ARCHITETTURA DISTRIBUITA PER IL COLLEZIONAMENTO TRAMITE SNMP DI DATI PER IL BILLING DI UNA RETE GEOGRAFICA: IL CASO SEABONE

CORSO DI SISTEMI PER L'ELABORAZIONE DELL'INFORMAZIONE:
GESTIONE DI RETE

Prof. Luca Deri

AA 2001/2002

Andrea Manzi <manzi@cli.di.unipi.it>
Marco Pasquali <pasquali@cli.di.unipi.it>

Sommario

1. Introduzione

Una delle aree di management che risente maggiormente dei cambiamenti in atto in questi ultimi tempi è sicuramente quella dell'accounting. Con questo termine intendiamo tutta la serie di informazioni che riguardano:

- il volume di traffico prodotto;
- l'allocazione e il monitoraggio della banda;
- la ripartizione dei costi fra gli utenti (*Billing*).

Fra i dati di accounting la nostra attenzione e` rivolta in particolare al monitoraggio dei dati necessari per la ripartizione dei costi fra i vari utenti della rete, appunto il billing.
I metodi finora usati per far pagare ai consumatori il loro utilizzo della rete stanno cambiando. Con la connessione a banda larga è normale avere un canone fisso per ottenere il servizio 24 ore su 24, ma l'orientamento del mercato è rivolto a controllare l'effettivo traffico prodotto dall'utente per poi applicare tariffe proporzionate.
Anche i tipi di servizi offerti dagli Internet Service Provider (ISP) stanno evolvendo. Mentre oggi la maggior parte del traffico internet è best-effort, con l'introduzione di servizi differenziati ogni tipo di richiesta sarà puntualmente garantita. Ovviamente dato che le applicazioni sfruttano in maniera diversa la rete (una videoconferenza utilizza molte più risorse rispetto al web browser), i clienti dovranno essere tassati in base ai servizi che richiedono. Tali servizi vengono raggruppati in classi che prendono il nome di classi di QoS (Qualità del servizio).
In un futuro non troppo lontano potremmo quindi connetterci tutto il giorno ad internet ed essere tassati non sul tempo di connessione, ma sull'effettivo volume di traffico prodotto.
In questo scenario è quindi essenziale proporre soluzioni che permettano il corretto collezionamento dei dati per il billing.
Il nostro documento è rivolto ad analizzare un sistema per la raccolta dei dati sopradescritti sulle reti portanti a cui si connettono i vari ISP, le cosiddette reti geografiche.
La struttura di qualsiasi rete geografica e' determinata da un numero elevato di router e switch. Proprio per la presenza di tutti questi dispositivi, un operazione di gestione che, effettuata su reti di piccole/medie dimensioni non desta grosse preoccupazioni, risulta essere difficoltosa su reti geografiche.
Per avere un'idea delle dimensioni di una rete geografica, e della difficoltà di gestione che ne deriva, prendiamo in considerazione la rete Seabone.

1.1 La Rete SEABONE

La rete geografica Seabone garantisce, grazie ai suoi numerosi punti di accesso, la connettività fra i vari paesi europei e un collegamento con gli Stati Uniti della velocità di 3.1 Gbit/s (Fig.1)

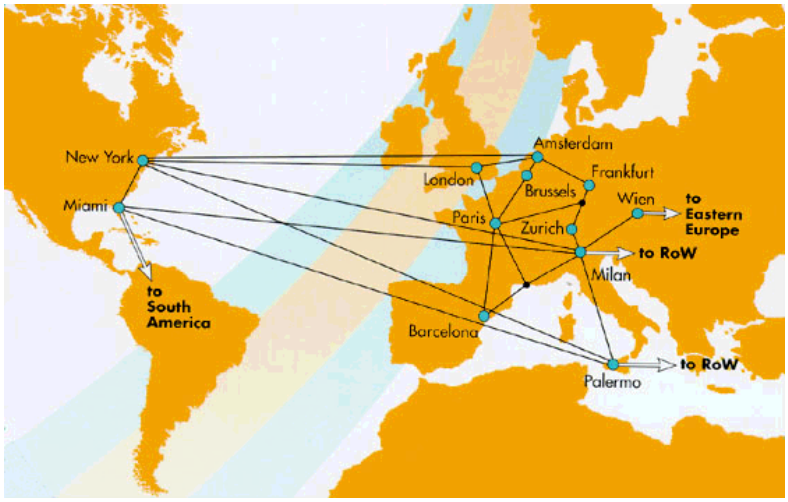

Fig. 1 Rappresentazione della rete SEABONE

Gli Internet Service Provider e le società di telecomunicazione, acquistando l'accesso ai vari POP di Seabone, sono certi di un servizio efficiente grazie ad alcune sue caratteristiche peculiari:

- Performance di rete elevate;
- Connettivita' con il mondo Internet;
- Copertura geografica;
- Servizi opzionali.

Un apparato cosi' imponente, lungo tutta la rete vengono infatti impiegati circa 12000 router e switch oltre ai migliaia di chilometri di fibra ottica, e' un ottimo punto di partenza per la presentazione del problema che vogliamo risolvere.

