Automatic Configuration and Reconfiguration of a Web-based Network Traffic Monitoring System
(웹 기반의 네트워크 트래픽 모니터링 시스템의 자동 구성 및 재구성 기법)

박 윤 규

ykpark1@tigris.postech.ac.kr

DP & NM, GSIT, POSTECH

Contents

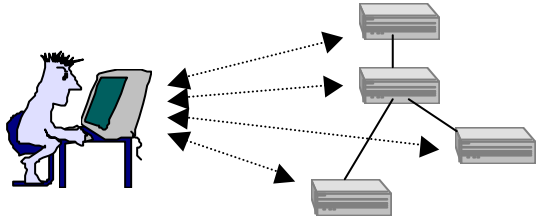
- Introduction
- Related Work
  - Auto Configuration and Reconfiguration
  - MRTG
  - Network Map
- System Requirements
- System Design
- Implementation
- Conclusion and Future Work

Introduction

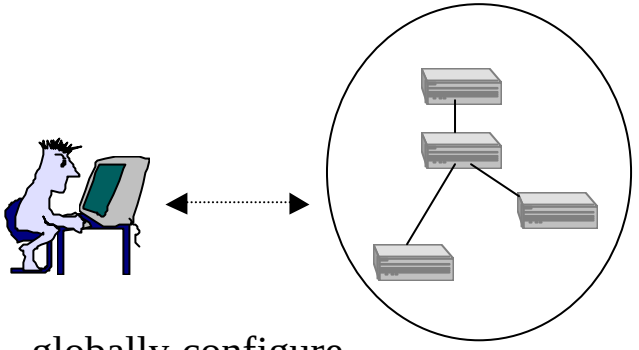
- As networks grow in size, speed, and complexity, the role of network management becomes increasingly important
- Importance of monitoring for network usage
  - use effectively present network resources
  - plan future networking
  - network traffic monitoring information can be supported
- Commercial 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

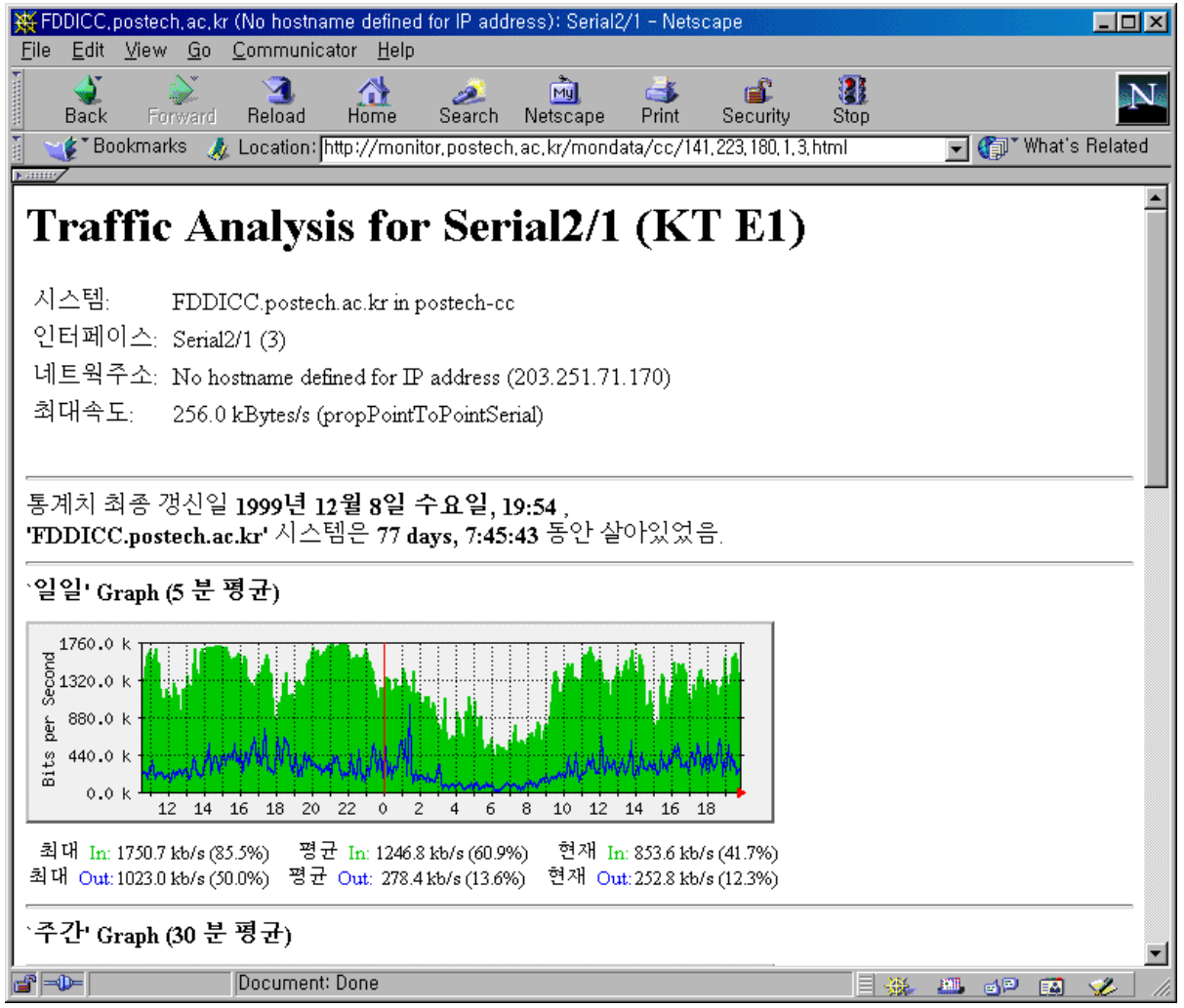
(configure simultaneously on multiple devices)

