Appunti delle lezioni di Gestione di Sistemi e Reti
2006-2007

Franco Sirovich

4. I Concetti di Gestione (di Rete) di SNMP

SNMP è letteralmente l'acronimo di Simple Network Management Protocol,quindi l'acronimo del nome di un protocollo. In realtà, nell'uso comune indica una *raccolta di specifiche per la gestione* di reti (e non solo), che contengono la specifica del protocdlo ma anche la definizione di:

- Concetti importanti per la gestione
- Modalità per effettuare la gestione
- Strutture dati che si devono usare nella gestione

All'interno di questo corso cercheremo sempre di generalizzare l'esposizione alla gestione in generale di sistemi, mentre è tradizione parlare sempre solo di "network management",anche se è chiaro che gran parte delle considerazioni fatte si applicano in generale a qualunque sistema. Per questo abuso del termine network management spesso si ritiene che SNMP sia utilizzabile esclusivamente per gestire reti di comunicazioni: SNMP è invece adatto a gestire qualunque sistema distribuito, sia a livello di rete che a livello applicativo e sistemistico.

4.1. Retroscena (background)

SNMP ha seguito lo stesso modello di evoluzione seguito da TCP/IP e da tutta Internet. Vediamone gli aspetti salienti.

4.1.1. Le origini di TCP/IP

È opportuno tenere presenti le origini di TCP/IP perché influenzarono pesantemente anche il modo di sviluppare SNMP. Le abbiamo esposte in altri corsi e quindi rimandiamo a quanto già esposto.

4.1.2. Le origini di SNMP

Agli inizi dello sviluppo di TCP/IP fu data poca attenzione al problema della gestione di Internet e

delle sue applicazioni. L'attenzione alla gestione si limitò a inserire in ICMP delle funzioni utili alla gestione, quali i messaggi di *echo request* e *echo reply*, i messaggi per annunciare/richiedere i router di default e le maschere di sottorete, il messaggio di *redirect*. Altre funzionalità di IP (*source routing* e *record routing*) potevano essere usate per la gestione, ma erano molto limitate, anche se sufficienti a costruire semplici applicazioni di gestione. Ma quando la crescita degli host divenne esplosiva (fine anni '80), l'uso di Internet per costruire sistemi *mission critical*, cioè il cui malfunzionamento era critico per l'organizzazione che li utilizzava, e il numero di operatori necessari della gestione diventò talmente grande che non si poteva contare che tutti fossero espertissimi, ci si rese conto che servivano strumenti seri di gestione.

Furono allora sviluppati vari tentativi, come sempre in Internet iniziando con approcci semplici per non appesantire e ritardare lo sviluppo. I passi significativi sono stati:

- *Simple Gateway Monitoring Protocol (SGMP)*, per monitorare in modo semplice i router, standardizzato nel 1987; era però troppo semplice e quindi passarono a qualcosa di più ambizioso.
- *High-Level Entity Management System (HEMS)*, generalizzazione dell'*Host Management Protocol (HMP)* che è stato il primo protocollo di gestione sperimentato.
- *CMIP over TCP/IP (CMOT)* per incorporare in Internet la maggior parte possibile del *Common Management Information Protocol* di *ISO-OSI*. CMIP è un protocollo estremamente sofisticato, e gli sviluppatori di Internet non erano convinti che servisse davvero tale complessità. Inoltre, CMIP usava uno stack di protocolli sottostanti assai più complesso di TCP/IP (CMIP ha tutti e 7 i livelli OSI) e quindi pensarono che almeno nello stack di protocollo andava semplificato, appoggiandolo direttamente su TCP. L'operazione era però complessa perché i due livelli mancanti in Internet (Sessione e Presentazione) offrono servizi molto importanti per CMIP. Nel frattempo, ...
- *SNMP* come evoluzione di SGMP, cioè ancora un tentativo di restare sul semplice ma meno banale di SGMP.

Lo sviluppo di SNMP e di CMOT proseguì a lungo in parallelo, ma la velocità con cui il gruppo che definiva SNMP produsse dei risultati e invece la complessità della stabilizzazione di CMOT, indusse nel 1988 la *IAB* (*Internet Architecture Board*) a decidere che SNMP era la soluzione nel breve termine e CMOT nel medio termine, riconoscendo quindi uno spazio ufficiale a SNMP.