1.2 Scopo del documento

Lo scopo del documento e` quello di definire un'architettura distribuita per il collezionamento dei dati per il billing, via SNMP, di una rete geografica come Seabone.
Premesso che i dati che intendiamo analizzare vengono raccolti dai vari dispositivi dislocati nella rete, il nostro compito consiste nella definizione di un metodo per inviare questi dati alla console di management, mantenendo come prerogativa la volontà di non degradare, con il nostro traffico, la performance della rete.
Abbiamo scelto di seguire la strada di un'architettura distribuita, perché un management centralizzato e' fortemente inefficiente nel caso di una rete geografica. Questo perché, se da una console di management si riescono ugualmente a collezionare i dati di accounting raccolti da tutti i dispositivi, la granularita' delle informazioni ricevute e` talmente ampia da non permetterne un'analisi accurata. Ad esempio, dovendo leggere dati da una rete come

Seabone, l'intervallo di tempo tra una misurazione e l'altra e` proporzionale al numero dei dispositivi (12000), perciò inammissibile per una gestione efficiente della rete.

2. Stato dell'Arte

L'architettura da noi progettata è caratterizzata da alcune funzionalità principali:

1. Misurazione del traffico Internet;
2. Conversione del traffico misurato in informazioni sugli utenti;
3. Elaborazione dei dati;
4. Memorizzazione dei dati.

Alla base della raccolta dei dati per il billing vi sono le misurazioni del traffico internet. Queste consistono in rilevamenti eseguiti sulle informazioni in transito sulla rete e descrivono l'utilizzo della rete stessa in funzione del tempo. Tali dati sono degli indici globali di rete, ciò significa che non sono specifici per il billing, ma riguardano molteplici aspetti della rete, di conseguenza devono essere convertiti per poter ricavare le informazioni desiderate (identificazione dell'utente, traffico generato da un determinato indirizzo, ecc.). I dati, una volta convertiti, possono essere utilizzati per l'elaborazione da parte di strumenti di billing (generalmente software che distribuisce i costi fra gli utenti in base a determinati parametri). Inoltre e` necessario memorizzare le informazioni, ricavate a seguito delle varie elaborazioni, in un archivio permanente in modo da mantenere una memoria storica sull'utilizzo della rete.

In questo capitolo vengono descritti gli strumenti esistenti e, per certi aspetti, indicati per la progettazione di un architettura che disponga di tali funzionalità.

2.1 SNMP

Il Simple Network Management Protocol (SNMP) è il protocollo di management più usato al momento. Il modello di rete SNMP e' caratterizzato da:

- Agent: Entità che si trovano dentro o molto vicino al sistema che deve essere monitorato. L'Agent deve monitorare la risorsa e salvare le informazioni più importanti riguardo i managed objects (astrazione delle caratteristiche delle risorse reali).
- Manager: legge le informazioni dagli agent e gestisce il verificarsi di eventi anomali che gli vengono segnalati dagli agent stessi. Un manager di solito processa i dati e può prendere decisioni in merito il tipo di informazione ricevuta.
- MIB (Management Information Base): una sorta di database dove vengono salvate le informazioni riguardo i managed objects. Il manager può richiedere o settare le instanze che sono nel MIB e che sono strutturate ad albero i cui differenti rami sono identificati da uno speciale numero (Object Identifier).

2.2 Management Distribuito

I nostri studi si sono orientati verso un'architettura che consenta di distribuire i compiti fra vari manager intermedi dislocati nella rete. Tali manager intermedi si trovano cosi` ad implementare sia le funzioni da manager, per le comunicazioni con i livelli sottostanti, sia le funzioni da agent in quanto è necessario che comunichino anche con i manager del livello superiore.
La piramide gerarchica dell'architettura distribuita è caratterizzata da:

1. Agent che implementano la raccolta dei dati;
2. Manager Intermedi, che leggono le informazioni dagli agent e devono comunicare con i loro superiori;
3. Console di Management che gestisce l'analisi dei dati ricevuti e l'instrumentazione dei dispositivi.

I vantaggi apportati da questo tipo di paradigma sono:

- Riduzione del traffico di management;
  grazie all'introduzione dei manager intermedi la maggior parte del traffico è confinata all'ambiente locale;
- Scalabilità dell'architettura di gestione;
  sia per il calcolo computazionale sia per la possibilità di ampliamento della rete.
- Bassa probalità di perdita di pacchetti di management;
  grazie alla breve distanza fra gli agent e i manager intermedi.
- Maggior robustezza dell'architettura;
  l'uso di molti manager intermedi limita il problema dei fallimenti. Se di fatto un manager intermedio, o addirittura la console di management, non funziona correttamente, le informazioni vengono archiviate in apposite strutture dei livelli sottostanti (MIB vedi Sez. 2.1).

