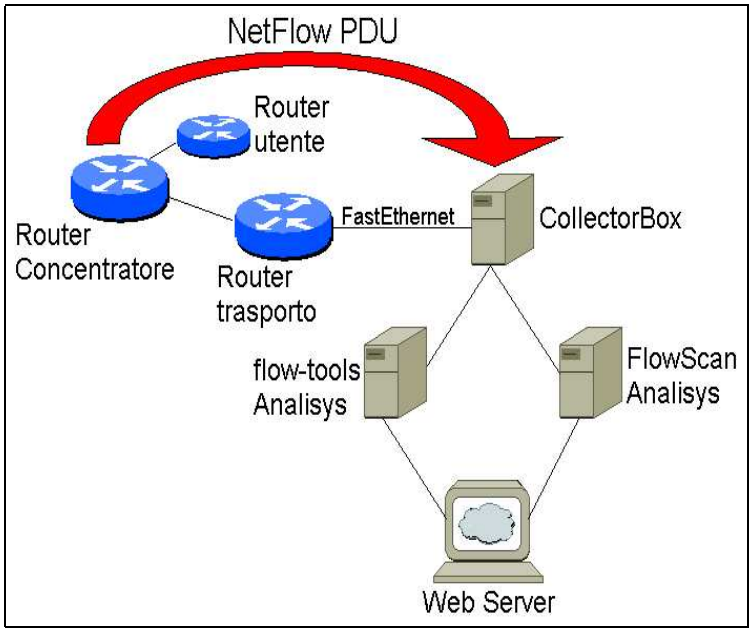


Figura 2: Architettura del sistema di monitoring

L'esportazione dei flussi viene svolta implementando NetFlow sul router concentratore, dove afferiscono i circuiti di cui si vuole analizzare il traffico. Quindi tutto il traffico in transito sul router concentratore viene esportato su una macchina dedicata alla collezione (collector-box). Per ottimizzare il meccanismo di export, sopratutto ricordando che i PDU NetFlow vengono trasmessi in UDP, è corretto che il collector-box venga posizionato in prossimità del router di trasporto, cui è collegato in FastEthernet o GigaEthernet. I flussi NetFlow hanno come sorgente l'interfaccia di Loopback del router concentratore e come destinazione la macchina dedicata alla collezione dei dati. La scelta dell'interfaccia di Loopback è conveniente per questioni di stabilità (il router ci manderà dati fino alla morte). La natura "input-based" di NetFlow presuppone che su tutte le interfacce del router venga abilitata l'esportazione, questo per permettere che venga collezionata la totalità del traffico in transito su di esso. L'aspetto della selezione dei flussi di traffico riferiti ai singoli router utente verrà affrontata più avanti nella fase di analisi.
Le due macchine di analisi sono state anch'esse posizionate nei PoP GARR con l'intento di velocizzare l'accesso ai dati presenti sul collector-box. La macchina denominata "flow-tools Analisys" è stata utilizzata per produrre dei report html, consultabili con un browser, che consentono diverse tipologie di controllo sul traffico. "Flowscan Analisys" è stata impiegata per la produzione degli andamenti del traffico in formato grafico. Sebbene i file html risultanti dalle diverse analisi siano presenti localmente sui calcolatori dedicati all'analisi, risulta utile impiegare una macchina con la funzione di web-server per la visualizzazione dei risultati.

3 Collezione dati

Il collector-box ha lo scopo di collezionare i flussi Netflow, incapsulati in pacchetti udp, esportati dal/i router e deve avere caratteristiche tali da minimizzare la perdita di tali dati. Il calcolatore messo a nostra disposizione, indipendentemente dalle necessità reali, è stato un Compaq Proliant dual proc pentium III 800 MHz, $1GB$ di RAM, $100GB$ di spazio disco e OS Linux Red-hat 7.1, kernel 2.4.18. Per la fase di collezione e archiviazione dei Netflow PDU (Protocol Data Unit) i software a disposizione sia open source che commerciali sono moltissimi e per un elenco aggiornato ed esaustivo rimandiamo a [3] e [14] . Per quanto ci riguarda la prima scelta è stata quella di utilizzare software open source perche' ci permette di avere un oggetto modificabile per esigenze ad hoc, consente la condivisione di problematiche attraverso mailing-lists e da' la possibilità di avere un prodotto facilmente aggiornabile. Il nostro studio dell'argomento ci ha portato dapprima a focalizzarci su Cflowd, software sviluppato dalla Caida [6] per poi passare a flow-tools [2]. flow-tools è un insieme di tools (scritti in linguaggio C), realizzati alla Ohio State University (OSU) a partire dal 1996 . Permette di filtrare tutti i campi del pacchetto Netflow direttamente sui raw-file in modo da razionalizzare l'analisi ed eliminare i dati superflui. È compatibile sia con apparati Juniper(che al posto di NetFlow utilizza una feature chiamata Cflowd, ma con caratteristiche identiche a NetFlow versione 5) che Cisco, è di facile configurazione e debugging, integra una serie di tools che gli permettono, tra l'altro, di fare il reply dei dati ad altri collectors sia in modo nativo che in seguito a filtraggi. Il collector-box riceve udp packets inviatigli direttamente dal router o dallo switch, ad una porta (configurata sul router) sulla quale il collector-tool sta in ascolto. flow-tool converte i raw-file Netflow in una rappresentazione che gli permette di avere un summary del flusso ricevuto dall'apparato remoto (flow record, 60 bytes). La maggior parte degli altri tools lavora su questo formato, sempre di tipo raw. Sul collector box da noi utilizzato è stata installato flow-tools versione 0.64(versione ancora beta).

L'analisi che abbiamo implementato necessita di più istanze dello stesso flusso che possono essere processate dai tool utilizzati. Questo serve a minimizzare le risorse computazionali in fase di analisi: l'approccio consiste nel processare solo i flussi che interessano, ovvero quelli che riguardano una singola utenza od un singolo link. A livello architetturale, per ovviare alla limitazione dei router CISCO 7500 di esportare flussi esclusivamente verso un unico collector, si è pensato di replicare più volte i pacchetti udp in arrivo dal router. Grazie a flow-fanout (una delle tante componenti di flow-tools) ogni pacchetto udp, contenente tutto il traffico del router concentratore, è stato replicato verso le due macchine di analisi, per un numero di volte pari ai processi di analisi necessari. I dati giunti alle macchine di analisi vengono processati e cancellati, mentre la copia originale dei flussi resta archiviata nel collector-box fino al raggiungimento di un parametro che definisce la massima occupazione di spazio disco permessa al processo.

4 Analisi

I dati a nostra disposizione, dopo la fase di collection, si prestano ad essere utilizzati per scopi differenti e molteplici, ma che possiamo suddividere in due categorie riassuntive:

- Network planning
- Suspicious activities

Con la prima intendiamo la possibilità di gestire e pianificare la rete in maniera più consapevole. Ciò è possibile con la conoscenza, ad esempio, delle matrici di traffico che coinvolgono i peer AS, o altri AS raggiunti tramite upstream provider.

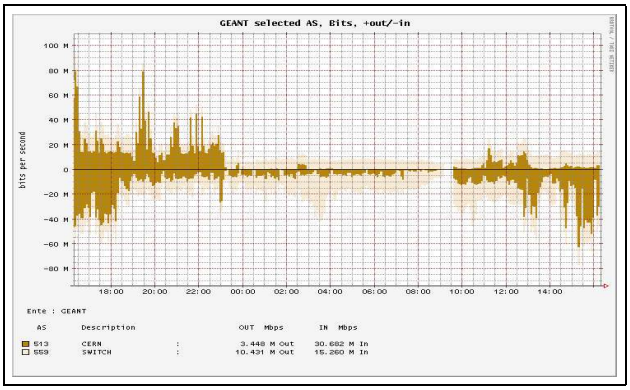


Figura 3: Matrice di traffico GARR–Cern, GARR-Switch attraverso GEANT

Un altro obiettivo importante di questo lavoro, riconducibile più alla gestione che al planning, è stato l'analisi del traffico per tipo di servizio (http, ftp, mail, dns, news, p2p, etc.) che permette di comprendere il rapporto sfruttamento di banda / applicazioni.

Le attività critiche sulla rete, o peggio, quelle patologiche rientrano nelle cosiddette *Suspicious Acivities* . Abbiamo dimostrato come le attività di port-scanning, DoS (Denial of Service), Flash Crowd e di occupazione anomala di banda possano essere registrate e rintracciate anche in quasi real-time. Gli strumenti software utilizzati nell'analizzare i dati sono stati due distinti, ma aventi entrambi in comune i file creati da flow-tools. Di seguito li presentiamo brevemente.

4.1 flow-report and show-reports

Questa sezione presenta i risultati ottenuti implementando i tool di analisi di flow-tools in cascata con script fatti da noi.

La figura precedente illustra l'implementazione del sistema. Sull'elaboratore nominato "analyzer-box" sono presenti i raw-file contenenti i dati Netflow, immagazzinati da flow-tools . È stata creata una maschera di configurazione del software per cui ogni ente è configurato in modo identico se non per le reti che lo contraddistinguono (laddove le reti non aggregabili sono più di tre si utilizza l'snmp interface index per motivi di efficienza). È stato creato un tool(daily-report) che a cron effettua reports differenti, a seconda delle esigenze, per ogni ente (flow-nfilter e flow-report). Una volta ottenuti i risultati, essi

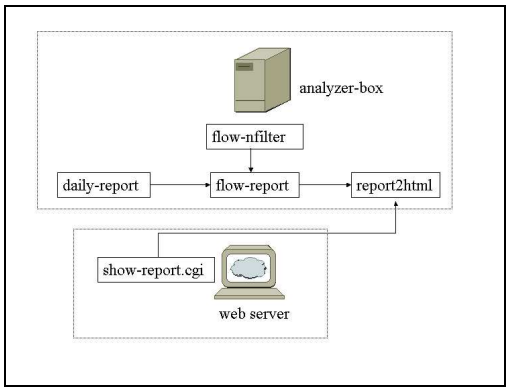


Figura 4: Architettura di visualizzazione dati

vengono convertiti in tabelle html (report2html) e archiviati in directory con un pattern utente/anno/anno-mese/anno-mese-giorno/data dove data è una delle due directory di destinazione corrispondenti l'una ai reports di 10 minuti(current) e l'altra a quelli sulle 12 ore. È stato creato dunque un cgi (show-report.cgi, in linguaggio perl) che permette all'utente di visualizzare i dati interrogando l'web server centrale.

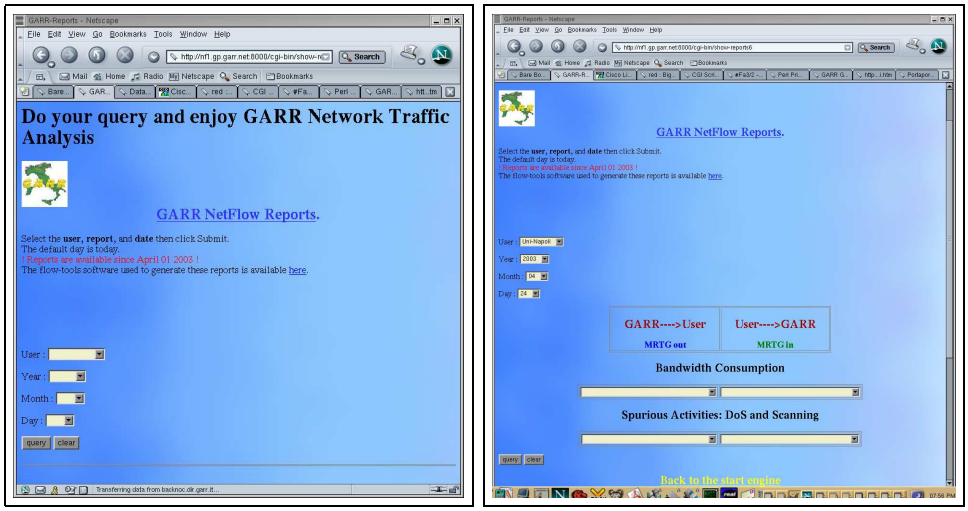
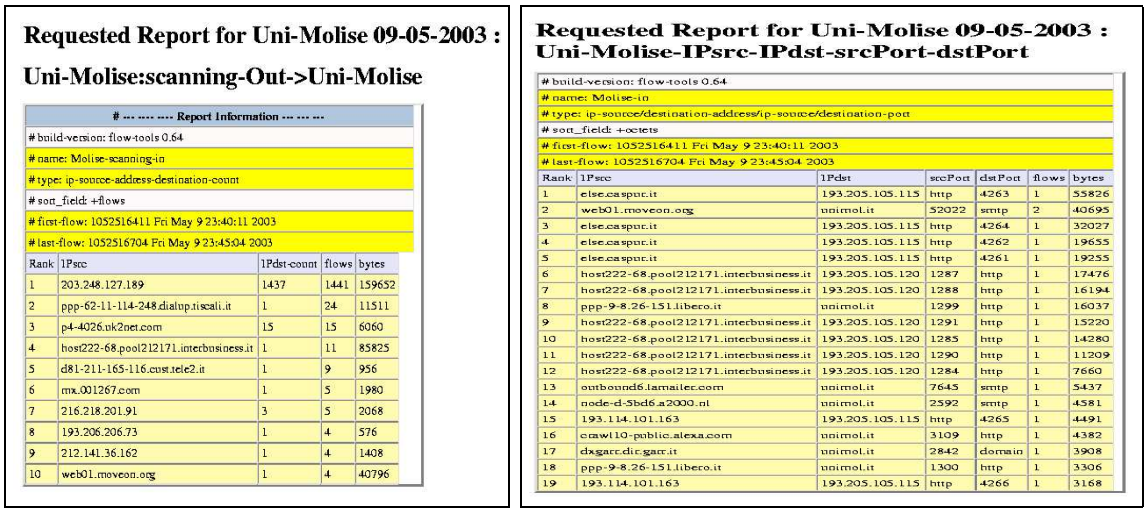


Figura 5: Home page di show-reports

I report disponibili sono stati decisi di granularità temporale pari a 10 minuti, e sono suddivisi per ogni ente per ogni direzione di traffico (GARR → User, User → GARR). Si è scelto di suddividere ulteriormente la struttura dei report secondo l'occupazione di banda e le attività di port-scanning e DoS. Il tool permette di avere informazioni su IPsorgente-IPdestinazione-PortaSorgente-PortaDestinazione, top sorgenti di traffico, matrice di AS, top servizi nelle 12 ore, presunti sorgenti e bersagli di DoS e di port-scanning.

