

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

Franco Sirovich

© Franco Sirovich ¹

1. Che cosa è un sistema di gestione?

1.1. Quando serve un sistema di gestione?

La cosa inizialmente più difficile da afferrare è il ruolo e lo scopo di un sistema di gestione, in quanto è un componente di un sistema informatico con il quale, nella esperienza comune di utilizzatori del sistema, non si viene a contatto. È un componente che viene usato dal personale che si occupa del corretto funzionamento del sistema, e non dagli utilizzatori finali del sistema. Questa distinzione fra "utilizzatori" va tenuta ben presente quando si parla di gestione: occorre non confondere gli utilizzatori del *sistema gestito* con gli utilizzatori del *sistema di gestione*.

I sistemi informatici vengono costruiti e messi in campo per rispondere alle esigenze dei loro utilizzatori. Per essere utili, oltre ad avere funzionalità ben concepite ed essere ben realizzati, devono "funzionare": un bel sistema che fa cose meravigliose quando funziona ma che è spesso (e soprattutto quando se ne ha bisogno) malfunzionante ("down"), non risponde alle esigenze degli utilizzatori. Ciò che è difficile credere, se non si ha esperienza, è che un sistema informatico deve essere *gestito* se si vuole che *continui a funzionare* come da sue specifiche: pare incredibile che lasciato a se stesso rapidamente diventi inutilizzabile. Per questa ragione si sviluppano *sistemi di gestione*.

Un Sistema di Gestione deve permettere di

- monitorare in maniera automatica tutto il sistema informatico,
- analizzare il guasto,
- aiutare a individuare la probabile causa,
- attivare operazioni automatiche di riparazione o di richiesta di intervento umano.

Il Sistema di Gestione non elimina la necessità di personale di gestione, ma permette una gestione più efficace e possibilmente *proattiva*, cioè che anticipa i problemi ed evita che essi si verifichino. Una gestione *reattiva* invece permette soltanto di reagire ad un problema quando questo si verifica. Chiaramente abbiamo bisogno sia di una gestione proattiva, che di una gestione reattiva perché una gestione proattiva non può evitare completamente che si verifichino problemi o guasti. I sistemi

¹ Quest'opera è stata rilasciata sotto la licenza Creative Commons Attribuzione-Non commerciale-Non opere derivate 2.5 Italia. Per leggere una copia della licenza visita il sito web <http://creativecommons.org/licenses/by-nc-nd/2.5/it/> o spedisci una lettera a Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

informatici devono essere gestiti, e per questo si sviluppano sistemi di gestione.

GSR è un corso sulla *gestione di sistemi distribuiti*, cioè sistemi costituiti da macchine ed applicazioni connesse in rete. Quando si deve gestire un sistema distribuito di deve anche gestire la rete che interconnette le macchine. Gli aspetti di *comunicazione* diventano quindi quelli più importanti, e quanto più numerosi sono i *componenti* di un sistema tanto più numerose sono le fonti di malfunzionamento. Quindi la gestione dei sistemi distribuiti è molto più complessa, ma anche più necessaria, della gestione di un sistema informatico che non utilizza sistemi di comunicazioni ed è quindi costituito da componenti che eseguono tutti su un unico sistema di calcolo.

GSR espone la tematica della gestione partendo dal punto di vista del sistema di comunicazione. Internet ha sviluppato una *architettura di gestione* (basata su opportuni *standard* detti, nella terminologia di Internet, *RFC* o *Request For Comments*) che si applica sia agli apparati di rete che alle macchine ed alle applicazioni. Questa architettura è nota con il termine *SNMP: Simple Network Management Protocol*, che letteralmente è il nome del protocollo applicativo usato in questa architettura; un esempio della figura retorica *sineddoche*, che consiste nell'usare la parte per indicare il tutto.

Internet non è stata la prima ad affrontare il problema della gestione. Ovviamente esistono standard e architetture *proprietarie* sviluppate dai costruttori che sono interessati a vendere ai clienti il sistema gestione assieme al sistema informatico da gestire. Le architetture proprietarie sono sostanzialmente simili a quelle di Internet: conoscere SNMP permette rapidamente di impadronirsi degli strumenti e delle metodologie di architetture proprietarie. Ma esiste anche una seconda architettura di gestione *aperta e vendor independent*, come SNMP, che è stata definita dall'ISO (*International Standard Organisation*) nell'ambito della serie di protocolli chiamata OSI (*Open System Interconnect*) e viene spesso indicata con il termine *OSIMIS (OSI Management Information System)*. OSIMIS è stata definita a partire dalle esigenze delle grandi reti di telecomunicazioni, ancora in uso nelle reti pubbliche geografiche basate su tecnologia SDH (Synchronous Data Hierarchy), ATM (Asynchronous Transfer Mode) e altre. OSIMIS è molto più complesso di SNMP, ma SNMP deve molte sue idee a OSI Management e quindi costituisce una buona introduzione a OSIMIS, che vi tornerà utile se doveste lavorare con OSIMIS.

Perché servono degli *standard* per gestire i sistemi informatici? Perché se tutti i componenti del sistema informatico implementano la stessa architettura di gestione, anche se sono realizzati da diversi produttori, quindi aderiscono ad uno *standard aperto e indipendente dal fornitore* il cliente può gestire diversi componenti con la stessa infrastruttura di gestione (vedremo che cosa è) ed è libero di acquistare i componenti del suo sistema da fornitori diversi, scegliendo l'offerta di volta in volta più conveniente. Le architetture proprietarie invece vincolano il cliente ad acquistare sempre e solo componenti dallo stesso fornitore perché gli altri fornitori sono messi in condizione di non avere le informazioni necessarie per gestire i propri componenti con l'architettura di gestione di un altro fornitore. Nel caso di sistemi distribuiti il numero e il tipo dei componenti del sistema è molto alto e quindi l'effetto "prigione" di una architettura proprietaria è ancora maggiore.

Storia di SNMP

SNMP (Simple Network Management Protocol, detto anche *SNMPv1*) fu definito inizialmente negli ultimi anni '80, avendo in mente, come dice il nome, uno standard estremamente semplice ed economico da implementare. L'aspetto economicità e semplicità è particolarmente rilevante per i sistemi di gestione, perché gli strumenti di gestione sono facilmente considerati un "sovrappeso", che si vorrebbe evitare di pagare (almeno, fino al primo grave malfunzionamento del sistema gestito!).