- 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)



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

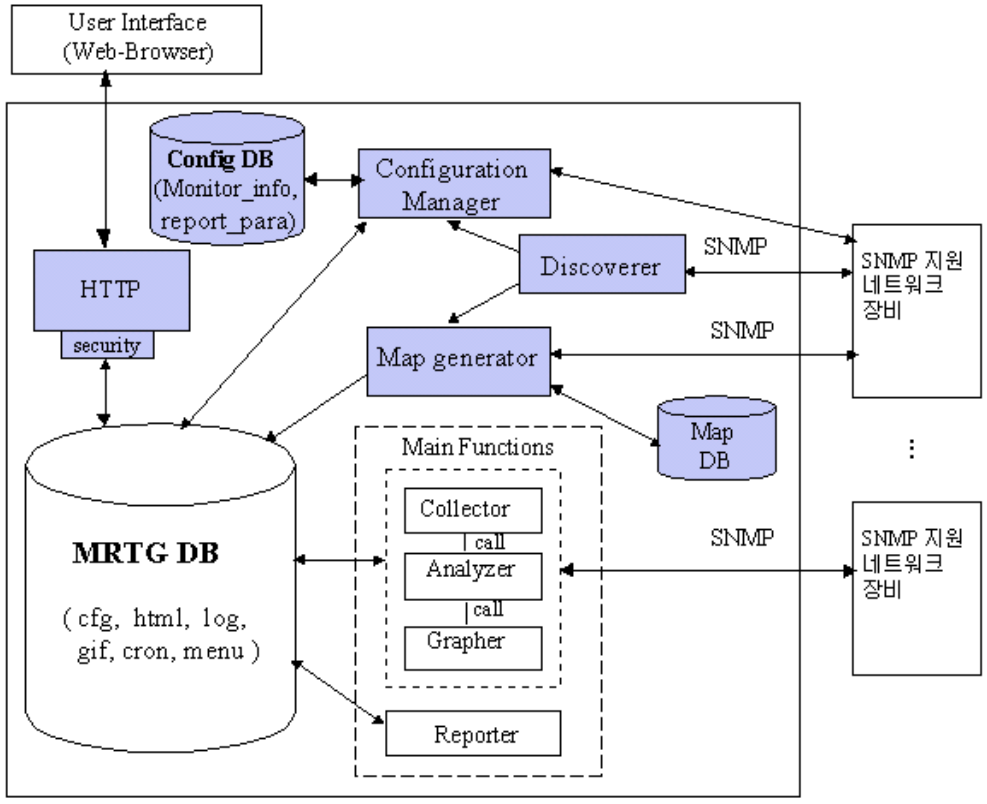
System Requirements

- 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

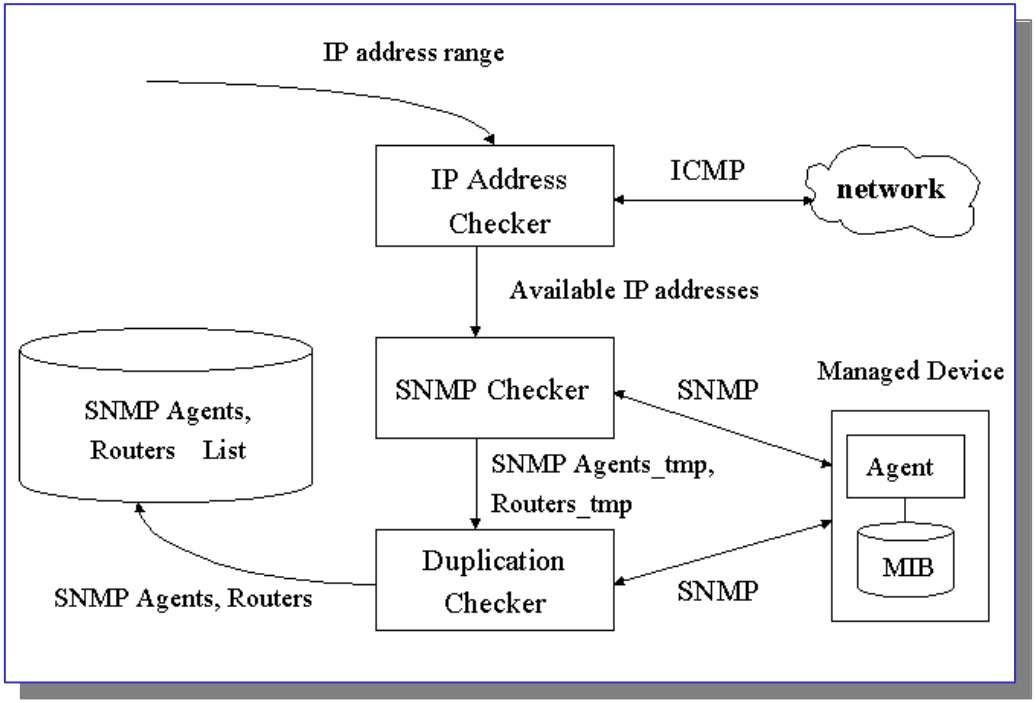
User Interface

Main Functions

System Design: Discoverer

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

Layer	Functionality
1	physical (repeaters)
2	datalink/subnetwork (bridges)
