Definizione di un'architettura client/server basata su SNMP per il monitoraggio del traffico che passa attraverso n router utilizzati per connettere un'azienda ad Internet.

Negro Giulio
Del Mutolo Nicola

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

A.A. 2001/2002

SOMMARIO:

1. INTRODUZIONE

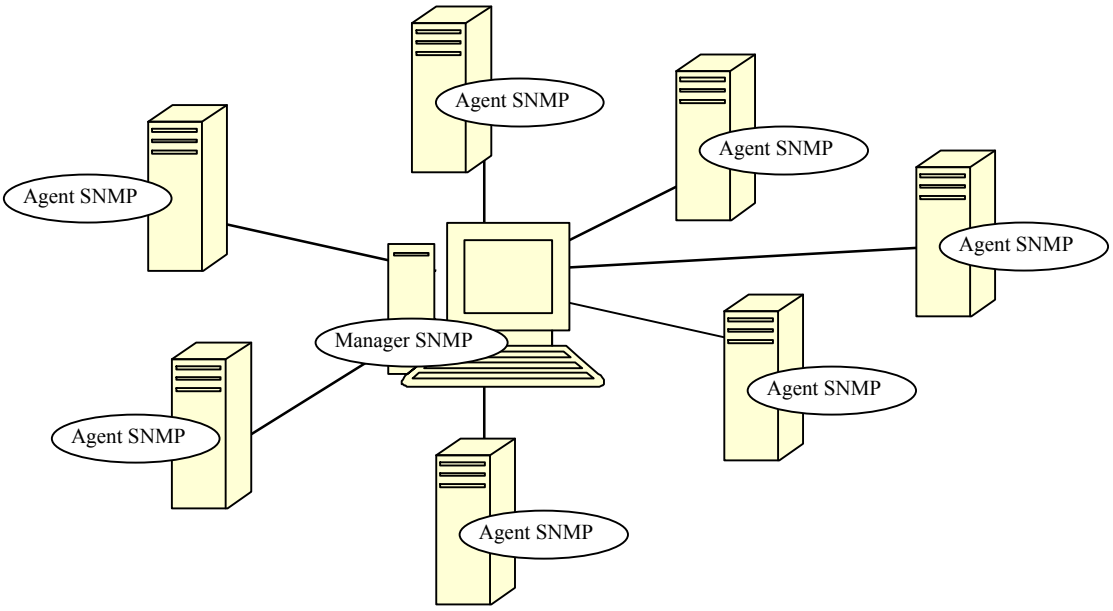
Il caso che andremo a studiare riguarda la realizzazione di un'applicazione client-server sviluppata utilizzando il protocollo SNMP, la quale consentirà ad un'azienda di monitorare i router che la connettono ad Internet.
Per monitoraggio intendiamo la raccolta di informazioni da tutti i router nei vari tempi di campionamento e la determinazione del loro stato.

1.1 AMBIENTAZIONE

Il tipo d'azienda presa in esame è un'azienda con sedi sparse su tutto il territorio nazionale avente almeno un router in ogni comune Italiano.
Il problema è quello di monitorare il traffico che passa attraverso questi router, gestendo alcuni casi di malfunzionamento degli stessi.
Con casi di malfunzionamento intendiamo problemi di traffico o di irraggiungibilità, che illustreremo in modo approfondito più avanti.