RMON (Remote Network Monitoring), costruito su SNMP, fu rilasciato nel 1991 e rivisto (RMON2) nel 1995. RMON permette, come dice il nome, di monitorare (cioè tenere sotto controllo) da remoto intere reti LAN, piuttosto che ciascuno dei componenti di rete; definisce quindi algoritmi e database per gestione di LAN.

La semplicità di SNMP ne favorì un grande successo commerciale, sia tra i costruttori che soprattutto fra gli utilizzatori, che ne apprezzarono l'aspetto "aperto" e "multivendor". Ma assieme al successo commerciale risultarono ben presto evidenti le troppe limitazioni dell'impostazione di SNMPv1.

Fu quindi preparata una seconda versione dello standard chiamata SNMPv2, che fu standardizzata nel 1993 e rivisto nel 1995. Questa seconda versione è caratterizzata da

- Maggiore efficienza nell'uso della infrastruttura di rete
- Maggiori funzionalità rispetto a SNMPv1

Ma anche SNMPv2 soffre di pesanti limitazioni funzionali e ciò portò alla definizione di SNMPv3 che fu standardizzato nel 1998, e che costituisce un progresso estremamente significativo rispetto a SNMPv2:

- Fornisce un quadro complessivo per le presenti e future versioni di SNMP
- Aggiunge funzionalità di sicurezza a SNMP

Inevitabilmente SNMPv3 risulta molto più complesso e costoso di SNMPv1 e di SNMPv2 e quindi ha avuto un successo commerciale molto inferiore.

Cercheremo di realizzare esercitazioni, che faranno uso di un sistema di gestione installato nel Laboratorio e di uso molto comune nelle aziende: *HP Open View*. Le esercitazioni non saranno fiscalizzate (anche questo anno): non devono essere "superate" per poter fare l'esame. L'esperienza con il corso di GSR per la laurea triennale ha mostrato che frequentare assiduamente le esercitazioni e comprendere bene tutto quello che vi viene presentato nelle esercitazioni è molto utile per passare l'esame senza fare una fatica eccessiva. Se avremo tempo e "risorse umane" offriremo le esercitazioni anche quest'anno.

- Lo scopo delle esercitazioni è
- Fornire strumenti di comprensione
- Fornire strumenti di validazione dello studio fatto e della comprensione raggiunta

Alla fine del Corso vorremmo che lo Studente fosse capace di

- Configurare un semplice agente di gestione
- Analizzare, progettare e realizzare piattaforme di gestione utilizzanti SNMP
- Compiere una valutazione dello stato di un sistema distribuito basandosi sugli elementi raccolti dal sistema di gestione.

La maturità e la profondità con cui questi obiettivi sono raggiungibili da uno studente della laurea magistrale è ovviamente molto maggiore di quella raggiungibile da uno studente della laurea triennale.

Ma quali sono, con maggior precisione, le funzionalità che deve offrire un sistema di gestione? È importante definirle perché uno standard di gestione deve essere funzionale a raggiungere questi obiettivi.

1.2. Requisiti di un Sistema di Gestione

I requisiti di un sistema di gestione sono stati definiti con chiarezza dall'ISO in ambito OSI, definendo le seguenti aree funzionali che un Sistema di Gestione dovrebbe rendere disponibili:

- Fault Management
- Accounting Management

- Configuration Management
- Performance Management
- Security Management

Queste aree di intervento sono state riconosciute generalmente valide in quanto "discendono" dai requisiti degli utenti (amministratori) riguardo ad un sistema di gestione; alcune di queste aree devono essere supportata da un *protocollo applicativo di gestione* (quale SNMP), altre da *applicazioni di gestione* che vengono sviluppate al di sopra del (utilizzando il) protocollo di gestione. Quindi il protocollo di gestione deve permettere di costruire applicazioni di gestione, ma non deve necessariamente realizzare completamente queste funzioni al suo interno.

Notate che anche se il protocollo di gestione è una *applicazione* in termini di architettura Internet, non risolve necessariamente *tutti* i problemi di gestione: deve invece costituire una *base* su cui costruire applicazioni di gestione che rispondano alle esigenze degli amministratori del sistema. Internet ha ripreso queste aree funzionali, ma la sua architettura di gestione non risponde in modo ugualmente soddisfacente a tutte.

Analizziamo le aree funzionali perché ci permettono di capire i requisiti di utente che un Sistema di Gestione dovrebbe soddisfare. Riprendiamo il tema della chiara definizione di *utente* del sistema di gestione. La prima risposta è che l'Utente è il *Gestore del Sistema*, che chiameremo genericamente "amministratore", anche se in molte realtà aziendali l'amministratore ricopre un rango più elevato, rispetto ai generici "sistemisti". Ma il Gestore del Sistema in realtà risponde alle richieste e alle proteste degli utilizzatori del Sistema che viene gestito, almeno *se* è un *buon* gestore e *se* l'azienda per cui lavora ha davvero come obiettivo la soddisfazione dei Clienti!

1.2.1. Fault Management

I guasti vanno ovviamente gestiti: cosa vuol dire? Quando si ha un guasto, occorre nel più breve tempo possibile:

- Determinare esattamente dove si è verificato
- Isolare il resto della rete/sistema dal guasto in modo che possa continuare a funzionare
- Riconfigurare la rete/sistema in modo da minimizzare l'impatto del componente guasto (è il cosiddetto *workaround*)
- Riparare o rimpiazzare il componente e riportare la rete/sistema al suo stato iniziale normale

Gli utenti finali si aspettano una soluzione immediata o comunque veloce, e si aspettano che le prestazioni e le funzionalità del sistema gestito siano il più possibili all'altezza del sistema senza il guasto. Quindi quello che occorre minimizzare è il *tempo di mancanza di servizio*, o *di servizio ridotto*, che costituisce il fattore determinante della *Qualità del Servizio (Quality of Service, QOS)*. Naturalmente, questa è la misura di una variabile "stocastica", non si presenta in ogni caso con lo stesso valore; occorre quindi minimizzare in primo luogo la *media annuale* del tempo di mancanza di servizio o di servizio ridotto, e in secondo luogo minimizzare la *varianza* nell'annodi questa misura o il *tempo massimo* nell'anno di mancanza di servizio.

