HELSINKI UNIVERSITY OF TECHNOLOGY
Networking laboratory

Adaptive, policy-based network prototype demo

IRoNet Seminar

17th February 2005

Introduction

- Demonstrating adaptive, policy-based network
- Basic elements
  - Flow analysis and classification + GUI (S. Kaikkonen)
  - Policy Server (J. Huttunen, ALTQ QUI by O. Fialka)
  - ALTQ and Policy control agent (P. Pulkkinen)

Flow analysis and classification

- Problem: how to do the traffic measurement & analysis so that one could differentiate the services to different classes automatically and periodically?
- Solution: traffic is captured from a link using a network tap, analyzed using a flow analyzer and the data is processed using intelligent algorithms

Coralreef

- One of the software packets developed by CAIDA
- Contains device drivers, a libcoral (library), several C- and Perl applications for different uses
- We use `crl_flow` appl. to analyze the flows (actually M. Luoma has added some features to it and we call it `flow`)
- Source can be a trace file or live ntw iface

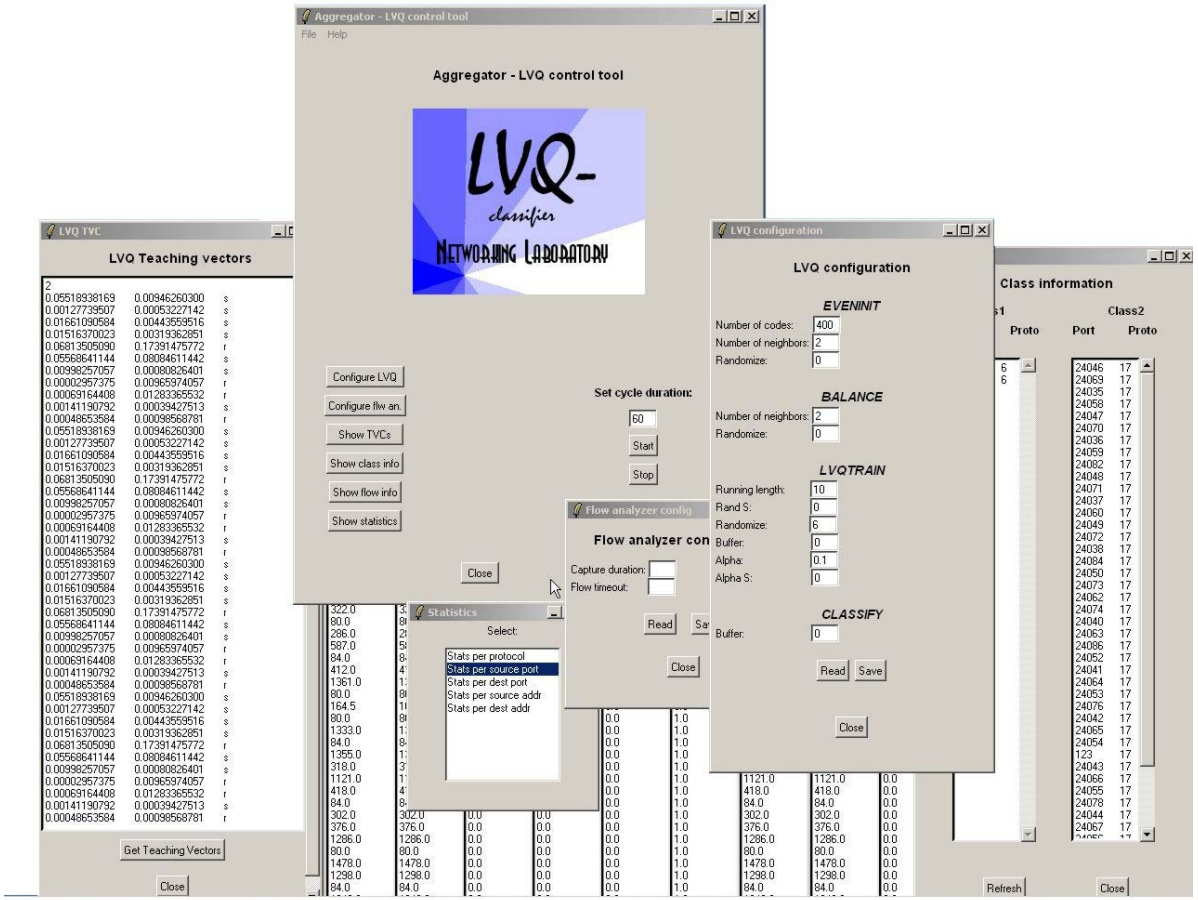
LVQ PAK

- Implementation of the Learning Vector Quantization algorithms