3	internet (IP routers)
4	end-to-end (IP hosts)
7	applications (mail relays)

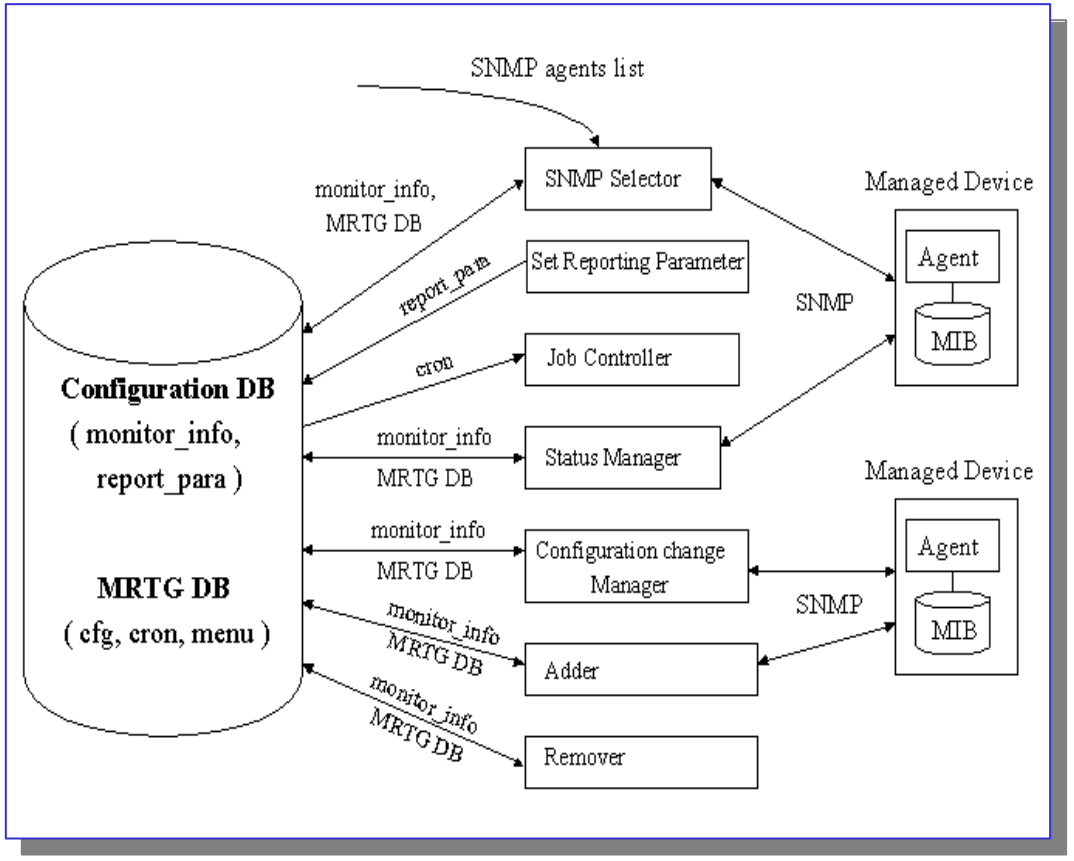


- Duplication Checker
  - select represent IP address in an agent

Router	2	254.2
	11	122.254
	22	180.254

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

ipRouteIfIndex

ipRoute	IfIndex
141.223.3.0 141.223.59.0 141.223.180.0	22

ipRouteNextHop

ipRoute	NextHop
141.223. 3.0	141.223.180.1
141.223. 59.0	141.223.180.1
141.223.180.0	141.223.180.254