Check con MRTG Si controlla l'ammontare di traffico per ogni interfaccia di modo da avere un cross-check con MRTG, soprattutto in considerazione della futura applicazione tariffaria basata sul 95^o percentile. Osserviamo che il tool permette di discriminare il traf-



Requested Report for Uni-Molise 09-05-2003 :
Uni-Molise:scanning-Out->Uni-Molise

# --- ---- ---- Report Information --- --- ---				
# build-version: flow-tools 0.64				
# name: Molise-scanning-in				
# type: ip-source-address-destination-count				
# sort_field: +flows				
# first-flow: 1052516411 Fri May 9 23:40:11 2003				
# last-flow: 1052516704 Fri May 9 23:45:04 2003				
Rank	IPsrc	IPdst-count	flows	bytes
1	203.248.127.189	1437	1441	159652
2	ppp-62-11-114-248.dialup.tiscali.it	1	24	11511
3	p4-4026.uk2net.com	15	15	6060
4	host222-68.pool212171.interbusiness.it	1	11	85825
5	d81-211-165-116.cust.tele2.it	1	9	956
6	mx.001267.com	1	5	1980
7	216.218.201.91	3	5	2068
8	193.206.206.73	1	4	576
9	212.141.36.162	1	4	1408
10	web01.moveon.org	1	4	40796

Requested Report for Uni-Molise 09-05-2003 :
Uni-Molise-IPsrc-IPdst-srcPort-dstPort

# build-version: flow-tools 0.64						
# name: Molise-in						
# type: ip-source/destination-address/ip-source/destination-port						
# sort_field: +octets						
# first-flow: 1052516411 Fri May 9 23:40:11 2003						
# last-flow: 1052516704 Fri May 9 23:45:04 2003						
Rank	IPsrc	IPdst	srcPort	dstPort	flows	bytes
1	else.caspur.it	193.205.105.115	http	4263	1	55826
2	web01.moveon.org	unimol.it	52022	smtp	2	40695
3	else.caspur.it	193.205.105.115	http	4264	1	32027
4	else.caspur.it	193.205.105.115	http	4262	1	19655
5	else.caspur.it	193.205.105.115	http	4261	1	19255
6	host222-68.pool212171.interbusiness.it	193.205.105.120	1287	http	1	17476
7	host222-68.pool212171.interbusiness.it	193.205.105.120	1288	http	1	16194
8	ppp-9-8.26-151.libero.it	unimol.it	1299	http	1	16037
9	host222-68.pool212171.interbusiness.it	193.205.105.120	1291	http	1	15220
10	host222-68.pool212171.interbusiness.it	193.205.105.120	1285	http	1	14280
11	host222-68.pool212171.interbusiness.it	193.205.105.120	1290	http	1	11209
12	host222-68.pool212171.interbusiness.it	193.205.105.120	1284	http	1	7660
13	outbound6.iamailer.com	unimol.it	7645	smtp	1	5437
14	node-d-5bd6.a2000.nl	unimol.it	2592	smtp	1	4581
15	193.114.101.163	193.205.105.115	http	4265	1	4491
16	crawl10-public.alexa.com	unimol.it	3109	http	1	4382
17	dxgarc.dic.garr.it	unimol.it	2842	domain	1	3908
18	ppp-9-8.26-151.libero.it	unimol.it	1300	http	1	3306
19	193.114.101.163	193.205.105.115	http	4266	1	3168

Figura 6: Esempio di risultato di show-reports

fico in termini di reti IP e di interfacce, di conseguenza si possono differenziare facilmente utenti che insistono sulla stessa interfaccia del router concentratore (L3).

Network planning Visualizzazione di matrice di traffico per AS in cui vengono evidenziati i flussi e l'ammontare della banda verso i link internazionali e di commodity nazionale.

Port scanning Se un indirizzo IP origina molti più flussi di quanti ne riceve, significa che probabilmente sta effettuando un Port-scanning. Si possono implementare dei filtri che selezionino il numero di flussi per singola coppia IPsrc-Ipdst.

Link saturi È immediato vedere, come nel caso dell'esempio dell'Università del Molise, quali sono gli indirizzi IP che generano maggior traffico e che quindi risultano essere i maggiori responsabili in termini di saturazione di banda 6. Viceversa si possono individuare gli IP che esportano maggior traffico, che possono risultare sorgenti di traffico illecito (scambio di file protetti da copyright).

Flash crowd Circostanze particolari, quali ad esempio il rilascio della nuova release di Red-Hat o un importante iniziativa da parte di un'ente, possono dar luogo all'impennarsi del numero dei flussi verso un particolare server web. Anche questa circostanza è facilmente evidenziabile con flow-tools.

Denial of Service Uno degli abusi più dannosi per la rete è sicuramente il Denial of Service. La sua forma più comune è il flooding di pacchetti verso un IP o una rete. MRTG non lo può evidenziare in quanto generalmente i pacchetti hanno dimensione molto piccole e l'ammontare di banda totale non satura necessariamente il link. Con flow-tools si può discriminare tale comportamento in 2 modi: il primo è triggerando il numero di flussi per IP, quando tale livello d'allarme è superato si provvede ad un meccanismo di segnalazione

automatica dell'incidente. Il secondo metodo è sui flussi tcp che hanno un numero anomalo di pacchetti con syn (questo metodo è comunque limitato da NetFlow). Esiste una terza via ed è rappresentata da MRTG: se si monitorizza il link (ad es. FastEthernet) su cui transita il traffico NetFlow, ci si accorge dell'anomalia molto presto, dato che il numero medio di flussi al secondo può crescere (nel caso di grandi aggregati) di 4-5 volte tanto; molto di più nel caso di aggregati più piccoli.

Dalla figura 7, relativa all'interfaccia dell'Università di Pavia, non si evince se la saturazione di banda corrisponda ad un traffico non patologico quale un DoS. Visualizziamo, sempre in figura 7, il grafico relativo alla FastEthernet su cui è attestato il collector-box.

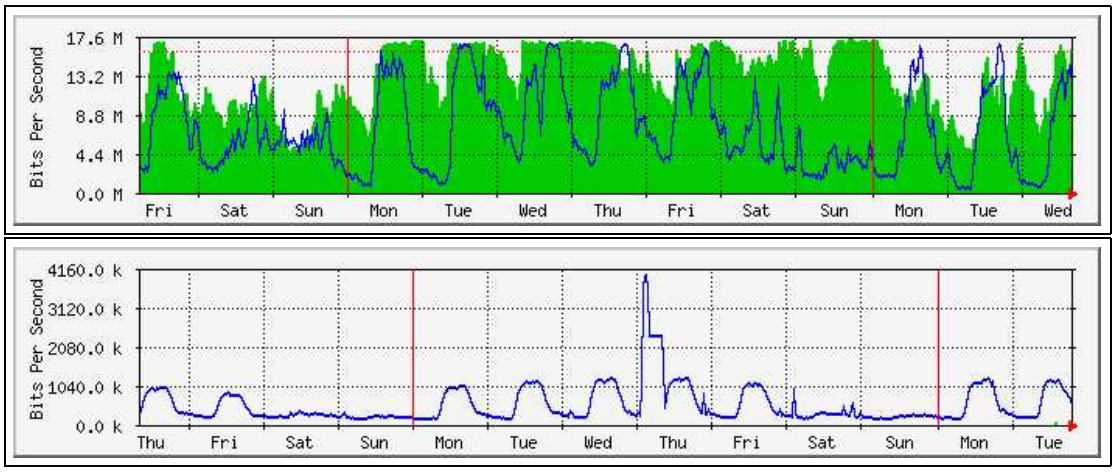


Figura 7: La prima figura si riferisce al traffico di accesso dell'utente, la seconda all'andamento del traffico NetFlow tra il router e il collector-box; è chiaro il riscontro di un'anomalia nell'acquisizione dei flussi NetFlow, che non risulta invece nel primo caso

Notiamo molto nitidamente che, benche' il link fosse saturo in giorni diversi, l'unica volta che la causa era imputabile ad un DoS si è registrata nella giornata di giovedì, momento in cui il traffico Netflow è aumentato di 4 volte rispetto ad un profilo molto regolare.

Studio del traffico Questo sistema permette di analizzare le applicazioni principali, quali http, ftp, mail, file-sharing ecc. sia per ente che per singolo IP. Oltre che per impatti sulla sicurezza questa funzione dà la possibilità al progettista di rete di capire i comportamenti degli utenti al fine di una migliore qualità del servizio.

4.2 FlowScan Analisys

Il secondo tipo di analisi è rivolto ad un'indagine real-time, che permetta di conoscere immediamente il pattern di traffico dettagliato (con informazioni sui singoli servizi o protocolli) di un singolo ente. In questa parte andiamo a presentare i risultati ottenuti grazie a FlowScan e cerchiamo inoltre di darne una breve presentazione. Rimandiamo all'appendice D per spiegazioni dettagliate sull'implementazione; sia per quel che riguarda il singolo utente (APM), sia per quel che riguarda le configurazioni che possono riguardare i componenti del NOC. Come vedremo infatti ci sono differenze sostanziali nel caso di monitoring su router di accesso piuttosto che di backbone.

Lo strumento software adatto a processare i dati raccolti da flow-tools è FlowScan [28], sviluppato da Dave Plonka, di University of Winsconsin-Madison. FlowScan è composto da una collezione di moduli e script perl; gestisce i dati collezionati da flow-tools e li riporta ad RRDTool, un database a perdita d'informazione costruito appositamente per archiviare e visualizzare dati. Ogni database comprende contatori per pacchetti, byte e flussi, oltre ad una serie di statistiche, campionate ad intervalli di 5 minuti, basate su uno dei seguenti attributi:

- Protocollo di livello 4 (TCP, UDP, ICMP)
- Servizio o applicazione (ftp, http, smtp, Kazaa...)
- Rete di classe A, B, C o blocco CIDR in cui risiede l'indirizzo IP locale
- AS sorgente e destinazione

All'interno del database sono collezionate anche le informazioni riguardanti il traffico totale, il multicast ed il traffico riferito a reti sconosciute. L'implementazione di FlowScan ovviamente presuppone l'installazione di un server web che consenta di visualizzare i report grafici. L'aspetto dello sviluppo grafico è stato affrontato con CUFlow, un modulo perl sviluppato appositamente per FlowScan.

In figura 8 presentiamo la home page del tool di analisi implementato, riferita ad un particolare ente della rete GARR; da qui è possibile scegliere le diverse opzioni che verranno presentate nel grafico.

- bit/s, flussi/s o pacchetti/s
- l'ampiezza del dominio temporale
- il formato dell'imagine prodotta (png o gif)
- la scelta di uno o più protocolli
- la scelta di uno o più servizi
- il traffico totale

Un esempio dell'andamento del traffico di un ente, in cui viene riportata la percentuale dei singoli servizi a 24 ore, è in figura 8. Un'aspetto importante riguarda la possibilità di visualizzazione del traffico non solo in termini di bits/sec ma anche di flussi/sec o pacchetti/sec. La presenza di spike nell'andamento del traffico per flusso può essere indicativa di Denial of Service o di anomalie nel traffico di rete, aspetto che può non risultare evidente da un profilo di traffico misurato in bit/s. Torna quindi chiara l'importanza di uno strumento di monitoring nel controllo di un buon funzionamento della rete.

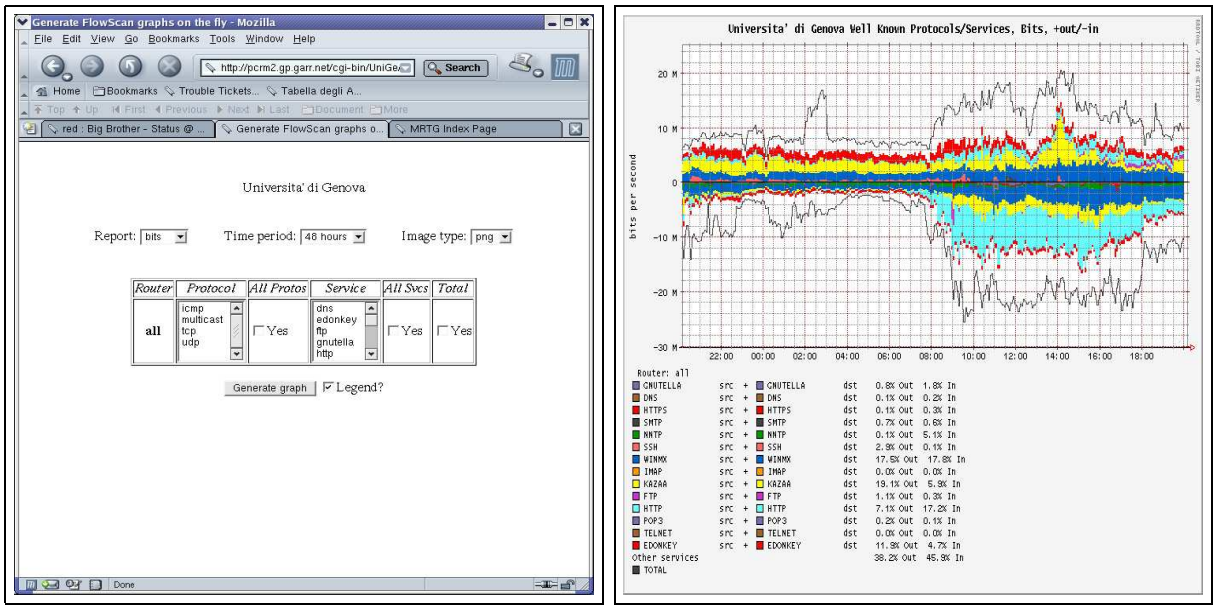
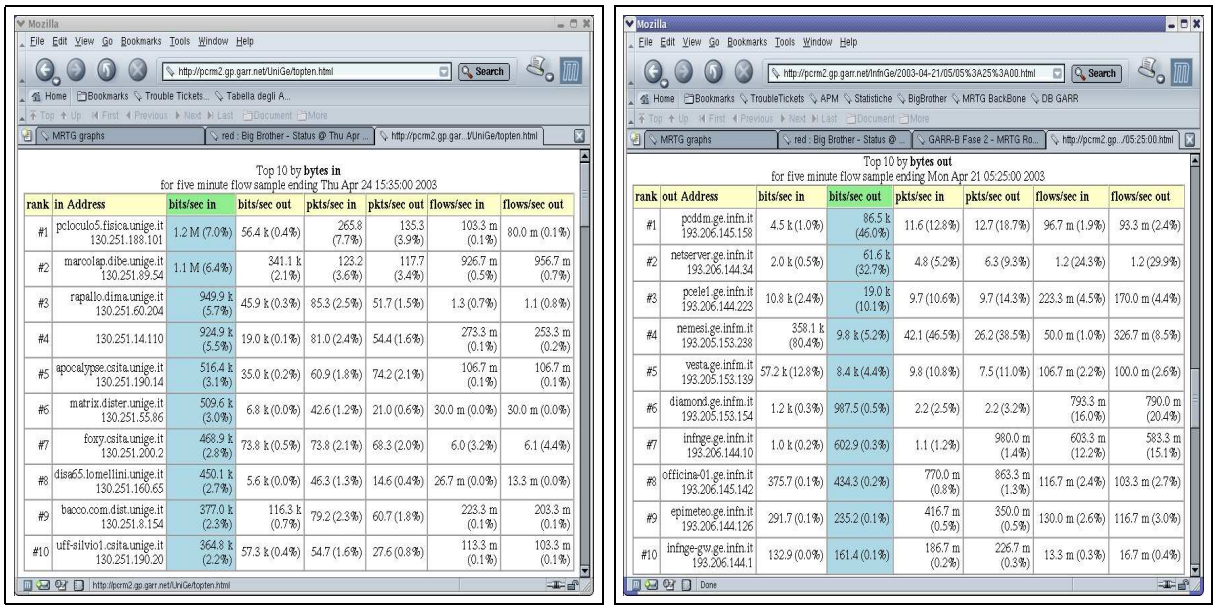


Figura 8: Home page di CUGrapher e andamento del traffico con le percentuali dei singoli servizi

FlowScan provvede inoltre, oltre alle viste grafiche, anche un controllo real-time sugli host che, ad intervalli di cinque minuti, hanno fatto il quantitativo maggiore di traffico. Per ogni flusso di rete vengono quindi aggiornati dei contatori che tengono traccia della quantità di bit/s, flussi/s e pacchetti/s scambiati negli ultimi cinque minuti. Il risultato è riportato in tabelle che, per ognuno degli n host, evidenziano la quantità di bits, pacchetti e flussi al secondo, in entrambe le direzione del traffico. La figura 9 riporta la tabella dei topten talker di Università di Genova tra le 15.30 e le 15.35 del 24 Aprile 2003. Il software permette l'archiviazione dei report in file html, con granularità a cinque minuti, che costituiscono un archivio storico delle sorgenti e destinazioni più attive all'interno dell'aggregato di rete preso in considerazione.

