ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Management of IP Networks: from SNMP Pull to Java Push

ICA fortnightly talk
May 10, 1999

Jean-Philippe Martin-Flatin
Swiss Federal Institute of Technology, Lausanne
Institute for computer Communications and Applications
ICA
martin-flatin@epfl.ch
http://icawww.epfl.ch/~jpmf/

Outline

- Problems with SNMP-based network management
- Proposed solution:
  - Web-based management
  - push model
- JAMAP
- Future work

Today’s management of IP networks

- SNMP frameworks (v1, v2c, v3)
  - manager-agent paradigm
  - polling (pull model)
  - notifications (push model)
- SNMP protocols (v1, v2c, v3)
- Network Management Platforms (NMPs): HP OpenView, Cabletron Spectrum, IBM Netview, Sun Solstice...

Mandatory NMP functions:	**Optional NMP functions:**
- network monitoring	- configuration
- data collection	- inventory
- notification handling	- ACLs
	- billing...

- Vendor- or device-specific add-ons (e.g. CiscoWorks)

NMP Problems (1/2)

- For customers:
  - too expensive (hardware and software):
    - ➠ dedicated hardware for network management
  - offer limited support for third-party RDBMSs
  - cost to migrate from Unix to Windows is too high:
    - ➠ Unix expertise required to maintain existing platforms
- For network equipment vendors:
  - the support of device-specific add-ons is too expensive:
    - ➠ many NMPs
    - ➠ many OSs
    - ➠ many add-ons

NMP Problems (2/2)

- For customers and network equipment vendors:
  - poor time-to-market for add-ons:
    - large vendors: several months after hardware release
    - startups: never --> need separate NMPs
  - versioning:
    - MIB upgrade in a network --> version mismatch between NMPand agents:
      - update NMP manually, device by device
        (no MIB-discovery protocol)
      - do not use new features of a MIB until all devices are upgraded

SNMP Problems (1/2)

- SNMP expertise is scarce and expensive (esp. SNMPv3)
- Scalability, network overhead and latency are adversely affected by some protocol design decisions:
  - BER encoding
  - SNMP table retrieval mechanism
  - OIDs take much more space than values
  - no compression
- Low-level semantics:
  - aimed at instrumentation
  - no standard high-level APIs
  - site-specific network applications developed from scratch:
    - ➠ bound to an NMP API, not a technology

SNMP Problems (2/2)

- Security:
  - SNMPv1 and SNMPv2c: no secure SNMP `get` or `set`
  - SNMPv3: not used yet
  - VPNs: expensive encryption hardware to manage remote subsidiaries
  - firewalls: UDP relays
- Unreliable transport protocol:
  - important SNMP notifications (unacknowledged) are lost for silly reasons (e.g. buffer overflow)
  - SNMPv3 informs (acknowledged) are not used yet
  - important mgmt data requires retransmissions at the application level
- Evolution of SNMP hampered by legacy systems:
  - “better replace than repair”

Proposed Solution (1/2)

- Change management framework:
  - SNMP-based management --> Web-based management
  - pull model --> push model
  - keep MIBs
- Change communication protocol:
  - SNMP --> HTTP
    - connectionless UDP --> persistent TCP connections
  - `gzip` compression
  - 1 OID per message --> n OIDs per push cycle
  - BER encoding --> MIME parts + strings

Proposed Solution (2/2)

- Change NMP:
  - split manager:
    - management server
    - management station
  - expensive NMP software --> inexpensive Java software
  - monolithic NMP --> distributed servlets
  - expensive add-ons --> less expensive Java applets
  - dedicated NMP hardware --> any hardware
  - few third-party RDBMSs --> all RDBMSs via JDBC