1.2 SOLUZIONI POSSIBILI

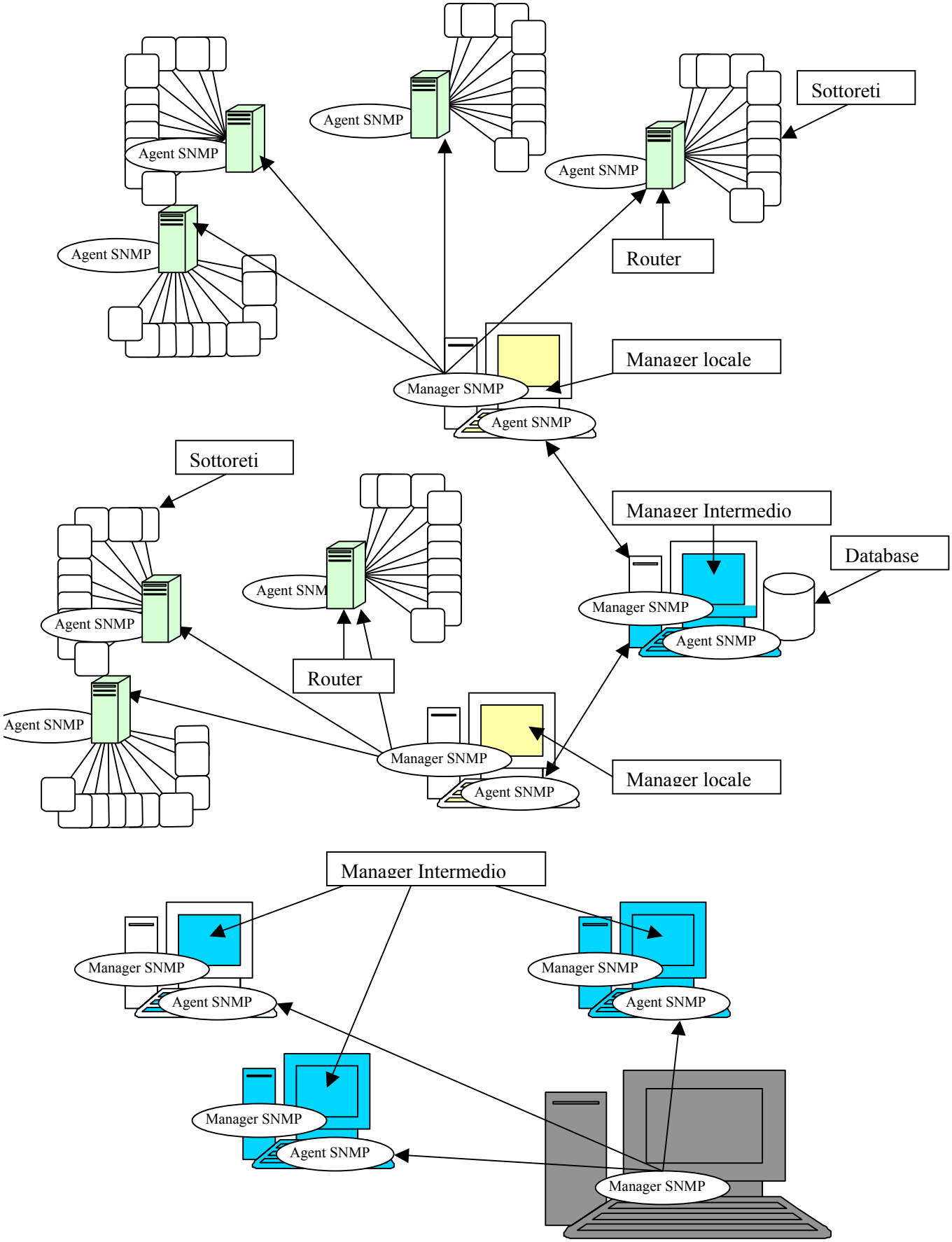
Il primo problema che dobbiamo risolvere è quello relativo al tipo di rete da utilizzare per collegare i vari router. Per evitare di utilizzare reti sovraccaricate dal traffico altrui e per evitare inconvenienti dovuti alla sicurezza utilizzeremo una rete di ***management*** dedicata, che interconnetta i router con i punti di monitoraggio.
Per implementare l'architettura in esame, è possibile utilizzare metodologie diverse. Una di queste potrebbe essere quella di adottare una politica di management centralizzato, dove ogni router è direttamente connesso ad una macchina centrale che monitorizza il traffico passante per ognuno di essi.