Nella nostra trattazione sono risultate di fondamentale importanza le mailing-list riferite ai singoli tool [19] [22] [23].



Top 10 by **bytes in**
for five minute flow sample ending Thu Apr 24 15:35:00 2003

rank	in Address	bits/sec in	bits/sec out	pkts/sec in	pkts/sec out	flows/sec in	flows/sec out
#1	pcloculo5.fisica.unige.it 130.251.188.101	1.2 M (7.0%)	56.4 k (0.4%)	265.8 (7.7%)	135.3 (3.9%)	103.3 m (0.1%)	80.0 m (0.1%)
#2	marcolap.dibe.unige.it 130.251.89.54	1.1 M (6.4%)	341.1 k (2.1%)	123.2 (3.6%)	117.7 (3.4%)	926.7 m (0.5%)	956.7 m (0.7%)
#3	rapallo.dima.unige.it 130.251.60.204	949.9 k (5.7%)	45.9 k (0.3%)	85.3 (2.5%)	51.7 (1.5%)	1.3 (0.7%)	1.1 (0.8%)
#4	130.251.14.110	924.9 k (5.5%)	19.0 k (0.1%)	81.0 (2.4%)	54.4 (1.6%)	273.3 m (0.1%)	253.3 m (0.2%)
#5	apocalypse.csita.unige.it 130.251.190.14	516.4 k (3.1%)	35.0 k (0.2%)	60.9 (1.8%)	74.2 (2.1%)	106.7 m (0.1%)	106.7 m (0.1%)
#6	matrix.dister.unige.it 130.251.55.86	509.6 k (3.0%)	6.8 k (0.0%)	42.6 (1.2%)	21.0 (0.6%)	30.0 m (0.0%)	30.0 m (0.0%)
#7	foxy.csita.unige.it 130.251.200.2	468.9 k (2.8%)	73.8 k (0.5%)	73.8 (2.1%)	68.3 (2.0%)	6.0 (3.2%)	6.1 (4.4%)
#8	disa65.lomellini.unige.it 130.251.160.65	450.1 k (2.7%)	5.6 k (0.0%)	46.3 (1.3%)	14.6 (0.4%)	26.7 m (0.0%)	13.3 m (0.0%)
#9	bacco.com.dist.unige.it 130.251.8.154	377.0 k (2.3%)	116.3 k (0.7%)	79.2 (2.3%)	60.7 (1.8%)	223.3 m (0.1%)	203.3 m (0.1%)
#10	uff-silvio1.csita.unige.it 130.251.190.20	364.8 k (2.2%)	57.3 k (0.4%)	54.7 (1.6%)	27.6 (0.8%)	113.3 m (0.1%)	103.3 m (0.1%)

Top 10 by **bytes out**
for five minute flow sample ending Mon Apr 21 05:25:00 2003

rank	out Address	bits/sec in	bits/sec out	pkts/sec in	pkts/sec out	flows/sec in	flows/sec out
#1	poddm.ge.infn.it 193.206.145.158	4.5 k (1.0%)	86.5 k (46.0%)	11.6 (12.8%)	12.7 (18.7%)	96.7 m (1.9%)	93.3 m (2.4%)
#2	netserver.ge.infn.it 193.206.144.34	2.0 k (0.5%)	61.6 k (32.7%)	4.8 (5.2%)	6.3 (9.3%)	1.2 (24.3%)	1.2 (29.9%)
#3	pcele1.ge.infn.it 193.206.144.223	10.8 k (2.4%)	19.0 k (10.1%)	9.7 (10.6%)	9.7 (14.3%)	223.3 m (4.5%)	170.0 m (4.4%)
#4	nemesi.ge.infn.it 193.205.153.238	358.1 k (80.4%)	9.8 k (5.2%)	42.1 (46.5%)	26.2 (38.5%)	50.0 m (1.0%)	326.7 m (8.5%)
#5	vesta.ge.infn.it 193.205.153.139	57.2 k (12.8%)	8.4 k (4.4%)	9.8 (10.8%)	7.5 (11.0%)	106.7 m (2.2%)	100.0 m (2.6%)
#6	diamond.ge.infn.it 193.205.153.154	1.2 k (0.3%)	987.5 (0.5%)	2.2 (2.5%)	2.2 (3.2%)	793.3 m (16.0%)	790.0 m (20.4%)
#7	infnge.ge.infn.it 193.206.144.10	1.0 k (0.2%)	602.9 (0.3%)	1.1 (1.2%)	980.0 m (1.4%)	603.3 m (12.2%)	583.3 m (15.1%)
#8	officina-01.ge.infn.it 193.206.145.142	375.7 (0.1%)	434.3 (0.2%)	770.0 m (0.8%)	863.3 m (1.3%)	116.7 m (2.4%)	103.3 m (2.7%)
#9	epimeteo.ge.infn.it 193.206.144.126	291.7 (0.1%)	235.2 (0.1%)	416.7 m (0.5%)	350.0 m (0.5%)	130.0 m (2.6%)	116.7 m (3.0%)
#10	infnge-gw.ge.infn.it 193.206.144.1	132.9 (0.0%)	161.4 (0.1%)	186.7 m (0.2%)	226.7 m (0.3%)	13.3 m (0.3%)	16.7 m (0.4%)

Figura 9: Tabella dei topten talkers negli ultimi 5 minuti, ordinati per quantità di bits/sec, in ingresso e uscita dal router utente

5 Prestazioni del sistema e dimensionamento

Le considerazioni che seguono vengono fatte in base all'analisi del traffico registrata con Netflow prolungata nel tempo (3-4 mesi) su Milano-RC e confrontata con Roma-RC e Napoli-RC; lo stesso tipo di analisi è stato fatto su MI-RTG e RM-RTG. Dato che i GigaRouter esportano il traffico campionato, la loro mole di traffico relativa a Netflow è ridotta (seguirà dettaglio); dei router concentratori presi in considerazione il traffico più consistente è stato quello di Milano-RC. Il comando *show ip flow export* ci permette di vedere la quantità di flussi esportati dal router e una serie di informazioni per il debugging.

RC-MILANO $ sh ip flow export
Flow export is enabled
Exporting flows to 193.xxx.xxx.72 (8100)
Exporting using source interface Loopback0
Version 5 flow records, origin-as
1367942252 flows exported in 45598077 udp datagrams
26651 flows failed due to lack of export packet
0 export packets were sent up to process level
0 export packets were punted to the RP
0 export packets were dropped due to no fib
566 export packets were dropped due to adjacency issues
8005 export packets were dropped enqueuing for the RP
15595 export packets were dropped due to IPC rate limiting
0 export packets were dropped due to output drops

Al tempo t_0 abbiamo azzerato i contatori e verificato il dato relativo ai flussi esportati 24 ore esatte dopo. Si è verificato che i flussi non fossero persi dal router controllando i contatori relativi alle statistiche dei flussi esportati ed il report di flow-tools sul collector-box (cross-check). Ripetendo questa operazione in giornate diverse con finestre temporali di ampiezza variabile e confrontando i dati ottenuti con quelli riportati dal collector abbiamo ricavato il numero medio e massimo di flussi esportati.

Esportazione UDP Queste operazioni hanno mostrato una media di 1350 $flussi/sec$ esportati dal router (Milano-RC) e un massimo di 4000 $flussi/sec$ in condizioni di traffico elevato ma non di DoS massiccio (caso patologico). Quest'ultimo caso, rilevato in momenti diversi, ha fatto riscontrare un'impennata dei flussi di un fattore compreso tra 4 e 5.5 volte il traffico usuale (in termini di flussi). In condizioni di elevato traffico e DoS contemporaneamente si sono registrati oltre $16000 flussi/sec$ (picco massimo). Netflow invia flussi (incapsulati in pacchetti udp), ciascuno pari a 48 $Bytes$. Una media di 1350 $flussi/sec$ fa sí che la media in bit/s che giunge al collector sia pari a $\sim$ 518 $Kbps$ nel caso medio e 1.5 $Mbps$ nel caso massimo (7.5 $Mbps$ caso DoS). Tenendo conto che l'effetto giorno-notte è abbastanza evidente nei flussi (non nell'utilizzo di banda da parte dell'utenza) a causa della mole di query http, risulta che il flusso medio nella fascia giornaliera è più del doppio rispetto a quello della fascia notturna. Il riscontro di quanto affermato lo si ha dai grafici mrtg della lan pilota di Milano-RTG alla quale è stato attestato il primo collector-box.

Spazio disco flow-tools dedica 60 $Bytes$ ad ogni flusso catturato dal collector-box. L'utilizzo del tool di cattura dei flussi (flow-capture) con l'opzione $-z5$ (compressione utilizzando zlib) ha dimostrato un rapporto di compressione pari a 3.2 (cioè per ogni flusso collezionato da flow-capture sono stati scritti su disco circa 19 $Bytes$ anzichè 60). Il massimo di compressione per cui sia evidente il risparmio di spazio-disco è stato il livello 6, che ha portato il rapporto di compressione a 4.8 (12.5 $Bytes$ su disco). Tuttavia l'incremento di compressione aumenta l'utilizzo di CPU di un fattore 2. Collezionando 1350 $flussi/sec$ medi sul router di Milano-RC si registra l'occupazione su disco di 2 $GigaBytes/giorno$.

GigaRouter Come detto in precedenza i GigaRouter supportano, allo stato attuale, la versione sampled di Netflow. Da studi fatti al Franhaufer Institute e alla Ohio State University risulta che per linee a 2.5 $Gigabps$, o superiori, un sampling $1 \div 100$ od anche $1 \div 1000$ porta a dei risultati quantitativi ragionevoli (differenze dal dato MRTG non superiori al 2%). Tale comportamento è stato riscontrato anche dai nostri test (bisogna tuttavia tenere in considerazione che i contatori dei router calcolano anche l'overhead del pacchetto, mentre NetFlow calcola solo il traffico IP). Come indicazione generale, è importante che il collector-box sia attestato il più vicino possibile al router tramite un cicuito dedicato (ad esempio FastEthernet). La vicinanza è dettata dalla natura connectionless di udp; i pacchetti udp che attraversano un link saturo o appena degradato, vengono irrimediabilmente perduti senza lasciare traccia.

Prestazioni degli apparati Dal punto di vista hardware (dual proc pentium III 800 MHz) e OS (Linux Red-hat 7.1, kernel 2.4.18) il collector-box utilizzato ha mostrato di essere abbastanza stabile . Il fatto di essere un biprocessore ha permesso di intervenire sulla macchina senza che gli interventi di routine siano mai stati critici per l'acquisizione dati. La cpu è risultata il parametro fondamentale, soprattutto a causa del lavoro di compres- sione sui dati. Talvolta, soprattutto nelle occasioni di DoS, si è assistito ad una perdita di flussi compresa tra lo 0.5 e il 3% , ma quasi nella totalità dei casi la causa è stata la perdita di pacchetti udp (si rende necessaria una modifica al Kernel per il resizing dei buffer).
La memoria RAM non ha presentato limitazioni (512 $MByte$) in quanto a dimensioni. La velocità di scrittura su disco ha mostrato di dover essere superiore a 10 $Mbps$ (caratteristica non critica per i dischi Ultra Wide SCSI).

Tipo di traffico	Traffico Reale Mbit/s	Traffico Reale flussi/s	NetFlow Kbit/s	NetFlow pacchetti/s
Medio	100	1062	407	36
Patologico (DoS)	100	$16 \cdot 10^3$	$6.4 \cdot 10^3$	218

Tabella 1: Rapporto tra traffico di produzione e traffico generato da NetFlow

Tipo di traffico	Traffico Reale Mbit/s (medi)	Traffico Reale flussi/s (medi)	Totali GBytes/giorno	Compressi GBytes/giorno
Medio	100	1062	5.13	1.56
Patologico (DoS)	100	$16 \cdot 10^3$	25	7.6

Tabella 2: Occupazione su disco del traffico generato da NetFlow

6 Conclusioni

In questo documento abbiamo presentato una possibile struttura di passive monitoring per una rete WAN. Strumento fondamentale è stato Netflow, feature dei router CISCO che ci ha permesso di implementare un sistema che non ha alterato le prestazioni della rete salvo in casi patologici, quali DoS. Si è illustrata a valle di Netflow una struttura di monitoring basata su software Open Source. In particolare si è rivelato estremamente utile, flessibile e affidabile, **flow-tools**, sia in qualità di collector-tool che di analyzer-tool, derivato da un lavoro della Ohio State University. Utilizzando strumenti disponibili in rete, quali Flow-Scan, ed altri realizzati da noi, abbiamo visualizzato i dati collezionati in modo da rendere immediata e di facile comprensione l'analisi. Quest'ultima è stata suddivisa in analisi di lungo periodo e analisi quasi real time. I report giornalieri hanno evidenziato la tipologia del traffico transitante da un ente predeterminato verso la rete GARR e viceversa. Abbiamo dimostrato come il traffico di file-sharing sia diventata un'applicazione determinante ai fini della gestione della banda per un determinato ente e che la diffusione di tali software ha cambiato il profilo di traffico di tipo monte-valle (giorno-notte); ci siamo soffermati sulla descrizione di alcuni tool popolari e abbiamo mostrato come si possa evidenziare, almeno in parte, tale traffico. Inoltre la capacità di evidenziare i flussi di traffico per AS , danno la possibilità di pianificare la rete in modo più consapevole. La parte real-time si è rivelata utilissima nell'identificare comportamenti anomali nella rete, in particolare i DoS, i port-scanning e gli IP sospetti di occupare banda oltre il lecito. Il progetto ci ha coinvolto nelle discussioni internazionali sull'argomento. Il confronto con la comunità delle NREN non solo europee, ma anche americane ed asiatiche, ci consente di affermare che la via da seguire sia quella segnata da IP-FIX, primo standard IETF che fissa il formato dei dati esportati dai router. Alla luce dei risultati ottenuti, abbiamo eseguito lo studio di un possibile sistema di monitoring per l'intera rete dimensionando la quantità delle macchine che servono e definendo la loro tipologia. La parte bibliografica, contenente vari puntatori, non è tutto quanto abbiamo visto, ma tutto ciò che abbiamo ritenuto più utile.