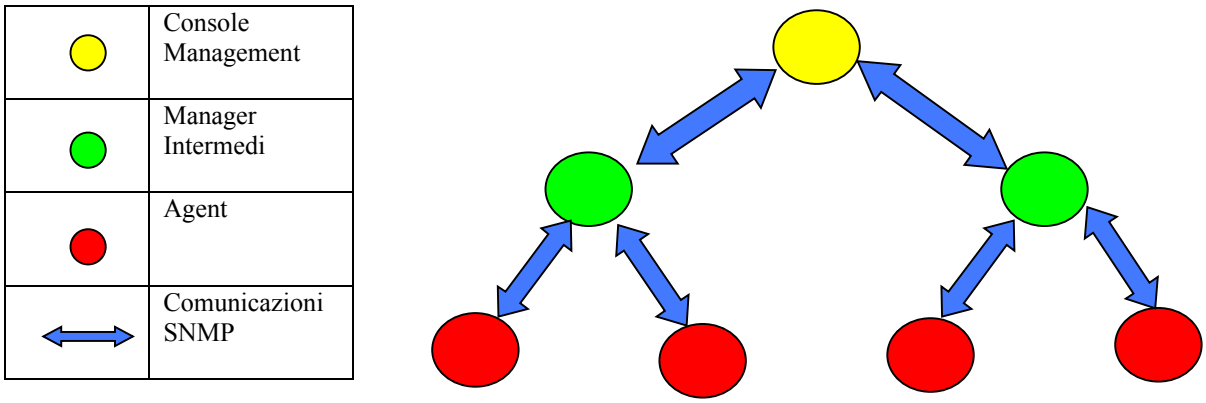

Fig. 2 Descrizione dell'architettura del sistema

2.3 Script-MIB

La distribuzione delle funzionalità di management avviene delegandole tra vari manager intermedi sparsi nella rete. In questo modo i servizi vengono avvicinati al luogo fisico dove sono necessari il che comporta meno lavoro per la Console di Management e soprattutto per la rete. Uno dei metodi per la distribuzione delle funzionalità di management è l'utilizzo e l'invocazione di script.
Lo Script-MIB è un particolare MIB che permette di caricare, memorizzare e fissare gli script in un ambiente sicuro. Il suo utilizzo e` necessario in quanto l'architettura da noi progettata (vedi Cap. 3) e` interamente basata sull'utilizzo di script.

Le sue caratteristiche principali sono:

- Creazione e aggiornamento veloce delle funzioni di management:
  - Sono implementate tramite script-software;
  - Gli script possono essere scritti in tutti i linguaggi di cui si definisce il supporto durante l'implementazione dello Script-MIB.

- Semplice distribuzione delle funzionalità di management. Gli script possono essere:
  - Inviati allo Script-MIB usando SNMP;
  - Presi da uno script repository (database che contiene tutti gli script necessari per il corretto funzionamento delle funzionalità di management) via http/ftp dallo stesso Script-MIB.

- Le funzioni di management vengono eseguite in un ambiente sicuro.
  - Lo Script-MIB viene controllato via SNMP;
  - Esistono speciali regole di accesso per modificare, aggiungere e cancellare gli script;
  - Le risorse del sistema sono protette contro accessi non autorizzati, che ne impediscono l'abuso.

- Permette l'attuazione di funzioni banali, ma ripetitive (settaggio dei parametri del MIB) in maniera molto semplice.

- Se implementa le funzioni da manager intermedio riduce il carico di dati scambiati sulla rete, processando molte informazioni di management nei livelli intermedi dell'architettura.

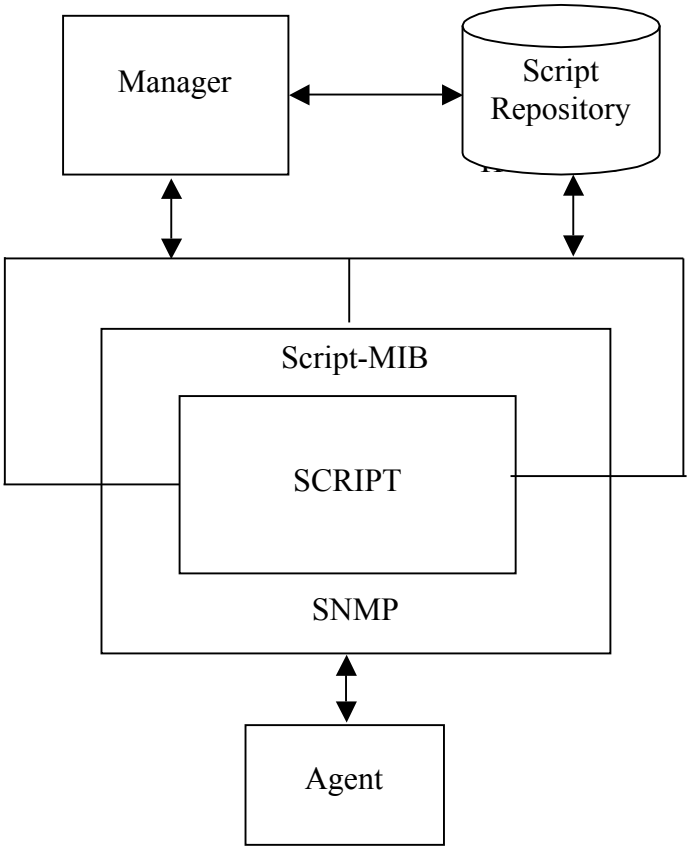

Fig. 3 Schema dello Script-Mib

2.4 NeTraMet

Il workgroup IETF RTFM (Real Time Flow Measurement) ha definito un architettura che consente di eseguire corrette ed efficienti misurazioni di flussi[1] sulla rete, composta da quattro parti:

1. Meter. Sono SNMP-Agent che consentono il monitoraggio dei flussi di traffico in transito sulla rete. I dati ricavati sono memorizzati in un MIB, nel caso specifico il Meter MIB.
2. Meter Reader. Questo sistema implementa la raccolta delle informazioni di traffico dal Meter MIB. Questi dati vengono processati o memorizzati in un database.
3. Manager. Configurano sia i meter che i meter reader, specificando i compiti che devono eseguire e i tempi.
4. Applicazioni di analisi. Effettuano l'analisi statistica, creano report e grafici sui dati, applicano le politiche per il billing, ecc..

In molte situazioni, il manager e l'applicazione di analisi sono raggruppati nello stesso sistema.

[1] RTFM definisce i flussi come il traffico generato in un intervallo di tempo da uno specifico componente della rete (in questo specifico caso i componenti della rete vanno intesi come: host, gruppi di host, porzioni di rete).

