

Management solutions for IP networks

Jan Forsl w, Ian Jarrett, Pdraig Moran and Bal zs Szviovski

In recent years, data traffic through the Internet has increased exponentially. IP networks have expanded in size and in speed to cope with increased demand. As a consequence, network management has become increasingly important.

At the same time, customers are asking for advanced IP services, such as IP-based virtual private networks (IP-VPN), voice-over-IP (VoIP) and electronic commerce (e-commerce). In particular, operators are investigating the potential of providing quality of service (QoS) over IP networks at less cost than over conventional ATM and frame-relay networks. The key criterion of a successful network operator has become the ability to offer and monitor new services across the network in a cost-effective manner.

In this article, the authors explore various considerations that an operator must bear in mind when developing an IP-management solution. The authors also describe some of the new IP-management products under development at Ericsson.

The IP-management framework

A variety of applications is required for managing an IP service throughout its life cycle. Some applications have been around for several years, such as IP address management by means of dynamic-host-configuration-protocol (DHCP) servers and remote authentication control using authentication, authorization and accounting (AAA) servers. Other management applications are newer, such as IPsec key distribution, which uses Internet key exchange (IKE) servers.

To encompass a wide variety of management applications, the IP-management

framework needs to have an open and scalable architecture. It needs to adhere to global standards that can be met by multiple vendors, and it needs to allow flow-through automation across multiple systems when services are activated and data is collected. Figure 1 provides a structure for classifying IP-management applications into a service-management life cycle.

The design servers—which are used for traffic planning in this structure—mostly include capabilities for simulating user and network behavior. The policy servers allow the large-scale provisioning of a given set of user services, such as IP-VPNs. The element- and network-management system (EMS/NMS) focuses on deployment-related aspects of the network, and the monitoring server ensures that promised service levels are supplied. The customer care and billing system handles the accounting aspects of the managed service. Finally, in a large network environment, a workflow server can be deployed to manage the progress of service fulfillment and assurance.

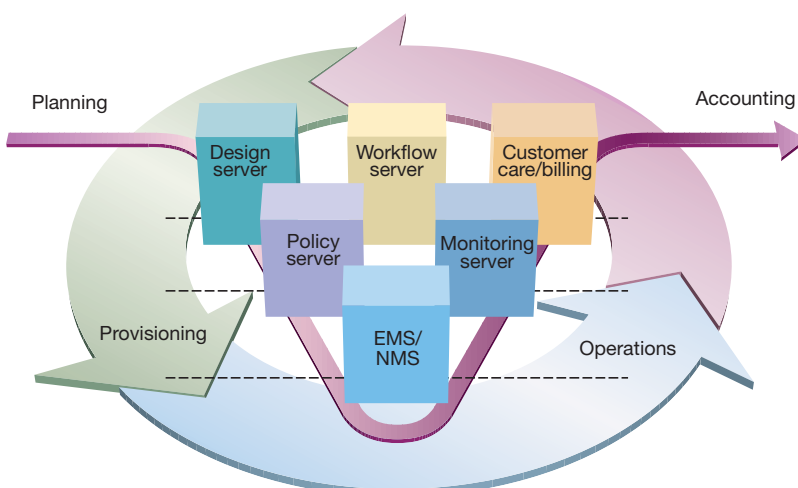
Open standards

Ericsson has chosen to base its IP-management framework on a combination of the common object request broker architecture (CORBA—as defined by the Object Management Group, OMG) and the common information model (CIM—as defined by the Distributed Management Task Force, DMTF) standards. CORBA is applied as an event channel for sharing dynamic data between management server applications in a distributed and scalable environment. The CIM is implemented in a lightweight-directory-access-protocol (LDAP) directory for publishing static configuration information between management-server applications. Each server process can be operated on multiple workstations to share load or to distribute it to different physical locations.

Umbrella applications

Apart from recording and charging for delivered services, the customer care and billing (CCB) server supports the processing of service orders through the definition of service-level agreements (SLA). In small installations, the CCB server can also take on the role of coordinating underlying management servers. In diverse network-management implementations, the installation of a workflow management (WFM) server can help to manage the progress of service-order processing, network design,

Figure 1
The service management life cycle.



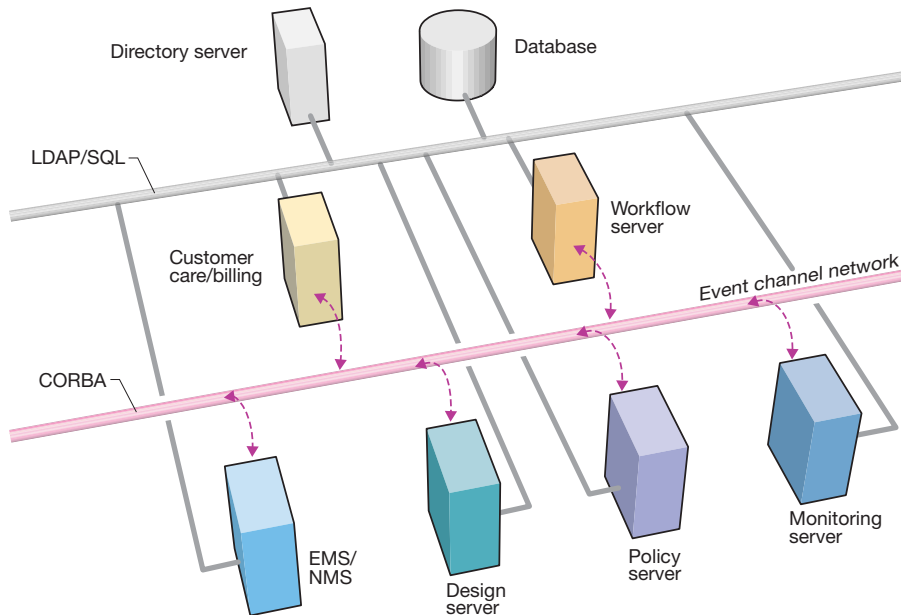


Figure 2
The IP-management framework.