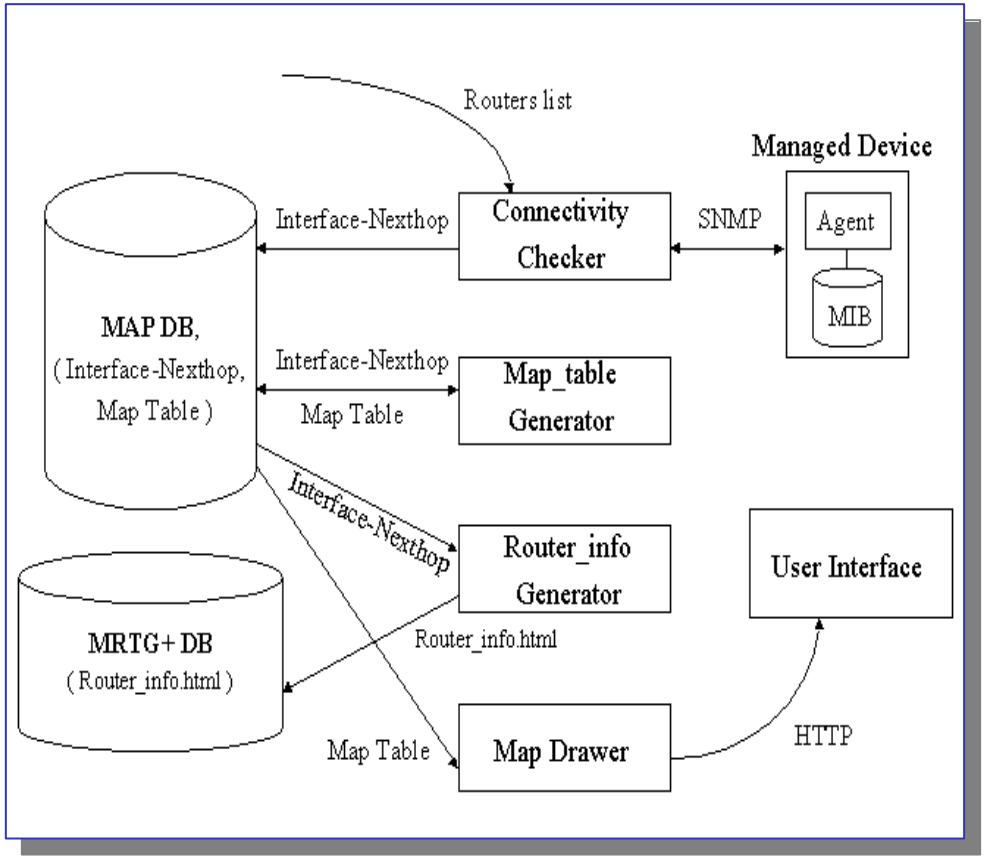
ipAdEntIfIndex

ipAdEnt	IfIndex
141.223.180.254	22

IfIndex	NextHop
22	141.223.180.1 141.223.180.254

IfIndex	NextHop
22	141.223.180.1

Interface-Nexthop



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

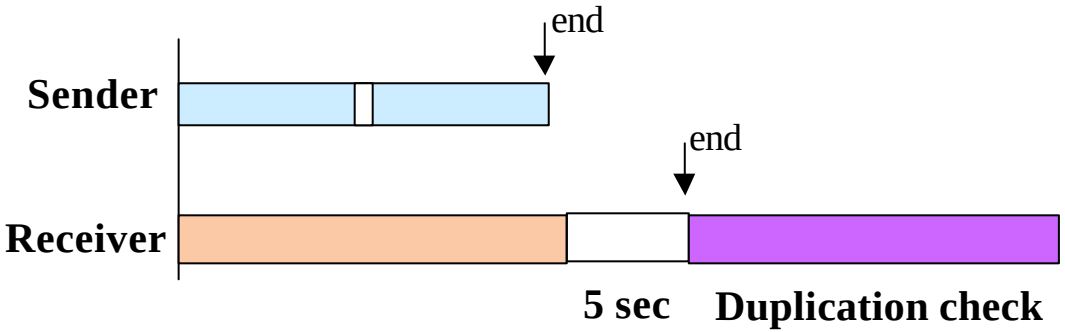
CPU	Intel Pentium
Memory	32 MB

- Software

Operation System	Linux release 6.0 Kernel 2.2.5-22
Language	C(gcc2.0), C++(egcs-2.91.66), Perl 5.003_03
Web server	Apache 1.3.9
Traffic monitoring tool	MRTG-2.7.1
Graphic libraries	GD-1.3, 1.7.3
SNMP manager tool	UCD-snmp-4.0.1