NeTraMet è un'implementazione Open Source (GPL) dell'architettura RTFM.
RTFM definisce un set di circa 40 attributi che possono essere imputati ad ogni flusso ed usati per la descrizione dei dati. I più importanti fra questi sono gli attributi dell'indirizzo, che specificano l'indirizzo ad un particolare livello della rete. Ad esempio per i pacchetti IP, il SourcePeerAddress e il DestPeerAddress definiscono i pacchetti da e verso indirizzi IP specifici.
Per configurare un RTFM meter, un amministratore deve prima creare un set di regole (*ruleset*) in modo da indicare:

- su quali flussi effettuare le misurazioni;
- qual è la sorgente dei flussi (sottorete, host);
- il livello di dettaglio delle misurazioni per ogni flusso.

Queste regole verranno applicate dal meter agli attributi di ogni pacchetto in transito.

Il nostro progetto si basa concettualmente sull'architettura definita dal NeTraMet. A tale architettura sono state però apportate delle modifiche sostanziali per conformarla alle esigenze di raccolta dei dati per il billing in reti geografiche come Seabone. Sono state aggiunte funzioni tipiche dello Script-MIB.
Di seguito verranno illustrate in dettaglio le scelte di progettazione che caratterizzano ogni componente della struttura.

3. Progettazione

La struttura dell'architettura distribuita da noi progettata e' caratterizzata da quattro livelli (Meter, Meter Reader, Accounting Manager, Console di Management). Viene di seguito descritta in ogni suo componente.

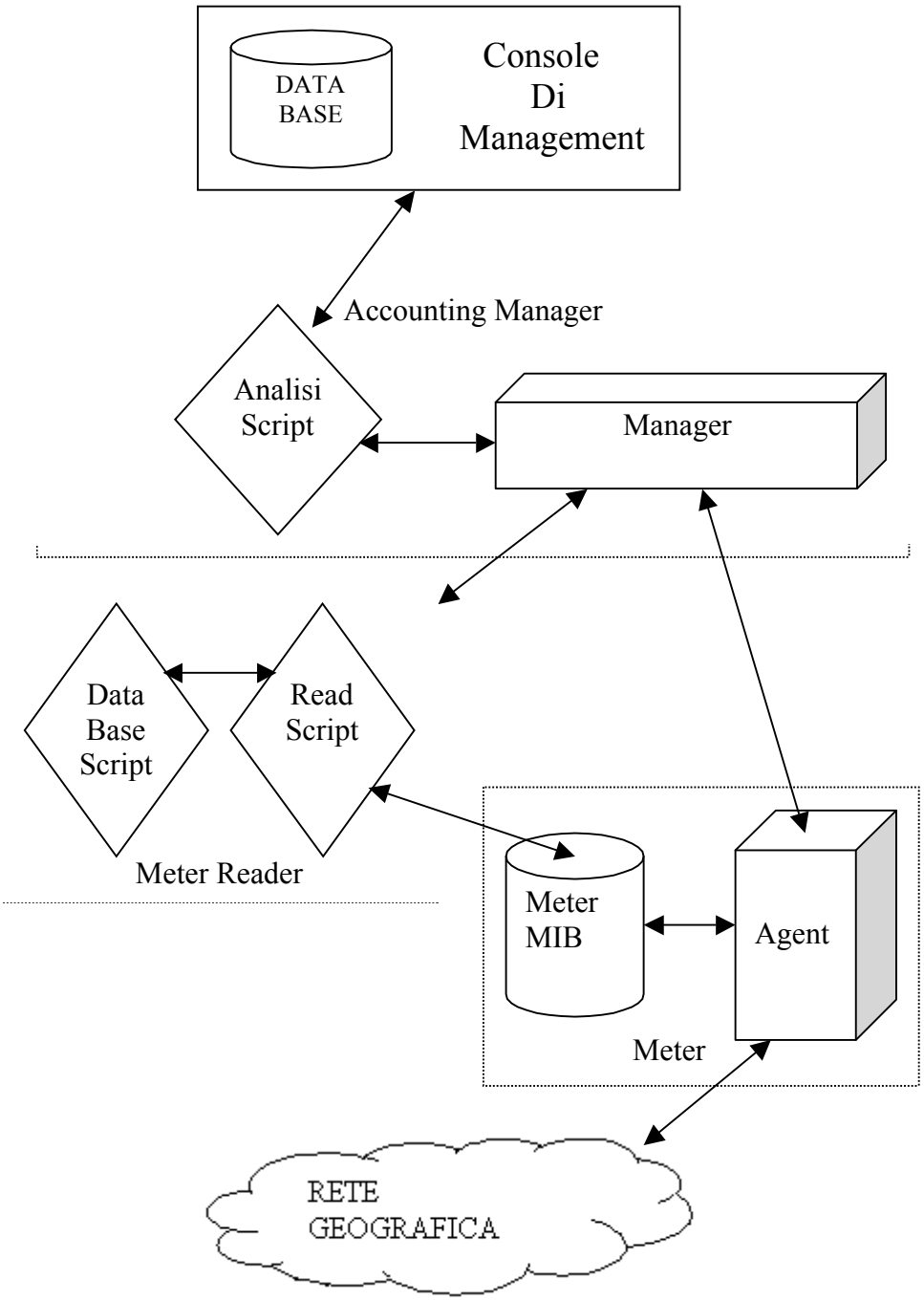

Fig.4 Schema delle comunicazioni fra i Meter, Meter Reader, Accounting Manager, Console di Management.

