Web-Based Network Management: From Pull to Push

Bell Laboratories, Lucent Technologies
October 29, 1998

Jean-Philippe Martin-Flatin
Swiss Federal Institute of Technology, Lausanne
Institute for computer Communications and Applications

martin-flatin@epfl.ch
http://icawww.epfl.ch/~jpmf/
Outline

• Pre-Web Network Management Platforms
• Problems
• Web-Based Management
• Pull Model
• Push Model
• Future Research
• Q&A
Network Management Platforms

- **OSI:**
  - 5 SMAs (FCAPS)

- **IP:**
  - 3 core functions (mandatory):
    - network monitoring
    - data collection
    - notification handling
  - other functions (optional): configuration, ACLs, billing...
  - SNMP management framework (SNMPv1, SNMPv2c, SNMPv3)
  - Examples: HP OpenView, Cabletron Spectrum, IBM Netview, Sun Solstice...
IP NMPs: 3 Core Functions

• Network monitoring:
  ■ detect faults in network devices and links:
    ➪ reactive w.r.t. network faults
    ➪ proactive w.r.t. complaints from users/customers

• Data collection:
  ■ gather data to build daily, weekly and monthly reports:
    ➪ proactive in the longer term

• Notification handling:
  ■ quick
  ■ react to events generated by agents (SNMP notifications)
  ■ react to events generated by the manager (event correlator)
Regular Management

• Ongoing network monitoring or data collection
• Automated
• 2 modes:
  ■ attended mode: operators gazing at GUIs (red-icon syndrome)
  ■ unattended mode: automated correlation, alarms (pager, email, telephone, siren...)
• Medium-sized to large networks
Ad Hoc Management

• Troubleshooting or configuration
• Manual
• Always attended mode: administrators and/or operators
• All networks
• Replaces regular management in small networks
Pre-Web NMPs: SNMP Management Framework

- Manager/agent paradigm
- Polling for data collection & network monitoring
- Unsolicited push for notification delivery
- SNMP communication protocol
- SMIv2 (ASN.1)
- BER encoding
- MIBs (generic, vendor-specific)
- ...
Pre-Web NMPs: Market Evolution

• Once upon a time, there were open systems [...]:
  ■ generic network equipment
  ■ generic management

• Market segmentation

• From generic MIBs to vendor-specific MIBs

• From generic management GUIs to vendor-specific management GUIs (add-ons)

• For customers, openness guaranteed by:
  ■ SNMP management framework
  ■ SNMP protocol
A Simple Model of Pre-Web NMPs
Data Collection

- Polling definitions and schedules repository
- Polled data repository
- Polling engine
- SNMP request
- SNMP reply
- NMS
- DBMS proxy
- Polled data repository
- SNMP client
- SNMP server
- Network device
- Administrator

- Polling definition GUI
- Polling scheduler GUI
- SNMP request
- SNMP reply
Report Generation

Diagram:

- **Report definitions and schedules repository**
- **Polled data repository**
- **Reports repository**
- **DBMS proxy**
- **Report definition GUI**
- **Report scheduler GUI**
- **DBMS proxy**
- **Report generator**
- **DBMS proxy**
- **Report visualizer GUI**
- **Report printing script**
- **NMS**
- **Administrator**
Notification Delivery and Event Handling

- SNMP trap server
- Event correlator
- DBMS proxy
- Event repository
- Events repository
- Event handlers
- Event handler definition GUI
- Network map GUI
- SNMP trap client
- Network device
- NMS
- Administrator
- Operator
Ad Hoc Management

- Virtual terminal (telnet)
- One-shot MIB-variable visualizer GUI
- Time series visualizer GUI
- MIB browser GUI
- Vendor-specific GUI
- Generic GUI
- Network map GUI
- Network topology discovery
- SNMP request
- SNMP reply
- SNMP server
- Network device
- Administrator or Operator
- One-shot MIB-variable visualizer GUI
Problems with Pre-Web NMPs (1/2)

• NMP vendors:
  ■ no problem, it’s big bucks time!

• Customers:
  ■ NMPs are too expensive (hardware and software):
    ➤ dedicated hardware for network management
  ■ limited support for third-party RDBMSs
  ■ need for Unix expertise to maintain existing platforms:
    ➤ cost to migrate to Windows is too high

• Network equipment vendors:
  ■ the support of many device-specific add-ons for many NMPs and many OSs is too expensive
Problems with Pre-Web NMPs (2/2)

• Customers and network equipment vendors:
  ■ poor time-to-market for add-ons:
    ➤ several months after hardware release if large market share
    ➤ never if small market share:
      - startup companies need to resort to separate NMPs
  ■ versioning:
    ➤ version mismatch between the add-on (NMP) and the MIBs (devices)
      while a vendor-specific MIB is gradually upgraded in a network:
      - update the NMP manually, device by device
        (no MIB-discovery protocol)
      - do not use new features of a MIB until all devices
        have been upgraded
Software Engineering Problems with Pre-Web Network Management (1/2)