HEMS fu abbandonato perché più complesso di SNMP ma meno ricco di funzionalità di CMIP, e quindi non aveva senso. Per aiutare la transizione a CMIP (usando CMOT come transitorio) IAB cercò di stabilire che SNMP e CMIP dovessero usare le stesse variabili di monitoring e di controllo e quindi lo stesso schema di database e lo stessa data base di gestione. Ma ciò si è dimostrato impossibile perché gli oggetti di SNMP sono in realtà semplici variabili dotate di tipo e attributi di accesso SNMP, mentre gli oggetti di CMIP e quindi di CMOT sono oggetti nel senso pieno della tecnologia object-oriented. Fu quindi deciso di lasciare che CMOT evolvesse separatamente e indipendentemente. Questa decisione invece di aiutare CMOT ad affermarsi in realtà lo portò a scomparire, perché divenne sempre più complesso da definire, e SNMP vinse "sul mercato": SNMP diventò rapidamente il protocollo disponibile su tutti i dispositivi semplici, host, server, router, hub. Ciò avvenne perché lo standard era disponibile (mentre CMOT non era disponibile o ritenuto instabile) e semplice da implementare e quindi a basso costo (mentre CMOT era ritenuto, e in gran parte era, complesso e costoso da implementare. Per SNMP avvenne quello che avvenne in generale per Internet: essere uno standard disponibile, semplice e aperto favorisce l'adozione da parte degli utilizzatori e di conseguenza dei produttori: gli standard OSI divennero completi e disponibili molto più tardi, erano aperti nel senso che non erano proprietari ma non aperti nel senso della pubblica disponibilità del testo dello standard e nel senso della costruzione della rete (infatti soffrirebbero molto meno dei problemi di sicurezza che affliggono Internet), e ritenuti più costosi perché più complessi. Resterebbe da fare un bilancio complessivo oggi che abbiamo visibilità dei costi che derivano dalla mancanza di sicurezza e dalla eccessiva semplicità di Internet, che devono essere

"recuperate" in altro modo ad alto livello.

Un importante sviluppo di SNMP (nel senso della architettura di gestione, non del protocollo) fu la specifica di una funzione di *Monitoring Remoto* (*RMON*) che permette di monitorare una (sotto)rete come un unica entità. Questa specifica rese il monitoring di SNMP particolarmente performante ed adeguato alle esigenze delle gestioni di reti di comunicazione locali.

Il successo di SNMP nella gestione di semplici sistemi ha ovviamente portato alla applicazione di SNMP anche al caso di dispositivi/sistemi più complessi, e la cui gestione deve necessariamente essere più complessa: in questa sua evoluzione SNMP ha mostrato dei grossi limiti. Alcuni limiti sono stati superati da SNMPv2 (SNMP versione 2) e altri ancora in SNMPv3 (SNMP versione 3). SNMPv3 è molto completo ed adeguato, ma ha trovato difficoltà nell'adozione commerciale a causa della sua complessità e quindi del suo costo. I costruttori tendono a vedere la gestione come un costo addizionale nello sviluppo dei prodotti, e a non fornirla se non quando è "fortemente" richiesta dagli utilizzatori, e in tal caso con un costo aggiuntivo. Dalla loro parte, gli utilizzatori pretendono che il supporto per la gestione sia incluso nel prodotto sia dal punto di vista sistemistico che di prezzo e sono poco disposti a pagare un prezzo addizionale per la gestione. Il problema è acuito dalla presenza sul mercato di utilizzatori che ancora non hanno compreso la importanza degli strumenti di gestione e quindi sono pronti ad accettare prodotti che non hanno strumenti di gestione, così "dando ragione" ai costruttori che considerazione la gestibilità dei loro prodotti come un opzione a pagamento.

4.1.3. Gli standard relativi a SNMP

Gli standard relativi al protocollo SNMP e alle informazioni di gestione da utilizzare nella gestione di dispositivi/sistemi sono in continua crescita, proprio perché si tende ad applicare SNMP a sempre nuovi tipi di sistemi, ma le tre specifiche fondamentali, che stanno all a base di tutta l'architettura SNMP, sono:

- *Structure of Management Information (SMI)* per reti basate su TCP/IP (RFC 1155) che descrive come devono essere definiti i *managed object* che sono contenuti nel *database di gestione*
- *Management Information Base (MIB-II)* che descrive i managed object *minimi* che devono essere contenuti nel database di gestione, e che sono indispensabili alla gestione stessa
- *Simple Network Management Protocol (SNMP)* che definisce il protocollo che deve essere usato per operare sui managed object

Anche gli standard fondamentali hanno subito una evoluzione (come si vede anche dal fatto che MIB-II porta la numerazione delle versioni nel titolo!) ma la evoluzione principale si è avuta sul fronte delle MIB specifiche per i vari tipi di dispositivi.

