Automatic Configuration and Reconfiguration of a Web-based Network Traffic Monitoring System

ykpark1@tigris.postech.ac.kr
DP & NM, GSIT, POSTECH

1999. 12. 17
Contents

• Introduction
• Related Work
  – Auto Configuration and Reconfiguration
  – MRTG
  – Network Map
• System Requirements
• System Design
• Implementation
• Conclusion and Future Work
Introduction

• As networks grow in size, speed, and complexity, the role of network management becomes increasingly important

• Importance of monitoring for network usage
  – use effectively present network resources
  – plan future networking
  – network traffic monitoring information can be supported

• Commercial tools
  – support various monitoring functions
  – difficult to learn and use

• Need a network traffic monitoring system
  – to be used and managed easily
  – support auto configuration and reconfiguration
Auto Configuration and Reconfiguration

- Simplify configuration
  
  directly configure each device

  globally configure
  (configure simultaneously on multiple devices)

- Minimal user intervention to manage
- Easy user interface
- Discovery
  - automatically finds specific types of devices to be monitored
- Monitor and apply configuration changes automatically
Multi Router Traffic Grapher (MRTG)

• Monitors the traffic load on network-links
• Generates HTML pages containing GIF images which provide a visual representation of this traffic

• Features
  – most UNIX, windows NT
  – easy customization
  – portable SNMP manger
  – daily, weekly, monthly, yearly graph
  – monitor any SNMP variable

• Shortcomings
  – not easy for configuration management
  – not support network map
Multi Router Traffic Grapher (MRTG)

Traffic Analysis for Serial2/1 (KT E1)

시스템: FDDICC:postech.ac.kr in postech-ce
인터페이스: Serial2/1 (3)
네트워크 주소: No hostname defined for IP address (203.251.71.170)
최대 속도: 256.0 kBytes/sec (PointToPointSerial)

통계치 최종 거시일 1999년 12월 8일 수요일, 19:54.
FDDICC:postech.ac.kr 시스템은 77 days, 7:45:43 동안 살아있었음.

일일* Graph (5분 평균)

최대 In: 1730.7 kbps (83.5%) 평균 In: 1245.8 kbps (60.5%) 현재 In: 833.6 kbps (42.7%)
최대 Out: 1023.8 kbps (30.0%) 평균 Out: 278.4 kbps (13.4%) 현재 Out: 252.8 kbps (12.3%)

주간* Graph (30분 평균)

1999. 12. 17
Network Map

• Necessary for an overview of the physical/logical topology
• Map generation problems
  – information services present that supply network related information
  – no place that in any manner provides information about the connectivity of the various network elements, manually generate
• MIBs for map
  – BGP, OSPF, MIB-II
• Considerations
  – connectivity
  – properties and functions
  – policy, network name and address related information
  – geometric and geographical information
System Requirements

• Automatic discovery
  – look for devices in the specified range of addresses
  – run discovery at any pre-determined time

• Selective monitoring configuration

• Manageability
  – add/remove management agents
  – job control
  – show/change current status
  – check/apply configuration changes
System Requirements (cont’d)

• Network map
  – shows the connectivity of each devices
  – supports the device information

• Web interface
  – de facto, standard user interface
  – easy to use, ubiquitous

• Security
  – differentiate manager from user in permission
  – manager : configuration, monitoring
  – user : monitoring
System Design: Architecture

Discoverer
Configuration Manager
Map Generator
User Interface
Main Functions
System Design: Discoverer

- IP Address Checker (ping test)
- SNMP Checker
  - get-request (System.sysServices)

<table>
<thead>
<tr>
<th>Layer</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>physical (repeaters)</td>
</tr>
<tr>
<td>2</td>
<td>datalink/subnetwork (bridges)</td>
</tr>
<tr>
<td>3</td>
<td>internet (IP routers)</td>
</tr>
<tr>
<td>4</td>
<td>end-to-end (IP hosts)</td>
</tr>
<tr>
<td>7</td>
<td>applications (mail relays)</td>
</tr>
</tbody>
</table>

- Duplication Checker
  - select represent IP address in an agent

1999. 12. 17
System Design: Configuration Manager

- SNMP Selector
  - select agents and configure (cfg, menu)
- Set Reporting Parameter
- Job Controller
  - use crontab
- Status Manager
  - show and modify detailed monitoring status
  - port-level management
- Configuration change Manager
- Adder/Remover
System Design: Map Generator

- Connectivity Checker

<table>
<thead>
<tr>
<th>ipRoute</th>
<th>IfIndex</th>
</tr>
</thead>
<tbody>
<tr>
<td>141.223.0</td>
<td>10</td>
</tr>
<tr>
<td>141.223.39.0</td>
<td>22</td>
</tr>
<tr>
<td>141.223.180.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ipRoute</th>
<th>NextHop</th>
</tr>
</thead>
<tbody>
<tr>
<td>141.223.3</td>
<td>10</td>
</tr>
<tr>
<td>141.223.39</td>
<td>22</td>
</tr>
<tr>
<td>141.223.180</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ipAdEnt</th>
<th>IfIndex</th>
</tr>
</thead>
<tbody>
<tr>
<td>141.223.160.254</td>
<td>22</td>
</tr>
</tbody>
</table>