service and network provisioning, problem-solving actions, and so on. Ericsson provides systems-integration support for several third-party CCB and WFM products, including the Internet Administration Framework from Solect and IP Netcharger from Ericsson Hewlett-Packard (EHPT). Because these are considered to be general-management products that are largely independent of networking technologies, we will not discuss them in greater detail here. Instead, we will focus on IP-specific management applications as shown in Figure 2.

How the applications interact

To provide IP-VPN or similar services, four main management applications need to interact. Each application is responsible for filling in parts of the tree in the LDAP directory schema. The complete schema encompasses configuration information on nodes, policies, users, and services. The use of a directory enables any of the applications to fetch the schema and stored data at any time.

The role of the element- and network-management system is to ensure that the basic topology is configured and stays up and running. Before proposed configurations can be deployed, the EMS/NMS must validate them in terms of syntax and network integrity. The EMS/NMS can also take part in the user service life cycle. That is, when a service is activated, the workflow

BOX A, ABBREVIATIONS

AAA	Authentication, authorization and accounting	ISP	Internet service provider
ATM	Asynchronous transfer mode	JDBC	Java database connectivity
BGP	Border gateway protocol	LDAP	Lightweight directory access protocol
CCB	Customer care and billing	LSP	Label switched path
CGI	Common gateway interface	MIB	Management information base
CIM	Common information model	MPLS	Multiprotocol label switching
CLI	Command line interface	NMS	Network management system
CMIP	Common management information protocol	NOC	Network operations center
COPS	Common open policy service protocol	NRM	Network resource manager
CORBA	Common object request broker architecture	OMG	Object Management Group
CoS	Class of service	OSPF	Open shortest path first
DEN	Directory-enabled network	PDH	Plesiochronous digital hierarchy
DHCP	Dynamic host configuration protocol	PDM	Policy deployment manager
DMTF	Distributed Management Task Force	PDP	Policy decision point
DNS	Domain name service	PEP	Policy enforcement point
DWDM	Dense wave division multiplexing	PFA	Packet and frame access
EMS	Element management system	POP3	Post office protocol version 3
FCAPS	Fault configuration accounting performance security	QoS	Quality of service
FDDI	Fiber distributed data interface	RADIUS	Remote authentication dial-in user service
GPS	Global positioning system	SDH	Synchronous digital hierarchy
HTTP	Hyper text transfer protocol	SLA	Service level agreement
ICMP	Internet control message protocol	SLM	Service level manager
IKE	Internet key exchange protocol	SMTP	Simple mail transfer protocol
INM	Internet network monitor	SNMP	Simple network management protocol
IP	Internet protocol	TCP	Transmission control protocol
IPsec	Secure Internet protocol	TE	Traffic engineering
IS-IS	Intermediate system-to-intermediate system	ToS	Type of service
ISM	Internet service monitor	VoIP	Voice over IP
		VPN	Virtual private network
		WFM	Workflow management
		WFQ	Weighted for queuing
		WRR	Weighted round robin

management needs to ensure—via the EMS/NMS—that the basic network capabilities are available. In the IP-VPN service scenario, this might entail enabling multi-protocol label switching (MPLS) on backbone router interfaces or creating redundant path topology on which user traffic can be routed.

The design server gives input for path placement by using simulation techniques to calculate optimized routes with guaranteed quality of service between sources and destinations. The design server takes input from the EMS/NMS (which provides configuration data) and the monitoring server (which supplies performance data).

The policy server performs correlation and configuration functions that are specific to the provisioning of user services (such as IP-VPN and VoIP) to the network. Compared to the network-element configurations provided by the EMS/NMS, the variety of configurations in the policy server is quite limited. But on the other hand, the policy server is optimized for a high volume of finite, repetitive tasks: it receives service requests

initiated by the CCB and reacts to alarms generated by the monitoring server.

The monitoring server collects data on system, application, and network performance, and correlates it with configured services, users, and topology, in order to monitor the delivered service-level agreement. If the monitoring server detects an SLA violation, it generates an event. The policy server and the CCB capture events, to perform corrective actions.

Against this background, let us now examine each application area in greater detail.