4.2. Concetti Fondamentali

4.2.1. Architettura di Gestione

Gli standard definiscono un modello per la gestione che contiene quattro elementi chiave:

- *Stazione di gestione*, detta anche *manager*
- *Agente di gestione*, detto anche per brevità *agent*
- *Base informativa di gestione* (*management information base, MIB*), cioè l'insieme dell'informazione che permette di effettuare la gestione di un dispositivo/sistema
- *Protocollo di gestione*, che permette di operare sulla informazione di gestione e in questo modo effettuare la gestione del dispositivo/sistema descritto dall'informazione di gestione

La *stazione di gestione* è in realtà una *elemento funzionale* che può essere implementato su una macchina stand-alone (un PC o una workstation) oppure su una macchina condivisa. Il termine "stazione" è ingannevole perché fa pensare ad un componente hardware; forse il termine "manager" è più neutrale anche se se può far pensare ad un operatore che effettua la gestione (noi useremo sempre i termini *operatore* o *amministratore* per indicare la persona o le persone che effettuano la gestione di un sistema).

Quale che sia il modo con cui viene implementata, la stazione di gestione deve fornire una *interfaccia utente* all'operatore. Come minimo la stazione di gestione avrà:

- Insieme di applicazioni di gestione per
  - Analisi dei dati
  - Risoluzione dei guasti
  - ...
- Una interfaccia utente che permetta sia il monitoring che il controllo della sistema gestito
- La capacità di tradurre i requisiti di gestione in monitoring e controllo effettivo, quindi strumenti operativi che permettano la effettuazione concreta della gestione
- Un database di informazioni estrate dalle MIB dei componenti gestiti

Solo gli ultimi due elementi funzionali sono soggetti a standardizzazione, perché sono quei componenti che necessitano di interoperabilità fra stazione di gestione e agente di gestione. I primi due componenti invece consistono in funzionalità ed elaborazioni esclusivamente interne alla stazione di gestione, e quindi qualunque sviluppatore di stazioni di gestione può operare come preferisce per avere successo sul mercato.

L'agente di gestione (*management agent*) è quell'elemento funzionale che permette alla stazione di gestione di gestire effettivamente i componenti che sono sottoposti a gestione. L'agente di gestione risponde alle richieste di informazione e di azione che gli giungono dalla stazione di gestione, e può fornire alla stazione di gestione, in modo asincrono e non sollecitato, importanti informazioni di gestione

Le risorse sono gestite rappresentandole come oggetti, che sono essenzialmente una *variabile* che rappresenta un semplice aspetto dei componenti gestiti dall'agente. La collezione di questo oggetti viene chiamata *management information base* (*MIB*). Può essere considerata una collezione di punti di accesso, situati sull'agent, che permettono alla managementstation di interagire con le risorse.

Tutte le risorse di un particolare tipo (ad es. i bridge) è bene che siano gestite tramite esattamente lo stesso insieme di management object. Infatti solo in questo modo le operazioni di gestione di una ampio parco di risorse fornite da costruttori diverso può essere gestito con le stesse procedure. È un vantaggio per gli utilizzatori poter scegliere di momento in momento il costruttore e il modello più conveniente senza pagare un alto prezzo nella gestione di questo parco di apparati disomogeneo. I costruttori sono invece interessati a rendere la gestione dei loro apparati diversa daquella dei concorrenti, sia per poter introdurre delle funzioni aggiuntive che forniscano loro vantaggi competitivi (e questa è una sana motivazione) sia per "catturare" gli utilizzatori rendendo loro costoso cambiare fornitore (e questa motivazione non è "sana"!). Ma l'unico modo di ottenere che apparati di costruttori diversi siano gestiti nello stesso modo è quello di definire uno standard (e questo è infatti stato fin dall'inizio l'obiettivo che ha prodotto lo sviluppo di Internet) e ottenere il sostegno degli utilizzatori che impongano l'adozione degli standard da parte dei costruttori, rifiutandosi di acquistare prodotti che non rispettano gli standard. Nel caso della gestione, è importante apprezzare che non basta standardizzare il protocollo che permette alla stazione di gestione di interagire con l'agente di gestione: occorre ancora di più standardizzare il modello dei dati di gestione che permettono di gestire

una particolare tipo di risorse. È come dire che per operare su un particolare database relazionale non basta conoscere il linguaggio SQL: occorre anche conoscere il particolare modello dei dati che sono immagazzinati nel DBMS.