In generale:

- I Meter svolgono le funzioni definite per il NeTraMet Meter (vedi Cap. 2 Sez. 2.4);
- I Meter Reader, installati su macchine dedicate, raccolgono le informazioni contenute nel Meter MIB e le archiviano nel proprio Reader MIB. Questi implementano al loro interno un Read-Script necessario per effettuare le funzioni da manager (richieste al Meter MIB), e un DataBase-Script necessario per le funzioni da agent (invio delle informazioni richieste dall'Analisi-Script);
- Gli Accounting Manager sono formati da due componenti:
  1. Manager, provvede all'instrumentazione sia dei Meter che dei Meter Reader.
  2. Analisi-Script necessario per l'elaborazione finale dei dati per il billing inviati dal Meter Reader e per l'implementazione del protocollo agent nei confronti della Console di Management.
- Console di Management che individua la postazione dell'amministratore centrale e gestisce tutta l'architettura sottostante.

Tutti i componenti definiti nel nostro progetto, che utilizzano degli script per il loro corretto funzionamento, si basano sull'uso di uno Script-MIB (vedi Cap. 2 Sez. 2.3)

3.1 Meter

I Meter, installati sulle device della rete, al loro interno vengono divisi concettualmente in due componenti:

1. L'Agent ha il compito di processare tutti i pacchetti in transito applicando delle regole definite dagli amministratori e memorizzate nel Meter MIB. Nel caso specifico, le regole di cui sopra, devono prevedere la scansione del pacchetto e l'individuazione di attributi, che saranno poi elaborati (vedi Sez. 3.2), quali:

   - indirizzo sorgente/destinazione;
   - protocollo di comunicazione;
   - porta sorgente/destinazione;
   - tipo di servizio[2] (QoS).

   Il primo attributo è necessario per l'identificazione della macchina da cui viene generato il flusso, che generalmente sarà correlata ad un responsabile a cui far arrivare il resoconto del traffico. Gli altri attributi sono necessari nel caso in cui i costi vengano imputati anche in base al tipo di servizio erogato.

2. Il Meter MIB è un database in cui vengono memorizzate le caratteristiche dei flussi in transito (vedi Cap. 2 Sez. 2.4). Così facendo si riduce notevolmente la quantità di informazioni da memorizzare. Questo in quanto i pacchetti con caratteristiche comuni sono identificati dallo stesso oggetto in cui si specificherà solo il volume di traffico prodotto.

[2] Nell'Ipv4 l'intestazione del pacchetto identifica il tipo di servizio, questo permette di distinguere la gestione dei pacchetti. Nell'Ipv6 la stessa funzione è svolta dal campo "classe di traffico" anch'esso presente nell'intestazione.

Gli oggetti facenti parte del Meter MIB possono essere suddivisi in 3 diversi gruppi a seconda delle loro caratteristiche. Tali gruppi sono:

1. Controllo;
2. Gestione dei flussi di traffico;
3. Politiche di memorizzazione.

Nel gruppo di controllo sono specificate le informazioni necessarie per la configurazione del MIB stesso. Ad esempio le informazioni:

- Generali,
  lo stato del MIB, la data di creazione, etc.
- Meter Reader di riferimento,
  per poter collocare il Meter nella giusta posizione nell'architettura gerarchica.
- Manager,
  per consentire l'instrumentazione del Meter.
- Regole da seguire,
  per l'accesso al Meter MIB da parte del Meter Reader di riferimento.

Il secondo gruppo riunisce informazioni necessarie all'elaborazione dei pacchetti in transito ed il terzo definisce le caratteristiche dei flussi che devono essere memorizzati.
Sono questi ultimi gruppi che consentono al Manager, definendo un set di regole inviate via SNMP, di instrumentare il Meter (vedi Cap. 2, Sez 2.4). Tali regole possono essere aggiornate dall'amministratore a seconda delle esigenze che si presentano di volta in volta.

3.2 Meter Reader

Le funzioni del Meter Reader sono definite da due script che interagiscono fra loro:

- DataBase-Script,
- Read-Script.

DataBase-Script

Il DataBase-Script svolge le funzioni tipiche di un agent. Fra i suoi compiti enunciamo:

- Memorizzazione dei dati nel Reader MIB.
- Interazione con Read-Script.
- Invio dei dati memorizzati in risposta alle richieste dell'accounting manager.

Il Reader MIB, definito per i Meter Reader, oltre al gruppo di controllo (simile a quello definito per i Meter MIB fatta eccezione per gli oggetti che identificano il Meter Reader di riferimento), deve prevedere un altro gruppo di oggetti che chiameremo "Responsabili". Questo gruppo viene definito da una tabella accessibile dal Manager (vedi Sez. 3.3) via SNMP.

Tale tabella è composta da:

1. Oggetti con caratteristiche comuni che identificano i responsabili delle macchine che generano traffico, posti sulle righe;
2. Le classi di QoS (vedi Cap 1), ed il traffico totale prodotto da ogni utente, poste sulle colonne.

		Classi di QoS			Traffico Totale
		HTTP	FTP	TELNET	
Responsabili	Bianchi Mario	……..	……	……..	……..
	Verdi Andrea				
	Rossi Luigi				

Tab. 1 Esempio della tabella Responsabili leggibile dagli umani.

Il DataBase-Script provvederà, se necessario, a ripartire le informazioni sul traffico prodotto, ricevute dal Read-Script, fra le varie classi di QoS. Inoltre la tabella sarà memorizzata nel MIB col formato adeguato per la sua trattazione (ad esempio il nome "Bianchi Mario" sarà sostituito da un numero).

Read-Script

I compiti principali del Read-Script sono:

- Interrogare il Meter MIB per l'acquisizione dei dati.
- Ricavare dai dati dei flussi le informazioni riguardanti gli utenti della rete.
- Interagire con DataBase-Script.