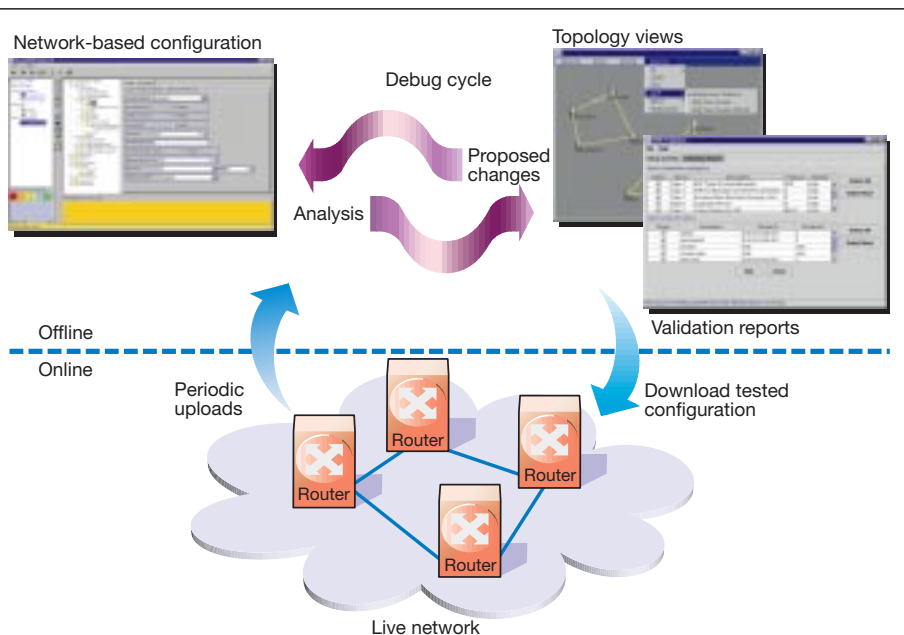
The element- and network-management system

The EMS/NMS applications are responsible for the deployment-related aspects of the network. The EMS normally handles all *fault configuration accounting performance security* (FCAPS) aspects on a per-network-element basis. As such, the element manager is sometimes directly integrated into the network element, to facilitate concurrent updates of the configuration-management module when new router features are rolled out. For this reason, the functions of the network-management system have often been restricted to network surveillance, equipment management, and performance reporting. Ordinarily, this is based on available simple network management protocol (SNMP) management information bases (MIB) in the routers.

However, as IP networks grow in size, configuration management becomes more difficult to maintain on a per-network-element basis, creating the need for an IP subnetwork manager. Ericsson has developed an IP subnetwork manager called the *Network Resource Manager* (NRM). Ordinarily, the IP subnetwork manager is integrated into the operator's NMS platform (HP OpenView, AdventNet WebNMS, Bull OpenMaster or Compaq TeMIP).

The IP subnetwork manager alleviates the problems of large configurations by providing a network-wide view of proposed configurations before they are deployed. Figure 3 shows how an IP subnetwork manager is used in an interactive fashion to establish a viable configuration baseline for the network. The operator enters the proposed configuration through Web-browser-like configuration displays, then checks it

Figure 3
Network configuration life cycle.



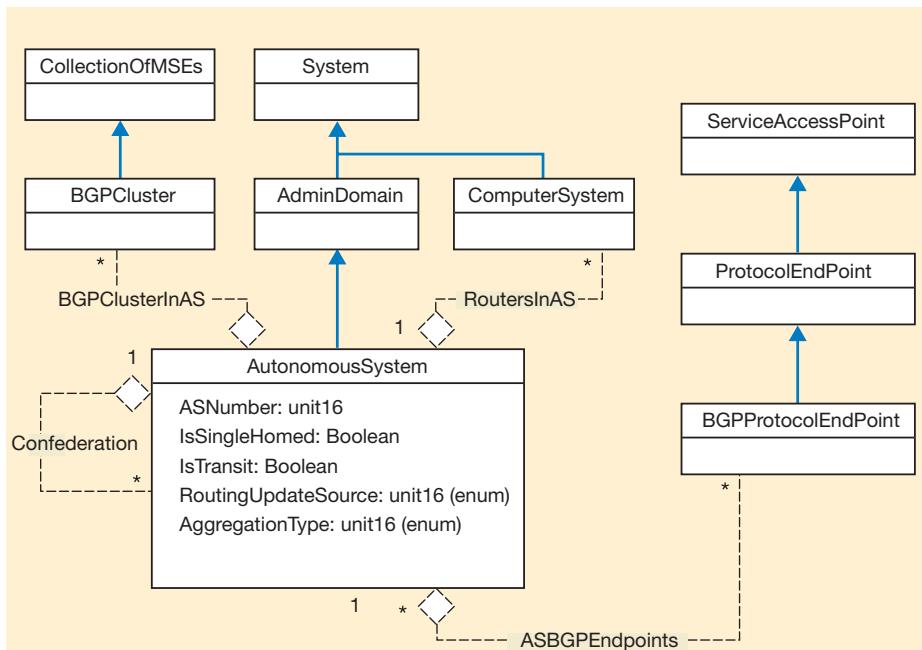


Figure 4
Example BGP sub-schema: autonomous systems.

in topology views and validation programs before exporting the configuration files to the routers.

Simple release handling

Because network element capabilities are usually updated frequently, the router model in the network resource manager must be easy to upgrade. Accordingly, a configuration specification language (from which data that describes the object can be generated automatically) has been defined. A parsing program uses the metadata to automatically update configuration displays and import/export functions according to the software release in the router.

Common information model

Standardized MIBs are too limited to be applied as the information model for configuring router networks, because their focus is mainly on the monitoring aspects of the network. Using a proprietary command line syntax, a configuration file syntax, or both, each router vendor provides its own information model for configuring routers. Simple scripts are often used to automate the parameter settings on a set of routers. To replace these scripts with a full-fledged IP subnetwork manager, a unified network-wide information model is needed with

which the configuration-management applications can interact. When a common information model is used for the different routers, the management applications can share common attributes. Thus, network-oriented applications do not need to be changed each time a new router is introduced. The main adaptation effort is instead moved to the element-parser stage in the import and export functions.