Questo tipo di architettura richiede l'installazione di un agent SNMP su ogni router che intendiamo monitorare e un manager SNMP sulla macchina centrale. Per recuperare le informazioni di cui ha bisogno, il manager farà ***polling*** sui vari agent leggendo le informazioni necessarie.
Anche se sulla carta questo tipo d'implementazione potrebbe sembrare semplice, si rivela tutt'altro che funzionante. I motivi del fallimento di questo modello applicato ad un caso del genere sono molteplici. Risulta evidente la difficoltà di quest'architettura di gestire grandi quantità di dati, poiché il tempo che impiegherebbe ad interrogare tutti i router del territorio nazionale diventerebbe elevato.
Facciamo un esempio:
Sapendo che il nostro manager centrale deve monitorare almeno un router per ogni comune italiano, possiamo fare una stima approssimativa di 4500 router totali da controllare. Considerando un tempo medio tra invio della richiesta e ricezione di una risposta da parte dell'agent di 0.5 secondi, il tempo che impiegherebbe il manager per completare un giro di interrogazioni sarebbe di circa 35-40 minuti! Teniamo anche conto che i numeri presi in considerazione sono altamente ottimistici, sia come quantità di router che come velocità di risposta. Per ovviare a questo problema potremmo fare in modo che il nostro manager centrale comunichi contemporaneamente con più macchine, ma la situazione non migliorerebbe molto, poiché la mole di dati da gestire rimarrebbe in ogni caso elevata.
Un'altra soluzione che abbiamo preso in considerazione e che si è dimostrata più funzionale per il nostro problema è quella di utilizzare una politica di ***management by delegation*.**
Utilizzando questa politica, le responsabilità ed il peso computazionale vengono delegate a macchine intermedie che fanno da tramite tra i router ed il manager centrale.
Considerando che il nostro campo di applicazione è quello di una rete nazionale sfrutteremo la suddivisione del territorio per spartire i compiti tra i nostri manager intermedi.
Prima di illustrare la soluzione scelta è necessario però stabilire alcune regole:

- Ogni router deve avere sotto di sé dalle 10 alle 15 reti locali, in modo che ogni rete locale copra all'incirca il territorio di un comune.
- Ogni router è monitorato da un manager/agent locale il quale avrà sotto di sé non più di 10 router. Quindi ogni manager/agent locale gestirà all'incirca i router di una provincia.
- Ogni manager/agent locale ha sopra di sé un manager/agent intermedio il quale interagirà con i manager/agent locali di due regioni.
- Ogni manager/agent intermedio interagisce con un unico manager centrale.

Con questa suddivisione dei compiti il carico di lavoro è ripartito quasi equamente fra tutti i vari manager/agent.
Adesso presentiamo due schemi. Nel primo mostreremo la situazione a livello locale, dove i manager/agent locali monitorizzano i router e inoltre interagiscono con i manager intermedi. Nel secondo schema mostreremo, invece, l'interazione tra manager/agent intermedi e il manager centrale.


Agent SNMP
Agent SNMP
Agent SNMP
Agent SNMP
Sottoreti
Router
Manager SNMP
Agent SNMP
Manager locale
Sottoreti
Agent SNMP
Agent SNMP
Manager Intermedio
Database
Manager SNMP
Agent SNMP
Router
Agent SNMP
Manager SNMP
Agent SNMP
Manager locale
Manager Intermedio
Manager SNMP
Agent SNMP
Manager SNMP
Agent SNMP
Manager SNMP
Agent SNMP
Manager SNMP

2. IMPLEMENTAZIONE

Passiamo ora ad illustrare l'implementazione del nostro modello. In questo paragrafo spiegheremo il comportamento e le interazioni tra i vari elementi della nostra architettura.
Come abbiamo detto in precedenza, i compiti sono distribuiti tra tre categorie di manager e agent. La prima categoria è quella dei manager locali i quali si occupano di monitorare i router di cui ci vogliamo interessare. La seconda categoria è quella dei manager intermedi, che interagiscono con i manager locali. Infine nella terza categoria abbiamo solo il manager centrale, il quale sorveglia la situazione. In realtà ci sarebbe anche una quarta categoria da prendere in considerazione, quella dell'agent SNMP installato su ogni router, dal quale il manager locale leggerà dei valori e stabilirà lo stato del router.

A questo punto dobbiamo definire quali sono i dati di cui intendiamo tener conto per eseguire il monitoraggio. Per questa implementazione abbiamo scelto di utilizzare per tutti i router il MIB II. Un'altra soluzione possibile era quella di utilizzare RMON, ma abbiamo preferito optare per la prima soluzione dato il numero non elevato di router da monitorare, per l'elevato costo di gestione di RMON e perché la nostra attenzione si è focalizzata solo su poche variabili, cioè su quelle che concorrono a determinare i valori del traffico in ingresso e del traffico in uscita dai vari router.