- Developed by Laboratory of Computer and Information Science, Neural Networks Research Centre, HUT
- Usage in traffic classification based on research by Mika Ilvesmäki, Networking Lab.
- We use the following programs of LVQ PAK:
  - eveninit to initialize the codebook vectors
  - balance to equalize the medians of shortest dist.
  - lvqtrain (olvq1 is used as a learning algorithm)
  - classify to do the classification

Our solution

- The abovementioned sw has been merged in the following way:
  - the traffic is directed from the tap to the Policy Server
  - `flow` uses hash tables to store the flow info → these hash tables are accessed to calculate the normalized values of packet sums and flow sums per different source ports / protocols → input for LVQ programs
  - also a lot of other (aggregate) statistical information is calculated based on the flow information

Graphical user interface



Policy Server

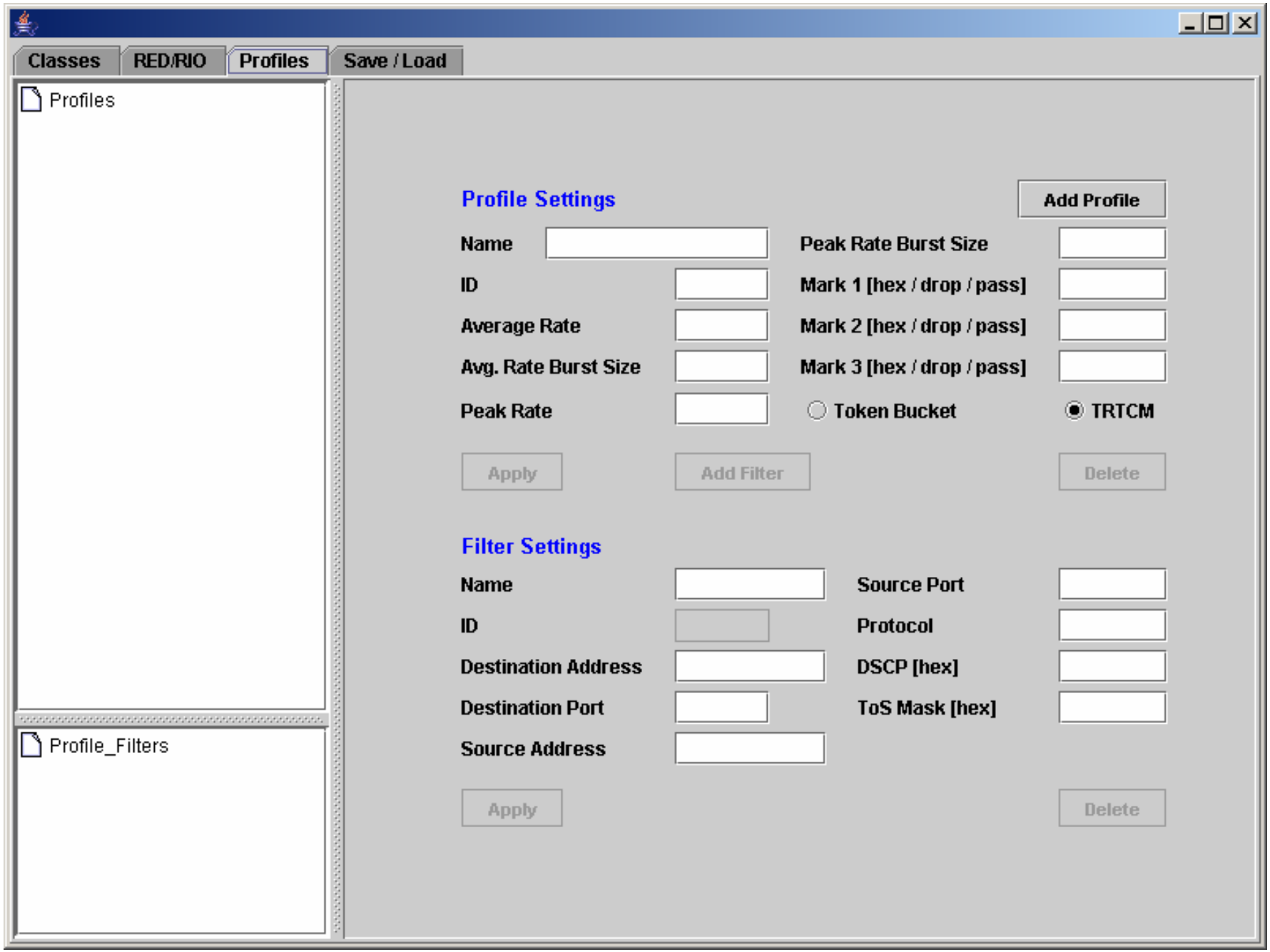
- Centralized MySQL database which stores network policies:
  - Classification rules (filters)
  - SLAs of individual users (profiles)
  - Router configurations