Implementation: Discoverer

- SNMP checker

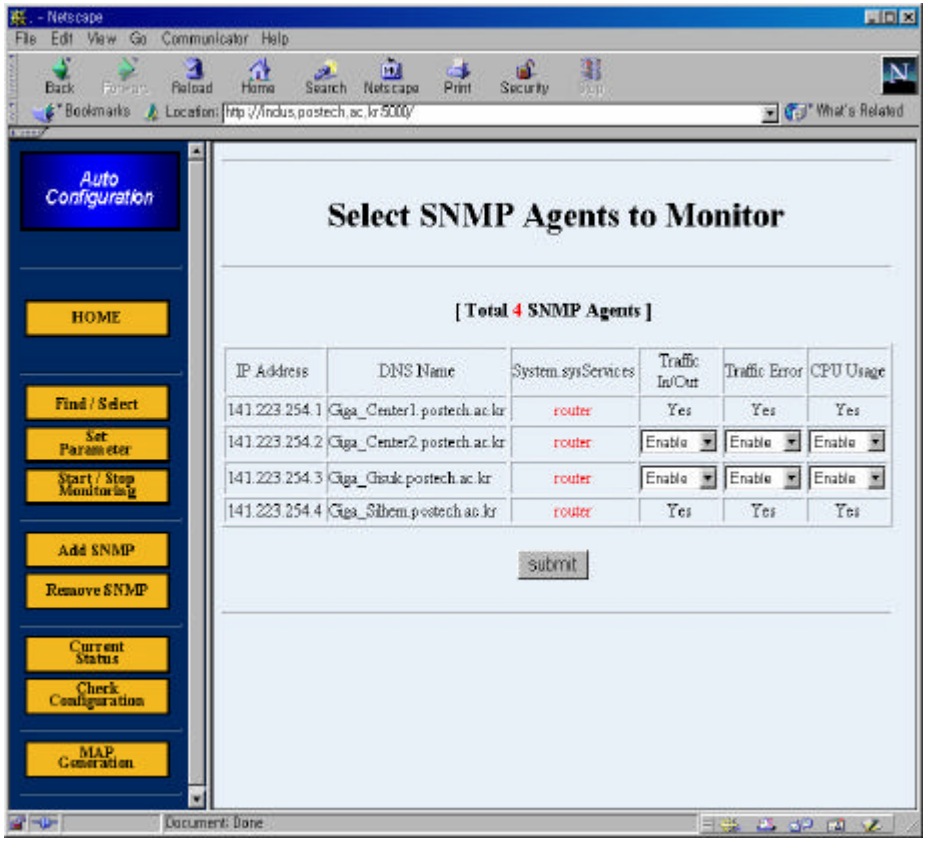


- Duplication check
  - using IpAdEntIfIndex table

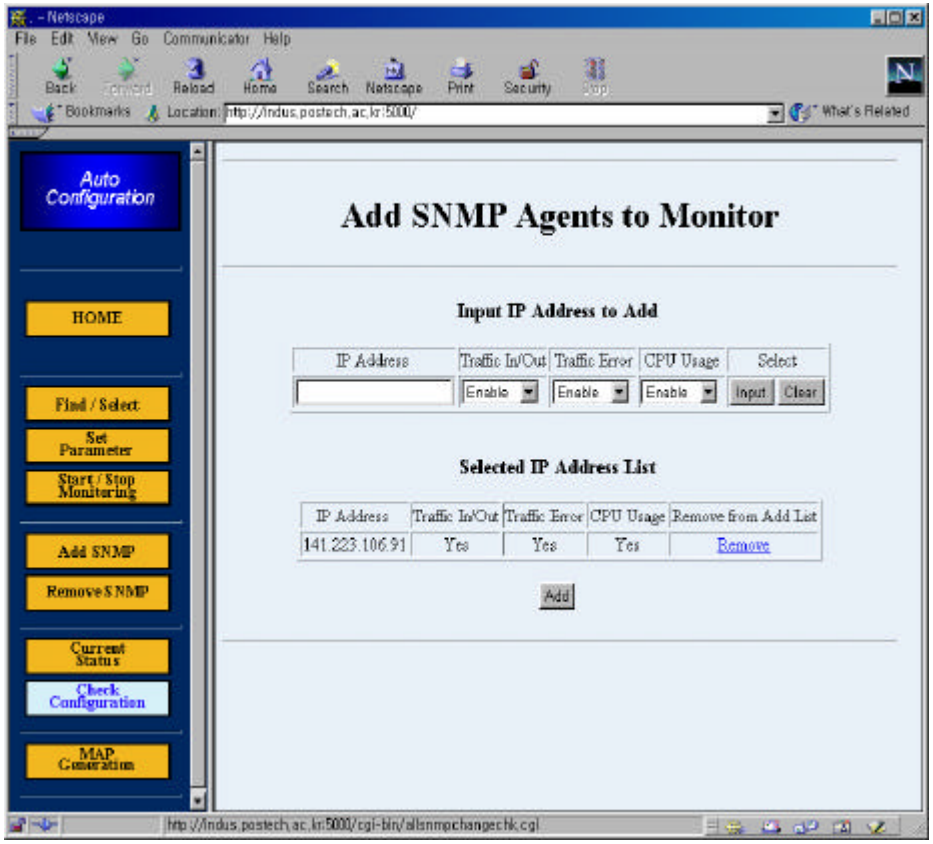
```
 global liveHostList, liveSNMPAgentList;
snmpfinder(retry, startIP, endIP)
{
 ////////// Ping part //////////////////
 thread_create(recvICMP);
 for (0 .. retry) {
  for IP = (startIP .. endIP)
