Our experience

- We have designed and implemented a QoS MIB/PIB for policy-based QoS control of routers.
  - Written in draft-kanada-diffserv-qospifmib-00.txt
  - Implemented as a MIB using SNMP
  - Implemented as a PIB using COPS
- We have experienced much difficulty.
  - Difficulty in understanding the structure of MIB/PIB
    - **Syntactic gap**: The syntax does not reflect the logical structure of policy rules, queues, …
  - Difficulty in implementing and using the MIB/PIB
    - **Semantic gap**: The unit of operation on a MIB is too small.
      - A policy rule should be handled as whole.
      - In SMI, each variable in a MIB is separately get/put.
        - The order and other *implicit* constraints must be satisfied.
      - This causes difficulty in mapping MIB operations to rule operations.
Problem

- **Service level**
  - SLAs
  - Policy server
  - Policy rules

- **Network level**
  - COPS Usage
  - API (IIOP)

- **Device level**
  - SNMP & MIB
  - COPS & PIB

*Large gaps* (syntactic & semantic)
— must be filled by something

Possible solution

- **Service level**
  - SLAs
  - Policy server
  - Policy rules

- **Network level**
  - Rule-based programming language
    - *(draft-kanada-???) — not yet available*
    - *or, MIB sequencer*
    - *(draft-miyake-???) — soon available*

- **Device level**
  - Network nodes (routers, switches, …)
Why do we need a programming language?

- A language has its syntax and semantics that are suited to a specific purpose.

- Policy-based configuration is programming.
  - Network nodes have been configured only using parameters (data).
  - We need programs for configuration, because the function to be configured is so complex.
    - If-then rules (policy rules) are programs.
    - Network nodes are going to be intelligent.
  - 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.

What kind of language?

- A rule-based language
  - Because a policy is a rule-based program.

- This language may be similar to languages for expert systems, such as OPS5 or Nexpert Object.
  - We may have to learn from AI and Knowledge Engineering.
Relation between the language and the protocols

This language may be used with 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.
  - COPS-PR already contains language syntax definitions.
    - e.g. `<Request> ::= <Common Header>
    - `<Client Handle`
    - `<Context = config request>`
    - `[<Named ClientSI: Provisioning >]`
    - `[<Integrity>]`

Why not language semantics definitions?
Why not generalize them?

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

- A new method of specifying protocols is required.