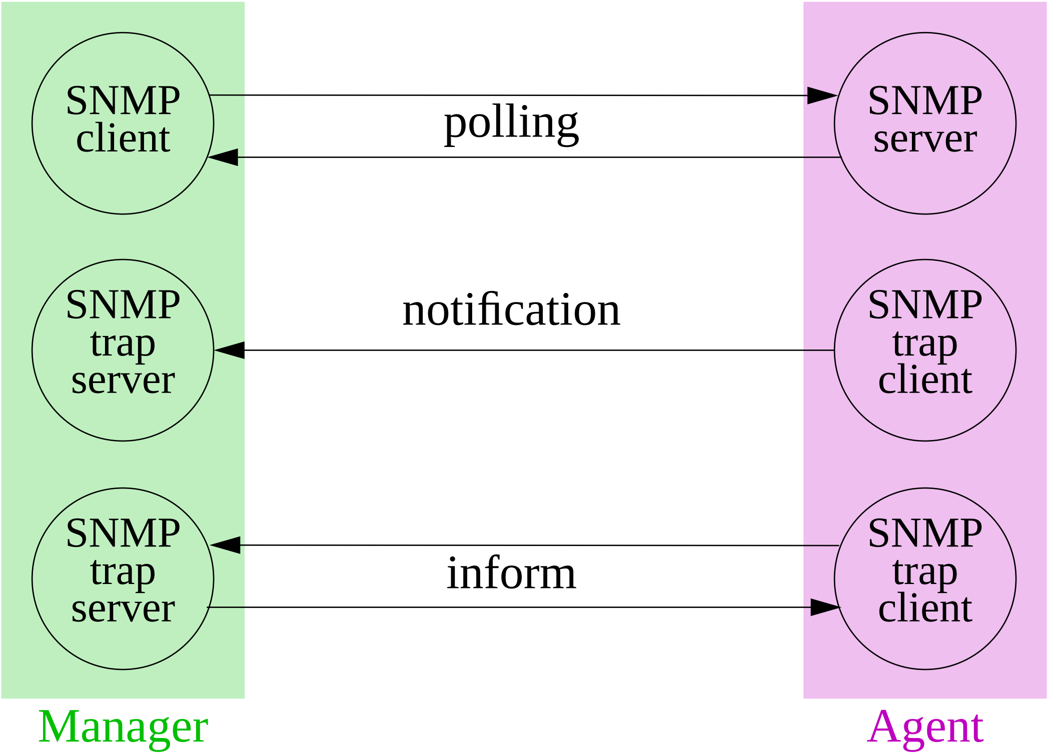
Web-Based Management

- Definition: network, systems, application and service management based on Web technologies
- Much choice:
    - HTML forms
    - CGI scripts (Perl, Tcl/Tk, shell, binaries)
    - Java applets, servlets and applications
    - Java Object Serialization
    - Java RMI (distributed objects)
    - Java IDL (CORBA)
    - JDBC (RDBMSs)
    - ...

Distribution in Java: Why Use HTTP?

- Four techniques to communicate between agents and managers:
  - HTTP
  - sockets
  - RMI
  - Java IDL (CORBA)
- Distributed objects in network management (RMI or CORBA):
  - telecoms = yes
  - Internet = no
- HTTP > sockets:
  - firewall setup slightly easier for nonexperts
  - natural communication between servlets of the mgmt server
  - same technology between agents and servers and within the servers

Pull vs. Push in SNMP-Based Management



From Pull to Push (1/2)