Il monitoring viene effettuato “leggendo” i valori dei management object; il controllo della configurazione si ottiene settando il valore di managed object; le azioni di gestione vengono invocate settando valori di particolari managed object, perché in SNMP non sono presenti elementi di protocollo che permettono esplicitamente di invocare azioni sull'agente.

Management station e management agent devono condividere un protocollo per poter interagire: *network management protocol*. In Internet questo protocollo è SNMP. E ovviamente un protocollo applicativo, che fornisce tre funzioni chiave:

- *get*: permette alla stazione di gestione di recuperare il valore di uno o più managed object
- *set*: Permette alla stazione di gestione di assegnare valori di uno o più managed object
- *trap*: permette all'agente di gestione di notificare alla stazione di gestione un evento significativo che si è verificato

Lo standard non specifica il numero degli elementi funzionali di gestione da utilizzare, ma per quanto riguarda le stazioni di gestione è prudente averne almeno due per evitare che un guasto di una stazione renda completamente impossibile la gestione del sistema. Per quanto riguarda gli agenti di gestione, è necessario che siano installati quanti sono necessari per gestire tutte le risorse critiche o importanti del sistema (vanno ovviamente identificate perché potrebbero essere diverse a seconda dell'uso del sistema. A parità di risorse di gestione, è meglio che il numero di agent sia il più piccolo possibile. Fino a che SNMP rimane "semplice" gli agenti di gestione possono anche essere numerosi (centinaia) ma ci sono limiti che non si possono superare (come vedremo), perché l'interazione fra la stazione e l'agente è costosa in termini di occupazione della rete di comunicazione e di potenza di calcolo sulla stazione.

4.2.2. Architettura del Protocollo di Gestione

SNMP è stato progettato come protocollo applicativo della suite TCP/IP di Internet; usa il protocollo di trasporto UDP. La prossima figura illustra una tipica configurazione:

