Active networks

■ What are AN (active networks)?
   • Networks that are customized to user needs.
   • Networks that downloads programs to their nodes (i.e., routers, switches, …)

■ The AN view is the programming point of view over networks.

■ If you are interested in the AN:
   • See IWAN’99 papers (1st Int’l Working Conference on Active Networks) published as an LNCS by Springer.
Relation between AN and resource allocation / configuration management

Policy-based resource allocation (configuration management) is the first step towards AN.
- Customizing resources
- Downloading policies to network nodes
  - Policies are rule-based programs with limited functionality.

Why is the programming point of view important?
- While network nodes could be configured only using parameters (data), the AN concept was not necessary.
- Now we need programs for configuration, because the function to be configured is so complex.
- Program semantics must be specified formally for the network to be interoperable.
  - Standard protocols do not guarantee interoperability any longer.
  - Protocols specify only very limited part of the semantics.

Typical programming interfaces and our practices for resource allocation

SNMP & MIB
- DiffServ MIB is discussed in IETF DiffServ WG.
- Kanada et al. (Hitachi) have designed a QoS MIB.
  - Described in draft-kanada-diffserv-qospifmib-00.txt.
  - We have implemented a previous version of the QoS MIB.

COPS & PIB
- Being standardized by IETF RAP WG.
- The QoS MIB (PIB) can be used through COPS.
  - We have implemented a previous version of the QoS PIB.

API
- Being standardized by IEEE P1520 (using CORBA IDL).
- Yoshizawa et al. (Hitachi America) are proposing drafts to IEEE P1520.
  - They are implementing the API.
Design of the QoS MIB

■ It is a step toward establishing programming methodology for active networks.

■ Features of the QoS MIB

• Modularity of rules
  • Modularity is very important for programming.
  • Each classifier, meter, or action is an if-then rule.
    – This separation has become possible by introducing “virtual flow labels”.
  • The order of rules is not significant.
    – The conditions are disjoint (no if-then-else).

• Classification of scheduling algorithms:
  1) First-in first-out (FIFO) scheduling, 2) Priority scheduling, 3) Packet-fair scheduling, 4) Byte-fair scheduling, 5) Bounded byte-fair scheduling

• Classification of dropping algorithms:
  1) Dropping all, 2) Tail dropping (non-early dropping), 3) Random early dropping (RED/WRED), 4) Deterministic early dropping (DED/WDED)

Problems in current programming interfaces

■ A problem in MIB (and PIB):
  Unsuitability for representing program semantics

• Mismatching of syntactic structures
  • No classification rule nor meter structure in the MIB.
  • This causes difficulty in understanding a MIB.

• Mismatching of unit of operation
  • Always single get/put on a variable is the unit of operation.
  • This causes difficulty in MIB/PIB implementation and usage
    – This caused serious bugs in our project.

■ A problem in API:
  Difficulty in passing optional data

• Number of parameters is fixed in C/C++-like languages.
  • It is difficult to design APIs that can be used even in exceptional cases.
  • This problem may be solved by using Lisp-like languages.
Possible solutions

To design a rule-based programming language for the interface.

- A policy is a rule-based program.
- Program semantics should be expressed by a programming language.
- This language may be similar to languages for expert systems, such as OPS5.
  - We may have to learn from AI and Knowledge Engineering.

Possible solutions (Cont’d)

This language may be implemented using any protocol.

- Either SNMP & MIB, COPS & PIB, API (IIOP), or other protocols.
- If COPS is used, the language semantics must be mapped to the COPS usage formally.

Or, the definition of a protocol must embed a language definition.

- A new method of specifying protocols is required.