- Protocol efficiency:
  - poor efficiency of BER encoding [Mitra 1994]
    [Neufeld and Vuong 1992]:
    ➤ addressed by PER (Packed Encoding Rules) in OSI
    ➤ SMIv1 and SMIv2 mandate BER encoding for all SNMP frameworks
  - poor efficiency of SNMP:
    ➤ no efficient table retrieval mechanism --> repeated message exchanges
    ➤ in varbind lists, OIDs take much more space than values
Software Engineering Problems with Pre-Web Network Management (2/2)

- Security:
  - lack of secure SNMP `get` or `set` in SNMPv1 and SNMPv2c
  - SNMPv3: just released, still to show its acceptance in the field
  - VPNs: need expensive encryption hardware to manage remote subsidiaries
  - firewalls: UDP relays are complex to set up and maintain [Chapman and Zwicky 1995]

- TCP vs. UDP
  - in theory, both OK to transport SNMP; in practice, only UDP
  - important SNMP notif. are lost for silly reasons (e.g. buffer overflow)
  - some mgmt data is more important than user data (e.g. heartbeats)
Web-Based Management

• What is it?
• What can Web technologies bring to pre-Web NMPs?
• Early solutions: HTML pages
• Today: pull model
• Tomorrow: push model
• Near future: mobile code (mgmt tasks delegated to agents)
What is Web-Based Management? (1/2)

• Marketing answer = WBEM
  ■ today, CIM schema but little Web [Thompson 1998]

• Proposed technical answer = integrated network, systems and service management based on Web technologies:
  ■ HTML forms
  ■ Java applets, servlets and applications
  ■ JDBC
  ■ Java RMI and Object Serialization
  ■ Java IDL
  ■ ...

• This talk = Web-Based Network Management
What is Web-Based Management? (2/2)

• Distribution in Java:
  • HTTP
  • sockets
  • RMI
  • Java IDL (CORBA)
    ➤ telecoms = yes
    ➤ Internet = no

• Typically:
  • applet to servlet: HTTP or sockets
  • applet to Java application: sockets or RMI
What can Web technologies Bring to Pre-Web NMPs?

• Get rid of the NMP
• Reduce network overhead of management data
• Reduce development costs of add-ons
• Reduce time-to-market of add-ons
• Put small and large equipment vendors in fair competition
• Simplify management of remote subsidiaries across a firewall
• Improve support for third-party RDBMSs
• ...

HTML Pages

• Secondary tasks:
  - automate and standardize problem reporting (helpdesk)
  - put daily, weekly and monthly reports online (paper-free office policy)
  - online help for network troubleshooting:
    ➤ administrators write symptom-driven HTML pages for operators, with pointers to online vendors’ documentation
    ➤ user-friendly access to management scripts (Perl, Tcl/Tk) and programs (ping, traceroute, netstat)
HTML-Based CLI

• Mapping between URLs and command line interface:
  ■ e.g. on Cisco routers [Bruins 1996]:
    ➤ http://routername/exec/show/interface/ethernet0/
    ➤ show interface ethernet0
  ■ generated via HTML forms, or embedded in symptom-driven HTML pages
Pull vs. Push

• Newspaper metaphor:
  ■ buy it everyday from the same newsdealer
  ■ receive it everyday by postal mail

• Pull model:
  ■ request/response paradigm
  ■ data transfer initiated by the manager
  ■ e.g., data polling in pre-Web NMPs (network monitoring and data collection)

• Push model:
  ■ publish/subscribe/distribute paradigm
  ■ parallel and independent data transfers initiated by agents
  ■ e.g., SNMP notifications in pre-Web NMPs
Pull Model

• Ad hoc management:
  ■ vendor-specific management GUIs coded as applets
    ➞ HTTP together with SNMP
    ➞ HTTP instead of SNMP
  ■ generic management GUIs coded as applets

• Regular management:
  ■ all GUIs coded as applets
  ■ data polling based on HTTP
HTTP Together With SNMP (1/3)

- [Bruins 1996]
- Device-specific management GUI (add-on) coded as an applet
- Management data transferred via SNMP
- Ad hoc management
HTTP Together With SNMP (2/3)
HTTP Together With SNMP (3/3)

TCP/HTTP/URL

1. TCP/HTTP/applet
2. UDP/SNMP/SNMP get
3. UDP/SNMP/SNMP data

NMS (manager)

Network device (agent)

Administrator or Operator

Any machine

Proxy

Web browser

HTTP

HTTP client

SNMP client

Applet

Vendor-specific GUI

Network map GUI

SNMP proxy

Vendor-specific GUI (applet)