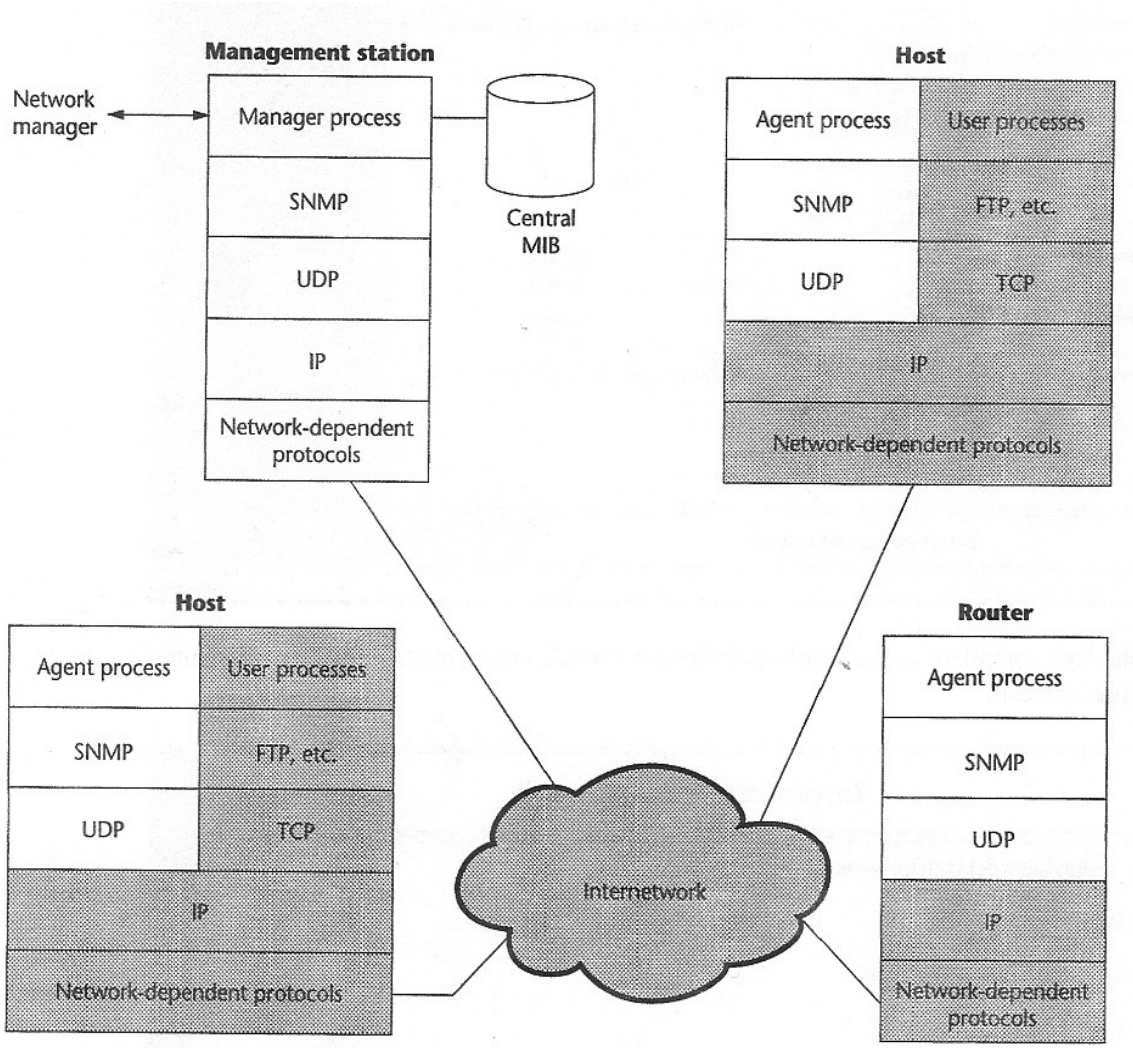

FIGURE 4.1 Configuration of SNMP

La figura evidenzia una stazione di gestione, dedicata e unica nel sistema. Questa configurazione è (o dovrebbe essere) molto rara (è illustrata così per semplicità) perché presenta un grave singolo punto di fallimento del sistema di gestione.

Facciamo attenzione che il termine "manager" viene usato sia per componenti software che per il gestore del sistema (umano). Un processo *manager* controlla l'accesso ad una MIB centrale che contiene l'informazione di gestione di tutto il sistema gestito. Questa collezione di informazione di gestione è in genere ospitata su database (non necessariamente,ma spesso relazionale) e contiene la "storicizzazione" delle informazioni di gestione che sono state considerate importanti per la gestione del sistema. Ciò significa che se una informazione è ritenuta importante, La MIB centrale conterrà i valori che questa informazione ha assunto nel passato, con una frequenza in genere più alta per i valori recenti (ad es. un valore ogni quarto d'ora, o la media del valore in ogni quarto d'ora), e via via più contenuta mano mano che si va nel passato (ad es., per i valori assunti l'anno passato posso accontentarmi di avere un valore alla settimana, o la media settimanale). Il database che ospita la MIB deve essere corredato di funzionalità e filtri per produrre report e e grafici sull'andamento di un valore della MIB nel tempo oppure in parti diverse del sistema.

La stazione di gestione fornisce una interfaccia utente al gestore del sistema, permettendogli non solo di esaminare i valori delle informazioni di gestione contenute negli agent, ma anche di invocare azioni di gestione programmate in un qualche opportuno linguaggio, e di analizzare ed esaminare i dati contenuti nella MIB centrale.

La management station realizza la gestione usando il protocollo SNMP, che a sua volta usa UDP, IP e i protocolli di rete fisica. Ogni agente di gestione deve implementare SNMP, UDP e IP, e i protocolli di rete fisica. Interpreta i messaggi SNMP inviatigli dalla stazione e controlla la MIB delle risorse che gestisce. Se supporta anche altre applicazioni Internet (ad es. FTP) deve avere anche TCP. Nell'esempio illustrato sopra, la parte ombreggiata indica l'ambiente che deve essere gestito. Vediamo più in dettaglio.