Le informazioni da elaborare vengono acquisite dai Meter Reader attraverso richieste SNMP ai Meter MIB, generalmente verranno inviate delle Get-Bulk così da ottimizzare il trasferimento dei dati.
I dati acquisiti in questo modo non forniscono informazioni specifiche per l'accounting in quanto identificano una serie di caratteristiche dei flussi in transito sulla rete (vedi Sez. 3.1).
Tutto ciò implica una progettazione del Read-Script che preveda delle funzioni per l'elaborazione dei dati di flusso, così da poter convertire tali dati in informazioni necessarie per imputare ad ogni responsabile i costi.

Il ciclo primario del Read-Script è costituito da 2 fasi:

1. Analisi delle informazioni ricevute per individuare l'indirizzo sorgente del flusso con la conseguente determinazione della macchina che ha generato traffico e l'identificazione del responsabile.

2. Interazione con il DataBase-Script per verificare se fra gli oggetti del gruppo "Responsabili" è presente l'utente appena identificato e:

    a. Se presente, inviare al DataBase-Script i dati per renderli disponibili all'accounting manager.

    b. Se non è presente, creare un nuovo oggetto nel gruppo "Responsabili" (si aggiunge una riga alla tabella) e inviare i dati al DataBase-Script per la memorizzazione.

E' possibile implementare in questo script dei meccanismi per garantire la sicurezza della rete.
Uno fra i più semplici, riportato come esempio, prevede che l'Accounting Manager invii "una tantum" un aggiornamento degli oggetti facenti parte del gruppo "Responsabili" (settaggio dei valori della tabella del MIB). In questo modo è possibile verificare l'intrusione nella rete di utenti non autorizzati.
Nel momento in cui viene identificato un oggetto, non presente nella tabella, la fase 2 b viene sostituita dall'invio di un segnale di allarme [TRAP], che deve essere definito nel MIB, all'Accounting Manager.

3.3 Accounting Manager

L'Accounting Manager, come detto in precedenza, è suddiviso in due componenti:

1. Analisi-Script;
2. Manager.

Analisi-Script

I compiti di questo script sono:

- Implementare il protocollo da agent per le comunicazioni con la Console di Management.
  A tale scopo l'Analisi-Script verrà dotato di funzioni che lo rendono in grado di prelevare e memorizzare informazioni nell'Accounting MIB definito per questo livello della gerarchia che stiamo progettando.
- Calcolare i costi da attribuire ad ogni utente della rete e memorizzarli in una tabella dell'Accounting MIB accessibile, sempre via SNMP, dalla Console di Management.
- Memorizzare le informazioni che il Manager dovrà interpretare per eseguire le operazioni corrispondenti.

L'introduzione della tabella di oggetti nella gerarchia dell'architettura consente un'ulteriore riduzione del volume dei dati che devono essere comunicati dai livelli più bassi a quello più alto.

Tale tabella definisce un gruppo, simile a quello che è stato definito per il Reader MIB (vedi Sez. 3.2), dove ogni oggetto, o riga della tabella, identifica un utente. In tale tabella, che chiameremo TaBil (Tabella di Billing), esiste una sola colonna nella quale vengono riportati i costi da attribuire all'utente corrispondente (Tabella 2). E' infatti compito dell'Analisi-Script acquisire, dalla tabella definita per il Meter Reader, i dati necessari per calcolare il costo da imputare ad ogni responsabile di macchina. Ciò viene fatto seguendo determinate regole indicate dall'amministratore centrale e ponderando i costi a seconda della classe di QoS del traffico prodotto. Una volta effettuati tali calcoli è possibile aggregare nel campo "COSTI" di TaBil il debito totale.

		COSTI
Utenti	Bianchi Mario	…….
	Rossi Luigi	…….

Tab. 2 Esempio di TaBil contenuta nell'Accounting MIB

Inoltre l'amministratore, dalla Console di Management, potrà modificare le informazioni contenute nell'Accounting MIB così da instrumentare il Manager a seconda delle esigenze che si presentano.

Manager

Il componente Manager ha il compito di:

- Gestire la parte di rete identificata dai Meter e Meter Reader che a lui fanno riferimento.
- Acquisire i dati di accounting dal Meter Reader.

L'acquisizione dei dati di accounting dal Meter Reader è speculare all'acquisizione dei dati da parte del Meter Reader stesso nei confronti del Meter (vedi Sez. 3.2), fatta eccezione per l'analisi dei dati che in questo caso viene implementata dall'Analisi-Script.
L'introduzione nell'architettura di tale componente è fondamentale per poter ripartire anche l'instrumentazione dei dispositivi oltre a quella delle informazioni.

Di fatto, nella struttura definita, i dati vengono scambiati in due direzioni:

1. Basso-Alto.
   Questo percorso è seguito dai dati di accounting che, ricavati da un'elaborazione del traffico di rete, devono essere trasmessi alla Console di Management.
2. Alto-Basso.
   Questo percorso è seguito dalle informazioni necessarie per l'instrumentazione dei componenti della struttura.

Il Manager dà la possibilità all'amministratore centrale di instrumentare tutti i componenti dell'architettura con un numero limitato di comunicazioni (proporzionale al numero di Accounting Manager). Questo in quanto, una volta che dalla console di Management viene programmato il Manager, questo provvederà a sua volta ad instrumentare tutti i componenti della gerarchia sottostante senza bisogno di comunicazioni aggiuntive fra Console di Management e Meter Reader/Meter.

3.4 Console di Management

La console di Management identifica la postazione degli amministratori centrali. Ciò significa che esiste un punto della rete da cui è possibile instrumentare tutti i componenti e a cui arrivano tutte le informazioni, in questo caso, per il billing.