Di seguito elenchiamo le variabili del MIB II a cui ci siamo interessati:

Traffico in Ingresso:
- IfInUcastPkts: Pacchetti in ingresso che non sono né broadcast né multicast.
- IfInNUcatPkts: Pacchetti multicast o broadcast.

La somma di questi due valori dà il numero dei pacchetti realmente passati dall'interfaccia di rete del router alla sua sottorete, vale a dire:

TrafficoTrasmessoInInput = IfInUcastPkts + IfInNUcatPkts**.**

Dobbiamo però tener conto anche di altre variabili:
- IfInDiscards: Pacchetti scartati dal router.
- IfInUnknownProtos: Pacchetti che utilizzano protocolli sconosciuti.
- IfInErrors: Pacchetti contenenti errori.

La somma dei valori sopraelencati restituisce il valore del traffico che non è stato trasmesso dal router alla sua sottorete, cioè:

TrafficoNonTrasmessoInInput = IfInDiscards + IfInUnknownProtos + IfInErrors.

La somma di ***TrafficoNonTrasmessoInInput*** e di ***TrafficoTrasmessoInInput*** restituisce il numero di pacchetti ricevuti dalla rete, che chiameremo ***TotPktsRicevRete.***
Le variabili ***TrafficoTrasmessoInInput, TrafficoNonTrasmessoInInput, TotPktsRicevRete*** sono determinate dal manager locale, il quale, tramite delle GET leggerà le variabili SNMP sopraelencate.

Traffico in Uscita:
- IfOutUcastPkts: Pacchetti in uscita che non sono né broadcast né multicast.
- IfOutNUcatPkts: Pacchetti multicast o broadcast.

La somma di questi due valori rappresenta il numero dei pacchetti realmente passati dall'interfaccia di rete del router ad Internet, vale a dire:

TrafficoTrasmessoInOutput = IfOutUcastPkts + IfOutNUcatPkts

Inoltre:

- IfInDiscards: Pacchetti scartati dal router.
- IfInErrors: Pacchetti contenenti errori.

La somma dei valori sopraelencati restituisce il traffico che non è stato trasmesso dal router ad Internet, che chiameremo ***TrafficoNonTrasmessoInOutput***.
La somma del traffico trasmesso e di quello non trasmesso in output rappresenta il numero dei pacchetti trasmessi dal router ad Internet e la variabile relativa sarà ***TotPktsTrasm.***
Anche in questo caso, come per l'input, le variabili ***TotPktsTrasm, TrafficoTrasmessoInOutput*** e ***TrafficoNonTrasmessoInOutput*** verranno determinate dal manager locale.

3. COMPORTAMENTO

Una volta stabilite le variabili in gioco é necessario definire il "comportamento" dei vari elementi della nostra architettura.
Per ogni elemento abbiamo fissato tre stati: **Verde**, **Giallo**, **Rosso.**
Gli stati si associano ai router, ai manager locali e a quelli intermedi. Il modo di determinare lo stato varia secondo la categoria e sarà illustrato più avanti.

Lo stato Verde, per i router, indica il loro corretto funzionamento. Lo stato Giallo indica che ci sono alcuni problemi, mentre quello Rosso indica che i problemi sono gravi.
I problemi che si possono definire con le variabili sopra citate sono i seguenti:

- **Traffico in Ingresso eccessivo.**
- **Traffico in Uscita eccessivo.**
- **Troppi Pacchetti in Ingresso Scartati.**
- **Troppi Pacchetti in Uscita Scartati.**
- **Router non risponde (situazione in cui non si sa se il router é UP o DOWN).**
- **Router irraggiungibile (router DOWN).**
- **Manager Irraggiungibile.**

In realtà gli ultimi tre problemi elencati non dipendono dalle variabili definite sopra, ma da possibili guasti sui collegamenti o da problemi tecnici. Al contrario i primi quattro problemi esposti dipendono, come è evidente, dal traffico in ingresso o in uscita dai router.