ALTQ

- Traffic management software
- Modifies the FreeBSD kernel
  - QoS support
- DiffServ
  - Traffic conditioning (metering, marking, shaping, dropping) at input interfaces
  - Queue management and scheduling at output interfaces

Policy Server / ALTQ GUI

- Administration tool
- Access to the Policy Server database
- Shows network policies stored in the database
- Enables configuration of the network policies
- Parameters equal to the ALTQ configuration

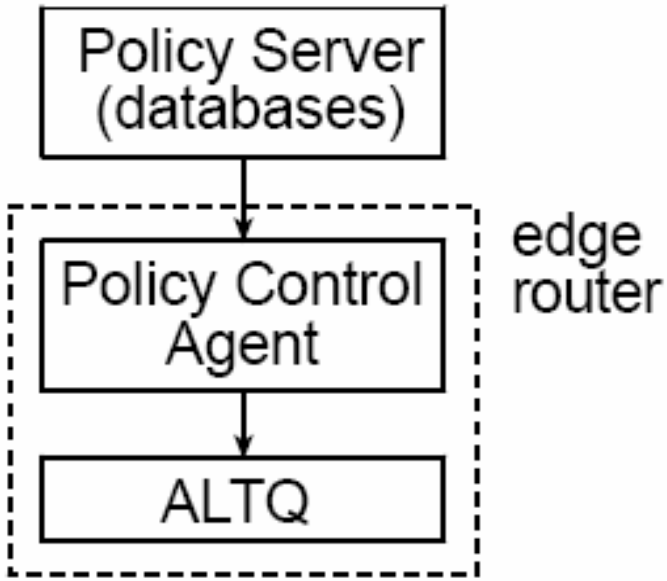

Classes
RED/RIO
Profiles
Save / Load
Profiles
Profile_Filters
Profile Settings
Add Profile
Name
ID
Average Rate
Avg. Rate Burst Size
Peak Rate
Peak Rate Burst Size
Mark 1 [hex / drop / pass]
Mark 2 [hex / drop / pass]
Mark 3 [hex / drop / pass]
Token Bucket
TRTCM
Apply
Add Filter
Delete
Filter Settings
Name
ID
Destination Address
Destination Port
Source Address
Source Port
Protocol
DSCP [hex]
ToS Mask [hex]
Apply
Delete

Policy Control Agent

- Communication module between the Policy Server and ALTQ
- Is located in every edge router
- Enables automatic and dynamic router configuration

Policy Control Agent (cont.)