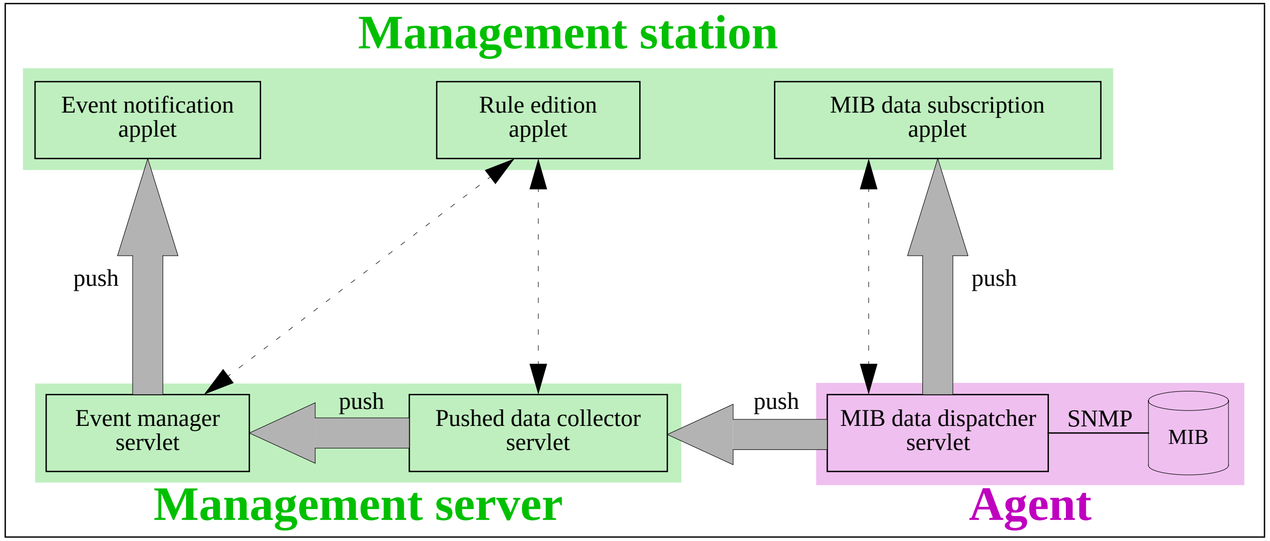
- Pull Model:
  - Request-response paradigm
  - Strict client-server communication model:
    - client requests data from server
    - server sends data to client
  - Client = manager
  - Servers = agents
  - Data transfers initiated by the client
  - Example in SNMP-based NMPs: data polling for network monitoring and data collection

From Pull to Push (2/2)

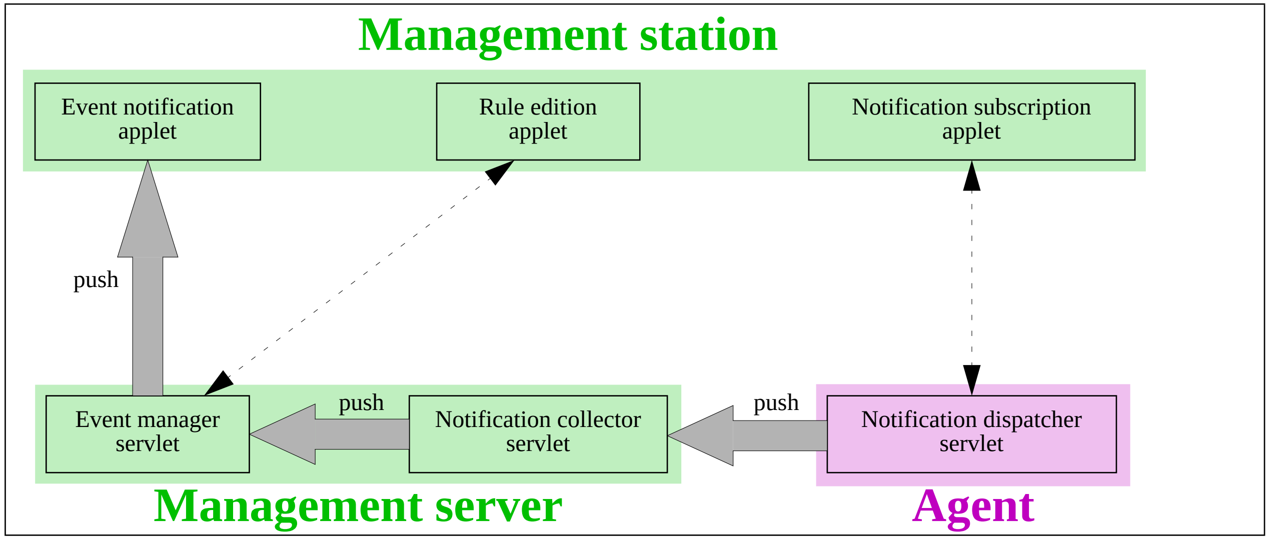
- Push Model:
  - Publish-subscribe paradigm:
    - 3 phases: publication, subscription and distribution
  - Pseudo client-server communication model:
    - client sends data to server
    - server may acknowledge (e.g. SNMPv3 informs) or not acknowledge (e.g. SNMPv1 traps and SNMPv2 notifications) receipt of this data
  - Clients = agents
  - Server = manager
  - Parallel and independent data transfers initiated by the clients