A Passive and Active Monitoring

Il passive-monitoring [30] è basato sull'utilizzo di hardware-device per monitorare il traffico in transito sulla rete. Questi possono essere dispositivi fisici quali sniffer (introdotti magari tramite splitter ottici o mirroring del traffico) o possono riferirsi a particolari implementazioni introdotte in altri dispositivi quali routers, switch o end-host. Esempi di tali strumenti includono il Remote Monitoring (RMON), il protocollo SNMP, ed i dispositivi che supportano NetFlow. Nel passive-monitoring i dispositivi sono interrogati periodicamente e le informazioni sono collezionate (nel caso dell'SNMP i dati vengono estratti dalla Management Information Base) per essere analizzate a posteriori con l'intento di valutare lo stato e le performance della rete. L'approccio passivo non incrementa l'utilizzo della rete in termini di banda per la sua particolare caratteristica di sfruttare direttamente il traffico di produzione. Tuttavia le richieste fatte per la collezione dei dati e la gestione di un sistema di passive-monitoring possono, in alcuni particolari casi (ad esempio netflow), generare traffico di certe dimensioni. L'aspetto critico del passive-monitoring è la gestione e la raccolta dei dati sopratutto nell'ipotesi che si vogliano estrapolare informazioni da tutti i pacchetti che compongono i flussi collezionati. Considerato che il passive-monitoring richiede l'analisi dei flussi di rete che compongono il traffico reale, risulta fondamentale affrontare con attenzione il problema della privacy o della sicurezza dei dati collezionati.

L'active-monitoring si basa sulla possibilità di iniettare pacchetti di prova nella rete o di trasmettere i pacchetti a server o ad applicazioni, monitorando e misurando il servizio di rete. L'active-monitoring permette un controllo esplicito sulla generazione dei pacchetti iniettati nella rete. Questo consente un totale controllo sulla natura del traffico generato, sulle tecniche di campionamento, sulla grandezza e sul tipo dei pacchetti (in modo da emulare varie tipologie di traffico), sul path e sulle funzioni scelte per essere monitorate. Sia il volume che gli altri parametri riferiti al traffico iniettato nella rete sono totalmente configurabili ed anche con volumi di traffico limitati è possibile ottenere risultati significativi. La simulazione di vari scenari è facile da implementare e, a partire da essa, è possibile testare il Quality of Service (QoS) ed il Service Level Agreements (SLAs).

B NetFlow

La nascita di NetFlow risale al 1996 quando CISCO rilascià un'immagine di IOS EFT con questa nuova feature [31]. La nozione di flow-profiling, intesa come monitoring del traffico di rete a partire dai singoli flussi, è stata introdotta nel mondo della ricerca sulle reti di calcolatori con l'intento di ottimizzare la conoscenza del traffico internet. Il flow-profiling è stato il metodo considerato ideale per applicazioni di caching e accounting, per la sua caratteristica di essere un buon compromesso tra la cattura della totalità dei pacchetti in transito su un'interfaccia ed il loro semplice conteggio. Risolve quindi il difficile problema dell'acquisire la quantità di dati giusta per ottenere informazioni sensibili. Non esiste ancora uno standard che definisca un protocollo adatto ad esportare informazioni dai router; CISCO ha implementato uno strumento, NetFlow, che permette di allocare una memoria cache in cui vengono memorizzati i flussi di traffico e da cui possono essere facilmente esportati. NetFlow esiste in diverse versioni, variabili per granularità di informazioni che

si ritengono importanti o per gli apparati di rete che vengono utilizzati. Qui di seguito è presentata la descrizione dettagliata del tool che rappresenta le fondamenta della totalità del lavoro che viene presentato.

Un flusso è definito da una sequenza unidirezionale di pacchetti in transito tra una determinata sorgente e destinazione, entrambi identificati, a livello network, da un indirizzo IP, a livello di trasporto, dai numeri di porta sorgente e destinazione. Nello specifico NetFlow [24] [25] [26] identifica un flusso combinando i seguenti sette campi:

- IP address sorgente
- IP address destinazione
- Numero di porta sorgente
- Numero di porta destinazione
- Tipo di protocollo di trasporto
- ToS byte
- Interfaccia logica d'ingresso (ifIndex)

Queste sette chiavi identificano un unico flusso. Nel caso un flusso abbia un solo campo differente da un altro, esso verrà considerato un nuovo flusso. Un flusso può contenere anche altri campi che dipendono dalla versione di NetFlow (ad es. l'AS number nel formato dei flussi esportati da NetFlow Versione 5), che è configurata sul router all'atto della configurazione.

B.1 NetFlow, definizione e benefici

Abilitando la modalità NetFlow il router processa ogni pacchetto di un nuovo flusso. Le informazioni provenienti dall'elaborazione del primo pacchetto di ogni flusso sono usate per popolare una tabella chiamata Flow Cache, che verrà poi utilizzata dai pacchetti successivi del flusso. NetFlow pu essere il motore di svariate applicazioni chiave quali:

- Applicazioni di Accounting e di Billing La collezione dei dati include anche dettagli come l'IP address, il numero dei pacchetti e dei byte per pacchetto, il timestamps, il type of services e l'application port. Queste informazioni possono essere utilizzate per quantificare l'occupazione e l'utilizzazione delle risorse per un eventuale billing o per cambiare in modo dinamico le policy di QoS o di sicurezza sulle varie applicazioni o flussi.
- Applicazioni di Network Planning e di Analisi La collezione dei dati è la fonte per tools sofisticati di Network Planning e Network Analisys in grado di aiutare alla pianificazione e all'ottimizzazione della Rete sia da un punto di vista economico che da un punto di vista di sicurezza e affidabilità.

- Applicazioni di Network Monitoring La collezione dei dati abilita capacità di monitoring sui flussi e sulle applicazioni, il che permette l'individuazione di problemi. Efficace strumento di troubleshooting e capacità di proattività alla Rete (prevenzione degli errori e risoluzione automatica prima che l'utenza si accorga del problema).

B.2 NetFlow Cache Management e Data Export

Il cuore di NetFlow è la gestione della Flow Cache, specialmente quando i router hanno un'elevata densità di flussi da gestire. Il NetFlow cache management software contiene una serie di algoritmi in grado di determinare se un pacchetto è parte di un flusso esistente, se bisogna generare una nuova entry nella Flow Cache o nel determinare se una entry della Flow Cache deve essere aggiornata o cancellata in quanto il flusso è terminato. Netflow permette la scelta dei tempi di mantenimento di un flusso all'interno della cache memory e prevede l'impostazione di un parametro temporale (timeout) che regola la durata temporale di un flusso prima che esso venga considerato completo, quindi pronto ad essere esportato. Il suddetto parametro di timeout può essere impostato sia per i flussi considerati active che per quelli inactive. Sono considerati inactive i flussi che non stanno più ricevendo pacchetti; active quelli che stanno subendo pacchetti di aggiornamento. Le regole per determinare se un entry è scaduta includono le seguenti modalità:

- Flussi inattivi per lunghi periodi di tempo (oltre il valore di inactive timeout) sono giudicati conclusi e rimossi dalla Flow Cache.
- Flussi active con una vita superiore al valore di active timeout sono giudicati conclusi e rimossi dalla Flow Cache.
- Connessioni TCP/IP che hanno ricevuto il comando di fine sessione (FIN) o che sono stati resettati (RST) vengono rimossi dalla flow cache.

Tutti i flussi giudicati conclusi sono raggruppati insieme in NetFlow Export UDP datagrams pronti per essere spediti ad una stazione di raccolta con funzionalità di archivio. Questo archivio è poi utilizzato da stazioni di management per la manipolazione delle informazioni secondo delle query ben precise atte ad estrapolare dati di consumo e di controllo.

Abilitando NetFlow su un'interfaccia, viene riservata una quantità di memoria tale da ospitare un numero determinato di entry nella flow-cache. Il numero di entry permesse nella flow-cache ha un valore di default che dipende dalla piattaforma e dalla quantità di DRAM presente sul router. Ogni singolo flusso (cache-entry) è pari a 48 *Bytes*.

Con il comando *ip flow-cache entries* è possibile configurare l'ampiezza della Flow-Cache per ospitare un numero di entry compreso tra 1024 e 524288. È importante sapere che i cambiamenti di capacità della Flow-Cache avvengono solamente dopo aver abilitato e disabilitato Netflow o successivamente ad un reboot del router.

I dati Netflow sono input-based, i flussi di dati vengono registrati solo per pacchetti entranti sull'interfaccia del router. Questa caratteristica di Netflow introduce notevoli complicazioni nell'analisi del traffico entrante e uscente da un'interfaccia.

Il router controlla la flow-cache una volta al secondo, e considera un flusso spirato tenendo conto delle seguenti istanze

- Trasporto completato (TCP FIN o RST)
- Flow-cache piena
- Il valore di timer timeout-inactive è spirato o il traffico è concluso
- Il valore di timeout-active è spirato o il traffico è concluso

L'inactive timer esporta un pacchetto con un valore di inattività di traffico di 15 sec. È possibile modificare il valore di inactive timer in un intervallo compreso tra 10 e 600 sec (Tabella). L'active timer esporta un pacchetto successivamente a 30 min di traffico attivo. È possibile modificare il valore di active timer in un intervallo compreso tra 1 e 60 min. Il contenuto della Flow-Cache viene visualizzato con il comando show ip cache flow.

B.3 Comandi di configurazione e visualizzazione di NetFlow sui router

L'applicazione di NetFlow su un router prevede sia una parte di configurazione globale che una specifica per ogni interfaccia da cui si vogliono esportare i flussi. In modalità di configurazione globale è necessario specificare i dettagli riguardanti i flussi UDP che verranno esportati (indirizzo IP sorgente e destinazione, porta di trasporto e versione di NetFlow). Il comando che permette l'abilitazione di NetFlow ip flow-export. Tramite il comando ip route cache flow possibile abilitare NetFlow su un'interfaccia. Nel caso di un'interfaccia fisica che comprenda sub-interface, è sufficiente configurare NetFlow sulla fisica perch tutte le interfacce virtuali vengano abilitate automaticamente. La modalità di esportazione dei flussi da uno specifico router viene abilitata, in modalità di configurazione globale, con i comandi presentati in tabella.

```
RC_MILANO#configure terminal
RC_MILANO(config)#ip flow-export source Loopback0
RC_MILANO(config)#ip flow-export version 5 origin-as
RC_MILANO(config)#ip flow-export destination 193.204.221.12 8100
RC_MILANO(config)#interface ATM1/0/0
RC_MILANO(config-if)#ip route-cache flow
RC_MILANO(config-if)#exit
```

Tabella 3: Esportazione dei flussi

Nel caso si vogliano esportare flussi con un meccanismo di sampling, la configurazione è la seguente.

In tabella sono presentati i comandi per la configurazione di NetFlow sui router e l'output di show ip flow export.

Con il comando show ip flow export si verifica la correttezza della configurazione oltre a controllare il numero effettivo di flussi che vengono esportati o droppati.

```
MILANO-RTG#configure terminal
MILANO-RTG(config)#ip flow-export source Loopback0
MILANO-RTG(config)#ip flow-export version 5 origin-as
MILANO-RTG(config)#ip flow-export destination 193.204.221.79 8002
MILANO-RTG(config)#ip flow-sampling-mode packet-interval 100
MILANO-RTG(config)#interface ATM1/0/0
MILANO-RTG(config-if)#ip route-cache flow sampled
```

Tabella 4: Configurazione sampling

```
RC_MILANO(config)#ip flow-cache timeout active 25
RC_MILANO(config)#ip flow-cache timeout inactive 400
```

Tabella 5: Impostazione dei parametri di cache timeout active e inactive

```
RC_MILANO(config)#ip flow-cache entries ?
  <1024-524288>  Entries
```

Tabella 6: Variazione del numero di entry nella cache di NetFlow

```
RC_MILANO#show ip flow export
Flow export is enabled
  Exporting flows to 193.204.221.12 (8100)
  Exporting using source interface Loopback0
  Version 5 flow records, origin-as
  99477631 flows exported in 3319499 udp datagrams
  0 flows failed due to lack of export packet
  0 export packets were sent up to process level
  0 export packets were punted to the RP
  0 export packets were dropped due to no fib
  0 export packets were dropped due to adjacency issues
  0 export packets were dropped enqueuing for the RP
  0 export packets were dropped due to IPC rate limiting
  0 export packets were dropped due to output drops
```

Tabella 7: show ip flow export

Per controllare il contenuto della cache si deve utilizzare il comando show ip cache flow. Viene riportata una serie di statistiche che puo' risultare utile per il troubleshooting sulla rete.

Per controllare il contenuto della cache si deve utilizzare il comando show ip cache flow. Viene riportata una serie di statistiche che puo' risultare utile per il troubleshooting sulla rete.

E' possibile visualizzare il contenuto della cache selezionando i soli flussi riguardanti una precisa interfaccia; in questo modo si può controllare in maniera real-time il traffico di un unico ente.

B.4 Versioni di NetFlow

I dati NetFlow vengono esportati in datagram UDP in quattro dei seguenti formati (a seconda della versione di NetFlow utilizzata): Version 1, Version 5, Version 7 e Versione 8. I datagram consistono di un header e di uno o pi flow-record. Il primo campo dell'header riporta la versione di NetFlow del datagram esportato, mentre il secondo contiene il numero di record di cui è composto il datagram. Nelle versione 5, 7, e 8 l'header comprende anche un sequence number, tale da permettere un controllo per i datagram persi nell'esportazione UDP. Il sequence number di un datagram è pari al s.n. del precedente sommato al numero di flussi del datagram precedente. In seguito alla ricezione di un datagram, viene effettuato un check sul s.n., in modo da risalire all'eventuale perdita di flussi persi. La versione che abbiamo utilizzato è la 5, adatta alle versioni di router presenti sulla rete GARR e riportante, oltre ai campi fondamentali già presenti nella Version 1, anche informazioni sul protocollo BGP e sugli Autonomous System. Le versione 7 ed 8 non sono state prese in considerazione essendo adatte alla famiglia dei Catalyst CISCO. Nelle figure vengono presentati l'header ed il flow-entry di NetFlow Version5.