HTTP server

Any machine

SNMP server

Network device

MIBs
HTTP Instead of SNMP (1/3)

- [Wellens and Auerbach 1996]
- Device-specific management GUI (add-on) coded as an applet
- Management data transferred via HTTP
- Ad hoc management
HTTP Instead of SNMP (2/3)
HTTP Instead of SNMP (3/3)
Generic GUIs Coded as Applets

Diagram:

- **Administrator or Operator**
- **Any machine**
- **WWW server (intranet or Internet)**
- **Web browser**
- **Network map GUI**
- **HTTP client**
- **Generic GUI (applet)**
- **HTTP server**
- **Vendor-specific GUI (applet)**
- **HTTP/SNMP gateway**
- **HTTP server**
- **MIBs**
Pull Model

- Ad hoc management:
  - vendor-specific management GUIs coded as applets
    - HTTP together with SNMP
    - HTTP instead of SNMP
  - generic management GUIs coded as applets

- Regular management:
  - all GUIs coded as applets
  - data polling based on HTTP
Web-Based Ad Hoc Mgmt & Pre-Web Regular Management

Administrator or Operator

Web browser

Vendor-specific GUI (applet)

Network map GUI

HTTP client

HTTP server

Vendor-specific GUI (applet)

HTTP/SNMP gateway

HTTP server

Generic GUI (applet)

Data polling

SNMP

Event handling

SNMP

SNMP trap client

Network device

WWW server (intranet or Internet)

Generic GUI (applet)

Any machine

Operator

NMS

SNMP server

MIBs
All GUIs Coded As Applets
Data Polling Based on HTTP
The Push Model

• 3 phases:
  ■ publish
  ■ subscribe
  ■ distribute

• Issues:
  ■ Positions of client and server now reversed:
    ➫ transfer of management data initiated by the agent
    ➫ client side of the persistent connection still on the manager side
    ➫ we want the server to initiate a transfer in a client/server architecture!
  ■ Firewalls: HTTP vs. sockets vs. RMI
  ■ Timeout of the persistent connection
Publish and Subscribe Phases

Diagram showing the processes involved in publish and subscribe phases. It includes components such as Web browser, Firewall, WWW server (intranet or Internet), Any machine, Data server, and Network device. The diagram illustrates the flow of data and interactions between these components.
Distribute Phase for Network Monitoring and Data Collection

How?

- Web browser
  - Network map GUI (applet)
  - (client)

- Data server
  - JDBC server

- Administrator or Operator
  - Java appl.
  - Network map registry
  - JDBC client
  - Event correlator
  - Pushed data interpreter
  - Pushed data collector

- Any machine
  - Event handler
  - Pushed data filter
  - Pushed data collector

- Network device
  - Push definitions and schedules repository
  - Push scheduler
  - MIB data dispatcher
  - MIBs

- Administrator
  - Pager Email Telephone Siren

- General purpose data repository
  - JDBC server

- Any machine
  - Network map GUI

- Firewall
  - Pushed data collector
  - Pushed data interpreter
  - Event correlator
  - JDBC client

- Data server
  - JDBC server

- Administrator or Operator
  - Java appl.
  - Network map registry

- Any machine
  - Event handler
  - Pushed data filter
  - Pushed data collector

- Network device
  - Push definitions and schedules repository
  - Push scheduler
  - MIB data dispatcher
  - MIBs

- Administrator
  - Pager Email Telephone Siren
Distribute Phase for Notifications

How?

Any machine

Web browser

Network map GUI

(applet)

Network map registry

Network device

Network monitor

Notification generator

Notification dispatcher

Java appl.

JDBC client

Event handler

Notification filter

Notification collector

Any machine

Administrator

JDBC server

Data server

Administrator

or Operator

Any machine

General purpose data repository
Sockets

• Advantages:
  ■ bi-directional
  ■ simple to implement

• Drawbacks:
  ■ persistent connection is unstable if socket timeout < push period
  ■ robustness: notifications delivery by the agent depends on a persistent connection created by another entity (the manager)
  ■ firewalls: require specific settings (UDP or TCP)
Java RMI (1/2)

Distribution via Java RMI for notification delivery

Distribution via Java RMI for data collection and network monitoring
Java RMI (2/2)

• Advantages:
  ■ bi-directional association between RMI client and RMI server (sockets underneath)
  ■ elegant design (fully OO network management)

• Drawbacks:
  ■ requires a full JVM in agents
  ■ RMI implementations are slow --> not scalable
  ■ firewalls: how to control ports used by RMI clients? (supposedly transparent to the application)
HTTP

- Advantages:
  - simple design: client and server on the good side
  - robustness: the agent can reconnect immediately in case of timeout, it does not have to count on the manager
  - firewalls: no change, or minor change (assuming Web access)

- Drawbacks:
  - HTTP server in Java appl.: large program made larger, more difficult to debug, slower to execute, with a larger footprint on the host
The Collapsed NMP

