**S−38.180 Palvelunlaatu Internetissä**

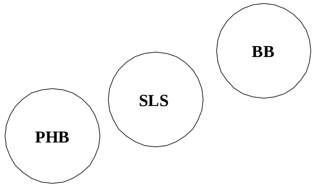
**S−38.180 Quality of Service in Internet**

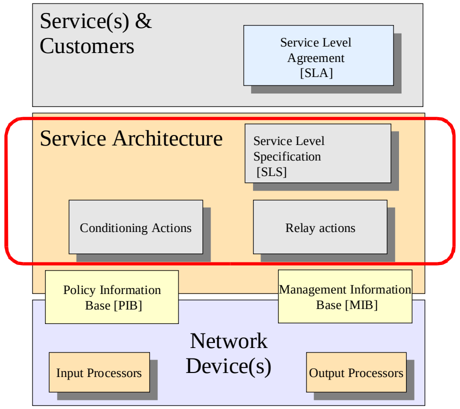
Luento 9: Differentiated Services II

Lecture 9: Differentiated Services II

# Today’s Topic

- This lecture is about Differentiated Services architecture





# Best Effort semantics

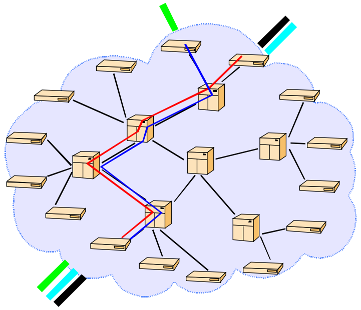
- Best Effort −service
  - All packets are treated equally
    - Forwarding is based on the destination address
    - Packets are queued into single FIFO queue
    - During the time of congestion packets are dropped
      - From the tail of the queue
        - When there is no space in the queue
        - When agerage queue length goes above threshold
  - Access to the network is sold to the customers

# Differentiated Services semantics

- Differentiated Services
  - Packets are differentiated to ***N*** parallel **Best Effort** networks
    - Each parallel network operates like basic Best Effort network with the exeption that there can be priorities and other semantics associated to the service.
  - ’QoS’ based network service is sold to the customer

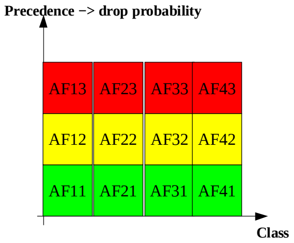
# EF semantics

- ’End−to−end’ service
  - Single domain end−to−end
  - Quality is defined by two constrains:
    - Provisioning
      - Class should be provisioned with enough resources to handle worst case aggregate
    - Sharing
      - No resource reservation for individual flows.
      - Under and overflows possible
      - Timing and delays can not be held or guaranteed



# AF semantics

- No end−to−end semantics
  - Service can be deployed
    - Point−to−point
    - Any−to−any
  - Uncontrollable resource usage inside the network
    - Problem of commons



| | | | |
|---|---|---|---|
| AF13 | AF23 | AF33 | AF43 |
| AF12 | AF22 | AF32 | AF42 |
| AF11 | AF21 | AF31 | AF41 |

# What a customer wants ...

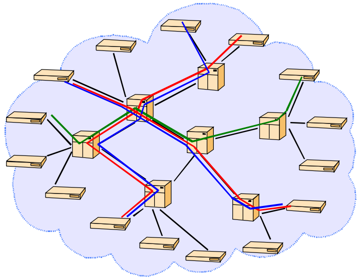
- Lets face the music
  - Customer **is** only interested in the **perceived quality**
    - How things are rolling compared
      - Minute ago
      - Year ago
  - Customer **is not** interested in the novel **technology** which is behind the service
  - This means end−to−end service quality

# End−to−end service

- What prohibits ???
  - Structure of DiffServ is based on local control (policies)
    - Classification based on the policies at the edge of the network
    - Forwarding based on the policies in the core of the network
  - We **can** stretch through single domain (ISP) with EF
  - We **may** stretch through single domain (ISP) with AF
- End−to−end
  - **Is not** within single ISP
  - It **is** between source and destination

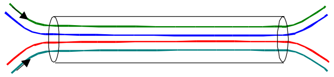
# Let us strech a little bit ...

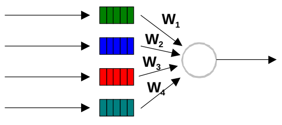
- If we want to have end−to−end semantics to the AF:
  - We need to control resources and offered load hand in hand
    - Load to a single link in some class increases
      - Can we adjust scheduling
      - Do we need to reroute some of the classes
        - » Class and constraint based routing



# Still stretching ...

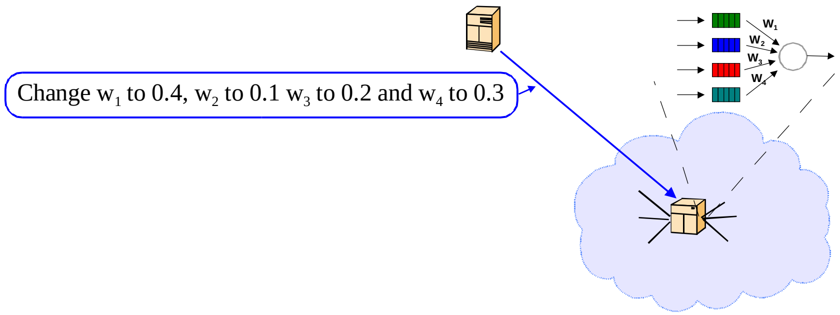
- Lets modify CBQ heuristics:
  - If class green is unsatisfied and class turquoise is unsatisfied but at the scale of the network only class green is unsatisfied we allow only green to borrow.
- Is this possible ?
  - Not with the logic which we have today build inside DiffServ
    - Single router does not know network scale situation
      - No state information associated