JAMAP: Global Picture

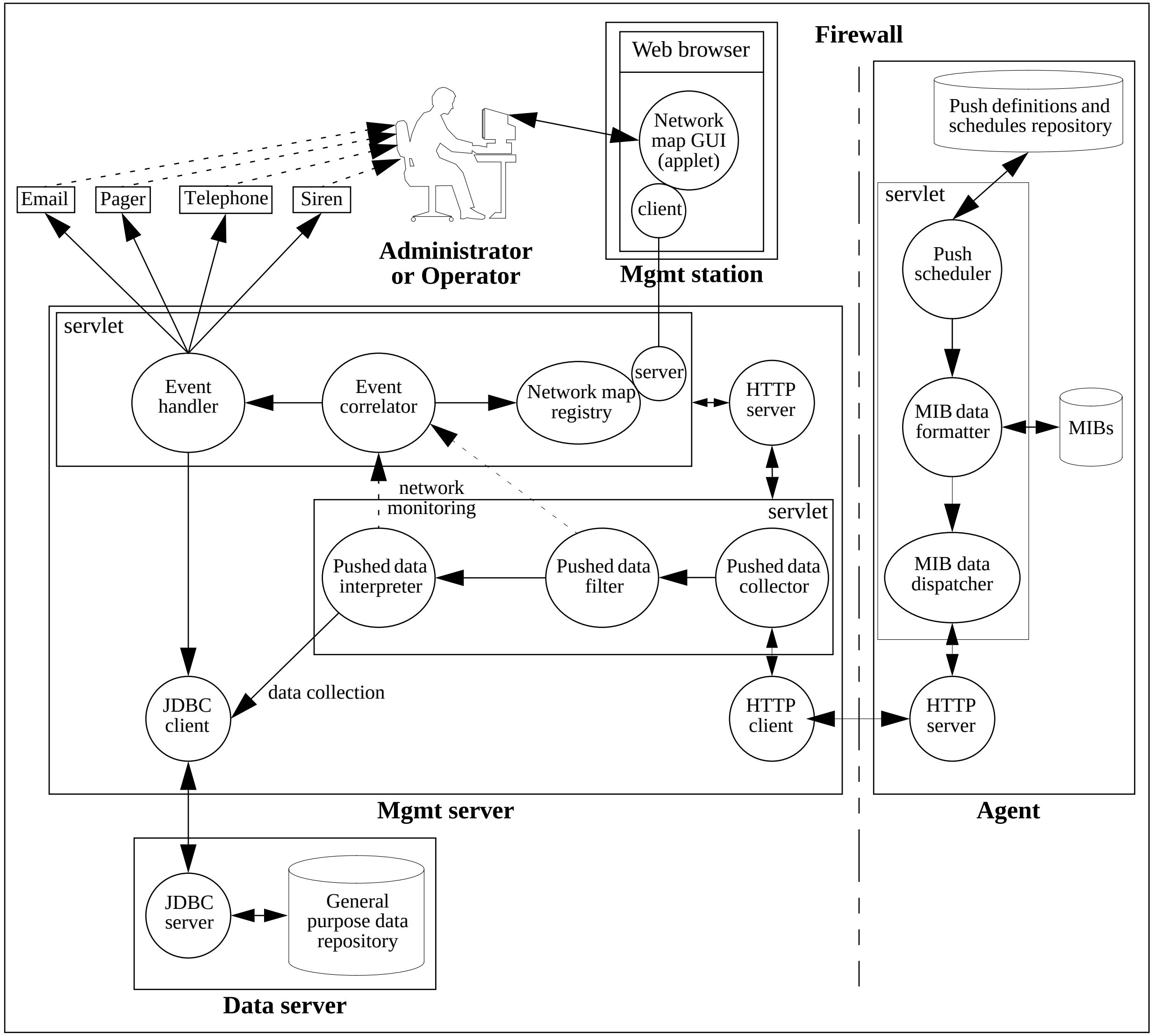
Monitoring and Data Collection


Management station
Event notification applet
Rule edition applet
MIB data subscription applet
push
push
push
push
Event manager servlet
Pushed data collector servlet
MIB data dispatcher servlet
SNMP
MIB
Management server
Agent