The directory-enabled-networks (DEN) initiative, and in particular the networks working group in the Distributed Management Task Force, is currently the main standardization effort for creating a network-wide router schema. The network resource manager, which extends this schema to cover all aspects of a router network, extracts all common aspects from the router tree to provide centralized manipulation capabilities. The border-gateway-protocol (BGP) sub-schema shown in Figure 4 exemplifies this. As can be seen, the autonomous system subtree is separate from the computer system subtree, and only an association is used between the two.

Baselines

A set of router configurations constitutes a baseline in the network resource manager. The stored baselines are categorized as *pro-*

posed, running and historical. Operators make their changes on the proposed baseline. The running baseline consists of an uploaded version from the router network. Uploads are triggered by the topology-auto-discovery function in the NMS, a scheduling function, or both. The historical baselines constitute stable and verified router configurations that are used when severe network problems occur.

Configuration displays

The configuration displays in the NRM share a navigational tree that lists all the routers and their items. New items can be instantiated on demand from a palette view. Similarly, all configuration fields, their value ranges, help text, and default values are displayed in a properties view. Items can be copied within and between routers.

A set of specific template displays is provided for rapidly configuring network-wide parameters across several routers. The template parameters, which are set once, affect all selected routers or interfaces. These group statements are major time savers when configuring network-wide properties, such as class-of-service (CoS) drop profiles, routing policies, and firewalls.

Protocol-sensitive topologies

Protocol-sensitive topologies help operators to evaluate the proposed configuration of the network. They constitute individual views of the IP topology and each of the support-

ed routing protocols—for example, open shortest path first (OSPF), intermediate system-to-intermediate system (IS-IS) and BGP. The topology is shown as it has been configured, not as discovered. Proposed configurations can be displayed before they are exported to the router network—for instance, to analyze a protocol-specific area partitioning, an interface-reservation bottleneck, or routing-enforcement leakage. By launching any of a set of available table views, operators can make visual queries on a specific link or node in a protocol view. The table views provide easy access to the most important data on the selected item.

Validation checks

The validation function reports on an integrity check that is made to detect common but hard-to-isolate configuration problems. The validation program in the network resource manager has two components: syntax error checks of individual router configurations (element view) and integrity checks between router configurations (network view).

The network resource manager comes pre-configured with a set of 100 rules. Operators can extend this set by defining new rules in a scripting language designed to facilitate routing configuration, data collection, and validation. The language allows the implementation of complex algorithms, to analyze configuration data and report on possible configuration problems.

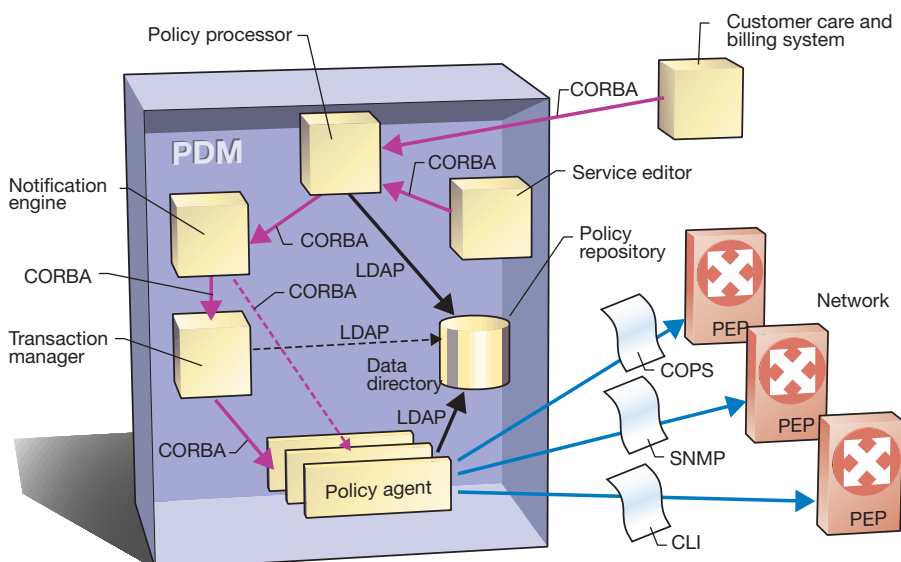
Secure import and export

When the operator is satisfied with the configuration, an import and export function automatically reconfigures the network. Configuration data is fetched from the database of the network resource manager and parsed into a set of configuration files. These are then transferred and loaded into each router.

The import/export function also facilitates the management of user-defined items, which contain a set of configuration statements in the router's native configuration language. Previously undefined configuration statements are processed as though they had been user-defined and displayed to the operator for action or correction.

If an error occurs in any part of the export transaction, the network resource manager can roll back to the previous configuration. The error is captured and displayed to the operator. The NRM also keeps track

Figure 5
High-level architecture of the PDM.



of any required sequences for the export transaction.