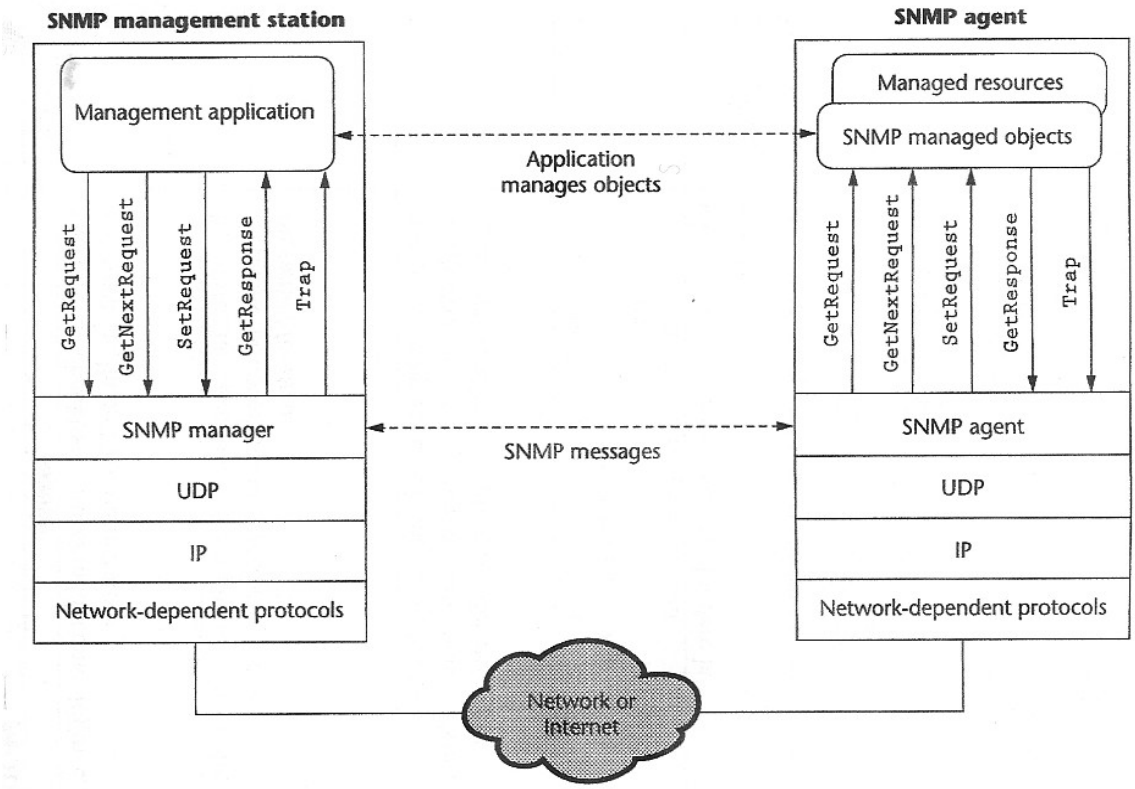

FIGURE 4.2 The role of SNMP

Come abbiamo visto anche a Reti 1, due entità di protocollo interagiscono scambiandosi *Protocol Data Unit* (*PDU*), cioè messaggi di protocollo. L'applicazione di gestione gestisce *managed objects*, quindi interagisce con le risorse gestite usando (in termini di) *managed objects*. A livello SNMP, il manager e l'agent si scambiano messaggi SNMP. Queste sono "interazioni orizzontali", effettuate affidando al livello sottostante (UDP e IP) il compito di recapitare le PDU.

L'applicazione di gestione può chiedere all'entità SNMP manager di inviare tre tipi di PDU:

- `GetRequest`
- `GetNextRequest`
- `SetRequest`

Le prime due sono richieste di informazione, la terza è una richiesta di assegnamento dell'attributo di managed objects. Tutti e tre i messaggi sono riscontrati dall'agent mediante la PDU `GetResponse`; le informazioni contenute nelle `GetResponse` sono consegnate all'applicazione di gestione.

L'agent può inviare anche PDU di `Trap`, reagendo ad eventi che avvengono nei managed object, e le informazioni contenute nelle `Trap` vengono passate all'applicazione di gestione.

SNMP è connection-less, così come lo è UDP; ci sono buone ragioni per essere connection-less e

usare un trasporto connection-less: verranno discusse in seguito.