Administrator or Operator

Any machine

Data server

General purpose data repository

JDBC server

Web browser

Network map GUI (applet)

HTTP client

MIB data subscription GUI (applet)

JDBC client

Notification subscription GUI (applet)

HTTP server

Any machine

Network device

Firewall

WWW server (intranet or Internet)

Network map GUI (applet)

HTTP server

Push scheduler servlet

Push definitions and schedules repository

HTTP server

Push definition servlet

Notification dispatcher client

MIB data subscription GUI (applet)

Notification subscriber GUI (applet)

Network map registry

Event correlator

Event handler

Java appl. server

Notification collector

JDBC client

Notification filter

JDBC client

Java appl. server

Network map registry

Event correlator

Event handler

Java appl. server

Pushed data collector

JDBC client

Pushed data filter

Pushed data interpreter

Any machine

Pager

Email

Telephone

Screen

JDBC client

Java appl. server

Pushed data collector

JDBC client

Pushed data filter

Pushed data interpreter

Any machine

Java appl. server

Pushed data collector

JDBC client

Pushed data filter

Pushed data interpreter

Any machine

Java appl. server

Pushed data collector

JDBC client

Pushed data filter

Pushed data interpreter

Any machine
Many Problems Have Been Solved (1/4)

- Customers:
  - platforms are too expensive (hardware and software)
    - no need for dedicated hardware
    - less expensive software (small collection of applets and scripts)
    - capitalize on previous investment (e.g., use existing RDBMS)
  - limited support of third-party RDBMS vendors
    - no need for peer-to-peer agreement, use JDBC instead
  - need for Unix expertise to maintain existing platforms:
    - cost to migrate to Windows is too high:
      - migration cost from/to any platform is minimal
      - GUIs and Java applications are platform independent
Many Problems Have Been Solved (2/4)

- Network equipment vendors:
  - the support of many device-specific add-ons for many NMPs and many OSs is too expensive:
    - single applet
- Customers and network equipment vendors:
  - poor time-to-market for add-ons:
    - zero time-to-market, whatever the market share
    - access to integrated network management for startup companies
  - versioning:
    - MIBs and applets upgraded together, device by device
Many Problems Have Been Solved (3/4)

• Protocol efficiency:
  ■ poor efficiency of BER encoding
    ➤ BER encoding no longer used
  ■ poor efficiency of SNMP
    ➤ SNMP as a communication protocol is replaced with HTTP
    ➤ more management data per packet with push or RMI
    ➤ management data can be compressed (gzip)
    ➤ with RMI, no MIB variables anymore (higher level of abstraction)
Many Problems Have Been Solved (4/4)

• Security:
  ■ management of VPNs: HTTP security may be used instead of encryption boxes
    ➞ still weak security
    ➞ at least better than community string
  ■ firewalls: HTTP simpler than SNMP

• TCP vs. UDP:
  ■ HTTP makes it possible to use TCP to transfer management data
  ■ reliable transport layer for SNMP notifications:
    ➞ important notifications are no longer lost for silly reasons
    ➞ still no guarantee of delivery
Bonus

- Redundant managers are simple to support with push:
  - one step toward fault-tolerance
New Problems

• NMP vendors:
  ■ loss of revenue
  ■ is a market of generic applets sustainable?
  ■ need to find niche markets:
    ➤ fault tolerance
    ➤ large networks where scalability is stretched to the limits
    ➤ real-time networks where responsiveness and speed are stretched to the limits

• Known problem:
  ■ clock synchronization

• Potential problem (to be investigated):
  ■ Java is slow --> scalability (e.g. JDBC)
Future Research

• Implementation of the push model
• BEAM project
Implementation of the Push Model

• Prototype: IP routers of Lightning (and others?)
• Develop a network management application in Java that implements the push model:
  ■ Sun’s JMAPI and Java DMK (M-beans)
  ■ AdventNet’s SNMP package
  ■ EmbeddedJava
• Demonstrate the advantage of using HTTP with firewalls
• Investigate the issue of scalability
BEAM Project (1/2)

- BEAM = Bulk & Easy Access to MIBs
- Objectives:
  - bulk MIB data transfers
  - keep MIBs
  - increase protocol efficiency:
    - get rid of the SNMP protocol
    - improve SMIv2
    - new MIME type for management data (transparent compression)
  - reduce network overhead
- Deliverables:
  - formal proposal to IETF in 1999
BEAM Project (2/2)

• BEAM + MIBs + push = new network management framework for the IP world

• Joint project between:
  - U. Braunschweig, Germany (Jürgen Schönwälder)
  - U. Twente, The Netherlands (Aiko Pras, Ron Sprenkels)
  - Telecom Italia (Luca Deri)
  - EPFL, Switzerland (JPMF)
Related Publications