La individuazione e la diagnosi del guasto devono essere il più veloci possibile, perché concorrono a formare il tempo di mancanza di servizio. Un buon Sistema di Gestione minimizza questi tempi, automatizzando il più possibile queste attività.

È importante capire fino dall'inizio che gli utenti finali possono solo segnalare che qualcosa non funziona perché il sistema non fornisce il servizio atteso; una segnalazione da parte degli utenti finali del sistema informatico è già un fallimento del sistema informatico. Le statistiche sul comportamento degli utenti indicano che solo una piccola percentuale degli utenti segnala il malfunzionamento ai fornitori del servizio: la maggior parte degli utenti semplicemente prende provvedimenti senza

protestare, ma questi "provvedimenti" non sono mai in favore del fornitore del servizio! Quindi basarsi sugli utenti finali per la segnalazione dei guasti è la peggior pratica che si possa adottare.

Un Sistema di Gestione invece può e deve tenere sotto controllo tutti i componenti del sistema e accorgersi da solo del loro malfunzionamento, prima che (e senza che) gli utenti finali segnalino il problema. Per questo una gestione proattiva è molto migliore di una gestione reattiva. Ovviamente, la riparazione del guasto deve essere effettuata dal personale di gestione, che però se non viene avvertito non può intervenire. Quindi tutti i meccanismi di segnalazione dei problemi al personale di gestione, ad es. tramite messaggi SMS, sono parte integrante del Fault Management.

Eseguire Fault Management richiede inevitabilmente interagire con il sistema gestito per raccogliere dati sul suo funzionamento per verificare il suo corretto funzionamento. Questa interazione con il sistema gestito non deve abbassare significativamente la prestazione del sistema.

1.2.2. Accounting Management

Per *accounting* si intende in generale *contabilizzazione* dei costi o dell'uso delle risorse del sistema. Anche se il sistema gestito non fornisce servizio a Clienti esterni all'Azienda (come invece succede nelle aziende che vendono il servizio) l'uso delle risorse del sistema deve (spesso) essere contabilizzato ed addebitato agli utilizzatori, perché

- Un utente o un gruppo di utenti può abusare delle risorse messe a disposizione e condivise
- Gli utenti possono fare un uso inefficiente delle risorse del sistema

Sempre più spesso l'uso delle risorse del sistema viene addebitato agli utenti interni del sistema gestito perché questa pratica permette di capire come e dove vengono usate le risorse. Se il gestore del sistema conosce il livello e il modo di utilizzo delle risorse è in migliori condizioni per pianificare la crescita del sistema, che è l'attività indispensabile per una gestione proattiva del sistema.

Il gestore del sistema deve essere in grado

- Di definire il tipo di informazioni di accounting che il sistema di gestione deve raccogliere nei vari punti del sistema
- L'intervallo di tempo per la raccolta delle informazioni, e l'invio ai livelli più alti di gestione dell'informazione
- Gli algoritmi usati per calcolare la contabilizzazione
- La frequenza e il formato dei report di accounting

Solo una parte molto limitata di queste funzionalità può essere fornita direttamente dal protocollo applicativo di gestione: questo è una delle aree in cui è necessaria la maggiore "customizzazione", o personalizzazione, della gestione di una sistema informatico, anche perché i dati raccolti devono essere forniti al sistema di contabilizzazione usato all'interno dell'azienda, che è ben lungi da essere "standard".

1.2.3. Configuration and Name Management

I sistemi moderni sono composti di componenti e sottosistemi logici che possono/devono essere configurati per effettuare una grande varietà di funzioni in una grandissima varietà di modi. Una volta deciso il modo di utilizzo del componente, il Sistema di Gestione deve aiutare il gestore a scegliere il software appropriato e il set dei valori delle variabili di configurazione che sono appropriati per il software usato e per le funzioni desiderate.

L'area del Configuration Management riguarda anche l'inizializzazione e lo shutdown di una rete, o di un sistema, o di un suo componente. Spesso è richiesto che operazioni sui componenti siano compiute automaticamente, ad es. *shutdown* o *startup* di una scheda di interfaccia di rete. Configuration Management deve permettere una facile riconfigurazione della rete, o la

configurazione della rete secondo modelli predefiniti e archiviati. Infatti, spesso è necessario tenere in archivio diverse configurazioni della rete o dei componenti e poter facilmente passare da una configurazione ad un'altra per rispondere a necessità che si verificano con una certa regolarità.

Un esempio di configurazione? Assegnare l'indirizzo IP ad una interfaccia, impostare il nome di dominio di una macchina, ecc.

1.2.4. Performance Management

La gestione delle prestazioni è un'area estremamente importante oggi, in quanto l'efficacia delle applicazioni spesso si basa sulle prestazioni di componenti del sistema che hanno un ruolo critico nel determinare le prestazioni complessive. Questa area può essere scomposta in *monitoring* e *control*. Nella gestione delle prestazioni, occorre essere in grado di rispondere a domande quali:

- Quale è il livello di utilizzo della capacità del componente?
- C'è traffico/carico eccessivo?
- A che livello è il throughput del componente? È ridotto a livelli eccessivi?
- Ci sono colli di bottiglia nel sistema?
- A che livello è il tempo di risposta del sistema o di un suo componente?

L'analisi delle prestazioni, e soprattutto la previsione dell'andamento delle stesse, è impossibile se non si dispone di un modello del sistema, anche se semplificato. La previsione si basa inevitabilmente sulla raccolta di statistiche sui dati di prestazione del sistema e dei suoi componenti, che devono essere raccolte con regolarità, e archiviate in un opportuno *database*.

1.2.5. Security Management

Riguarda la gestione di meccanismi di protezione e di controllo degli accessi, quali la generazione, distribuzione e memorizzazione di

- Chiavi di cifratura
- Password
- Informazione di controllo degli accessi

La gestione della sicurezza richiede in primo luogo di monitorare e controllare l'accesso alle reti e ai sistemi. È quindi essenziale tenere sotto controllo ed analizzare i log di accesso e dei tentativi di violazione della sicurezza. Un tentativo fallito di accesso al sistema gestito può essere solo causato dalla digitazione errata delle credenziali di accesso, ma una serie ripetuta e frequente di tentativi di accesso falliti è sicuramente segno di un tentativo di accesso fraudolento. È essenziale essere in grado di rilevare i tentativi di accesso fraudolento per mettere in atto misure di protezioni specifiche contro l'attaccante.