B.5 Sampled Netflow

Concludiamo la discussione su NetFlow, illustrando gli aspetti riguardanti il campionamento che si sono rivelati particolarmente importanti nello studio di monitoring che abbiamo affrontato. Questo particolare aspetto è riferito ai router Cisco della famiglia 12000 [25], che hanno l'efficacia di demandare la maggior parte delle decisioni direttamente all'hardware delle linecard, meccanismo che migliora notevolmente il throughput dei pacchetti. Nel momento in cui viene abilitato NetFlow sui Cisco 12000, il meccanismo di forwarding fatto dall'hardware viene by-passato al software. Questa situazione si riflette in un incremento sostanziale dell'utilizzo di CPU, rilevato in presenza di traffico anomalo (DoS). Questo aspetto assume importanza solo sulle linecard dei 12000 che utilizzano una versione di NetFlow software based. Le linecard dei 12000 Engine 4 e 5 hanno un'implementazione di NetFlow basata su hardware ASIC (application-specific integrated circuit), che permette di switchare i pacchetti NetFlow al line-rate. Inoltre essendo tutto basato sull'hardware, non si presentano problemi di diminuzione delle performance sulla CPU della line-card [25]. Quando viene implementato NetFlow software-based sui 12000, è fortemente raccomandata l'applicazione del sampling. La feature di sampling permette al router di campionare uno di x pacchetti IP forwardati. I pacchetti campionati vanno

```
RC_MILANO#show ip cache flow
IP packet size distribution (23062M total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .438 .036 .029 .012 .008 .008 .006 .005 .004 .005 .004 .004 .003 .003

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .003 .002 .025 .041 .354 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 4456704 bytes
  7827 active, 57709 inactive, 46651245 added
  712192266 ager polls, 0 flow alloc failures
  Active flows timeout in 10 minutes
  Inactive flows timeout in 120 seconds
  Last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet      324169      0.4        24   212     11.5      14.7      46.9
TCP-FTP        4262580      6.1         9   204     57.2       7.9      36.5
TCP-FTPD        741963      1.0       526   852    558.5      38.1      16.5
TCP-WWW      371787246    532.0        12   545   6816.0       6.1      27.9
TCP-SMTP       7225291     10.3        34   638    357.7      16.5      33.2
TCP-X            34364      0.0       314   302     15.4      53.8      34.5
TCP-BGP          21628      0.0         6    49      0.2     171.2      34.9
TCP-NNTP        260951      0.3      1423   901    531.7      91.1      24.5
TCP-Frag         25895      0.0         9    64      0.3       5.2      61.5
TCP-other    234619118    335.7        68   644  23008.3      32.0      30.3
UDP-DNS       28612165     40.9         2    84    116.3      11.7      78.0
UDP-NTP        1255839      1.7         1    75      3.2      43.4      69.7
UDP-TFTP         12716      0.0         3    52      0.0       9.3     117.7
UDP-Frag        527981      0.7        35   939     26.5      13.9      95.6
UDP-other    311443312    445.7         2   176   1287.9      12.8      77.3
ICMP          38838177     55.5         3   142    189.7      13.6      84.2
IPINIP             834      0.0        12   266      0.0       8.3     118.3
GRE              22324      0.0        53   212      1.7     157.5      17.3
IP-other         12777      0.0       656   144     12.0     193.9      23.6
Total:      1000029330   1431.1        23   607  32994.8      14.9      47.6

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
AT8/0/0.1     192.168.1.2     AT0/0/0.178   192.167.21.42   06 125D 0050     3
AT8/1/0.1     205.188.7.72    AT0/0/0.178   192.167.29.191  06 1446 0509     4
AT8/0/0.1     61.64.172.75    AT0/0/0.178   192.167.22.87   06 4109 0050     1
...
```

Tabella 8: show ip cache flow

```
RC_MILANO#if-con 0 con
Entering CONSOLE for VIP2 R5K 0
Type ^C^C^C or if-quit to end this session


VIP-Slot0>sh ip cache flow | include Se0/1/1
SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Se0/1/1       193.206.129.154 AT8/1/0.1     211.114.16.4    01 0000 030D     1
Se0/1/1       193.206.129.154 AT8/0/0.1     61.182.254.162  01 0000 030D     1
...
```

Tabella 9: show ip cache flow sulla VIP 0 del router

quindi a riempire la Flow-Cache. L'abilitazione del sampling sostanzialmente decrementa l'utilizzo di CPU dei pacchetti NetFlow, permettendo uno switching pi veloce ed efficiente. È chiara la complicazione introdotta dal sampling in applicazioni di passive-monitoring; sopratutto in un tipo di analisi basata sui flussi, in cui però viene attuato un campionamento per pachetti. Questo infatti, limitando l'analisi ad una percentuale dei pacchetti, implica una trattazione statistica nel caso si vogliano estrapolare informazioni riferite alla totalità del traffico in transito sul router. In seguito viene trattata l'accuratezza dell'analisi del traffico a partire da un base di dati campionata.

B.6 IPFIX

IPFIX (Internet Protocol Flow Informaion eXport) è un working-group facente capo a IETF nato appositamente per trovare uno standard univoco riguardo l'esportazione e l'analisi di flussi di rete IP[29]. Esiste infatti un numero elevato di sistemi di esportazione di informazioni di flussi IP di uso comune sulle reti; questi sistemi differiscono significativamente tra loro sebbene buona parte di essi abbia adottato un meccanismo di trasporto comune. Alcune differenze ancora presenti rendono tuttavia complicato lo sviluppo di tool di analisi che possano avere uno sviluppo generalizzato. Rimane quindi viva la necessità presentata sia dal mondo industriale che dalle National Research Network di raggiungere uno standard comune riferito agli apparati di rete, quali router o switch, da cui si possano esportare flussi di rete in un formato standard. Un sistema di esportazione di flussi d'informazione include un data-model ed un protocollo di trasporto. Le informazioni contenute nel flusso esportato sono di due tipi:

Attributi derivanti dall'header del pacchetto IP, quali indirizzo sorgente e destinazione, protocollo e porta d'applicazione.

Attributi spesso conosciuti solo dall'exporter, quali le porte d'ingresso e d'uscita, l'IP subnet mask e gli AS number e talvolta anche informazioni sui sub-IP-layer.

Il gruppo di lavoro vuole definire un protocollo grazie al quale i flussi possono essere esportati con una sequenza temporale, da un exporter ad un collection-engine ed un'architettura che ne permetta l'implementazione. Il protocollo scelto deve essere adatto a

runnare su di un protocollo approvato da IETF, con un controllo della congestione quale TCP o SCTP (Stream Control Transmission Protocol).

Gli obiettivi primari di IPFIX sono:

- Definire la nozione di flusso IP standard.
- Escogitare una codifica dati che supporti l'analisi sia di flussi IPv4 che IPv6 con politiche di routing sia unicast che multicast.
- Considerare l'esportazione di flussi IP basata su sampling dei pacchetti.
- Identificare una politica di privacy e sicurezza che preservi l'esportazione dei flussi. In particolare determinare una tecnologia sicura di export, e.g. TLS (Transport Layer Security).
- Transport mapping
- Assicurarsi che il sistema di flow export sia affidabile, che minimizzi la probabilità di perdita dei flussi e che riporti messaggi di log in presenza di perdite d'informazione.

La mailing-list di IPFIX è nata appositamente per raggiungere gli obbiettivi di standardizzazione cui si è fatto riferimento. La lista è ospitata dalla Division of Information Technology's Network Services group dell'Università del Wisconsin - Madison. L'email della lista è ipfix@net.doit.wisc.edu .

C flow-tools

flow-tools è un insieme di tool, realizzati alla Ohio State University (OSU) a partire dal 1996 da Mark Fullmer, per collezionare (Netflow-collector), filtrare, visualizzare ed analizzare i cosiddetti Netflow PDUs (Protocol Data Unit). La filosofia di utilizzo dei tool è di tipo Unix pipeline, cioè permette la cascata dei comandi, compresi quelli presenti nel sistema operativo. flow-capture è il primo e fondamentale strumento per implementare la raccolta dati in un collector-box. Come è stato illustrato precedentemente il collector-box riceve udp packet inviatigli direttamente dal router o dallo switch ad una porta precisa sulla quale il collector sta in ascolto. flow-capture converte i raw-files Netflow in una rappresentazione che gli permette di avere un summary del flusso ricevuto dall'apparato remoto (flow record, 60 bytes). La maggior parte degli altri tool lavora su questo formato, sempre di tipo raw. Una caratteristica importante è che viene utilizzata la libreria di archiviazione zlib, consentendo una compressione considerevole della mole di dati ricevuti dal collector (è un'operazione che va tarata sulla propria macchina considerando il rapporto spazio-disco vs CPU). Attualmente il modo in cui è utilizzato sulle macchine GARR è tale da consentire un rapporto $\sim$ 1 : 3.3 circa tra il dato compresso e quello reale. I benefici principali di questo software sono i seguenti:

- Permette di filtrare tutti i campi del pacchetto Netflow direttamente sui raw-file in modo da razionalizzare l'analisi ed eliminare i dati superflui

- È compatibile sia con apparati Juniper che Cisco
- È di facile configurazione e debugging
- Integra una serie di tool che gli permettono, tra l'altro, di fare il reply dei dati ad altri collector sia in modo nativo che inseguito a filtraggi.

Installazione Software su Collector-BOX

1. Scaricare il file zlib-1.1.4.tar.gz o versioni più recenti
2. Installare tar -zxvf zlib-1.1.4.tar.gz
   cd zlib-1.1.4
   ./configure
   make
   make install
3. Scaricare il pacchetto flow-tools cercando di utilizzare una versione stabile (Attualmente 0.62–se si volesse installare la 0.64 , bisogna provvedere a patcharla). Il file sarà del tipo flow-tools-0.62.tar.gz
4. Installare tar -zxvf zlib-1.1.4.tar.gz
   cd zlib-1.1.4
   ./configure -prefix=directory-di-destinazione-prescelta
   make
   make install
5. È importante ricordarsi di mettere nel PATH la directory contenente i file binary di flow-tools: editare il file $.bash\_profile$ dell'utente proprietario del processo flow-capture e aggiungere alla variabile PATH :/daqualcheparte/Flow-Tools/bin

Att.ne: flow-tools di default cerca di fare l'installazione in /usr/local/ e di accedere a /var .
Mentre il primo passo è scavalcabile con l'opzione $--prefix$ all'atto dell'installazione, la seconda richiederebbe l'editing del file di configurazione per cambiare PATH e non averne uno assoluto. Una soluzione intermedia consiste nell'avere i permessi in scrittura nelle directory richieste da flow-tools.

C.1 Collezione dati

1. Creare un directory padre per ogni router che si desidera collezionare: Ad esempio per milano-rtg:
   mkdir /home/netflow/Statistiche/Milano-rtg.
2. Assicurarsi dello spazio libero a disposizione sul disco
   df -h .
3. Decidere quanto spazio riservare ai dati di ciascun router.

4. Fase di cattura flow-capture -z4 -V5 -n288 -w/home/netflow/Statistiche/Milano-rtg -E5G -S5 193.204.221.78/193.206.129.252/8700 .

Significato delle opzioni utilizzate

-z: livello di compressione desiderato; minimo=0, max=9. Con un livello pari a 5 la macchina nf1.gp.garr.net (Compaq proliant) Comprime con circa un rapporto 1:3.3 ; di fatto aumentare il livello di compressione oltre il 6 non aumenta di molto la compressione, peraltro incrementa l'impiego di cpu. Di conseguenza è necessario un tuning che bilanci CPU e spazio disco.

-V : versione di netflow (quella da noi usata è la 5)

-n : numero di rotazioni giornaliere: numero di volte che flow-capture crea un nuovo file per giorno(288 volte corrispondono a 5 minuti)

-w : Seleziona directory di destinazione dei raw-files

-E: Seleziona lo spazio disco limite da utilizzare: si possono utilizzare le lettere b, K,M,G
come multipli (es. 5G corrisponde a 5 Gigabyte). All'esaurimento dello spazio selezionato, flow-tools automaticamente cancella i raw-files più vecchi (FIFO).

-S: messaggi di log ogni intervallo selezionato (5 minuti nell'esempio) che riportano i totali per il dato intervallo (in termine di flussi, pacchetti, bytes)

Infine si seleziona ip-collector-box/ip-router/porta, dove:
ip-collector-box = ip della macchina che fa da collector(su cui viene runnato flow-capture)
ip-router = indirizzo ip del router che invia dati netflow(Loopback)
porta = porta udp che abbiamo designato sul router in fase di configurazione di netflow export

Automaticamente, con le impostazioni date precedentemente si dovrebbe, ottenere la struttura (per ad es. l'08/01/2003)
/home/netflow/Statistiche/Milano-rtg/2003/2003-01/2003-01-08

Nella directory foglia (cioè yyyy-mm-dd) troviamo:

-rw-r–r–	1	root	root	100	Jan	8	17:16	ft-v05.2003-01-08.171125+0100
-rw-r–r–	1	root	root	100	Jan	8	17:21	ft-v05.2003-01-08.171624+0100
-rw-r–r–	1	root	root	100	Jan	8	17:26	ft-v05.2003-01-08.172123+0100
-rw-r–r–	1	root	root	100	Jan	8	17:31	ft-v05.2003-01-08.172622+0100
-rw-r–r–	1	root	root	92	Jan	8	17:31	tmp-v05.2003-01-08.173121+0100

I file sono identificabili facilmente dall'orario che fa parte del nome e segue il punto, espresso in hhmmss. Il file tmp è quello non ancora chiuso da flow-capture.

Debugging

Prima di passare alla estrazione dei dati e alla relativa analisi è bene assicurarsi che stiamo acquisendo i dati in modo corretto.

1. Verifica traffico

   (a) Assicurarsi che a livello Netflow sia tutto regolare (vedi sezione Netflow)

   (b) Assicurarsi che il collector-box sia raggiungibile dal router, ma che non sia aperto all'esterno

   (c) Sembra banale, ma assicurarsi che ip-sorgente e ip-router siano effettivamente quelli corretti

   (d) Attenzione ai DoS: controllare tramite MRTG l'aumento del traffico sul link dedicato

2. Visualizzazione dati

Ovviamente questo tutorial non pretende di spiegare tutte le funzionalità di flow-tools, per le quali si rimanda alle references e ai file man. Premilinarmente a questa discussione è comunque importante capire la filosofia di utilizzo dei componenti di flow-tools. Tool con funzioni diverse possono essere messi in cascata alla Unix-pipe. Il primo strumento fondamentale è flow-cat. A differenza di cat di Unix, esso permette di tagliare le varie intestazioni dei files in modo che siano elaborabili dal tool di visualizzazione più file contemporaneamente. Di conseguenza il suo utilizzo sarà sempre

flow-cat $ft - v05.2003 - 01 - 08.171125 + 0100 \mid \cdots$
Nel caso si vogliano indicare tutti i files della directory, o di una certa ora, basta utilizzare il carattere * nella posizione opportuna
flow-cat ft-v05.2003-01-08.* $\mid \cdots$

Per avere informazioni sintetiche sul numero di flussi che stiamo acquisendo Utilizziamo l'opzione -p (preload header) e in cascata ad esso flow-header
[netflow@nf2 2002-12-10]$ flow-cat -p ft-v05.2002-12-10.151150+0100 | flow-header

Visualizzazione immediata : flow-print.

Partiamo da un esempio:
$[netflow@nf22002 - 12 - 10]$ flow-cat ft-v05.2002-12-10.151150+0100 | flow-print |more

mode:	streaming
capture start:	Tue Dec 10 15:11:50 2002
capture end:	Tue Dec 10 15:16:49 2002
capture period:	299 seconds
compress:	off
byte order:	little
stream version:	3
export version:	5
lost flows:	0
corrupt packets:	0
capture flows:	434073

srcIP	dstIP	prot	srcPort	dstPort	octets	packets
192.107.72.245	152.163.208.249	6	2459	80	394	5
193.206.120.164	80.117.222.208	6	3655	3389	142	1
192.107.75.158	212.110.12.177	6	6418	80	40	1
193.206.120.164	147.102.46.105	17	3655	3602	63	1
141.108.3.252	193.205.222.141	17	7000	7001	197	2

Questo output è analogo a quanto si vede sul router CISCO con *sh ip cache flow* l'unica differenza sta nel fatto che la rappresentazione dei numeri di porte e protocollo sono in decimale anziche' in esadecimale come avviene invece su Cisco IOS.

Opzioni da utilizzare con flow-print
-n : Use symbolic names where appropriate.
L'output di prima diventa:
$[netflow@nf22002-12-10]$ \$ flow-cat ft-v05.2002-12-10.151150+0100
| flow-print -n | more .

srcIP	dstIP	prot	srcPort	dstPort	octets	packets
192.107.72.245	152.163.208.249	tcp	2459	http	394	5
193.206.120.164	80.117.222.208	tcp	3655	3389	142	1
192.107.75.158	212.110.12.177	tcp	6418	http	40	1
193.206.120.164	147.102.46.105	udp	3655	3602	63	1
141.108.3.252	193.205.222.141	udp	afs3-fi	afs3-ca	197	2

Oss.: I nomi vengono associati andando a leggere i files /etc/services e /etc/protocols
-f :
3 1 line, no interfaces, decimal ports
4 1 line with AS number
$[netflow@nf22002-12-10]$\$ flow-cat 2003-06-06.235501+0200 | flow-print -f5| more

Nell'output mostrato si evidenzia l'informazione completa sul flusso, compreso il time-stamp di inizio e di fine al millisecondo, e le interfacce sorgente e destinazione.