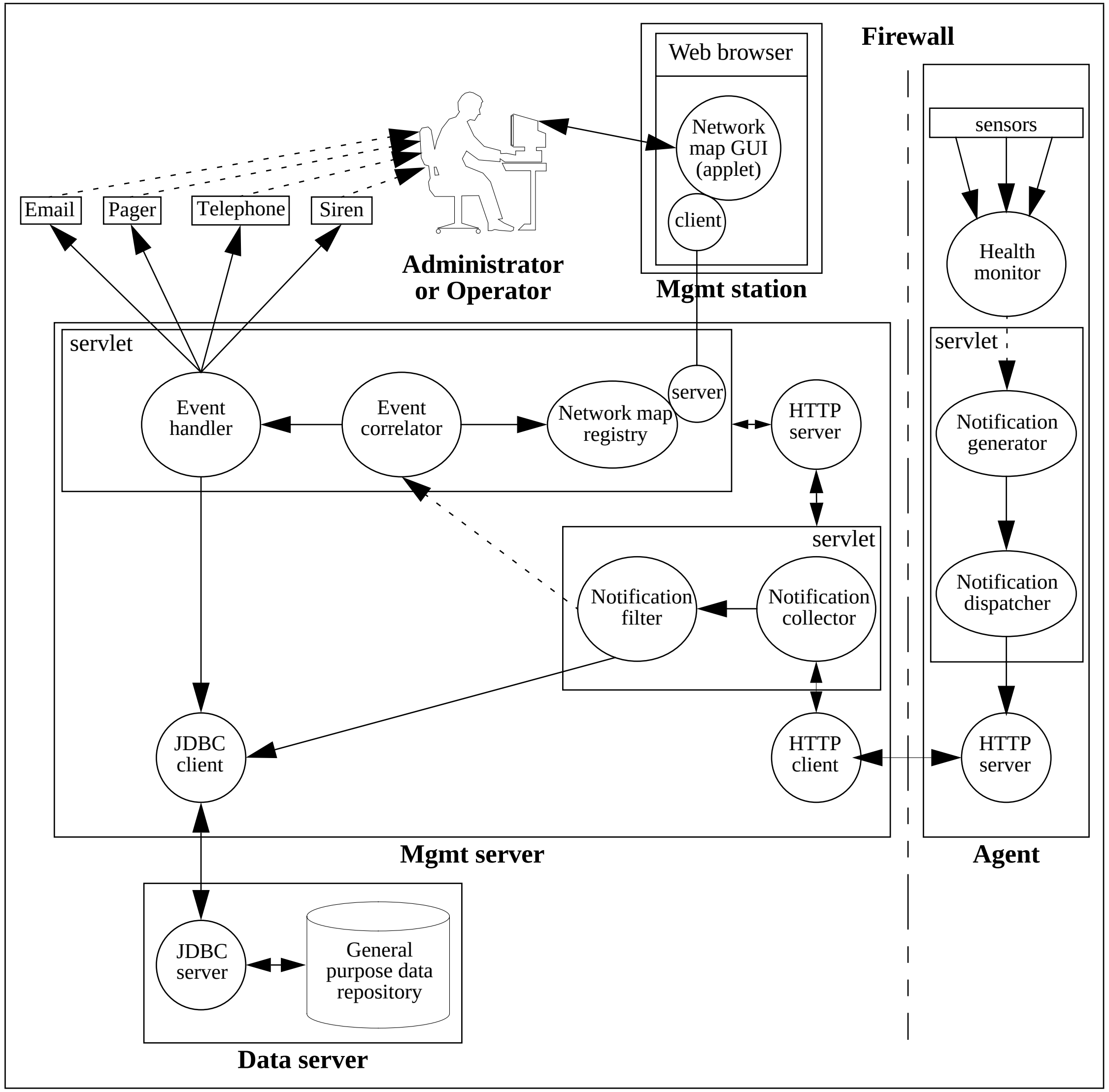
Notifications


Management station
Event notification applet
Rule edition applet
Notification subscription applet
push
push
push
Event manager servlet
Notification collector servlet
Notification dispatcher servlet
Management server
Agent

JAMAP: Monitoring and Data Collection



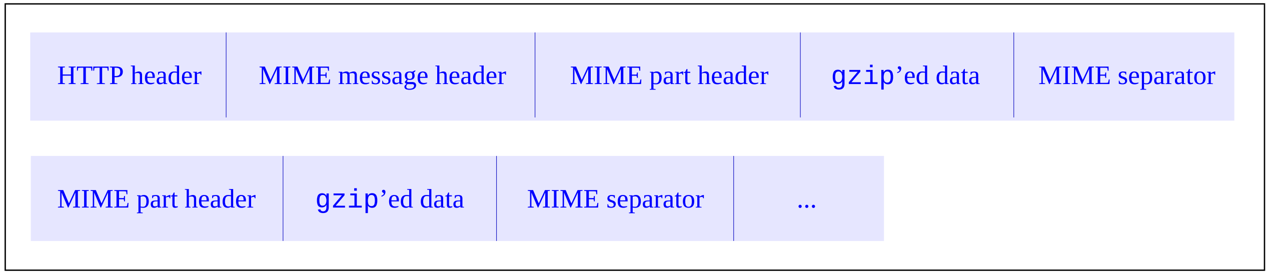
JAMAP: Notifications



Issues

- Firewalls: connection should be created by internal manager, not external agent
- 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!
- Persistent connections:
  - need to control timeout by HTTP server
  - need to reconnect in case of teardown

HTTP and MIME



HTTP header | MIME message header | MIME part header | `gzip`'ed data | MIME separator

MIME part header | `gzip`'ed data | MIME separator | ...

- MIME = Multipurpose Internet Mail Extensions
- Advantages:
  - simple to implement
  - firewalls: minor change (assuming Web access already)
- Drawbacks:
  - manager must detect network outage to set up new connection:
    - send keepalives if no data after 9 minutes
    - blind during 9 minutes

Related Work

- Architectures:
  - Bruins, Deri, Harrison *et al.*, Maston, Mullaney, Thompson
- Prototypes:
  - Marvel by Anerousis, CyberAgent by Burns and Quinn, Webbin by Barillaud *et al.*, WbASM by Kasteleijn, NetFinity by Reed *et al.*
- Commercial offers:
  - JDMK by Sun (JMAPI)
  - AdventNet
  - Metrix
  - NMPs moving to Web-based configuration management

Conclusion (1/2)

What do we gain by going from SNMP pull to Java push to manage IP networks?

- Get rid of the expensive NMP
- Get rid of SNMP, use well-known Web technologies instead
- Reduce network overhead of management data
- Reduce development costs of add-ons
- Zero 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

Conclusion (2/2)

What does it cost to go from SNMP pull to Java push to manage IP networks?

- network equipment vendors must add 2 things to their equipment:
  - a push system
  - a scheduling system
- administrators need to synchronize the clocks of the managers and the agents (e.g. with NTP)
- we need professional-grade software on the manager side
  - JAMAP is simply a prototype

Future Work

- Performance analysis
- Scalability: is Java too slow?
- XML: higher level semantics
- EmbeddedJava:
  - promises: lightweight JVM and RMI
  - dangers: royalties to Sun? controlled by Sun?