1.3. L'architettura di un sistema di gestione

L'architettura di un sistema di gestione è descritta (da Stallings) nella prossima figura. La figura è complessa perché dobbiamo/vogliamo raffigurare non solo i componenti di gestione ma anche quelli gestiti; inoltre, si ha come sempre la necessità di rappresentare sia gli aspetti "hardware" del sistema (quante e quali macchine ci sono) ma anche i componenti software che eseguono sulle varie macchine. Stallings usa il termine *Network Management System (NMS)* per indicare l'intero sistema che effettua la gestione, e ovviamente il NMS è costituito da due componenti:

- Una applicazione, con una interfaccia grafica potente, che permette di eseguire tutti i tipi di compiti di gestione del sistema che sono necessari e che viene chiamata *Network Management Application (NMA)*

- Un insieme di entità che gestiscono i dispositivi, che sono quasi sempre incorporate nei dispositivi stessi; e che sono chiamate *Network Management Entity (NME)*.

Al di là dei termini, si riconoscono i due tipi di componenti che abbiamo introdotto nel paragrafo precedente, e cioè Manager e Agent. Ricordiamo che NMA e NME sono "applicazioni" per quanto riguarda il protocollo di trasporto. La NMA è talmente complessa che sicuramente esegue come una applicazione del sistema operativo ospite. La NME può in certi casi essere parte del sistema operativo se deve interagire molto "a basso livello" con le risorse che deve gestire.

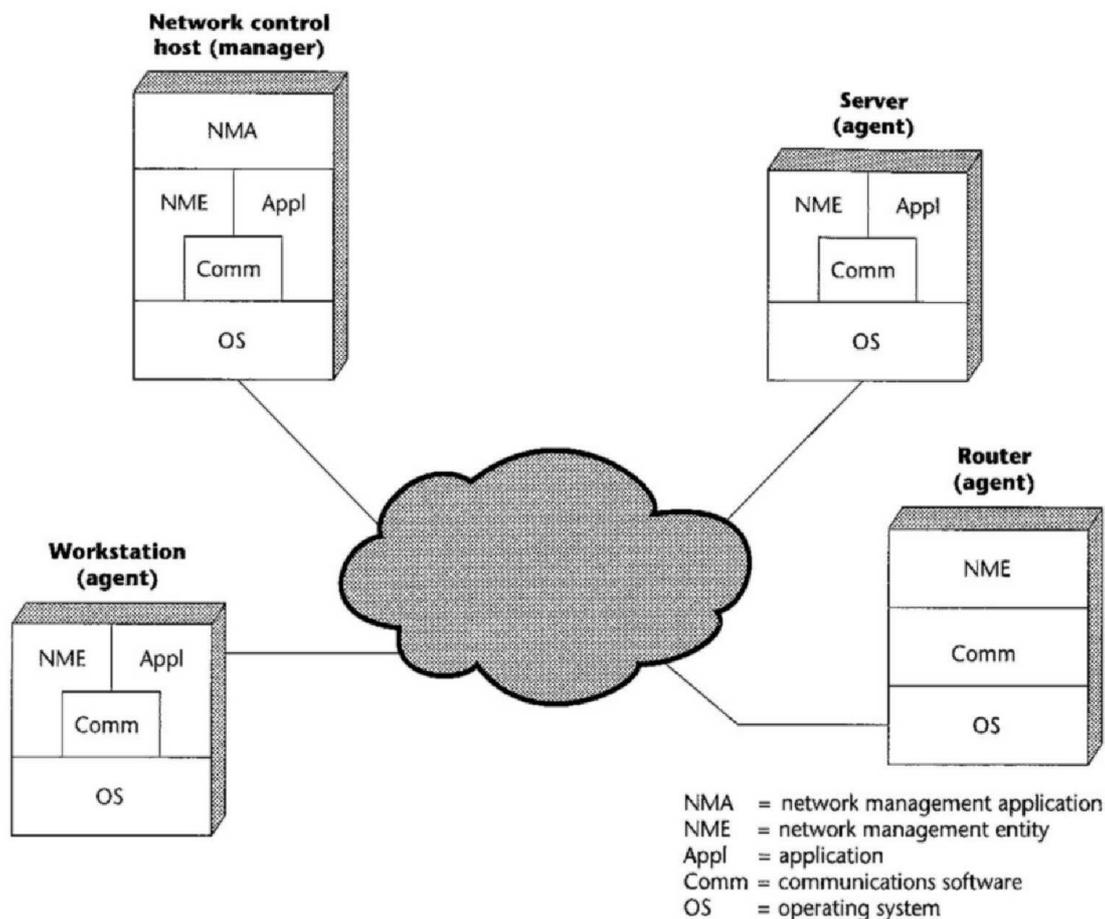


FIGURE 1.1 Elements of a network management system

La figura permette di fare un certo numero di considerazioni interessanti. Vediamo che nel sistema c'è una workstation che è un oggetto gestito, evidentemente perché il suo corretto funzionamento è considerato importante. Ma la macchina su cui esegue la NMA stessa non è degna di essere gestita anche lei? Non è di notevole importanza? Se tale macchina non funziona il sistema di gestione non funziona; quindi anche la macchina che Stallings chiama *Network Control Host (NCH)* e che ospita una NMA deve essere gestita. Ma per essere gestita deve avere un componente NME che permette alla NMA di gestire la stessa macchina su cui esegue. Naturalmente la gestione avverrà tramite protocollo applicativo, che non ha problemi a funzionare trasparentemente fra applicazioni che eseguono sulla stessa macchina.

Questa figura evidenzia elementi che la figura di Mauro non evidenzia (attenzione alle didascalie nella figura!), ma non evidenzia il ruolo server/client dei componenti, internet-wise, cioè secondo la architettura internet di applicazioni distribuite.

- Si noti che in un server, l'applicazione sotto gestione è un importante oggetto da gestire: la NME (Agent) deve permettere di gestire la applicazione, non solo l'hardware o il SO della macchina.

