QoS paradigm

• Three concurrent problems:
  – Construction of proper forwarding treatments
    • What kind of service structure best characterises user requirements and operator business
  – Decision of proper forwarding class for the application data stream
    • How network becomes aware of individual data streams and their requirements
  – Engineering the network for committed quality level
    • How network can guarantee service level agreements made for the customers

IRoNet prototype

• Is based on following ideas and assumptions:
  – There is a need for a ‘QoS’ capable network architecture
  – This architecture need NOT to provide hard quality for the users
    • i.e. no signaling is required nor are connections reserved
  – Users do not care actual details of the service which they use rather they are interested in using similar network than today
IRoNet prototype

• Our solution is based on:
  – Division of traffic into a number of classes based on their characteristics
    • Real-time conversations (VoIP etc)
      – Low jitter
    • Short interactive flows (HTTP etc)
      – High throughput
    • Long lasting flows (FTP etc)
      – Low packet loss
    – Provisioning of resources for each class based on the usage and policy
    – Allowing a single user to move within the network

IRoNet prototype

• Our prototype has following components
  – FreeBSD based DiffServ edge routers
    • ALTQ is used to engineer the services
  – NECSOM MediaSwitch DiffServ / MPLS core routers
    • Linux traffic control is used to engineer the services
  – CoralReef measurement probes and network policy generators
  – Policy Server with MySQL database farm

Edge routers

• Standard PC hardware
  – Celeron 433MHz / 128MB
  – D-Link multiport 10/100 Ethernet NICs
  – FreeBSD 4.5 operating system
    • ALTQ-package to add all necessary QoS functions
      – Classification
      – Marking
      – Queueing
      – Scheduling
  • ALTQ
    – Alternate Queueing for BSD
      • http://www.csl.sony.co.jp/person/kjc/kjc/software.html
      – Modifies OS-kernel
      – Software modules reside between network device driver and IP forwarding daemon
Core routers

- NECSOM media switch
  - Based on Frame Synchronized Ring architecture developed in VTT
  - Max 12 10/100 Ethernet interfaces
    - Each carrying own processor
  - Linux operating system
    - Linux traffic control package to add all necessary QoS functions

FreeBSD policy agent

- Basically software that communicates with
  - ALTQ API for manipulating interface dependent parameters
    - Filter lists for different classes
    - Profile values for metering, policing and marking
  - Kernel routing table for installing precalculated routes
  - Policy server for current information about provisioning

Policy Server

- Database server storing
  - Records of individual users service level agreements
    - User is allowed to use
      - 256kbps webtraffic
      - 64kbps VoIP
  - Network device information
    - Routers and their configurations
  - Topology information
    - Physical topology
    - Routes
Policy Server

- Routing and provisioning server
  - QoS routing simulator performing optimization of routes
  - Combination of offline and online operations
    - Real-time collection of network status data (LSA)
    - Non-real-time processing of the information
      » Simulation of best possible implementation of policy
      » Installing the routes into the network database

Policy Server

- Network profile server
  - Processing of measurement data collected inside the network
    - Centralized processing
      » All traffic data is postprocessed inside the server
    - Distributed processing
      » Most of the data is preprocessed within the network

Policy Server

- Network profiles are generated based on the traffic flow observed on the representative points within the network
- Policies (filters) for differentiating the traffic into the classes are generated based on the network profiles

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Flow Analysis</th>
<th>Aggregate Analysis</th>
<th>Clustering</th>
<th>Storing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow Analysis</td>
<td>Reconstruction of the traffic</td>
<td>Statistical manipulation of the flow statistics</td>
<td>Computational intelligence methods to reduce the dimensionality</td>
<td>Decision process for clusters</td>
</tr>
<tr>
<td>Aggregate Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clustering</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Where are we now ...

- So far we have deviced first version of the edge router, policy server and measurement probes
  - Capabilities which they currently have
    - **Edge router**
      - Full user plane operation
      - Communication channel to the dB
    - **Policy server**
      - User dB
        - SLA
        - Authentication
      - Network policy dB
        - Filterlists making the differentiation

Where are we now ...

- **Measurement probe**
  - Line rate capturing capability
    - tested up to 155Mbps b/dir
    - now in process of modifying software for 2.4Gbps b/dir
  - Flow and aggregate statistics processing
Performance data

- Edge router forwarding capability
  - Sensitivity to packet size
  - Sensitivity to added processing

![Graph of forwarding capability vs. packet size with two lines representing Without ALTQ and With ALTQ]

- Edge router forwarding time
  - Sensitivity to packet size
    - Small packets
    - Large packets
  - Sensitivity to added processing
    - Without ALTQ
    - With ALTQ

![Graph of forwarding time vs. frame size with two lines representing Mean and 90% Load with different colors for Without ALTQ and With ALTQ]

Performance data

- Edge router isolation capability
  - No differentiation
  - Four classes
    - Class 1: 4 Mbps
    - Class 2: 2.5 Mbps
    - Class 3: 1.5 Mbps
    - Class 4: 1 Mbps

![Bar chart showing isolation capability with vertical bars for each class]
Performance data

- Edge router isolation capability
  - No differentiation
  - Four classes
    - Class 1: 4 Mbps
    - Class 2: 2.5 Mbps
    - Class 3: 1.5 Mbps
    - Class 4: 1 Mbps

Conclusions

- Prototype is building up piece by piece – however, it takes a while before we have all pieces together
- Next year we will have routing and provisioning server, and core router ready
- New ideas have came up and will be added to the prototype
  - SIP control for adding possibility to signal connections based on time charging

- Questions ???