# Still a little bit further

- What if we have intelligence (bandwidth broker) outside the network which would control the scheduling of classes



Change $w_1$ to 0.4, $w_2$ to 0.1 $w_3$ to 0.2 and $w_4$ to 0.3

# Bandwidth Broker

- Outside intelligence which controls the network provisioning
  - Makes possible to offer end−to−end semantics
    - **Domain wide**
      - Thats what we just talked about (however there are still some caps in the story)
    - **Inter−domain**
      - We need to
        - » translate domain specific service attributes at the border of two domains (pretty fixed)
        - » Dynamically adjust resource requests to the other domain...

# Inter−domain issues

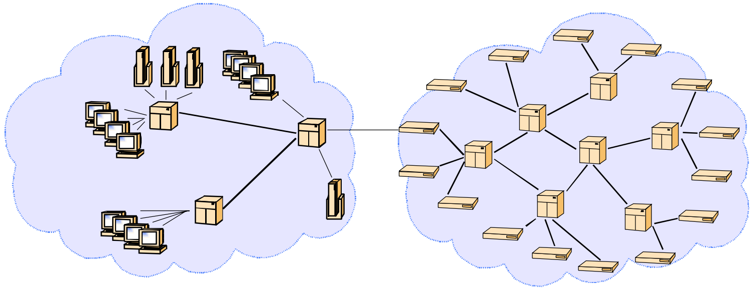
- Inter−domain traffic forwarding is based on bilateral of multilateral peering agreements
  - These tend to be business of lawyers and therefore rather static
  - Our demand is varying rapibly and therefore we need to be dynamic
    - Peering agreements must change to more flexible
      - Rule of thumb: more money −> more lawyers −> more static
      - We need to brake that rule by defining peering more dynamically
        - » One idea: charging should be based on the aggregate traffic in the classes and rate of change requests

# Other issues

- What is potential problem in this scenario:

**Corporate A**
- **100Mbps LAN**
- **2Mbps WAN**

**ISP B**
- **2.4Gbps DiffServ net**

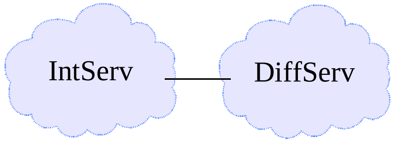


# Other issues

- 2Mbps access link is eaily overloaded when both sides have higher capacities
  - Access link is not DiffServ if ISP does not deliver customer premises equipements.
- Corporate LAN may cause service degradation to the traffic passing out the corporate LAN
  - Solution is to use some mechanism to guarantee that traffic is not degraded inside high speed LAN
    - IntServ

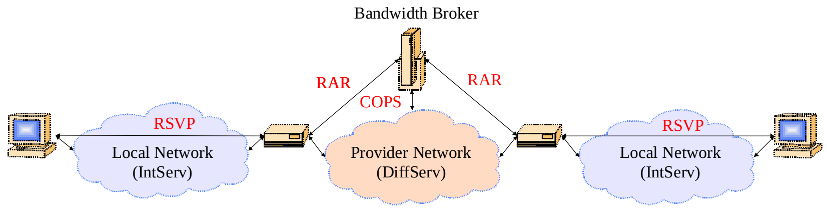
# IntServ / DiffServ co−existence

- We need to be able to pass reservation attributes to and from IntServ cloud.
  - IntServ cloud may be
    - Corporation
      - Outbound / inbound traffic is delivered as guaranteed traffic
        - » Mapping to DiffServ classes based on policy
    - Other ISP having IntServ as backbone
      - Mapping between IntServ and DiffServ classes



# IntServ / DiffServ co−existence

- Bandwidth Broker can be used to do this also
  - Edge router has dual capabilities
    - Passes RSVP messages to the BB to be processed to the domain specifig weight and filter modifications



# Reality check

- Are we rotating things back to IntServ ?
  - BB:s require knowledge from the network (offered load, provisioning)
    - By measuring itself
    - By signaling from the users
  - BB:s modify conditioning and forwarding actions of network routers
- What is the difference to the IntServ ?
  - If we provide end−to−end service we need fixed routes and resources that at the minimum match the requirements
    - We need state information somewhere
      - Centralized − DiffServ BB:s
      - Distributed − IntServ routers

# Reality check

- Is it so that we tend to re−invent the wheel
  - Sometimes it may not be bad thing
  - Sometimes we dare to say it straight to the people

http://www.caspiannetworks .com

# Conclusion

- Differentiated Services is service architecture which allows to build N locically separated Best Effort networks into a single physical network
- Differentiated Services provides tools to offer QoS which is only assured
- Differentiated Services does not provide end−to−end semantics to the services which are build upon it
- End−to−end semantics are only achieved with outside intelligence − like bandwidth brokers