   sendICMP(ECHO_REQUEST, IP);   sleep(1);
 }
 thread_join(recvICMP);

  ////////// 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

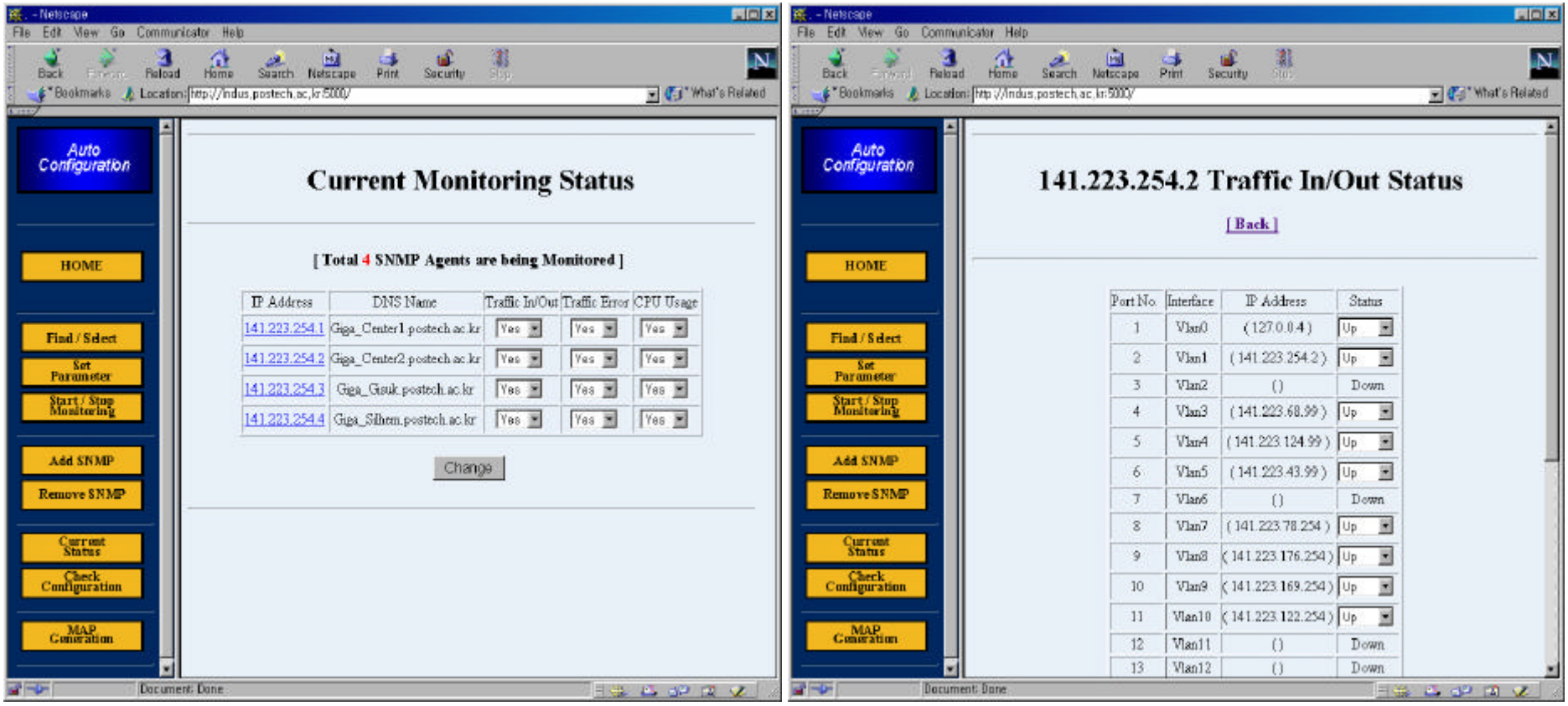


Adder



Implementation: Configuration Manager (cont'd)

Status Manager



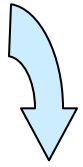
Implementation: Map Generator

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)
}
```