Start	End	Sif	SrcIPaddress	SrcP	DIf
0606.23:54:58.177	0606.23:55:03.673	69	151.97.230.10	0	80
0606.23:54:58.181	0606.23:55:03.673	69	151.97.230.10	0	80
0606.23:54:58.181	0606.23:55:03.673	69	151.97.230.10	0	80
0606.23:54:58.181	0606.23:55:03.673	69	151.97.230.10	0	80
0606.23:54:58.193	0606.23:55:03.685	69	151.97.230.10	0	80

DstIPaddress	DstP	P	Fl	Pkts	Octets
194.236.63.172	2048	1	0	2	74
194.236.63.174	2048	1	0	2	74
194.236.63.175	2048	1	0	2	74
194.236.63.176	2048	1	0	2	74
194.236.63.177	2048	1	0	2	74

C.2 Visualizzazione e analisi

Visualizzare l'intero contenuto di 5 minuti di traffico su un router può avere senso solo dal punto di vista degli abusi (vedremo in seguito) e del debugging (ci si assicura che i dati siano consistenti). Per selezionare l'informazione esistono flow-filter e flow-nfilter.
Flow-filter
Questo strumento permette al volo di filtrare, secondo vari parametri, l'aggregato di traffico selezionato.
Probabilmente l'utilità maggiore è nell'utilizzo quasi real-time e perciò ritengo che un filtro sull'interfaccia (in o out) sia l'opzione più utile:

-i input_filter :Input interface filter, ie -i0 to permit traffic from interface 0
-I output_filter :Output interface filter, ie -I0 to permit traffic to interface 0.

$bash-2.05\$$ flow-cat $ft-v05.2002-12-10.15*|$flow-filter -i 5 | flow-stat -P -f9 -S3

```
# — —- —- Report Information — — —
 # Fields:       Percent Total
 # Symbols:      Disabled
 # Sorting:      Descending Field 3
 # Name:         Source IP                #
 #
 # Args:         flow-stat -P -f9 -S3
#
 # IPaddr            flows     octets    packets
 #
 193.206.158.32      8.824     99.766    99.805
 193.206.158.142     88.235    0.212     0.183
 193.206.158.144     2.941     0.022     0.012
```

Nell'esempio si è selezionata l'interfaccia di ingresso 5 che corrisponde a Direzione GARR link1.
Anzichè visualizzare l'output con flow-print si è utilizzato flow-stat. Flow-stat è un'utility

0 Overall Summary
1 Average packet size distribution
2 Packets per flow distribution
3 Octets per flow distribution
5 UDP/TCP destination port
6 UDP/TCP source port
7 UDP/TCP port
8 Destination IP
9 Source IP

Tabella 10: Opzioni di flow-stat

che permette di generare reports basati su :
IP-address, porte, pacchetti, bytes, interfacce del router , next hops, AS, ToS. Nell'esempio si è utilizzata l'opzione -f 9 che seleziona gli IP sorgenti e l'output è ordinato secondo l'opzione -S3 (sort), ovvero bytes (octets) in ordine decrescente. Le opzioni più utilizzate sono in tabella 10

Flow-nfilter

L'utility flow-nfilter filtra i flussi basati secondo criteri definibili dall'utente. I filtri sono composti da primitive e una definizione. Le definizioni contengono righe, raggruppate secondo operazioni logiche di tipo AND o OR, che matchino i flussi usando le primitive selezionate.
ES:
filter-primitive inizio
type time-date
permit gt December 09, 2002 16:00:00

Le primitive dei filtri iniziano con la parola chiave filter-primitive seguita da una parola simbolica. Ogni primitiva prevede un tipo definito (sotto) e segue la parola chiave type. Infine segue una lista di parole chiave permit o deny seguite da un argomento che sarà successivamente valutato per stabilire se il flusso è accettato o rifiutato (tipo acl). L'azione di default è deny. La match keyword in una definizione seleziona il criterio per matchare la primitiva . Dopo tante parole oscure, vediamo un esempio:
filter-primitive Geant
type ifindex
permit 32
filter-primitive start
type time
permit gt 16:00
filter-primitive end
type time
permit lt 19:30

filter-definition ente_in
match input-interface Geant
match start-time start
match end-time end

È stata definita la primitiva Geant che seleziona l'ifindex di Geant (permit). (Per trovarla snmpwalk milano-rtg.garr.net public IfAlias)
Di seguito si vuole selezionare una data di inizio di analisi ed una di fine.Si sono perciò definite le primitive start ed end. La start permette i flussi con data maggiore o uguale alle 16:00, la end invece permette i flussi con data inferiore alle 19:30. Infine la definizione ente_in matcha tutti i flussi che hanno come interfaccia di input quella definita in Geant e che inizino a start e finiscono al tempo end.

C.3 Implementazione

Nel documento è riportato lo schema a blocchi degli script utilizzati per generare i report su base giornaliera. Gli script sono nostri (spesso collezionando parti di codice distribuiti via mailing-list) (figura 5).

Ad esempio se selezioniamo Uni-Napoli Fed II, 2002, December,19 e nella colonna RU→RC selezioniamo Top IP Src (bytes)
Visualizziamo:

#Top IP Source ordered by bytes
— —- —- Report Information — — —
Title: Uni-Napoli - 2002/12/19
Fields: Percent Total
Symbols: Disabled
Sorting: Descending Field 2
Name: Source IP
Args: flow-stat -w -P -p -f9 -S2 -T Uni-Napoli - 2002/12/19
mode: streaming
compress: off
byte order: little
stream version: 3
export version: 5
#

# IPaddr	flows	octets	packets	duration
192.167.11.200	0.061	6.300	8.387	0.667
143.225.140.51	0.057	6.170	3.149	0.941
143.225.178.4	5.520	5.812	4.869	7.648
192.167.11.33	0.633	5.419	2.905	0.927
192.132.34.17	0.324	5.398	4.365	0.603
143.225.252.205	0.608	3.452	2.275	1.604
143.225.178.3	1.267	3.368	2.363	1.413
192.132.34.114	0.218	3.347	1.882	0.732
143.225.167.8	0.035	3.291	1.766	0.756
192.132.34.151	0.019	2.925	1.432	0.069
143.225.253.86	0.123	2.791	1.791	0.897
143.225.155.209	0.017	2.738	1.416	0.280

In pratica abbiamo ottenuto quali sono i principali IP che generano traffico in termini di quantità di bytes nella direzione RC→RU, il dato è espresso in termini percentuali sul traffico totale.
È notevole come i primi 12 IP della lista generino da soli il 50% del traffico per un ente delle dimensioni di Uni-Napoli in un tempo di osservazione di 24 h .

A questo punto dovrebbe essere chiaro come le possibilità siano davvero ampie di avere

reports di vario tipo.
Di seguito riportiamo quelle che dal punto di vista del NOC possono essere le informazioni di maggior utilità.

C.4 Esempi

* Lista dei top 10 talkers (in termini di bytes %)
Roma-RC 13/03/2003 ore 18-19
bash-2.05$ flow-cat ft-v05.2003-03-13.18* | flow-stat -p -P -f9 -S2| head -30

# IPaddr	flows	octets	packets
#			
193.206.8.3	29.912	22.700	27.796
193.206.195.129	5.410	12.556	8.459
193.43.65.134	9.276	9.850	9.362
193.43.65.133	8.701	8.567	4.272
193.204.199.2	0.180	6.815	1.928
193.204.111.2	3.060	4.600	4.088
193.206.158.127	4.200	3.066	3.888
192.107.86.249	0.028	2.709	2.761
192.107.80.28	2.945	2.590	1.201
193.206.158.125	0.024	2.185	0.601
193.204.90.41	0.785	1.750	0.842

* Lista delle top 10 dest ports (in termini di bytes %)
Roma-RC 13/03/2003 ore 18-19
bash-2.05$ flow-cat ft-v05.2003-03-13.18* | flow-stat -p -P -n -f5 -S2| head -30

# port	flows	octets	packets
#			
WinMx	0.551	16.831	13.241
31112	0.001	6.348	2.034
http	22.629	3.633	12.712
smtp	0.462	2.142	1.348
chargen	0.001	1.789	0.469
4767	0.003	1.607	0.633
2995	0.002	1.491	0.545
Kazaa	0.481	1.258	1.071
msg-icp	0.001	1.108	0.291
4293	0.003	0.827	0.310
snmp	0.493	0.585	2.742

* Lista delle top 10 source ports (in termini di bytes %)
Roma-RC 13/03/2003 ore 18-19
bash-2.05$ flow-cat ft-v05.2003-03-13.18* | flow-stat -p -P -n -f6 -S2| head -30

port	flows	octets	packets
http	47.886	46.441	39.748
49233	0.002	6.279	1.543
WinMx	0.223	3.452	3.301
4658	0.036	3.050	1.136
16699	0.013	2.894	2.606
4949	0.001	1.428	0.434
46620	0.001	0.547	0.188
4215	0.001	0.512	0.135
4210	0.002	0.502	0.132
4213	0.002	0.501	0.132
4194	0.001	0.500	0.131

* Lista dei protocolli(in termini di bytes %)
Roma-RC 13/03/2003 ore 18-19
bash-2.05$ flow-cat $ft - v05.2003 - 03 - 13.18*$ | flow-stat -p -P -n -f12 -S2

protocol	flows	octets	packets
tcp	80.987	97.532	93.022
udp	14.375	1.823	6.112
icmp	4.566	0.643	0.856
ipv6	0.026	0.001	0.003
gre	0.006	0.001	0.004
igp	0.020	0.000	0.001
pim	0.021	0.000	0.001

* Lista ip-ip(in termini di bytes %)
bash-2.05$ flow-cat $ft - v05.2003 - 03 - 13.1*$ | flow-stat -p -P -n -f10 -S3 |head -30
Roma-RC 13/03/2003 ore 10-19

src IPaddr	dst IPaddr	flows	octets	packets
#				
193.206.195.129	213.140.17.151	0.002	2.447	0.686
193.204.199.2	134.95.220.207	0.001	2.280	0.574
192.107.86.9	192.107.75.158	0.019	1.075	0.387
193.204.108.60	61.82.57.47	0.000	0.812	0.320
193.204.108.60	210.194.132.181	0.000	0.649	0.211
192.107.86.249	217.133.244.54	0.000	0.598	0.262
192.107.80.28	81.67.160.123	0.000	0.596	0.215
193.204.90.81	193.206.138.36	2.553	0.541	1.707
193.204.90.82	193.206.138.36	2.401	0.532	1.675
193.206.131.222	151.37.178.225	0.000	0.508	0.216
193.204.90.49	213.156.50.135	0.000	0.486	0.311

D FlowScan

Abbiamo visto che le funzionalità di NetFlow e di flow-tools risultano necessarie per quel che riguarda l'esportazione dei flussi da un router e la raccolta su di un garbage-collector. Una volta raccolte le informazioni, in formato raw, è necessario affrontare il problema dell'organizzazione di questi dati e della loro visualizzazione. FlowScan è composto da una collezione di moduli e script perl e funge da giunzione per un insieme di altri strumenti open-source quali un collection-engine, un database altamente performante ed un tool di visualizzazione. Una volta assemblati tutti i pacchetti, il sistema produce immagini grafiche adatte ad essere consultate via web. Queste provvedono una vista continua e real-time del traffico di rete cui si è interessati.

Il principio di funzionamento di FlowScan si fonda sull'analisi dei raw file raccolti da flow-tools, in particolare sulla ricerca all'interno di essi di parametri fondamentali che vengono impostati all'atto della configurazione e quindi sull'aggiornamento di contatori che riportano la quantità esatta di bytes di un'applicazione o di una particolare rete cui si è interessati. Il funzionamento di FlowScan è quindi abbastanza semplice, cerca all'interno dei raw-file le informazioni cui si è interessati e mantiene dei contatori che riflettono il contenuto dei flussi. Questa miriade di contatori viene riportata a RRDTool, un database a perdita d'informazione costruito appositamente per archiviare e visualizzare dati.

Le componenti di FlowScan sono diverse. La parte primaria è il programma denominato flowscan, è uno script perl che svolge la funzione di processo centrale all'interno del sistema. flowscan esegue i report-module scelti all'atto della configurazione. Questi report module derivano dalla classe FlowScan e sono definiti nel modulo perl FlowScan.pm. I report module sono tre: CampusIO, SubnetIO e CUFlow. CampusIO e SubNetIO sono simili, differiscono solo per quel che riguarda la complessità del sistema da gestire. SubNetIO permette infatti di monitorare Local Area Network più complicate rispetto a CampusIO e di dividere la base dati acquisita per reti che possono risultare piu' importanti rispetto ad altre. Nel nostro caso viene utilizzato il modulo CUFlow, che combina le caratterische di CampusIO e SubNetIO oltre a processare i dati in maniera molto piu' veloce rispetto a entrambi.

La seconda componente software è RRDtool [17] che viene sfruttato da FlowScan per collezionare i dati ordinati ed automaticamente aggregarli per medie temporali. I dati vengono collezionati in file *.rrd* che costituiscono un database con le informazioni essenziali sui flussi IP. RRD è un database di tipo round-robin, progettato per ospitare una quantità finita di dati che vengono riscritti con un processo circolare nel momento in cui tutto lo spazio allocato è stato occupato. Un'utility importante, compresa nel pacchetto FlowScan è flowdumper, che viene usata per esaminare il contenuto dei flussi manualmente. L'ultimo software necessario all'analisi è CUGrapher [18], compreso nella distribuzione di FlowScan e necessario per graficare il contenuto dei file *.rrd*. CUGrapher permette di differenziare il traffico per protocollo, servizio, TOS, router e rete e quindi di generare report su intervalli di 5 minuti o su periodi temporali più estesi. CUGrapher è un tool grafico cgi, che genera immagini real-time a partire dalle scelte dell'utente, attingendo ai dati forniti da CUFlow. L'implementazione di CUGrapher ovviamente presuppone l'installazione di un server web che consenta di visualizzare i report grafici.

D.1 Implementazione dell'architettura per il sito utente

La parte implementativa necessaria alla produzione dei grafici real-time è presentata di seguito; per una descrizione dettagliata dell'intera implementazione si rimanda a [27].

Dal router viene esportato il flusso verso il collector-box con i comandi riportati in tabella 11.

```
ROUTER(config)#ip flow-export source Loopback0
ROUTER(config)#ip flow-export version 5
ROUTER(config)#ip flow-export destination 193.xxx.xxx.71 8100
ROUTER(config)#ip flow-cache timeout active 1
```

Tabella 11: Comandi da configurare sul router per attivare l'esportazione verso il collector-box

Ovviamente è necessario abilitare il comando *ip route-cache flow* su tutte le interfacce del router. Inoltre è molto importante configurare il parametro di active-timeout a 1 minuto, per evitare che il router consideri lo stesso flusso più volte, causando spike anomali negli andamenti del traffico.

Con questa serie di comandi, il router esporta il contenuto della flow-cache verso l'indirizzo IP specificato. La macchina 193.xxx.xxx.71 fungerà quindi da collector-box.