4.2.3. Polling Guidato da Trap (Trap-Directed Polling)

Quando una stazione di gestione è responsabile per un gran numero di agenti, non è accettabile che periodicamente richieda il valore di tutti i managed object di tutti gli agenti, per molte ragioni:

- Tutte queste informazioni non sono in realtà necessarie, e certamente non con una alta frequenza
- Procurarsi con alta frequenza questa mole di informazioni provoca un aumento del traffico sulla rete di telecomunicazione che, anche se fosse accettabile in termini di capacità, rende eccessivo il costo della gestione
- Procurarsi con alta frequenza questa mole di informazioni richiede un elevata quantità di CPU non solo sulla macchina che ospita la stazione di gestione, il che non è tanto grave perché ve ne sono poche nella rete, ma negli agenti di gestione; questo carico di CPU è assai più grave perché richiede un aumento del dimensionamento del dispositivo/risorsa che ne aumenta il costo

La strategia suggerita (polling guidato da trap) è invece la seguente:

- Molto raramente (ad es. una volta al giorno) la stazione esegue un polling completo di tutti gli agent, richiedendo non tutte le informazioni disponibili nella MIB degli agent ma solo quelle che sono ritenute essenziali per la gestione del sistema (ad es. di configurazione e di prestazione)
- Costruita questa *linea di riferimento* (*baseline*) si fa affidamento su eventi segnalati dall'agent, oppure, come vedremo, su un periodico polling di poche informazioni critiche, che segnalino il non corretto funzionamento del sistema
- Una volta avuta informazione della esistenza di un possibile problema, la stazione di gestione effettua polling mirati per accertare l'esistenza di un problema e individuarne le cause

Questa strategia di trap directed polling conduce a:

- Riduzione della capacità della rete utilizzata dalla gestione
- Riduzione della potenza di calcolo necessaria sugli agent

Si noti che questa strategia richiede una attenta analisi preliminare

- dei dati delle MIB che sono rilevanti per gli obiettivi di gestione del sistema (che variano da caso a caso, a seconda dell'uso e della rilevanza del sistema sotto gestione);
- dei danni provocati da ciascun malfunzionamento per decidere su quali occorre intervenire con prontezza e in quali tempi per effettuarne la riparazione;
- delle informazioni che ci permettono di rilevare il malfunzionamento, o ci danno indicazione della possibile esistenza di tale malfunzionamento;
- delle informazioni che ci permettono di accertare la esistenza del malfunzionamento;
- delle informazioni che ci permettono dì effettuare la diagnosi, ossia di individuare le cause che devono essere rimosse (il guasto).

Queste analisi conducono a risultati diversi a seconda della composizione del sistema da gestire e della rilevanza del sistema all'interno della organizzazione che lo utilizza per i propri scopi. Non è quindi possibile fare questa analisi in modo generale con risultati che valgano per qualunque sistema.

4.2.4. Proxy

L'uso di SNMP richiede che l'agent supporti tutta lo stack SNMP/UDP/IP: questa requisito può essere difficile o troppo costoso da soddisfare perché:

1. Dispositivi di piccola potenza di calcolo e memoria potrebbero non essere in grado di ospitare ed eseguire tutto il software necessario per un agent di gestione; anche se dispongono di funzionalità che usate opportunamente potrebbero permettere un certo livello di gestione del dispositivo.
2. Il dispositivo supporta TCP/IP ma per altre ragioni non si vuole che abbia il carico aggiuntivo di ospitare un agente SNMP: tipicamente la esecuzione di un agente gestione richiede una maggiore memoria e una maggiore potenza di calcolo. Non solo queste richieste aumentano il costo del dispositivo, ma lo aumentano in modo non lineare "a gradini". Il costruttore potrebbe trovarsi in una situazione in cui il ridotto aumento di memoria e potenza di calcolo lo costringe a installare un banco di memoria aggiuntivo oppure passare al modello superiore di CPU. Con un incremento di costo molto maggiore della "quota" che poi è effettivamente utilizzata dall'agent.
3. Il dispositivo potrebbe essere gestibile con protocolli diversi da SNMP, perché il costruttore non ha ritenuto opportuno dotarlo di gestione SNMP oppure perché è un dispositivo "vecchio" (*legacy*).

Per queste ragioni è stato sviluppato il concetto di *proxy agent* illustrato nella prossima figura.