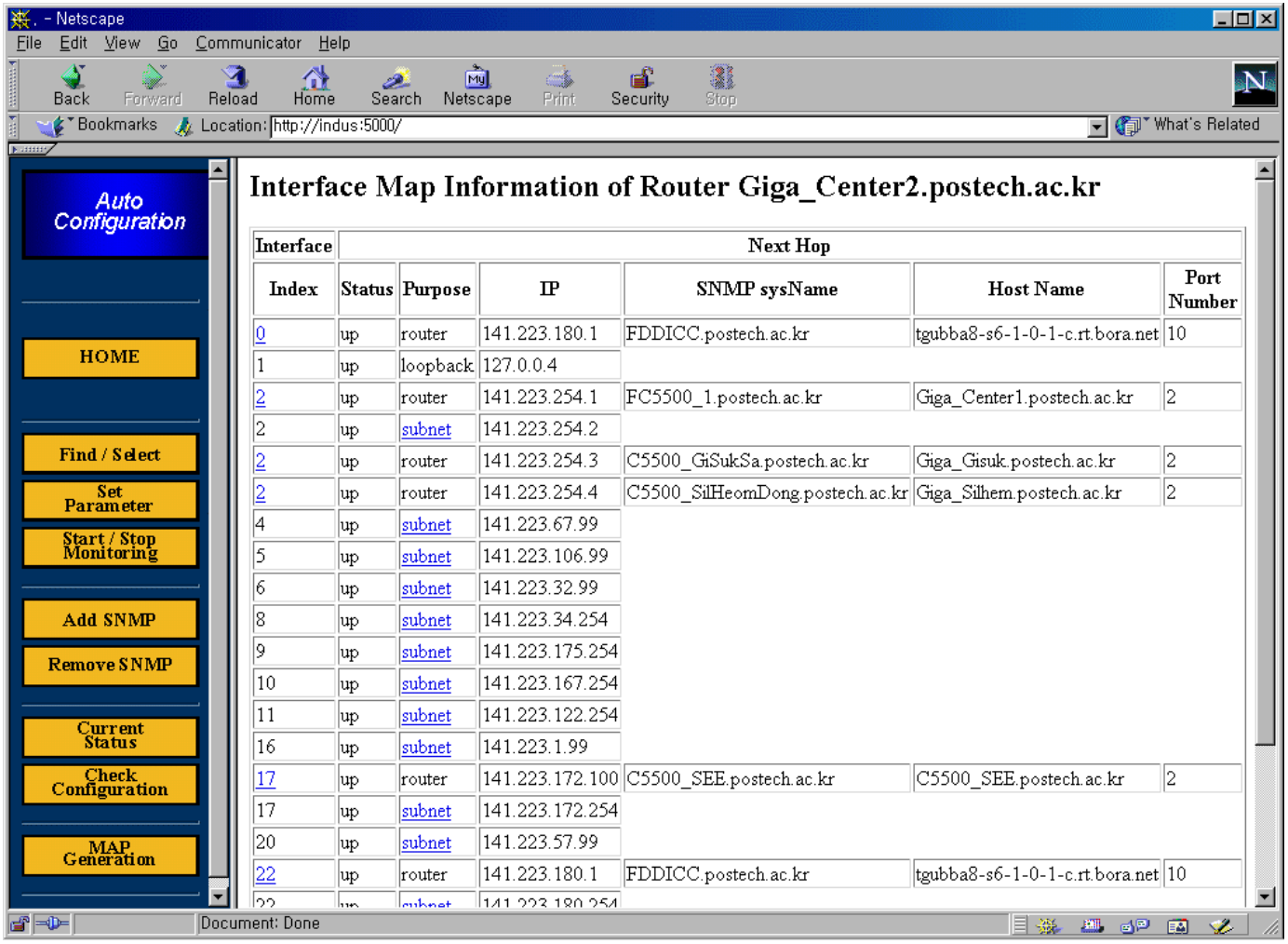
Map table Generator



```
# Number of nodes
4

# Node location and image
Giga_Center1.postech.ac.kr:195:128:Cisco5500.png:50
Giga_Center2.postech.ac.kr:130:203:Cisco5500.png:50
Giga_Gisuk.postech.ac.kr:100:156:Cisco5500.png:50
Giga_Silhem.postech.ac.kr:100:100:Cisco5500.png:50

# Map size
295:303

# Connection
:Giga_Center1.postech.ac.kr<--->Giga_Center2.postech.ac.kr
:Giga_Center1.postech.ac.kr<--->Giga_Gisuk.postech.ac.kr
:Giga_Center1.postech.ac.kr<--->Giga_Silhem.postech.ac.kr
:Giga_Center2.postech.ac.kr<--->Giga_Gisuk.postech.ac.kr
:Giga_Center2.postech.ac.kr<--->Giga_Silhem.postech.ac.kr
:Giga_Gisuk.postech.ac.kr<--->Giga_Silhem.postech.ac.kr
```

Implementation: Map Generator (cont'd)