Il collector-box dev'essere quindi abilitato alla raccolta dei flussi di traffico ed alla loro elaborazione. Come già più volte accennato il software necessario e' flow-tools. All'interno del pacchetto di flow-tools, il programma adatto alla cattura dei flussi è flow-capture. In tabella 12 viene riportato un esempio di come si possa eseguire correttamente flow-capture per collezionare i flussi di traffico sul collector-box.

```
[netflow@pc bin]$ ./flow-capture -z5 -V5 -n287 -N0
-w/usr/local/netflow/collector/Router -E2G 0/193.xxx.xxx.71/8771
```

Tabella 12: Tutto il traffico in transito sul router viene collezionato sul collector-box nella directory /usr/local/netflow/collector/Router

Ecco le diverse opzioni di flow-capture: -z5 si riferisce al fattore di compressione dei dati (secondo l'algoritmo proprio di zlib), -V5 è la versione di NetFlow implementata sul router, -n287 corrisponde al numero di file giornalieri che si vogliono creare (287 corrisponde ad un file ogni 5 minuti), -N0 fa in modo che i file vengano conservati tutti all'interno della stessa directory; -w è la directory in cui i file vengono collezionati, -E2G indica lo spazio disco messo a disposizione dal sistema (oltre i 2 GigaBytes i dati verranno riscritti). L'ultima parte della riga di comando si riferisce all'indirizzo IP destinazione (0 indica il localhost), l'IP sorgente e la porta di servizio su cui viene aperta la socket.

Una volta collezionati i file, contenenti i flussi di traffico, rimane solamente da runnare FlowScan. La parte di installazione è spiegata dettagliatamente al link [28].

La configurazione è molto semplice e si riferisce solamente a due file; i file di configurazione si trovano nella directory /var/local/flows/bin/ e sono flowscan.cf e CUFlow.cf .

Nel file flowscan.cf vengono passate a FlowScan le informazioni riguardanti la directory in cui trovare i raw file acquisiti con flow-tools e il report module che si vuole implementare (nel nostro caso CUFlow) oltre a due parametri opzionali di importanza marginale.

Nel file CUFlow.cf risiedono le informazioni

- Subnet, sono le reti di cui si vuole monitorare il traffico, tramite le quali il sistema rivela la direzione di propagazione di un pacchetto
- Network, sono le reti cui si particolarmente interessati, e di cui si vogliono avere dei file .rrd dedicati
- Output Dir, è la directory in cui si vogliono mettere i file .rrd
- Scoreboard n, directory e file necessari al calcolo dei topten talkers ad intervalli di 5 minuti
- AggregateScore n, directory e file necessari al calcolo dei topten talkers su tutto l'arco temporale
- Router, è il router da cui si esportano i flussi
- Services, sono i servizi cui si è interessati
- Protocols, sono i protocolli cui si è interessati

A questo punto è possibile runnare flowscan (lo script eseguibile si trova nella stessa directory dei file di configurazione), per visualizzare i grafici è sufficiente copiare il file CUGrapher.pl nella directory cgi-bin di apache */usr/local/apache/cgi-bin/* ed editarlo nei seguenti campi:

- $rrddir (directory in cui ci sono i file .rrd)
- $organization (nome dell'utente di cui sia sta monitarando il traffico; è solo una label che farà da titolo nella pagina web).

Una volta configurato correttamente il sistema, è possibile vedere i grafici alla url http://127.0.0.1/cgi-bin/CUGrapher.pl

D.2 Implementazione dell'architettura sui router di backbone

Come estrapolare le informazioni di traffico dei singoli utenti da un flusso NetFlow? Il traffico esportato dai router con NetFlow comprende ovviamente tutti i flussi che transitano sul router stesso. Vediamo com'è possibile filtrare solo le informazioni riferite ad una particolare rete di indirizzi per poi gestirle in maniera indipendente.

L'implementazione di NetFlow sui router è identica alla parte descritta precedentemente, è invece radicalmente diversa la parte di collezione dati. Il concetto fondamentale riguarda la moltiplicazione del flusso proveniente dal router per il numero di utenti che si vogliono monitorare; una volta creati n flussi identici si può lavorare distintamente su ognuno di essi per estrapolare solo le informazioni cui si è interessati.

La duplicazione dei flussi viene fatta sul collector-box, grazie a flow-fanout (una delle tante parti di flow-tools). Ammettiamo che l'esportazione dei flussi sia sempre quella di tabella 11. In questo caso sul collector-box è corretto runnare una riga di comando simile a quella in tabella 13

```
[netflow@pc bin]$ ./flow-fanout 0/193.xxx.xxx.126/8100
0/193.xxx.xxx.72/8777 0/193.xxx.xxx.72/8778 0/193.xxx.xxx.72/8779
```

Tabella 13: Il collector-box riceve il flusso proveniente dal router sulla porta 8100, quindi replica ed esporta tre istanze dello stesso flusso verso la macchina di analisi (sulle porte 8777, 8778, 8779

Vediamo nel dettaglio cosa significa la riga di comando in tabella 13, con l'aiuto della figura 10.

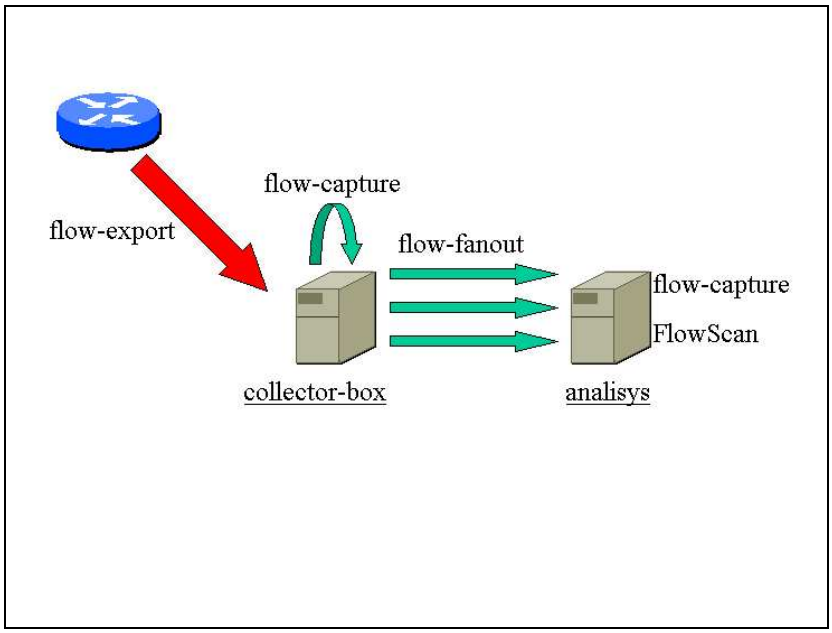


Figura 10: Il flusso derivante dal router di backbone viene moltiplicato in più istanze dal collector-box, che vengono inviate alla macchina di analisi

Il flusso ricevuto da 193.xxx.xxx.126 (Router) sulla porta 8100 viene replicato 3 volte alla macchina di analisi (193.xxx.xxx.72) su porte diverse (8777, 8778, 8779). Quindi la macchina di analisi riceve tre istanze identiche dello stesso flusso su porte diverse, provvederà quindi a filtrare ogni flusso per estrarre solo le informazioni riferite ad un preciso aggregato di reti. In questo modo è possibile estrarre le informazioni riferite ai singoli utenti. I comandi necessari alla cattura e al filtraggio sono riportati di sotto.

Analizziamo in dettaglio solo una singola riga.

Il flusso ricevuto sulla porta 8778, viene filtrato (-fUniv2 -Ffiltro) e collezionato nella directory /usr/local/netflow/collector/Univ2. Il filtraggio avviene con l'utilizzo del file Univ2 che viene riportatoin tabella 15. -Ftest si riferisce alla parte del file detta filter-definition in cui si specifica l'informazione che si vuole estrapolare (ip sorgente o destinazione appartenente alle reti specificate nella prima parte del file).

```
[netflow@pc bin]$ ./flow-capture -z5 -V5 -n287 -N0 -fUniv1 -Ffiltro
-w/usr/local/netflow/collector/Univ1 -E2G 0/193.xxx.xxx.71/8777
[netflow@pc bin]$ ./flow-capture -z5 -V5 -n287 -N0 -fUniv2 -Ffiltro
-w/usr/local/netflow/collector/Univ2 -E2G 0/193.xxx.xxx.71/8778
[netflow@pc bin]$ ./flow-capture -z5 -V5 -n287 -N0 -fUniv3 -Ffiltro
-w/usr/local/netflow/collector/Univ3 -E2G 0/193.xxx.xxx.71/8779
```

Tabella 14: Le istanze provenienti dal collector-box vengono filtrate per essere poi processate da FlowScan

```
[netflow@pc bin]$ less Univ2

filter-primitive rete
type ip-address-mask
permit 193.xxx.xxx.0 255.255.255.0
permit 193.yyy.yyy.0 255.255.248.0

filter-definition test
match ip-source-address rete
or
match ip-destination-address rete
```

Tabella 15: Esempio di file filtro, che permette di estrapolare da un flusso solo le informazioni riguardanti le reti menzionate

Quindi nella directory /usr/local/netflow/collector/Univ2, verranno collezionati i file di flow-tools contenenti solo le informazioni sul traffico delle reti riportate in tabella 15. A questo punto rimane semplicemente da runnare FlowScan su ciascuna delle directory in cui vengono messi i file dei singoli utenti.

E Traffico Peer to Peer

Il traffico Peer2Peer sta diventando negli ultimi tempi argomento di forte riflessione da parte degli amministratori di rete. Dall'avvento di Napster (che di fatto non era un vero software p2p) in poi la comunità Internet si è dedicata all'utilizzo di software che permettessero la condivisione di file tra milioni di utenti. La novità principale di questi software è l'abilità di ricercare nella comunità p2p file come se fossero contenuti in un enorme database centralizzato, quando invece i contenuti sono archiviati solamente nelle macchine degli utenti. Per ogni dettaglio in merito consigliamo di partire dal sito di O'Reilly [15]. Al di là di considerazioni legate all'uso legale o illegale che se ne possa fare, sicuramente quello che affligge oggi gli amministratori di rete è lo scambio di file coperti da copyright, in genere file mp3 (musicali) e ultimamente file avi (video). Al momento, pi che il carattere legale del problema (che deve ancora assumere contorni ben definiti), ciò che impatta direttamente nell'utilizzo della rete è l'occupazione di banda che cresce in modo esponenziale se il software p2p non è settato in modo fair dall'utente. Un utilizzo indiscriminato di queste applicazioni compromette la disponibilità di banda per gli altri utenti e inficia la qualità di servizio non solo di semplici CDN a 2Mbps ma anche di linee a grande capacità di banda. Va da se che i profili di traffico dell'intera rete che fino a poco più di un anno fa evidenziavano un comportamento giorno-notte di tipo sinusoidale, per enti quali le università molto evidente e abbastanza regolare, oggi sono poco definibili. Un host che abbia al suo interno anche un solo film avi, con attivo il software p2p anche la notte, provoca un consistente traffico verso quesi paesi, quali gli USA ad esempio, dove è ancora giorno. Vediamo ad esempio che il traffico GARR ↔ GEANT (solo reti della ricerca) è ancora abbastanza monte-valle (figura 11).

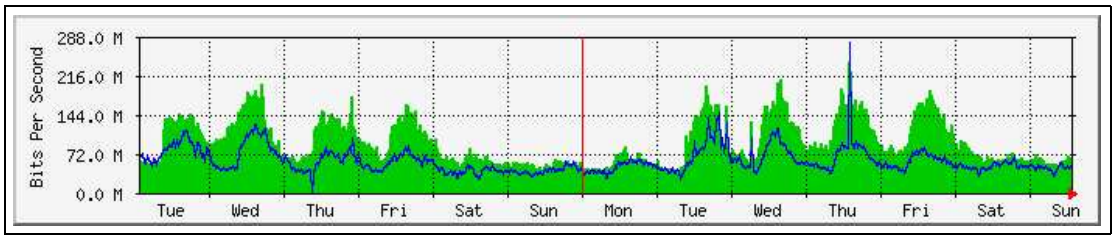


Figura 11: Andamento del traffico mensile sul link di commodity ricerca internazionale di Geant

Quello con Global-Crossing (commodity internazionale) evidenzia una sproporzione di traffico GARR ↔ GX (linea blu) rispetto alla direzione opposta (linea verde) nelle ore notturne (figura 12).

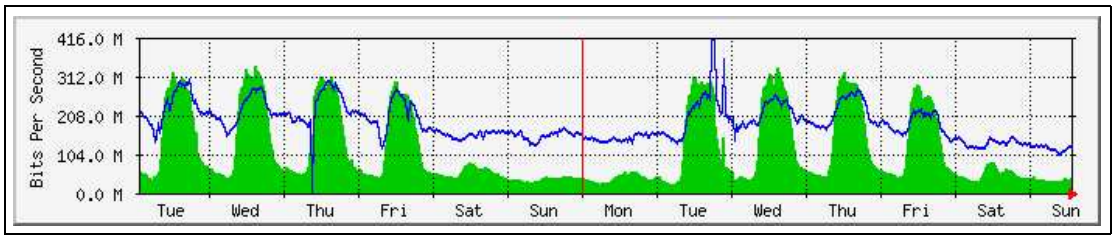


Figura 12: Andamento del traffico mensile sul link di commodity internazionale di GlobalCrossing

E.1 P2P computing

Internet è il più grande sistema di calcolo decentralizzato del mondo. Lo è dalla nascita, benche' negli anni la rete e i suoi fruitori siano mutati profondamente. In particolare all'inizio degli anni '90 sempre più sistemi in rete erano completamente centralizzati. L'avvento del world wide web (www) è stato il principale artefice di questa tendenza: web server centralizzati concentrati in pochi o addirittura unici siti. Vediamo un pò di topologie possibili di reti p2p, prendendo ad esempio le applicazioni più diffuse nella rete. A farla da padrone sono due software, entrambi per Windows, che sono Kazaa e WinMx, seguiti da Gnutella, eDonkey2000, BearShare (Linux) [8], [10], [12]. In particolare il sito dell'Università di Chicago [10] mostra una possibile policy da utilizzare con i propri utenti. Napster: l'esplosione delle applicazioni p2p è decisamente decollata con l'avvento di Napster. Scaricando il software dal server di Napster, si poteva accedere alla comunità Napster condividendo i file desiderati. La topologia di Napster (vedi figura 13), tuttavia, così come quella di SETI@Home, si basava su una query ad un motore di ricerca centralizzato che aggiornava in real time le utenze presenti nella comunità e gli header dei file da condividere(mp3 nel 99% dei casi) . Inviata la richiesta al server questi inviava al client le info sugli utenti in possesso delle informazioni ricercate e da qui in poi il processo di download e/o upload del file diventava indipendente dal server dando origine ad una comunicazione p2p tra gli host.