L'aspetto più interessante della Console di Management è quello che riguarda il meccanismo di programmazione dei dispositivi della gerarchia.
Inviando una nuova versione dell'Analisi-Script è possibile aggiornare le tariffe da applicare alle varie classi di QoS o al traffico totale prodotto da ogni utente. Inoltre, in questo modo, è possibile aggiornare anche il componente Manager del livello degli Accounting Manager (vedi Sez. 3.3).
Inviando all'Accounting Manager degli script che dovranno sostituire quelli esistenti nel Meter Reader (vedi Sez. 3.2) è possibile aggiornare anche quest'ultimo componente. Sarà dunque compito di ogni Manager, presente nel secondo livello, quello di instrumentare, per conto degli amministratori centrali, i Meter Reader. Questa operazione è necessaria, ad

esempio, per aggiornare la tabella del gruppo responsabili ed implementare il meccanismo di sicurezza sopradescritto (vedi Sez. 3.2).
Allo stesso modo dei Meter Reader è possibile instrumentare i Meter per avere le misurazioni che gli amministratori ritengono più adeguate.

L'altro aspetto che caratterizza la Console di Management è quello relativo all'imputazione dei costi ad ogni utente della rete. E' di fatto compito degli amministratori centrali procurarsi i dati di accounting, interrogando le varie TaBil del livello degli Accounting Manager, per inviare il resoconto del traffico e relativi costi, ad ogni utente.
A questo scopo la Console di Management sarà fornita di un database nel quale vengono memorizzati i dati di ogni responsabile e che verrà interrogato ogni qual volta è necessario inviare il resoconto.
Il meccanismo di sicurezza descritto nella sezione 3.2 si basa sul contenuto di questo database. Nel momento in cui viene aggiunto un utente si aggiorna la base di dati della Console di Management e, quest'ultima, provvederà ad inviare l'aggiornamento ai livelli inferiori della gerarchia implementando i meccanismi visti in precedenza.

4. Note per l'implementazione

Lasciamo a futuri documenti la trattazione piu' specifica di ogni particolare dell'architettura ed eventualmente una sua implementazione, che dato il poco tempo disponibile, non abbiamo potuto affrontare. Possiamo pero' darne delle linee guida che ci sembrano molto interessanti.
Per prima cosa consigliamo di utilizzare SNMP versione 3 in quanto mette a disposizione dei meccanismi necessari a garantire una maggiore sicurezza dell'architettura.
Si ha infatti la possibilità, con l'utilizzo di questa versione del protocollo, di indicare già al momento della definizione del MIB delle viste d'accesso al sistema. Queste permettono di suddividere le informazioni in modo tale che solo chi è autorizzato ad accedervi ne sia effettivamente in grado.
Inoltre, grazie al criptaggio di tutti i dati contenuti nei pacchetti inviati da questa versione del protocollo, è garantita la segretezza dei dati di accounting che riguardano gli utenti della rete.
Sempre per garantire una maggior sicurezza consigliamo di progettare una rete di management che possa essere utilizzata per instrumentare i componenti dell'architettura. Questa rete, distinta da quella usata per la trasmissione dei dati, viene utilizzata unicamente dagli amministratori che quindi possono accedere ai vari componenti anche nel caso di malfunzionamenti della rete portante, e non incorrono in pericoli che possono essere propri di reti a cui accede un numero notevole di utenti.
Per quanto riguarda il meter reader consigliamo un'implementazione dello Script-MIB chiamata JASMIN (Java Script MIB Implementation). La sua struttura e' composta da:

- Un agent che permette le comunicazioni con i manager di più alto livello via SNMP.
- Il kernel, indipendente dalla piattaforma di supporto per JASMIN, che comunica con l'agent attraverso un sub-agent.
- Il sistema di runtime che comunica con il kernel attraverso l'interfaccia SMX. Questo sottosistema è necessario per definire quali linguaggi di programmazione vengono supportati da JASMIN.

Uno dei problemi del JASMIN è però l'assenza di specifiche per la comunicazione con altre entità della rete. E' possibile soltanto rispondere alle richieste fatte dai manager di più alto livello, ma non comunicare con agent sottostanti, come nel nostro caso di utilizzo dello Script-MIB nei manager intermedi. Per questo sono state progettate delle librerie per la JVM (Java Virtual Machine) che implementano lo stack SNMP. In questo modo tutti gli script possono utilizzare la stessa libreria evitando di dover definire le specifiche SNMP per ognuno di essi.
Una fra le librerie più complete, che potrà essere utilizzata in caso di implementazione, è quella sviluppata dalla Advent-net. Fra le sue caratteristiche enunciamo il completo supporto di ogni versione dell'SNMP.

5. Caso di Studio: Seabone

L'architettura progettata può avere applicazione sulla rete Seabone da noi trattata nella Sez. 1.1

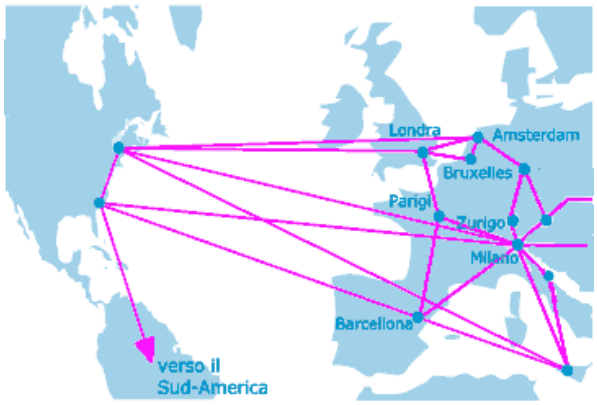

Fig. 5 Caso di studio: Seabone

Anche se non conosciamo la posizione precisa dei router sulla rete, il nostro compito e' di individuare la locazione ideale delle varie entità dell'architettura, in modo da raggiungere:

1. Il maggior grado di scalabilità possibile.
2. Il bilanciamento del carico delle informazioni scambiate fra i vari livelli della gerarchia.
3. L'ottimizzazione delle distanze fra i vari componenti.