Un NMS aggiunge qualcosa al sistema da gestire; almeno una NMA (Manager), e le NME (Agent). Spesso le NME sono già contenute nei dispositivi da gestire e al massimo si devono "configurare". Quindi la gestione di un sistema "costa" qualcosa e introduce componenti aggiuntivi a quelli del sistema gestito.

Un NMS deve permettere di avere una vista completa di tutto il sistema sottoposto a gestione, in una architettura unificante, nella quale ogni componente del sistema (applicazione, workstation, server, router, switch, bridge, link, etc.) è mostrato, etichettato, indicato con il suo indirizzo, etc.

1.3.1. Attività della NME e della NMA

I componenti del sistema di gestione hanno diversi compiti. Finora abbiamo parlato assai più dei compiti della NMA, ma le NME sono essenziali perché la NMA offra i servizi desiderati agli amministratori del sistema gestito. Ogni NME (Agent) svolge i seguenti compiti

- Raccoglie e mantiene informazioni sulle attività del dispositivo (applicazione, host, elemento di rete,...) che gestisce;
- Immagazzina queste informazioni localmente, in genere con certi limiti sulla quantità di informazioni che mantiene;
- Risponde a comandi da parte della NMA, che gli chiedono di:
 - Trasmettere le informazioni raccolte;
 - Modificare un parametro della configurazione del dispositivo;
 - Fornire informazioni di stato, ad es. valori di configurazione del dispositivo o stato di suoi componenti;
 - Generare traffico artificiale per fare dei test;
- Invia in modo autonomo alla NMA messaggi di avviso (trap) quando è necessario

La logica applicativa della gestione è implementata nella NMA, che quindi diventa un componente estremamente critico del sistema gestione, per cui non ci può essere un solo host su cui esegue la NMA (Manager), perché se tale host ha un guasto il sistema è fuori controllo

Il Network Management Center (NMC) consiste quindi di almeno due Network Control Host (NCH). NMA e NME comunicano usando un protocollo di gestione. Dato che NMA e NME si basano sul sistema operativo della macchina ospite e sul software di comunicazione, i venditori di sistemi operativi spesso offrono NMS che sono *proprietary*, cioè progettati per essere usati solo sul software del venditore. Recentemente sono emersi sistemi di gestione standardizzati per poter essere usati su apparecchiature di venditori diversi

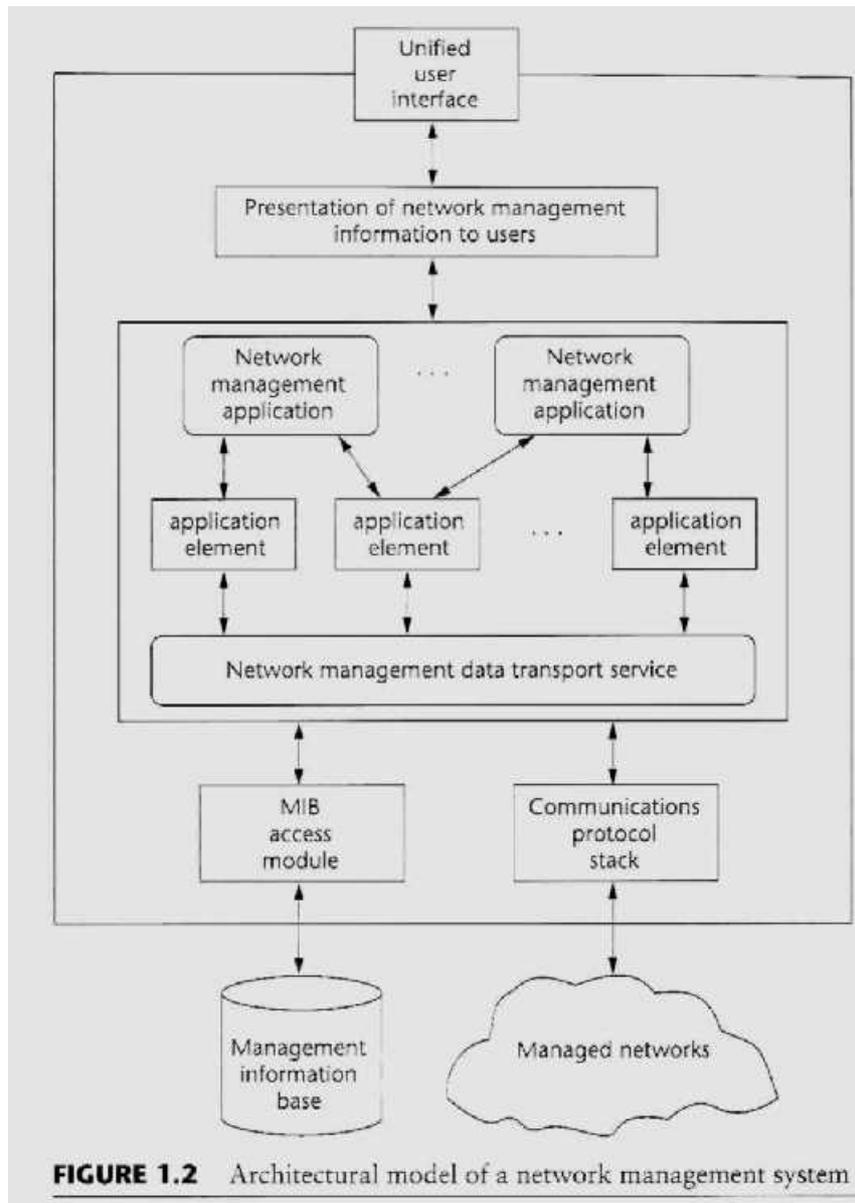
1.3.2. Architettura di una NMA

Come ci si può aspettare la NMA in se stessa ha una architettura complessa. I componenti possono essere classificati nei seguenti tipi:

- Software per la presentazione dei dati all'utente
- Network management software
- Software di comunicazione e di data base

Una NMA è tipicamente usata in modo interattivo dagli amministratori del sistema e quindi deve avere del software per l'interazione con gli amministratori. Questi aspetti grafici possono anche essere molto sofisticati per esempio per mostrare grafici dell'andamento nel tempo di variabili importanti quali il tempo di risposta del sistema, l'occupazione delle risorse, ecc. La gestione di un sistema richiede l'esecuzione di elaborazioni specifiche come pure la esecuzione di attività programmate dagli amministratori; quindi una buona NMA deve fornire un ampio corredo di software specifico per la

gestione di sistemi. Infine, i dati relativi ai dispositivi gestiti devono essere reperiti tramite le NME (usando uno stack di comunicazione) ma anche salvati in un database per archiviazione storica a per poter pianificare la crescita di potenza del sistema gestito (capacity planning). La prossima figura tenta di illustrare la architettura interna di una NMA.