The policy server

Policy server technology provides a mechanism for mapping business ideas, services, and concepts to the underlying network configurations. Through a mapping procedure and a policy rule engine, complex business rules can be represented in a way that facilitates a “relatively” straightforward realization in the network. Services like QoS bearer service provision or secure tunnel configuration can easily be specified in high-level terms. Example: “Provide GOLD service for all SAP traffic from site A to site B.” The policy server creates logical rules that represent how this can be realized in the network and applies the appropriate configuration operations to the network. It even takes into account the time span during which the policy is to apply.

The policy technology is also being expanded to invoke conditional or reactive policies. Thus, if a particular link fails or QoS demands are not being met, the underlying network resources can be reconfigured, or low-priority connections can be dropped.

Ericsson’s policy deployment manager (PDM) incorporates policy server technology that gives operators control of the QoS configuration in the network. This solution is being extended further to provide the configuration of customer VPNs; that is, it will deal with every aspect of the customer-provisioning flow from the entry of service orders by the customer service representative to the configuration of the customer sub link in the edge router (Figure 5).

The architecture of the PDM is component-oriented. Its generic design facilitates support for policy management to many different application spaces, such as quality of service, security and address handling. The PDM plays a central role in providing customer services in an IP network. The functionality provided by the system is offered to high-level systems through a CORBA interface. This is particularly appropriate where CCB systems have already been deployed.

The service editor

A service editor facilitates the configuration of customer services. In terms of QoS, this might be the definition of a “Gold” or “VoIP” service. When the service has been

defined (information on how it is to be deployed in the network), the service editor specifies differentiated service (DiffServ) settings, bandwidth requirements, and so on. The definition also allows services to be defined in terms of parameters. For example, the exact bandwidth associated with the “Gold” service can be determined by the customer subscription. The service editor can also specify global information, such as label definitions: “*Daytime* starts at 09:00 and ends at 18:00 hours.”

The policy agents

The general architecture of the PDM is based on a central policy processor that maps customer business service requirements to applicable internal policy rules. Policy agents deal with the actual deployment of these rules. There will be many agents in a network. Agents are distributed systems that implement and apply rules to the network. Their tasks include

- mapping policies that pertain to configuration requests to specific network elements;
- scheduling requests;
- removing configurations that no longer apply; and
- applying dynamic changes to the network.

Policy agents can be deployed to control geographical areas of the network, subsets of network types, or policy-enforcement points (PEP), or logical functions, such as QoS or IPsec configuration. Initially, communication between the policy agent and the PEP will take place via the command line interface (CLI); in subsequent releases, the CLI-based communication will be augmented with the simple network management protocol and the common open-policy service (COPS) protocol.

The notification server

The policy agents also interact with a notification server, which distributes dynamic information in the system. Policy agents subscribe to notification events that relate to their role. When a new customer subscription is created, the notification server is informed. It then informs all agents that are interested in the event, so that they can introduce the associated policy as appropriate. Planned extensions to the LDAP/DEN architecture propose that this notification function should be made part of directory services, whereas the PDM currently implements it as an external component.

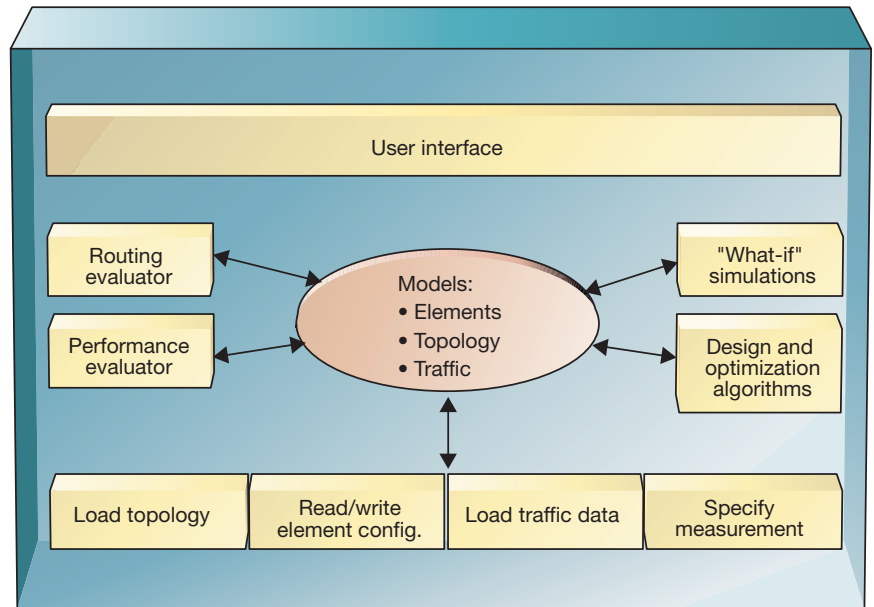


Figure 6
Functional blocks of the TE Tool.

The transaction server

A transaction manager component controls how policies are applied to the network. In many situations, a policy affects the configuration of many features in many different nodes. Should any operation fail, a rollback procedure is initiated and the network manager is informed. The policy application is an all-or-nothing mechanism.

On-demand service requests

The PDM will initially be deployed to provide typical customer services. Nonetheless, it will also support on-demand requests; for instance, customers connected to a broadband service can request a certain quality of service for a specific period of time—say, video stream from a local video server for three hours. To begin with, these requests will be made via a Web interface; later on they will be made via signaling.

Other enhancements being investigated include bandwidth brokering. In this case, the application of policies relates to the business rules associated with an application and to the network's ability to support the application.