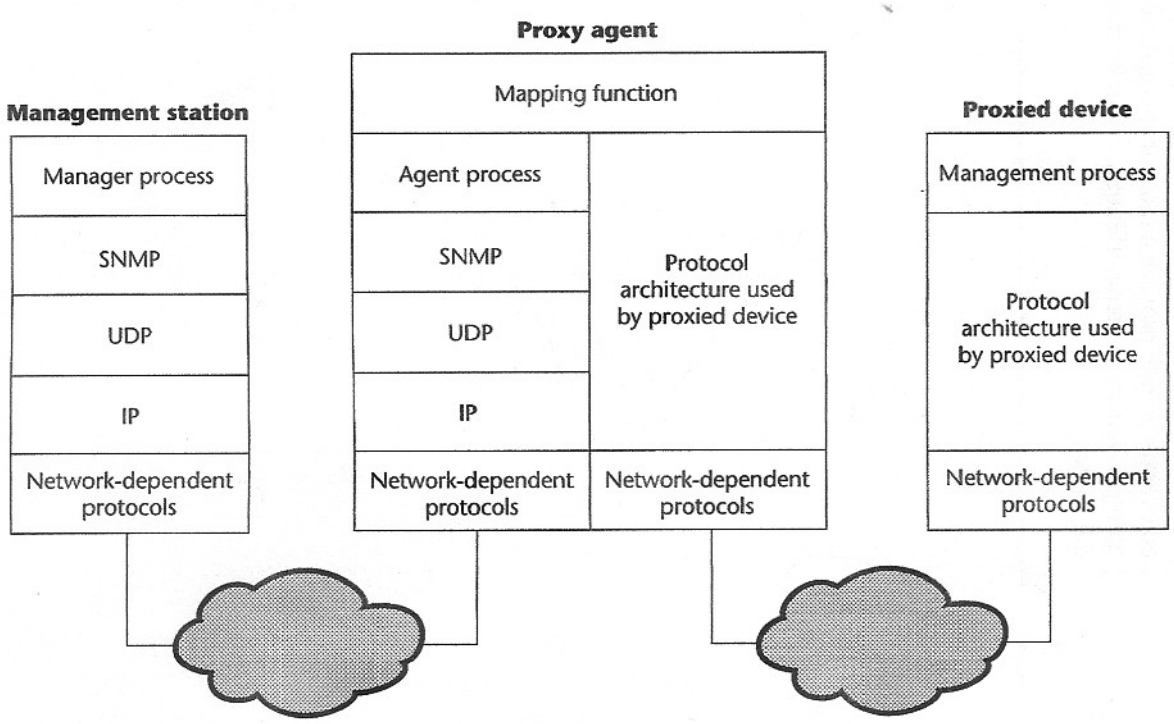

FIGURE 4.3 Proxy configuration

Questa figura fa riferimento da un dispositivo che comunque possieda una architettura di gestione, anche se diversa da SNMP. I casi (1) e (2) precedenti ricadono in questa casistica considerando i comandi che il dispositivo è in grado di accettare come comandi che realizzano un qualche livello di gestione del dispositivo (in genere molto limitato se non è stata sviluppata una vera e propria architettura di gestione. Un dispositivo sottosistema che non offra alcun comando invocabile da remoto che ne permetta un qualche livello di gestione ... non potrà essere gestito in alcun modo, a meno di sviluppare un agent di gestione direttamente "a bordo" del dispositivo; ma anche in questo

caso il dispositivo dovrà offrire della interfacce programmative che permettano di interagire con le risorse che intendiamo gestire e che ne permettano la gestione.

La cosa forse più importante da notare è la *funzione di traduzione* (*mapping*) che risulta necessaria perché il "modello" con cui SNMP gestisce il dispositivo è praticamente in ogni caso diverso da quello che il dispositivo offre tramite i suoi comandi di gestione. Non vi sarà quindi una corrispondenza uno-a-uno fra operazioni SNMP e comandi di gestione "nativi" né una corrispondenza uno-a-uno fra informazioni di gestione SNMP e informazioni di gestione offerti in modo nativo dal dispositivo. Occorre quindi "calcolare" le informazioni SNMP a partire dalla informazioni native e implementare i comandi SNMP in termini dei comandi nativi. Più precisamente la funzione di traduzione trasforma:

- Una richiesta SNMP in una o più richieste nella architettura di protocollo usata dal sistema "proxato"
- Una o più risposte del sistema proxato in una risposta da inviare alla stazione di gestione
- Una o più segnalazioni di eventi eccezionali verificatisi sul dispositivo in una o più segnalazioni di eventi SNMP

I proxy sono una realtà che, lungi dallo sparire con la diffusione di SNMP, si incontra sempre di più. Infatti, la diffusione di SNMP rende molto comune la disponibilità di un ambiente di gestione basato su SNMP (stazione di gestione) molto ricco di funzionalità: è quindi sempre più conveniente trattare in modo uniforme e con strumenti potenti anche applicazioni per le quali gli standard di gestione non sono ancora stati sviluppati. In questi casi, lo sviluppo di un proxy è la strada più veloce e meno costosa per sottoporre a gestione queste "risorse".