La figura ci permette di notare alcuni aspetti importanti di una applicazione di gestione.

1.3.3. Sistema unificato di interazione con l'utente

Una NMA è costituita da numerosi componenti (come vedremo), molti dei quali devono interagire con l'utilizzatore della NMA; è importante che lo stile di interazione con l'utente sia uniforme sia nella grafica che nella impostazione dei dialoghi e dei menu tramite i quali si accede alle funzioni dei singoli componenti. Per questa ragione la figura rappresenta la NMA come racchiusa all'interno di una *Unified user interface* che costituisce il "canale di comunicazione" fra utente e NMA.

1.3.4. Presentazione della informazione di gestione

Anche questo è un componente che conviene unificare perché componenti diversi producono gli stessi tipi di informazione di gestione da mostrare all'utente. Le elaborazioni necessarie per

"costruire" i dati da mostrare all'utente devono essere unificate, sia per risparmiare sviluppo di codice sia per dare uniformità alla interfaccia utente, che deve presentare sempre nello stesso modo dati della stessa natura.

In questo strato di *Presentation of network management information* vengono anche implementate le modalità di dialogo con l'utente che devono essere condotte in modo unificato e coerente. La uniformità della interfaccia utente nei vari componenti del sistema ha come ulteriori benefici effetto la riduzione della necessità di formazione del personale e la riduzione degli errori operativi.

La presentazione delle informazioni di gestione presenta una addizionale difficoltà nel progetto della modalità di interazione con l'utente. Poiché le informazioni di gestione sono estremamente numerose e di grandi dimensioni, vi è il pericolo che il gestore sia inondato di informazione, e perda di vista le cose importanti nella sua situazione. Occorre quindi fornire strumenti per organizzare, riassumere e semplificare questa enorme quantità di informazione, con elevate possibilità di personalizzazione.

1.3.5. Network Management Software

Il software di gestione vero e proprio può avere complessità molto variabile da NMS a NMS, ma sempre più comunemente è molto complesso perché deve rispondere a requisiti molti diversi e sofisticati. Nella figura è rappresentato il caso di un software di gestione complesso e ricco di funzionalità.

1.3.5.1. Applicazioni di gestione

Nel riquadro più interno della figura vengono evidenziate un certo numero di *Network Management Application* che sono i componenti che realizzano la logica applicativa della NMA. Le NMA sono ambienti applicativi estremamente complessi, che vengono sviluppati nell'arco di lunghi periodi, aggiungendo moduli o programmi che svolgono funzioni specifiche, che devono essere richiamabili dall'utente in modo unificato.

L'automazione di compiti di gestione tramite applicazioni specifiche sviluppate dagli amministratori del sistema dovrebbe essere disponibile agli amministratori alla stessa stregua di applicazioni native della NMA.

1.3.5.2. Application Element

Le applicazioni che forniscono servizi direttamente fruibili dagli amministratori fanno uso di componenti, chiamati *Application Element*, il cui ri-uso favorisce la riduzione di costo e la robustezza (mancanza di errori) dell'intera NMA.

Gli application element costituiscono anche uno strumento di astrazione rispetto ai servizi di più basso livello su cui è costruito l'ambiente della NMA.

1.3.5.3. Servizio di trasporto di dati di management

Lo strato sottostante viene indicato con questo termine perché questo è in fondo il suo compito: mettere la NMA in comunicazione con le informazioni di gestione che le sono necessarie. Queste informazioni possono provenire da varie fonti e per questo è necessario un livello di astrazione. A questo livello possono anche essere realizzate modalità di accesso ai dati che non sono offerte dai servizi sottostanti ma che possono essere implementate mediante le funzioni da essi offerte: ad es. operazioni di aggregazione di dati.

Quali possono essere le fonti di informazioni di gestione?

1.3.5.4. Stack di protocolli di comunicazione

Il sistema gestito è ovviamente la forma primaria e più immediata di informazioni di gestione. Gli NME mantengono, almeno in parte, la informazione di gestione relativa ai dispositivi che gestiscono

e quindi la NMA deve certamente poter interagire con il sistema gestito mediante un opportuno stack di protocolli di comunicazione.

Molto spesso una NMA deve essere sviluppata in modo da poter essere usata con diverse architetture di gestione, proprietarie o standard, e quindi con diversi stack di protocolli di comunicazione; inoltre, deve spesso essere disponibile su diversi sistemi operativi e diverse architetture hardware. Per questa ragione lo stack di protocolli di comunicazione non può essere interfacciato direttamente agli strati superiori della NMA ed è necessario uno strato di astrazione e di unificazione dei vari servizi offerti nei vari ambienti in cui la NMA dovrà eseguire, che verrà poi linkato alle librerie corrette.

1.3.5.5. Modulo di accesso alla MIB (Management Information Base)

La figura evidenzia che questo modulo permette l'accesso ad un database che contiene informazione di gestione: la *Management Information Base*. Questo è tipicamente ospitato da un data base relazionale commerciale oppure open source, a seconda della politica commerciale del fornitore della NMA.

Le applicazioni di gestione devono operare su dati che descrivono i dispositivi in un certo istante, ma devono anche archiviare i dati raccolti *storicizzandoli*, perché interessa il loro andamento nel tempo. Sui dati storici possono essere effettuate statistiche e analisi di tendenza.

I dati istantanei che riguardano i dispositivi di gestione si ottengono usando lo stack di comunicazione e contattando le NME, ma la base storica dei dati rilevanti deve essere costruita dalla NMA contattando periodicamente le NME e memorizzando i dati ottenuti in un data base.