The design server

IP networks offer new means of managing traffic. Two important new features are

quality of service and explicit path placement. The task of the design server is to tune the network by means of these new network-management features. The traffic engineering tool (TE Tool) is one example of a design server developed by Ericsson (Figure 6).

The TE Tool interfaces with the network- and element-management system and the monitoring server to build a global view of network status. The TE Tool uploads network and element configurations via the EMS/NMS interface. Also, based on the simulation results, it uses the EMS/NMS to invoke configuration changes in the network. The TE Tool uses the monitoring server interface to access measurements of end-to-end traffic volumes and delays on a per-QoS-class basis. The functions of the TE Tool can be broken down into three main categories:

- path design;
- network-performance optimization; and
- “what-if” analysis.

Path design and QoS tuning

Before new traffic demands (in the form of VPNs or low-delay VoIP trunks) with associated QoS and traffic load requirements can be put on the network, a check must be made to ensure that the network has sufficient resources. In particular, two tasks must be performed: an end-to-end path must be found and the QoS scheduling parameters must be

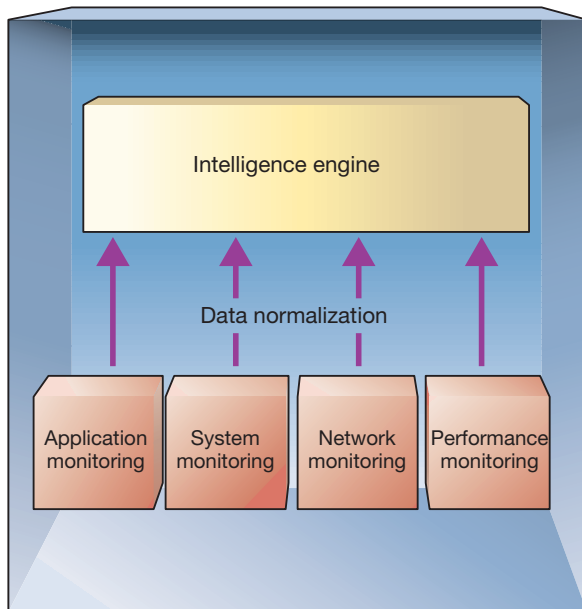


Figure 7
Data normalization.

tuned at the router interfaces—for example, the minimum service rates of buffers served by the weighted-round-robin (WRR) or weighted-fair-queuing (WFQ) schedulers. The end-to-end path can be restricted to satisfy a set of constraints or policies stipulated by the user (transmission media, QoS level). Operators can also use the TE Tool to reroute label-switched paths (LSP).

Network performance optimization

If the monitoring server detects a performance problem, it pinpoints the source and, if possible, re-engineers part of the network. One way of optimizing performance is to reroute traffic from congested paths through less congested parts of the network. As with path design, special policies should be taken into consideration, including preferences for certain paths, and attributes such as link delay, leasing costs, and QoS constraints. The TE Tool finds the alternative that is most cost-effective, requires least intervention, and offers the fastest transition from the old configuration to the new one.

“What-if” analysis

An important performance metric of an operational network is its ability to handle unforeseen events. Operators need to know how robust the network is, and if it can stay within defined performance intervals dur-

ing normal operation as well as when network elements fail or traffic increases. The TE Tool’s “what-if” analysis can simulate events of this kind and help pinpoint weaknesses and potential problem areas. Operators can use the tool to simulate a range of hypothetical events—such as congested paths, QoS violation, and connectivity failure—and to evaluate the network’s response.

The monitoring server

Ericsson’s service-level manager (SLM) provides the functionality of the monitoring server. Within the IP-management architecture, the SLM provides bespoke interfaces to underlying managed components to collect, collate and analyze data from a variety of networks, network elements, and applications. The SLM normalizes the data before it passes it on to its central intelligence engine (Figure 7).

Since the underlying managed components might support a variety of management protocols, interfaces, and data formats, the SLM supports numerous flexible interfaces with which to connect to them. How the SLM gets this information depends on the interface supported by the managed component; for example, an IP router might provide a convenient SNMP interface from

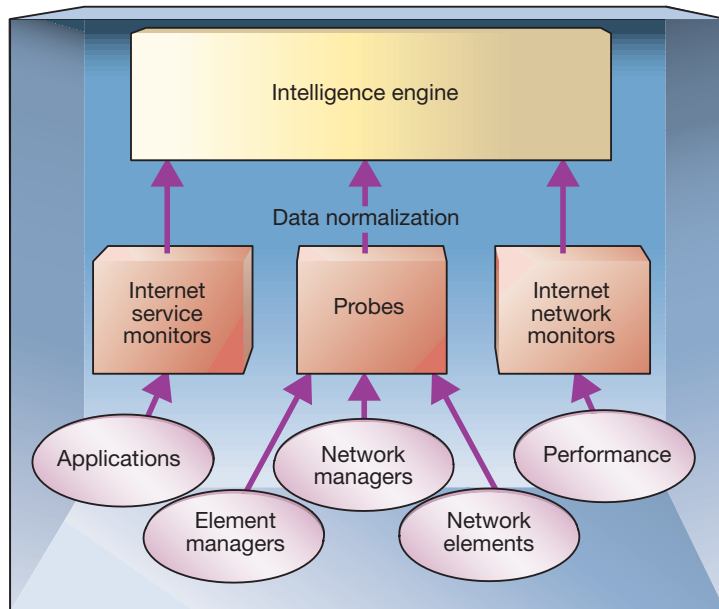


Figure 8
Data collection architecture in the service-level manager.

which to receive information, whereas a Web server might not. The SLM overcomes these problems by supplying a variety of flexible applications that can either monitor the managed components directly or via an intermediate element- or network-management application (Figure 8).