---

1999. 12. 17

POSTECH
DP&NM Lab.
System Design: Map Generator (cont’d)

- Map table Generator

  DNS name : x position : y position : image file : image size
  : : : : :
  x position : y position

  : DNS name <---> DNS name
  : : :

- Router information Generator
  - get-request to nexthop router
  - system name, connected port number, DNS name

- Map Drawer
  - editing, link with router information
System Design: User Interface

- Web browser
- Additional Web server
  - change daemon name, cgi-bin directory, port number
  - authentication service
Implementation: Environments

• Hardware

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel Pentium</td>
</tr>
<tr>
<td>Memory</td>
<td>32 MB</td>
</tr>
</tbody>
</table>

• Software

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation System</td>
<td>Linux release 6.0 Kernel 2.2.5-22</td>
</tr>
<tr>
<td>Language</td>
<td>C(gcc2.0), C++(egcs-2.91.66), Perl 5.003_03</td>
</tr>
<tr>
<td>Web server</td>
<td>Apache 1.3.9</td>
</tr>
<tr>
<td>Traffic monitoring tool</td>
<td>MRTG-2.7.1</td>
</tr>
<tr>
<td>Graphic libraries</td>
<td>GD-1.3, 1.7.3</td>
</tr>
<tr>
<td>SNMP manager tool</td>
<td>UCD-snmp-4.0.1</td>
</tr>
</tbody>
</table>
Implementation: Discoverer

- **SNMP checker**

- **Duplication check**
  - using IpAdEntIfIndex table

```cpp
global liveHostList, liveSNMPAgentList;
snmpfinder(retry, startIP, endIP)
{
   ////////// Ping part //////////
   thread_create(recvICMP);
   for (0 .. retry) {
      for IP = (startIP .. endIP)
         sendICMP(ECHO_REQUEST, IP);   sleep(1);
   }
   thread_join(recvICMP);

   ////////// SNMP part //////////
   thread_create(recvSNMP);
   for (0 .. retry) {
      foreach IP in liveHostList
         sendSNMPGet(SYSSERVICES, IP);  sleep(1);
   }
   thread_join(recvSNMP);

   ////////// Duplication Check //////////
   foreach IP in liveSNMPAgentList {
      if(dupTable.has(IP)) continue;
      else {
         resultTable.insert(IP);
         dupTable.insert(SNMPWalk(IPADENT_IFINDEX, IP));
      }
   }
   return resultTable;
}
```
Implementation: Configuration Manager

Selective monitoring configuration

Adder
Implementation: Configuration Manager (cont’d)

Status Manager
Implementation: Map Generator

Connectivity Checker

makeIfIndexNextHop()
{
ipAdEntIfIndexTable = snmpWalk(IPADENT_IFINDEX);
ipRouteIfIndexTable = snmpWalk(IPROUTE_IFINDEX);
ipRouteNextHopTable = snmpWalk(IPROUTE_NEXTHOP);

ifIndexIpRouteTable = reverseKeyData(ipRouteIfIndexTable);
ifIndexIpAdEntTable = reverseKeyData(ipAdEntIfIndexTable);

foreach key in ifIndexIpRouteTable {
  nextHop = ipRouteNextHopTable[key];
  ifIndexNextHopTable[key] = nextHop;
}

foreach key in ifIndexNextHopTable {
  if(ifIndexNextHopTable[key] == ifIndexIpAdEntTable[key])
    deleteData(ifIndexNextHopTable[key]);
}
}

sub setRouterLocation(connectionTable)
{
  foreach key in connectionTable {
    if(searchedRouter.has(key)) continue;
    locationList[key] = (...);
    searchedRouter.insert(key);

    foreach data in connectionTable[key] {
      locationList[data] = (...);
      searchedRouter.insert(data);
    }
  }

  Find minimum position in locationList
  Move this position to (100, 100)
}

Map table Generator
Implementation: Map Generator (cont’d)

### Router Information Generator

<table>
<thead>
<tr>
<th>Interface Map Information of Router Giga_Center2.postech.ac.kr</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Index</strong></td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>22</td>
</tr>
</tbody>
</table>
Implementation: Map Generator (cont’d)

Generated Network Map

Selectable images
Cisco2900.png  Cisco5500.png  Cisco7500.png  Router1.png  Router2.png

Location Information

<table>
<thead>
<tr>
<th>Name</th>
<th>x</th>
<th>y</th>
<th>Image</th>
<th>Ratio(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Giga-Center1.postech.ac.kr</td>
<td>155</td>
<td>120</td>
<td>Cisco5500.png</td>
<td>50</td>
</tr>
<tr>
<td>Giga-Center2.postech.ac.kr</td>
<td>130</td>
<td>203</td>
<td>Cisco5500.png</td>
<td>50</td>
</tr>
</tbody>
</table>
Conclusion and Future Work

• Easy to use and manage for network traffic monitoring system
• Auto configuration and reconfiguration
  – discovery
  – setup and manage configuration DB
  – network map
  – Web interface, minimum user intervention

• Future work
  – various MIB-II and private MIB for more performance monitoring
  – support connectivity information of layer 2 devices