Il modulo ben separato di accesso a database è necessario perché la NMA deve poter essere venduta pronta per usare database diversi di costruttori diversi, perché una azienda tipicamente ha già scelto un data base e non vuole acquistare un diverso data base solo per poter fare la gestione del sistema informatico. I diversi produttori di data base spesso introducono piccole differenze nelle interfacce di accesso al loro data base per guadagnare vantaggi competitivi rispetto ai concorrenti. Anche i driver di database "standard" come ODBC presentano spesso sottili differenze da un database ad un altro.

L'architettura di una NMA mette quindi in rilievo l'importanza di questa base di dati di informazioni di gestione. Le applicazioni devono poter lavorare con facilità e in modo trasparente sia sui dati storici che sui dati raccolti in un dato istante dai dispositivi. Per questa ragione il servizio di trasporto dei dati di management si appoggia da un lato sul DB, e dall'altro sullo stack di comunicazione: per nascondere agli strati superiori il modo con cui i dati sono ottenuti. I dati istantanei sono ottenuti contattando l'Agent sul dispositivo, mentre i dati riferiti ad un istante nel passato sono ottenuti dallo storico sul database. Gli Agent sui dispositivi quindi realizzano al loro interno la parte istantanea (corrente) della MIB, che il Manager può prelevare mediante il servizio di trasporto dei dati di management.

1.4. La Management Information Base (MIB)

Come si può apprezzare, il termine MIB viene quindi usato in modo *ambiguo* per riferirsi a volte ai dati di gestione memorizzati sul database, e altre volte per riferirsi ai dati di gestione che concettualmente sono immagazzinati all'interno degli Agent (delle NME), e si dice che "la MIB è dentro agli Agent". Il protocollo di management offre operazioni di raccolta dei dati della MIB mantenuta nel dispositivo (come illustrato nella parte bassa della figura precedente).

Dentro alla MIB, i dati devono obbedire ad un particolare *schema dei dati*, come in tutti i data base. Lo schema dei dati all'interno del database può essere liberamente definito dai progettisti della NMA, al fine di ottenere le migliori funzionalità e prestazioni. Invece i dati all'interno degli Agent è bene che siano *standardizzati*, in modo che dispositivi della stessa natura siano gestibili nello stesso

modo usando lo stesso codice e la stessa logica. Se due dispositivi della stessa natura, ad es. due host, sono descritti da dati di gestione che hanno uno schema diverso, la NMA deve sviluppare due moduli diversi per gestire questi due dispositivi anche se sono in realtà "la stessa cosa". Questo è un problema tipico per ogni applicazione che usa un database: lo schema dei dati deve essere "conosciuto" dall'applicazione perché essa funzioni come l'hanno progettata gli sviluppatori.

Può non essere sempre facile decidere se due dispositivi sono della stessa natura, ma nella maggior parte dei casi è facile capire che sono diversi: non ci sono dubbi che un *acquario di pesci* (supponendo che volessimo gestirlo perché il suo corretto funzionamento è essenziale per mantenere alto il morale dei lavoratori e quindi avere successo sul mercato) non può essere gestito usando le stesse informazioni che invece permettono di gestire un *router*, che necessita a sua volta di informazioni diverse da quelle necessarie per gestire un *database management system*. Questa dipendenza della NMA dallo schema dei dati di gestione spiega anche la proliferazione delle applicazioni di gestione: quando ho necessità di implementare logiche di gestione sofisticate, è inevitabile che si debba scrivere codice che è specifico per i dati che specificamente descrivono il dispositivo. Per mettere in un grafico l'andamento di una misura posso anche non sapere che cosa significa la misura (ad es., la temperatura dell'acqua dell'acquario, il numero di pacchetti al secondo in ingresso ad una interfaccia, il numero di transazioni al secondo eseguite dal DBMS) ma se voglio che la mia NMA esegua azioni e controlli più sofisticati ci sarà il momento in cui la gestione di un acquario diverge da quella di un router.

Quindi è importante definire lo schema dei dati istantanei resi disponibili dalle NME (Agent) in modo che dispositivi della stessa natura, anche se prodotti da costruttori diversi offrano gli stessi dati con lo stesso significato: cioè offrano lo stesso *modello*. Invece i costruttori hanno interessi diversi, soprattutto quando offrono dispositivi complessi che richiedono una opportuna gestione: in questo caso sono interessati a sviluppare essi stessi un buon sistema di gestione che aumenti il valore dei loro dispositivi e che però non possa essere usato per gestire dispositivi simili dei concorrenti, che ne trarrebbero vantaggio senza avere fatto gli investimenti. Inoltre, se la gestione dei componenti del sistema informatico è proprietaria, il Cliente avrà difficoltà a cambiare fornitore e passare ad un fornitore concorrente. Quindi i costruttori sono interessati a sviluppare architetture di gestione proprietarie, anche nel modello dei dati di gestione. L'interesse degli utenti è invece quello che ci siano buoni standard di gestione *aperti e vendor-independent*, e che i costruttori adottino questi standard aperti per i loro dispositivi. Per questa ragione Internet ha sviluppato la "sua" *architettura di gestione aperta*, e questa architettura include la definizione dei modelli, cioè del loro schema dei dati, dei dispositivi più diffusi (non a caso non ha ancora sviluppato lo schema dei dati di gestione degli acquari per pesci).

Come per tutti gli schemi di DB, deve essere usato un formalismo opportuno per definire lo schema dei dati; nei database questo formalismo si indica spesso con il termine *Data Definition Language*. Se vogliamo definire degli schemi di dati di gestione standard per dispositivi di diversa natura, occorre che anche il formalismo per esprimere questi schemi e la modalità con cui verranno definiti siano standardizzati. Il linguaggio e le modalità da adottare per esprimere lo schema delle MIB dei dispositivi è definito in Internet da uno standard (RFC) chiamato *Structure of Management Information (SMI)*.

Il protocollo di management offre operazioni di raccolta dei dati della MIB mantenuta nel dispositivo (vedere la figura precedente). Si può pensare di realizzare un NMS che sia capace di gestire dispositivi di costruttori diversi solo se le MIB di dispositivi dello stesso tipo sono uguali; e non potranno essere uguali se la SMI usata per definire la MIB è già essa stessa diversa.