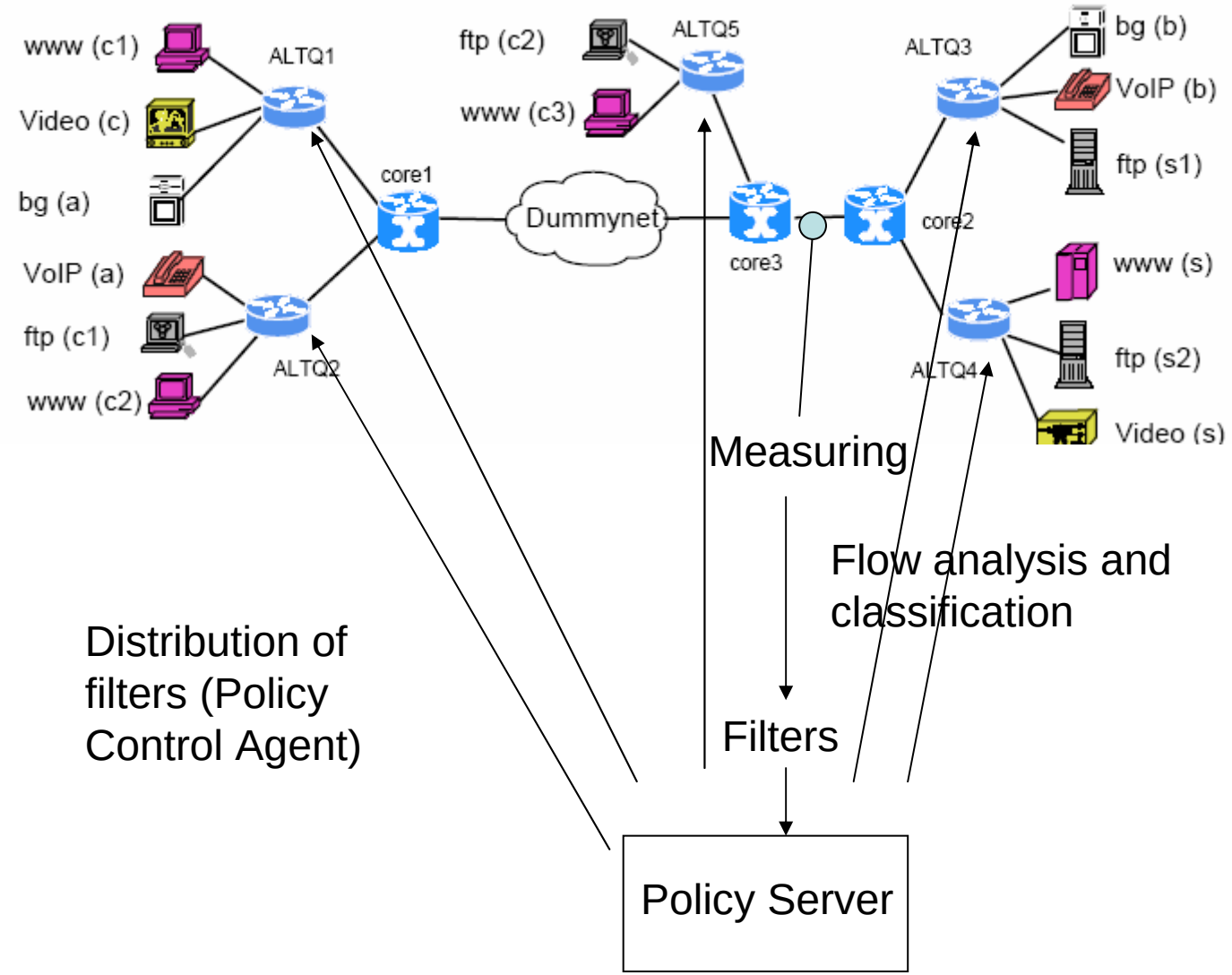
- Configures ALTQ according to the parameters received from the policy server



SIP support

- SIP connections handled separately
- Central SIP proxy with the authentication database
  - modified SER (SIP Express Router)
- When a SIP call is established, SIP proxy forwards appropriate filters to the edge routers of both parties
  - special user profile for SIP calls

Our current prototype



Equipment

- Two prototyping environments:
  - General purpose PC hardware (main prototyping environment)
    - FreeBSD 5.2 with ALTQ
    - AMD 1300 MHz
  - Necsom media switches
    - Distributed platform running embedded Linux on the network interfaces
    - Fast forwarding in the core network

Equipment (cont.)

- Traffic generators and measurement devices
  - SmartBits 600
  - Avalanche / Reflector
  - PC hardware
- Emulated applications:
  - FTP, WWW, Streaming video, VoIP etc.
- We are able to generate various type of traffic and analyze it

Demonstration

- We create following traffic to the network:
  - Streaming video (50 users)
  - WWW (200 users)
  - Background traffic
- Two classes
- Divide traffic into these classes using the classifier

Conclusions

- The initial version of the adaptive policy based network is ready
  - All pieces of the puzzle are implemented and integrated together
- It would be interesting to test the prototype in a real operational network with various users and applications
  - It is very difficult to create realistic traffic mix in an isolated test network