Internet service monitors

Internet service monitors (ISM) are components that allow the SLM to monitor Internet applications. Examples include

- domain name servers (DNS);
- remote-access dial-in user services (RADIUS) servers;
- hypertext transfer protocol (HTTP) servers;
- simple mail transfer protocol (SMTP) servers;
- post-office protocol (POP3) servers; and
- other applications that support transmission-control-protocol (TCP) sockets.

Each ISM can monitor multiple applications and converse with them in their native protocol. ISMs can be configured to send regular requests to monitored applications and record the time it takes for an application to

respond. For example, a RADIUS monitor can attempt to authenticate itself with a RADIUS server, and an HTTP monitor can download a Web page, Java applet or common gateway interface (CGI) form from a Web server. The ISM times these interactions and reports on their performance to the intelligence engine in the SLM.

The Internet service monitors actively contact monitored applications at regular intervals to test their response. This is particularly useful for Internet service providers (ISP) who host (and monitor) Internet applications. The ISMs allow ISPs constantly to monitor the response and quality of the Internet applications they host. ISMs report the response time of the applications they monitor through Web-based graphics that break the responses down into detail (connect time, response time and download time). They also support thresholds for each recorded attribute. Every response that is recorded by an ISM monitor is converted into an event and forwarded to the SLM intelligence engine where it can be correlated with and compared against other events in the monitoring server.

Probes

Probes are flexible software components that attach themselves to the event streams of a variety of network- and element-management systems and network elements. However, unlike ISMs, probes are passive and do not generate management traffic. Currently, the probes support more than 120 different classes of network-element, element-manager, and network-management applications, including the following Ericsson products:

- IP routers (AXI 520, AXI 540, AXC 623, AXC 627, AXC 711);
- ATM switches (AXD 301);
- frame-relay devices (Eripax PFA); and
- transport components (DWDM, SDH, PDH).

Other standard Internet components are provided by vendors, such as Cisco Systems, Sun Microsystems, 3Com, Microsoft, and Oracle, and open standards, such as the SNMP and common management information protocol (CMIP).

During the normal working life of a network-element, element-manager, or network-management application, a vast quantity of events can be generated as alerts, alarms, SNMP traps, log files, console messages, audit trails, or debug lines. These events are usually stored locally, close to the

source that generated them, creating a complex management nightmare for the operators who have to maintain them. SLM probes overcome this problem by attaching themselves to the source of the events. They then forward the events in a normalized format as an SLM alert to the SLM intelligence engine. Since multiple probes can receive events from multiple sources concurrently, a continual stream of event-based information can be normalized and forwarded to the SLM intelligence engine.

Probes use TCP/IP to create a reliable session between the intelligence engine and the source of an event. This guarantees the transmission and reception of events. The probes also function as a heartbeat, immediately notifying the central SLM intelligence engine and its operators of network outages. Notwithstanding, the primary role of the probe is to gather, normalize, and forward data.

Internet network monitors

The Internet network monitor (INM), which is used to provide accurate performance information for the monitoring server, is a distributed application consisting of three components that communicate with one another over an open CORBA interface (Figure 9). This architecture allows the INM to monitor delay in regular and traffic-engineered networks that carry IP traffic. The INM's statistical-sampling algorithms guarantee the accuracy of management data and overcome many of the deficiencies inherent in traditional polling-based performance-management applications (SNMP or CMIP).

The INM supports a variety of management applications via application-specific gates, which translate performance data requests into CORBA events. The INM agent is a central scheduler that controls exactly when performance tests should commence, and it instructs protocol-specific collectors to collect samples as appropriate. The INM architecture uses statistical sampling when it makes performance tests—samples of collections can be generated at random intervals over a given time span; the results of the entire sample (as opposed to one individual poll) are then returned to the manager. This sampling of thousands greatly increases accuracy.

One advantage of the INM is its accuracy at performing delay tests in IP networks. Not only are samples of delay measurements taken at random intervals, but the test pack-