Illustriamo adesso le mansioni dei manager locali, dei manager intermedi e del manager centrale.

3.1 I MANAGER LOCALI

I manager locali devono avere sotto di loro 10 router (il numero è indicativo; per le zone con tassi di traffico molto elevato, il numero può essere maggiore o viceversa per le zone con poco traffico).
Come illustrato nello schema sopra, i vari manager locali hanno il compito di ottenere informazioni dai router, per poi elaborarle e definirne lo stato.
Per ottenere queste informazioni il manager, tramite delle ***GET*** (utilizzando una politica di *polling),* legge i valori che ci interessano.

Passiamo ora ad analizzare l'interazione tra manager locali ed agent.
I manager locali faranno polling sui router, eseguendo delle GET al tempo t/4 e 3/4t (come tempo di campionamento t utilizzeremo 5 minuti), ed al tempo t invieranno al manager intermedio, tramite una TRAP, lo stato dei router ed i valori delle variabili relative al traffico.

Un'alternativa a questa politica, è quella di far fare polling al manager intermedio. Il difetto di questa tecnica, sono i tempi di attesa nel caso in cui non si riescano a reperire le informazioni dal manager locale, al contrario utilizzando le TRAP sono i manager locali a dover inviare le informazioni al tempo t.
Il tempo t non deve essere lo stesso per ogni manager locale, in quanto se così fosse il manager intermedio si troverebbe sommerso dalle TRAP di tutti i manager locali, rischiando di trovarsi al tempo t sovraccarico di lavoro e praticamente inattivo fino allo scadere del successivo quanto di tempo. La scelta del quanto di tempo t é a discrezione dell'amministratore del sistema.

Si suppone che, in caso di funzionamento corretto della rete e assenza di problemi nei router, essi rispondano alle interrogazioni effettuate dal manager locale al tempo t/4, entro il tempo 3/4t e a quelle effettuate al tempo 3/4t, entro il tempo t.
Sono utilizzati due tempi di campionamento per poter dare una visione più veritiera sullo stato dei router, poiché se al tempo t/4 un router non risponde, il suo stato da verde passa a giallo e se anche al tempo 3/4t non risponde, il suo stato diventa rosso, altrimenti torna verde. Al contrario, se un router, che al tempo t/4 aveva risposto, non risponde al tempo 3/4t, al tempo t il manager intermedio viene informato dal manager locale che lo stato del router XY è giallo, dichiarando quale é il problema riscontrato. (Router = XY, Stato = Giallo, Problema = Router non risponde).

I tempi t/4 e 3/4t potrebbero sembrare troppo brevi, ma sono giustificati dal fatto che il manager locale deve gestire un piccolo numero di router ed inoltre la distanza tra manager e agent non è molto grande.

Quanto detto sinora si riferisce ai problemi relativi ai collegamenti. Per definire lo stato del router bisogna tener conto anche del traffico in ingresso e in uscita.

Per quanto riguarda la determinazione dello stato dei router in base al traffico in ingresso ed in uscita, il manager locale esegue dei controlli sulle variabili che ogni volta legge da ognuno di essi. Logicamente questi controlli vengono effettuati per ogni singolo router, per poter determinare lo stato sia in base al traffico in ingresso che a quello di uscita.
Per effettuare questi controlli il manager locale prende in considerazione due variabili soglia (***Threshold***):

- ***ThresholdYellow***
- ***ThresholdRed***

Il primo controllo che il manager effettua é quello relativo al traffico in ingresso.
I valori delle due soglie sono i seguenti:

- ***ThresholdYellow*** = 85% della banda.
- ***ThresholdRed*** = 100% della banda.