Il giusto posizionamento della console di management e' di fondamentale importanza per la raccolta veloce ed efficiente dei dati.
Nel caso specifico, il nodo centrale della rete si trova a Milano. Da qui si ha la possibilità di raggiungere i nodi piu' importanti in modo diretto (esempio Milano-New York, Milano-Palermo, Milano-Barcellona). Riteniamo che, per l'architettura da noi progettata questa sia la locazione migliore per la console di management.
Per quanto riguarda gli accounting manager, la loro posizione ideale, e' quanto piu' possibile vicino alla console di management (raggiungibile cioè con collegamenti diretti o quasi). Tuttavia e' necessario bilanciare il numero di meter reader e di meter che devono essere interrogati e instrumentati dal componente preso in considerazione. Gli accounting manager, definiti dal nostro progetto, identificano delle sottoreti, quindi crediamo di poter identificare ognuna di queste sottoreti con le grande città collegate da Seabone (New York, Miami, Palermo, Londra, Amsterdam, ecc.).
Immaginando che in ogni sottorete sia presente un numero ancora elevato di dispositivi, questi devono essere ripartiti fra vari meter reader. Questa ripartizione deve essere effettuata in maniera scrupolosa analizzando le caratteristiche di ogni collegamento e i dispositivo installato.
Ovviamente ognuno di questi dispositivi deve essere in grado di supportare l'implementazione dei NeTraMet meter per le misurazioni del traffico.

6. Commenti Finali

Dell'architettura definita in questo documento è necessario specificare in questa sezione quelli che sono gli aspetti critici e argomenti di studio per il miglioramento di tutta la struttura.
Un aspetto di fondamentale importanza è quello definito dalle comunicazioni fra i livelli.
Se, di fatto, i dati di accounting vengono scambiati fra i componenti tramite SNMP, nel momento in cui si rende necessario l'aggiornamento di uno script, tale comunicazione deve essere supportata da un protocollo diverso (vedi Cap.2 Sez. 2.3).
Questo rende la comunicazione dispendiosa sia in termini di tempo che di occupazione della banda.
La scelta di utilizzare degli script è stata seguita comunque in quanto il loro aggiornamento è un'operazione poco frequente. Questa si verifica nel caso in cui:

- Le tariffe da applicare vengono modificate dagli amministratori,
  aggiornamento Analisi-Script.
- Cambiamento delle classi di QoS definite in precedenza,
  aggiornamento del DataBase-Script.
- Modifica dei meccanismi di conversione dei dati di flusso in informazioni utente,
  aggiornamento dello Read-Script.
- Aggiornamento della versione del meter,
  aggiornamento dello Read-Script.

Quest'ultimo fatto implica un'ulteriore comunicazione poco efficiente dovuta appunto all'aggiornamento del meter.

L'utilizzo delle tabelle definite nei vari componenti dell'architettura, rende le comunicazioni fra i vari moduli veloci ed efficienti. Avremmo potuto utilizzare un database per la memorizzazione dei dati, in questo modo avremmo potuto usufruire di una memoria storica dell'impiego della rete. Questa progettazione avrebbe pero' appesantito le comunicazioni fra i vari moduli in quanto i manager intermedi avrebbero dovuto inoltrare le loro richieste agli script, che a loro volta, avrebbero dovuto interrogare il database prima di poter inviare la risposta.
Con l'introduzione delle tabelle, i dati sono direttamente accessibili ai manager intermedi e alla console di management via SNMP.
Una considerazione correlata all'utilizzo delle tabelle deve essere fatta riguardo i possibili punti di fallimento (vedi Cap. 2 Sez. 2.2). In caso di malfunzionamenti dei componenti intermedi o della console di management, le informazioni non vengono prelevate e rimangono memorizzate nel MIB. Questo meccanismo garantisce la persistenza dei dati per un limitato arco di tempo, in quanto, in presenza di un notevole volume di dati, la dimensione delle instanze del MIB non e` sufficiente alla memorizzazione.

Un altro aspetto che non è stato approfondito in questo documento è quello che riguarda i meccanismi per la sicurezza. Quello accennato è un modo molto semplice, e a volte poco efficace, per impedire intrusioni da parte di persone non autorizzate.
Anche i meccanismi messi a disposizione dall'SNMPv3 non sono in grado di fornire un adeguato livello di sicurezza per reti così grandi e di così grande importanza come Seabone.

In realtà non sono stati presi in esame casi in cui la sicurezza può essere compromessa e, di conseguenza, è stato tralasciato lo studio di meccanismi efficienti.

5. Bibliografia

[1] L. Deri e J. Schönwälder, "Sistemi di Elaborazione dell'Informazione: Gestione di Rete", Febbraio 2001.

[2] J. Case, M. Fedor, M. Schoffstall, J. Davin, RFC 1157, Maggio 1990 www.ietf.org/rfc/rfc1157.txt;.

[3] http://www.seabone.net/

[4] Ruben Marsman, "Development of prototype-scripts for Script-MIB", Enschede 25 maggio 2000

[5] http://www.caida.org/tools/measurements/netramet