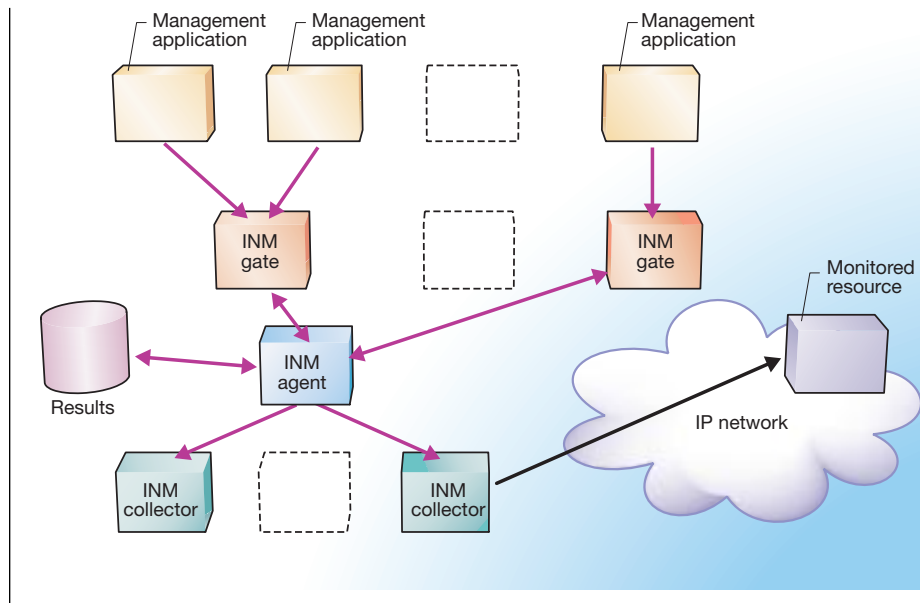


Figure 9
Internet network monitor.

ets are constructed of random sizes with random payloads. This ensures that samples experience the same delays as regular user traffic. The INM also supports the measurement of quality of service within an IP network, by allowing the management application to specify the type of service (ToS) that should be used for each sample—different ToS values should receive different QoS characteristics; the INM monitors and indicates whether or not this is so.

The INM, which reports accurate delay figures for one-way and two-way paths, also supports LSPs in modern traffic-engineered networks that use MPLS. The unique features in the Internet control message protocol (ICMP) collector allow samples of one-way and two-way test packets to be measured. Consequently, operators can use the INM to monitor and report on the QoS found in their traffic-engineered networks. Further support for the global positioning system (GPS) allows the ICMP collectors to synchronize themselves with a global time source, thereby providing accurate delay information across the Internet and around the globe.

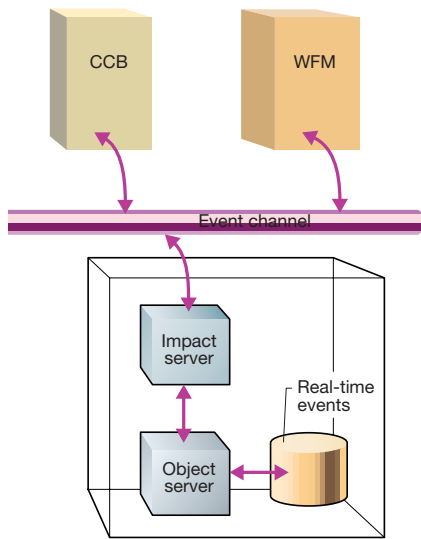


Figure 10
Interaction with other systems.

Intelligence engine

The intelligence engine provides the correlation and service-monitoring functionality required by the monitoring server. It consists of object and impact servers, which are additional components of the SLM (Figure 10).

The monitoring server considers everything to be an event. The object server—which receives events in real time from ISMs, probes, and INM components—categorizes and correlates events in accordance with applicable business rules. These rules allow the object server to determine which events are important, which are redundant, and which can be ignored. The object server can then instruct the impact server to identify—using the hierarchical impact-relationship model that it maintains—which monitored resources or services are affected by particular events. In accordance with actions defined in the hierarchical impact-relationship model, the impact server can request information off the event channel from external data stores, such as those maintained by the customer care, billing, or workflow systems. This allows the impact server to determine whether real-time events affect configured services, service levels and customers.

The impact server can also fetch additional information held in external sources and pass it back to the object server, which can combine customer and service-related information in its event database. Network operators who use the system can thus see, in real time, which customers and services are affected by the events.

Conclusion

As router-based service-provider networks evolve, the need for powerful, comprehensive network-management tools increases. Router networks have expanded in size, speed and complexity. Today, a handful of experts can no longer configure and monitor an entire ISP network through the command line interface on each router.

To tackle the challenge of managing router networks, Ericsson is introducing

new routing protocols, routing options, and carrier-class network-management tools, which are characterized by

- considerable automation in configuring the network; and
- effective capabilities for filtering and correlating diverse sets of network data.

The use of carrier-class management tools for network provisioning and surveillance is not new. Indeed, solutions of this kind have been used in large telephony and transmission networks for quite some time. Ericsson's experience of telephony and transmission networks has thus proved invaluable in the creation of an IP management suite that is adapted to the convergence of

- Internet handling of voice and data; and
- fixed and wireless communication services.

Ericsson's approach to the development of IP-network-management solutions has been to focus on network- and systems-management processes since these are often tailored to a specific subset of network elements—an added benefit of this approach is that it yields a synergetic effect from internal router development. Accordingly, for the network-planning and design-management processes, Ericsson is developing the traffic engineering tool (TE Tool); for network provisioning, Ericsson has developed the network resource manager (NRM); in the area of network data management, Ericsson has created a powerful package around the service-level manager (SLM); and within service configuration, Ericsson will soon introduce the policy-deployment manager (PDM).

This comprehensive product portfolio is further strengthened by strategic partnerships within the areas of customer care and service operations. In these partnerships (with Ericsson Hewlett-Packard, Solect, and several other well-known vendors), Ericsson takes full responsibility for integrating Ericsson-specific network elements and service offerings into the best-of-breed system platforms developed by our partners. The end product is a solid management portfolio for IP networks.

TRADEMARKS

Java™ is a trademark owned by Sun Microsystems Inc. in the United States and other countries.

TeMIP™ is a trademark owned by Compaq Computer Corporation.