Stabilite le soglie, il manager locale determinerà lo stato del router in base al seguente algoritmo:

situazione iniziale: StatoRouter = Verde;

```
StatoRouter = Verde; Problema = "";
while ( (TotPktsRicevRete ) >= ThresholdYellow) {
    if ( (TotPktsRicevRete ) >= ThresholdRed) {
       StatoRouter = Rosso; Problema = "Traffico in Ingresso eccessivo";
    }
     StatoRouter = Giallo; Problema = "Traffico in Ingresso eccessivo";
```

```
}
```

Per determinare lo stato del router in base al traffico di ingresso ed ai pacchetti scartati, il manager locale esegue i seguenti controlli:

```
situazione iniziale: StatoRouter = Verde;

StatoRouter = Verde; Problema = "";
while ( ( TrafficoTrasmessoInInput / TotPktsRicevRete ) <= ThresholdYellow) {
    if ( ( TrafficoTrasmessoInInput / TotPktsRicevRete ) <= ThresholdRed) {
        StatoRouter = Rosso; Problema = "Troppi pacchetti in Ingresso Scartati";
    }
     StatoRouter = Giallo; Problema = "Troppi pacchetti in Ingresso Scartati";
}
```

Allo stesso modo, per determinare lo stato del router in base al traffico di uscita il manager locale esegue il seguente algoritmo:

```
situazione iniziale: StatoRouter = Verde;

StatoRouter = Verde; Problema = "";
while ( (TotPktsTrasm ) >= ThresholdYellow) {
    if ( (TotPktsTrasm ) >= ThresholdRed) {
        StatoRouter = Rosso; Problema = "Traffico in Uscita eccessivo";
    }
     StatoRouter = Giallo; Problema = "Traffico in Uscita eccessivo";
}
```

Per quello che riguarda il rapporto tra pacchetti realmente trasmessi e totale dei pacchetti l'algoritmo é il seguente:

```
situazione iniziale: StatoRouter = Verde;

StatoRouter = Verde;
while ( ( TrafficoTrasmessoInOuput / TotPktsTrasm ) <= ThresholdYellow) {
    if ( ( TrafficoTrasmessoInOuput / TotPktsTrasm ) <= ThresholdRed) {
        StatoRouter = Rosso;
    }
     StatoRouter = Giallo;
}
```

Lo stato dei manager locali dipende dallo stato dei router da loro monitorati.
Per determinare lo stato dei manager locali dovremo determinare un "peso" da associare ad ogni stato.

Stato Verde = 1; **Stato Giallo** = 0,5; **Stato Rosso** = 0.
(Gli stati sopra indicati si riferiscono ai routers.).
Lo stato del manager locale sara' determinato nel seguente modo:
Stato Verde: Nessun problema, cioè la somma dei valori associati agli stati dei routers é uguale al numero di router monitorati dal manager locale.

Stato Giallo: Se la somma dei valori associati agli stati dei routers è maggiore o uguale a ⌊3/4+1⌋ del loro totale.

Stato Rosso: Se la somma dei valori associati agli stati dei routers è minore di ⌊3/4+1⌋del loro totale.

Facciamo un esempio:
Poniamo che un manager locale monitorizzi 15 routers e supponiamo che 8 siano nello stato verde, quindi non aventi alcun problema, 5 nello stato giallo, ovvero con qualche problema, 2 nello stato rosso.Quindi lo stato del manager locale diventa:

1*8 + 0.5*5 + 0*2 = 10.5 che e' strettamente minore di ⌊3/4+1⌋di 15. Da quanto detto lo stato del manager locale è rosso.

3.2 MANAGER INTERMEDI

I manager intermedi aspettano, da tutti i manager locali sotto di loro, le TRAP contenenti le informazioni sul traffico dei router. Queste informazioni sono memorizzate in un database per effettuare le varie statistiche di cui parleremo più avanti.
Se al tempo t/4 un manager intermedio non ha ancora ricevuto nessuna TRAP da un manager locale allora esegue una GET per recuperare le informazioni di cui ha bisogno. Se la GET non ha risposta invia al manager centrale una TRAP in cui indica il problema riscontrato. (Managerlocale=XXY, Problema=Manager Irraggiungibile).
Quando il manager intermedio riceve delle TRAP con delle informazioni che indicano il cambiamento di stato di un router, accede al DataBase e cambia lo stato di questo segnando il problema riscontrato.
Anche i manager Intermedi hanno uno stato che è determinato da quello dei manager locali e quindi dallo stato dei router da loro monitorati.

Come fatto in precedenza associamo un "peso" allo stato di ogni manager locale per determinare quello del manager intermedio.
Stato Verde = 1; **Stato Giallo** = 0,5; **Stato Rosso** = 0.