1.5. Gestione "Distribuita"

La concezione del sistema di gestione come basato su un unico NMS centralizzato non è soddisfacente. Infatti i gestori del sistema devono poter lavorare in sedi diverse, o su LAN diverse

della stessa sede, e comunque su macchine diverse. Non è accettabile che gli amministratori debbano recarsi nella sala macchine dove è situato il server che ospita il NMS e d'altra parte i Management Server devono essere replicati per ragioni di affidabilità e di continuità del servizio.

Gli amministratori accedono alle applicazioni che costituiscono il NMS tramite terminale remoto grafico. Le applicazioni di gestione accedono agli Agent sui dispositivi direttamente via protocollo di gestione, o tramite Element Manager che fanno da mediatori rispetto ai dispositivi. Lo scenario è quindi assai complesso, come illustrato nella prossima figura.

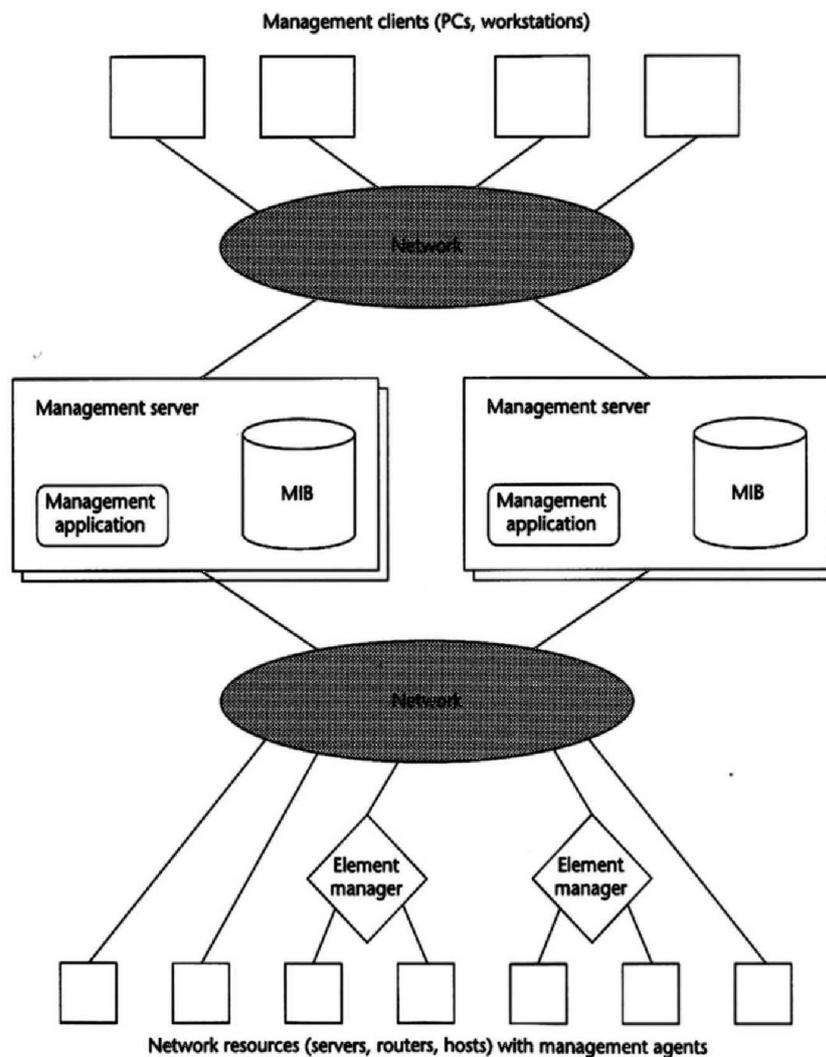


FIGURE 1.3 Typical distributed management system architecture

Gli Element Manager sono componenti che possono essere necessarie per varie ragioni. Una ragione è che i dispositivi da gestire sia gestiti tramite una architettura di gestione proprietaria diversa da SNMP; in tal caso gli Element Manager funzionano da *gateway applicativi* o *proxy*. Ma gli Element Manager possono essere necessari quando i dispositivi non possono ospitare una architettura Internet di gestione a causa della limitazione delle risorse di calcolo: in tal caso gli Element Manager sono degli Agent che interagiscono con le risorse da gestire mediante interfacce ad hoc, che possono richiedere o meno dei protocolli di comunicazione.

Abbiamo quindi uno schema di interazione come illustrato nella prossima figura:

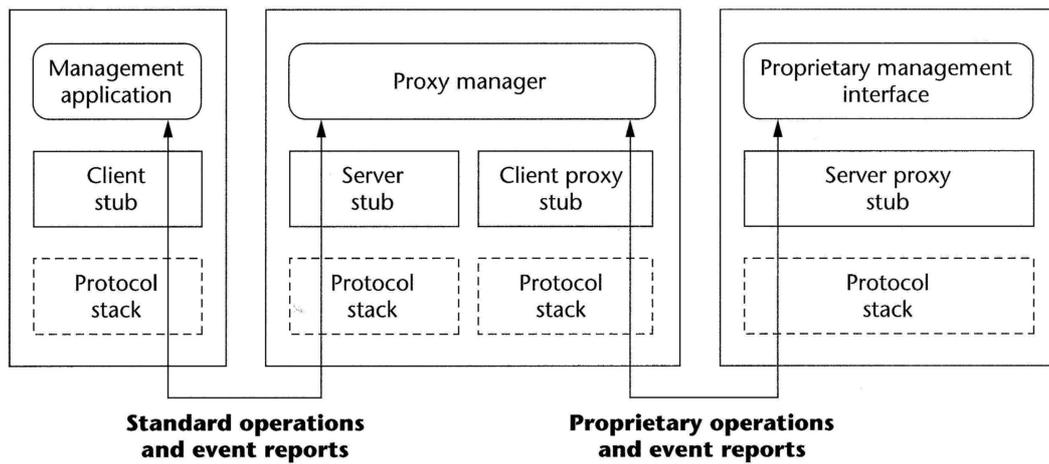


FIGURE 1.4 Proxy manager architecture

L'NMA contatta il proxy agent (o proxy manager, nel gergo di Stallings), che traduce la richiesta del manager in una forma appropriata per il componente di interesse. Le risposte provenienti dal sistema interessato vengono a loro volta tradotte nello standard di gestione del manager e inviate al manager. Viene spesso adottato uno schema a RPC (Remote Procedure Call) per la realizzazione di queste architetture sistemistiche.