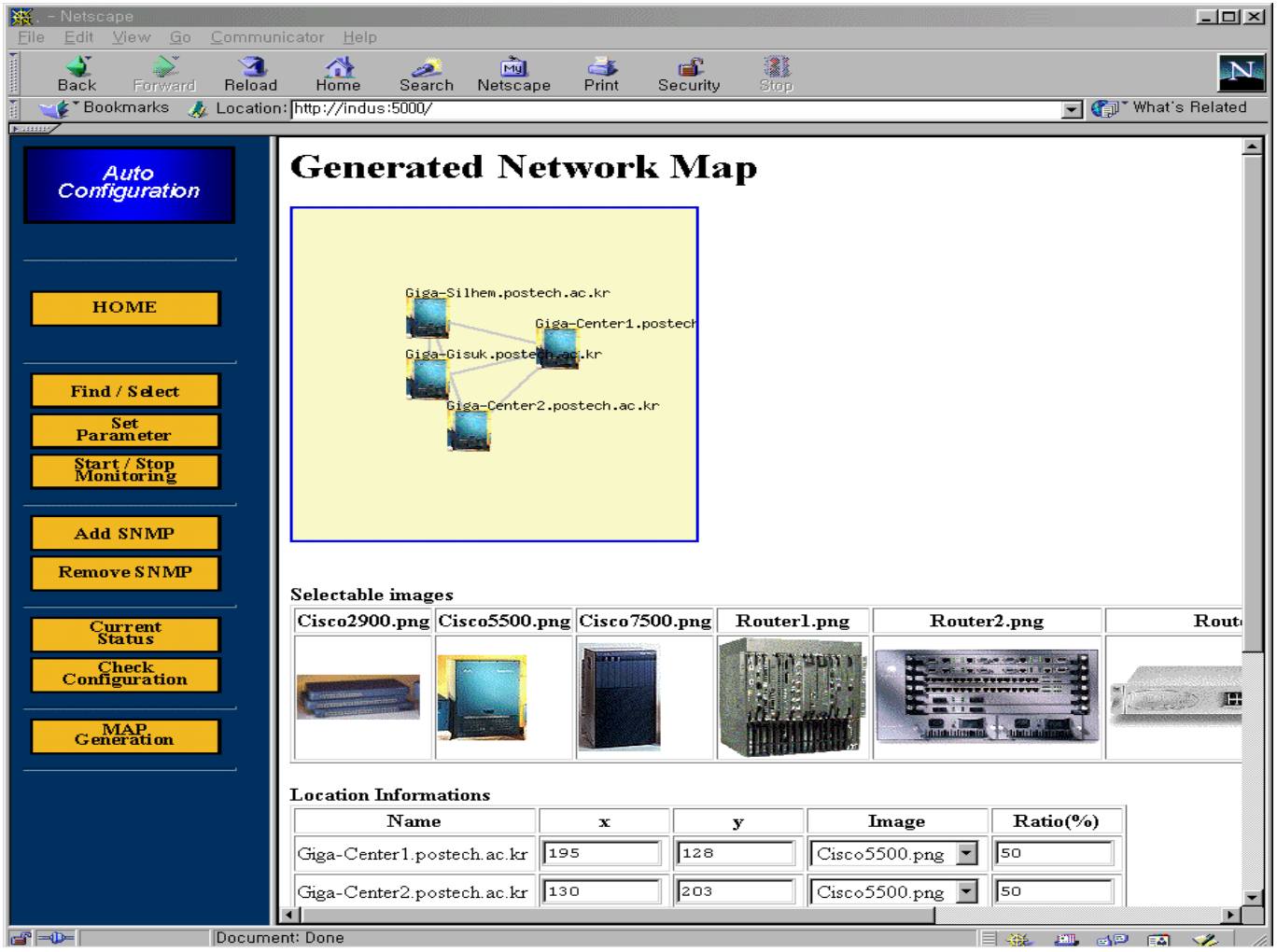
Router information Generator



Interface	Next Hop					
Index	Status	Purpose	IP	SNMP sysName	Host Name	Port Number
0	up	router	141.223.180.1	FDDICC.postech.ac.kr	tgubba8-s6-1-0-1-c.rt.bora.net	10
1	up	loopback	127.0.0.4			
2	up	router	141.223.254.1	FC5500_1.postech.ac.kr	Giga_Center1.postech.ac.kr	2
2	up	subnet	141.223.254.2			
2	up	router	141.223.254.3	C5500_GiSukSa.postech.ac.kr	Giga_Gisuk.postech.ac.kr	2
2	up	router	141.223.254.4	C5500_SilHeomDong.postech.ac.kr	Giga_Silhem.postech.ac.kr	2
4	up	subnet	141.223.67.99			
5	up	subnet	141.223.106.99			
6	up	subnet	141.223.32.99			
8	up	subnet	141.223.34.254			
9	up	subnet	141.223.175.254			
10	up	subnet	141.223.167.254			
11	up	subnet	141.223.122.254			
16	up	subnet	141.223.1.99			
17	up	router	141.223.172.100	C5500_SEE.postech.ac.kr	C5500_SEE.postech.ac.kr	2
17	up	subnet	141.223.172.254			
20	up	subnet	141.223.57.99			
22	up	router	141.223.180.1	FDDICC.postech.ac.kr	tgubba8-s6-1-0-1-c.rt.bora.net	10
22	up	subnet	141.223.180.254			

Implementation: Map Generator (cont'd)

Map Drawer



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