Lo stato dei manager intermedi è determinato come indicato di seguito:
Stato Verde: Nessun problema, vale a dire che la somma dei valori associati agli stati dei manager locali e' uguale al numero dei manager locali.

Stato Giallo: Se la somma dei valori associati agli stati dei manager locali è maggiore o uguale a ⌊3/4+1⌋del loro totale.

Stato Rosso: Se la somma dei valori associati agli stati dei manager locali è minore di ⌊3/4+1⌋del loro totale.

3.3 IL MANAGER CENTRALE

Il comportamento del manager Centrale invece è molto semplice. Per ipotesi supponiamo che tutto funzioni al meglio. Nel caso in cui sia segnalato al manager centrale un cambiamento di stato (Rosso -> Verde; Verde -> Giallo; Giallo -> Rosso) di una porzione di rete, esso può andare ad interrogare il manager intermedio associato a quella porzione di rete.
Un'alternativa potrebbe essere quella di permettere al manager centrale di interrogare direttamente il manager locale che dovrebbe però avere un MIB. Questa soluzione renderebbe inutile la presenza del Data Base nei manager intermedi.
Queste due soluzioni differiscono solo per il metodo di gestire le informazioni. Nella prima le informazioni vengono gestite tramite un DB che contiene informazioni relative ai router di più zone, mentre nella seconda le informazioni sono organizzate in un MIB che contiene le informazioni dei soli router locali di una zona.
Queste due politiche praticamente si equivalgono, la scelta è quindi molto soggettiva. La nostra preferenza è andata alla prima, soprattutto per i costi minori relativi alla gestione del DB e al minor numero di questi rispetto ai molti MIB.

Il compito del manager centrale è solo quello di leggere, quando lo ritiene necessario, i valori memorizzati nei vari manager intermedi. Il manager centrale, infatti, può essere interessato al monitoraggio del traffico in una determinata zona, di conseguenza le informazioni riguardanti la zona suddetta le richiederà al manager intermedio che la controlla.

4. CONCLUSIONI

Lo scopo di questo progetto è quello di fornire un'architettura per il monitoraggio del traffico che passa su n (n>1) router, utilizzati per connettere un'azienda ad internet. Ovviamente le scelte implementative fatte non sono sicuramente né le uniche né le migliori possibili, in quanto il caso in esame ha una complessità elevata e le variabili in gioco sono molteplici. Dobbiamo dire, però, che effettuando tali scelte, abbiamo cercato di avere una visione del problema quanto più vicina alla realtà.
Tra le varie tecniche di monitoraggio che potevamo scegliere, abbiamo adottato quella che ci dava un miglior rapporto tra costi e prestazioni.
Il quadro globale mostra sia l'utilizzo del management by delegation, sia paradigmi derivati da altre politiche di monitoraggio.
Il merge di tutte queste tecniche ci ha portato alla realizzazione di un'architettura semplice, funzionale ed espandibile in futuro.

5. SVILUPPI FUTURI

L'architettura che abbiamo sviluppato gestisce solo un parte delle problematiche che si incontrano nella gestione di una rete come quella in esame.
Gli ampliamenti che si possono effettuare sono molteplici. Il primo é quello di utilizzare RMON invece che il MIB II, in quanto esso si adatta molto bene al nostro problema e gestisce molte più situazioni di quelli prese in considerazione.

Un altro ampliamento possibile a questa architettura sarebbe quello di poter segnalare eventuali attacchi ai router.
Se si decidesse di utilizzare RMON, le possibili statistiche aumenterebbero decisamente e ci consentirebbero di avere un quadro della situazione più ampio di quello offerto dall'architettura attuale.
Alcune di queste modifiche potrebbero comportare ad esempio la nuova ripartizione dei manager sul territorio nazionale.

6. REFERENZE

J. Schonwalder, L. Deri "Sistemi di Elaborazione dell'Informazione: Gestione di Rete".
J. Kurose & K. Ross. "Internet e reti di calcolatori". McGraw-Hill 2001
GARR - Rete dell'Università e della Ricerca Scientifica Italiana. www.garr.it Sezione statistiche.
RFC 1213 MIB-II - K. McCloghrie
RFC 2819 RMON - S. Waldbusser