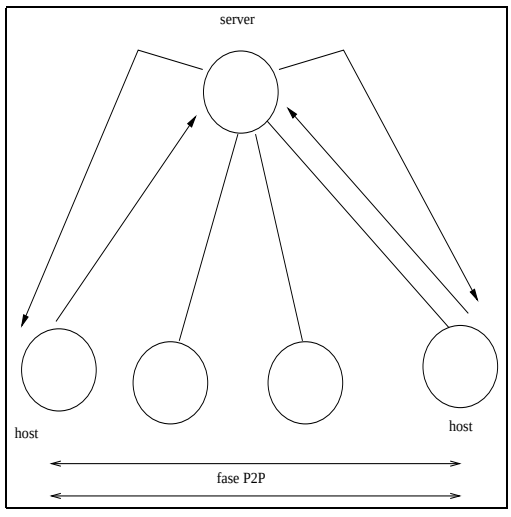


Figura 13: Topologia del protocollo Napster: ricerca centralizzata e scambio peer2peer

Gnutella, sviluppato per il mondo Unix nel marzo 2000 da parte della Nullsoft (la stessa di Winamp), rappresenta la pietra miliare nella diffusione di topologie di rete decentralizzate. Prima di essere un software, Gnutella è un vero e proprio protocollo per reti p2p. Il modo di funzionamento di Gnutella è abbastanza semplice nella sostanza (vedi figura 14). Conosciuto l'IP del primo peer1 (che potendo essere all'occorrenza sia client che server viene detto servant) viene a questi mandato un ping (in genere sulla porta 6346); il servant risponde (pong) inviando le informazioni su se stesso (ip address, porta, numero di file da sharare) e sui nodi di cui è a conoscenza; contestualmente invia a questi ultimi le informazioni relative al nodo che lo ha contattato. Dunque il processo di query avviene tramite la diffusione del messaggio attraverso tutti i peer che a loro volta li inviano ad altri peer... il tutto si ferma allo scadere del ttl (fissato a 7 dal protocollo). Considerando che sono previste fino a 25 connessioni attive contemporaneamente su un servant, la rete potrebbe essere inondata da 25^7 pacchetti per una singola query (per fortuna questo solitamente non avviene). Per approfondimenti rimandiamo a [8], [10], [12].

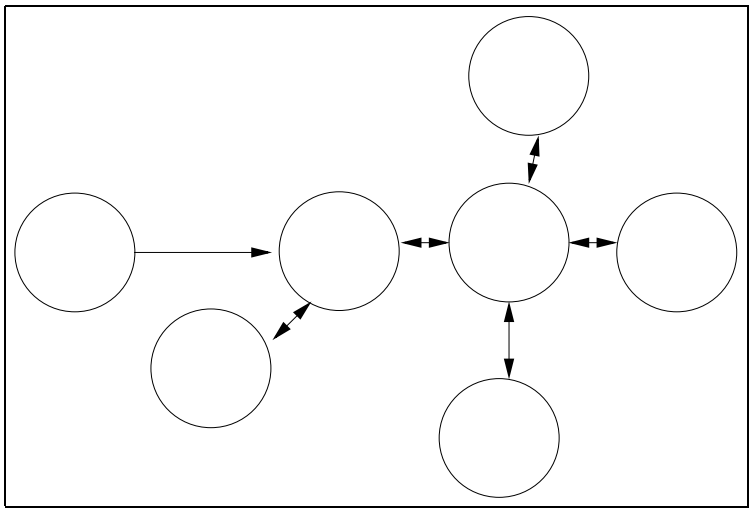


Figura 14: Topologia del protocollo Gnutella: rete decentralizzata

Da tempo KaZaA è il software preferito da milioni di utenti nella rete a causa della sua facile installazione e dell'efficiente prestazione nella ricerca di quanto desiderato. Il funzionamento di KaZaA può essere visto come un ibrido tra Napster e Gnutella. La query non viene fatta ad un server centrale (vedi figura 15), nè in flooding ai propri peer, ma ad alcuni hosts che fungono da supernodi. Ogni utente con un collegamento veloce può agire da supernodo (volendo può disabilitare questa opzione). Il supernodo tiene traccia dei suoi host e di altri supernodi, così invia la query ai suoi host ed ai supernodi da lui conosciuti che a loro volta ripetono l'operazione. La ridondanza delle query è dunque limitata rispetto a Gnutella.

E.2 Come riconoscere il traffico p2p

Fin qui abbiamo descritto i principali protocolli di file-sharing senza entrare nel dettaglio degli header scambiati tra host in una sessione tcp/ip. Per discernere il traffico p2p con Netflow la cosa pi semplice è selezionare le well-known-port utilizzate in modo diverso

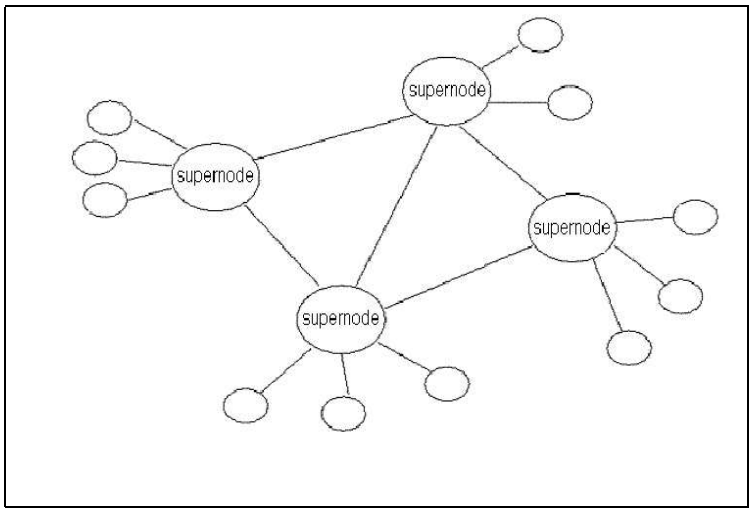


Figura 15: Topologia del protocollo Kazaa: situazione ibrida

dai vari software. Questo sistema ci dà un'informazione sulla quantità di traffico di file-sharing molto approssimativa e sicuramente difettosa (spiegheremo tra poco il perche'). In tabella 16 sono riportate quelle che attualmente sono le porte più utilizzate:

Kazaa	1214	tcp,udp
Gnutella	6345-6347	tcp
eDonkey2000	4661-4665	tcp,udp
WinMx	6257, 6699	tcp

Tabella 16: Porte e protocolli utilizzati dai più comuni software di file-sharing

Mentre possiamo affermare che il 99% dei flussi tra le porte riportate in tabella rappresenta traffico di file-sharing, non possiamo dire che sia sufficiente la ricerca di queste porte sulla rete per l'individuazione della totalità del file-sharing. Nel caso di WinMx o di eDonkey2000 una consistente quantità del traffico viene vista attraverso le porte segnalate, per la loro natura centralizzata. Per Gnutella si possono individuare tramite le porte 6345-6347 soprattutto i ping, cioè le connessioni di un nuovo host ai servant già presenti sulla gnutellanet; il download/upload dei file può avvenire invece, dopo una negoziazione tra i peer, su porte differenti. Kazaa ha un comportamento ibrido; gli utenti che utilizzano le versioni più vecchie sono legate alla porta 1214 sia in fase di discovery che in quella di file-sharing. Le versioni di Kazaa più aggiornate tendono ad utilizzare la 1214 solo per le query ai supernode, mentre la porta su cui aprire le connessioni con i peer è definibile dall'utente. La porta 80 sta assumendo un ruolo preponderante dato che tale servizio (http) non può, per ovvi motivi, essere filtrato dagli amministratori di rete.

L'esempio in tabella 17 è l'estrapolazione di alcuni flussi da Napoli-RC sulla porta 1214.

A parte i primi tre flussi che hanno dimensioni del MByte, gli altri sono molto piccoli in termini di dimensione. Se invece filtriamo i flussi IP-IP in base alla dimensione dei dati scambiati, otteniamo il risultato di tabella 18.

Abbiamo sottolineato le porte corrispondenti a quelle indicate per software p2p; si

srcIP	dstIP	prot	srcPort	dstPort	octets	packets
80.116.129.234	140.164.10.99	6	1214	2008	4483698	3001
193.205.113.27	172.189.65.149	6	2391	1214	1174027	864
193.205.118.57	195.137.100.120	6	3348	1214	2096904	1401
213.213.6.178	193.205.63.2	6	1214	80	525	5
193.205.105.125	10.1.140.248	6	3306	1214	144	3
192.87.198.110	193.206.114.45	17	1214	1702	35	1
80.222.95.189	194.119.194.105	17	1214	1259	63	1
194.119.194.105	80.222.95.189	17	1259	1214	35	1
194.119.194.122	62.16.198.237	17	1089	1214	35	1
62.11.109.190	140.164.56.124	6	1843	1214	48	1
140.164.11.128	80.116.191.21	6	3861	1214	144	3
193.205.63.209	130.161.179.143	17	3951	1214	189	3
193.205.63.209	80.181.164.227	17	3951	1214	140	4

Tabella 17: Esempio di flussi in transito sul router concentratore di Napoli sulla porta 1214

srcIP	dstIP	prot	srcPort	dstPort	octets	packets
62.211.4.97	193.206.103.134	6	6699	1224	1790214	2319
208.63.206.223	140.164.56.203	6	21028	9404	1587003	2974
213.45.22.5	193.206.103.134	6	6699	1217	1302683	1655
62.211.175.127	193.206.115.195	6	3382	4662	2083615	1645
62.195.50.171	140.164.41.95	6	2921	6699	1469964	2027
217.41.35.35	140.164.14.30	6	80	59410	2113198	1586
140.164.10.99	80.200.43.162	6	1214	2038	1011325	691
151.30.146.30	193.206.103.134	6	6699	1232	1090632	1971
66.250.223.51	193.206.101.2	6	80	1968	1873391	1253
193.206.122.208	193.206.115.93	6	16005	1307	2312112	2464
62.211.77.168	193.205.107.176	17	2682	4390	3876168	2718
151.29.203.102	194.119.194.7	6	64162	6699	6763996	6256

Tabella 18: Coppie IPsrc-IPdst che fanno più traffico (in termini di bytes scambiati); in rosso è evidenziata la porta 80, qui non usata per sessioni http

vede che quanto affermato su WinMx (6699) e eDonkey2000 (4662) viene confermato. In tab.18 è stata evidenziata la porta 80 (in rosso); agli indirizzi in questione non ci risultano server web attivi; questo può essere un segnale che il suo utilizzo non è http. Infine ecco un esempio di come si possa vedere chi più probabilmente sta scaricando media-file. I flussi in tabella 19 sono stati estrapolati in base all'ammontare di bytes (nell'esempio si sono scelti quelli superiori a 50MB).

srcIP	dstIP	prot	srcPort	dstPort	octets	packets
193.206.114.100	151.97.52.127	6	20078	1136	84333532	56226
193.206.114.100	151.97.52.127	6	20078	1136	89652412	59771
193.206.114.100	151.97.52.127	6	20078	1136	111304636	74206
193.206.114.100	151.97.52.127	6	20078	1136	71553620	47707
137.78.99.30	140.164.50.190	6	80	3549	59652486	39869
212.187.169.138	193.205.102.233	6	80	1239	70269517	46850
212.187.169.138	193.205.102.233	6	80	1239	76637644	51096
212.160.105.99	193.206.108.6	6	56383	61869	50053620	33372

Tabella 19: Flussi superiori a 50 MB, sospetto di download/upload di media-file

Molti APM ci chiedono cosa fare o come fermare il traffico non istituzionale. Qualunque sia la scelta (sia che si voglia limitarlo, sia che si voglia dropparlo) la questione che abbiamo posto è come identificarlo. Abbiamo visto che bloccare le porte o fare delle code con bassa priorità per il traffico da o verso tali porte non sempre è una soluzione efficace; comunque è di veloce applicazione e in casi di banda satura è la prima cosa da fare. Per quanto riguardano le situazioni mimetiche, consigliamo di creare dei triggers sui flussi o sui bytes tra una coppia di IP in modo che si possa per ogni singolo caso fare un'indagine più approfondita. L'ultima tabella analizzata ne è un esempio. Comunque per avere altre idee in merito, ad esempio sui servers centrali di WinMx, o altro, sono molto utili i riferimenti bibliografici [9] e [10].

Riferimenti bibliografici

[1] http://entropy.brni-jhu.org/linuxsetup34.html; Spiegazione di configurazione di cflowd e flowscan

[2] http://www.splintered.net/sw/flow-tools/; Pagina principale di Flow-Tools con presentazioni, software, puntatori

[3] http://www.switch.ch/tf-tant/floma/software.html; Puntatori a netflow e tool vari(by S.Leinen)

[4] http://www.switch.ch/tf-tant/floma/references.html#ipfix Puntatori a Ipfix (IP Flow Information eXport) (by S.Leinen) P.Barford,D.Plonka, Characteristics of network traffic flow anomalies, Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement Workshop 2001 , San Francisco, California, USA

[5] http://www.scd.ucar.edu/nets/docs/reports/trips/2002/jc-200201-i2-jt-meeting-trip-report.htm#flowtools, Tutorial di Flow-Tools by M.Fullmer

[6] http://www.caida.org , sito di partenza di molti tools di monitoring, tra cui Cflowd

[7] http://net.doit.wisc.edu/ plonka/FlowScan/ , sito di Flowscan

[8] http://www.oofle.com/iptables/filesharing.html , tecniche di controllo del traffico P2P

[9] http://www.ja.net/CERT/JANET-CERT/prevention/peer-to-peer.html , lista di applicazioni e well-Known ports per P2P

[10] http://security.uchicago.edu/peer-to-peer/no_fileshare.shtml , come bloccare alcune applicazioni P2P , e posizione in merito da parte dell'università di Chicago

[11] http://www.canarie.ca/canet4/monitoring , sito della rete della ricerca Canadese

[12] http://cnscenter.future.co.kr/hot-topic/p2p.html, sito di partenza con tantissimi puntatori al mondo P2P

[13] http://www.linuxgeek.org/netflow-howto.php , How to build detailed Network Usage Reports using RRDTool, flow-tools, FlowScan, and CUFlow

[14] http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html , puntatori aggiornati a software di monitoring

[15] http://openp2p.com/ , punto di partenza per il mondo peer to peer, a cura di O'reilly (leggi, tutorial, info tecniche, filosofia)

[16] http://www.sans.org/rr/threats/gnutella.php , overview del protocollo gnutella

[17] http://www.rrdtool.org/ RRDTool home page

[18] http://www.columbia.edu/acis/networks/advanced/CUFlow/ CUFlow home page.

[19] http://net.doit.wisc.edu/ plonka/list/flowscan/ FlowScan mailing list home page.

[20] http://wwwstats.net.wisc.edu Esempi di grafici ottenuti con FlowScan e CUFlow

[21] http://flows.ikano.com Esempi di grafici ottenuti con FlowScan e CUFlow

[22] https://www1.columbia.edu/sec/bboard/mj/cuflow-users/ CUFlow mailing list archive. The mailing list is cuflow-users@columbia.edu

[23] http://people.ee.ethz.ch/ oetiker/webtools/rrdtool/mailinglists.html Mailing-list di RRDtool

[24] http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm Servizi e Applicazioni NetFlow

[25] http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/ntfo_wp.htm Analisi performance NetFlow

[26] http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.pdf Tutorial CISCO NetFlow

[27] http://www.linuxgeek.org/netflow-howto.php; How to build detailed Network Usage Reports using RRDTool, flow-tools, FlowScan, and CUFlow; Robert S. Galloway 2003

[28] http://net.doit.wisc.edu/ plonka/lisa/FlowScan; 2000 Dave Plonka - FlowScan: A Network Traffic Flow Reporting and Visualization Tool

[29] http://www.ietf.org/html.charters/ipfix-charter.html; 2003 IP Flow Information Export (ipfix)

[30] http://www.slac.stanford.edu/comp/net/wan-mon/passive-vs-active.html 2001; Les Cottrell - Passive vs. Active Monitoring

[31] http://www.splintered.net/sw/flow-tools/papers/osu-flow-tools.pdf; Mark Fullmer, Steve Roaming