Today’s Topic

- This part of the lecture is about Differentiated Services architecture

Internet today

- Current Internet:
  - ‘Best Effort’ service
    - Equal opportunities (competitive resource sharing)
    - Equal missouries (uncontrolled delays and packet losses)
  - Trend:
    - Internet is becoming commercial network with services leveling the commercial incentives

Best Effort Service

- Ideological background
  - Network is used only with good intent and need
- Turned to battle field
  - As fast and soon as possible
- Customer model
  - Access to the ‘Internet’
    - Possibility to use shared information resources
- Basis
  - Connectionless packet forwarding
Best Effort Router

- Packets are forwarded based on their **destination address**
- Scheduling and queueing
  - FCFS
- Equal treatment

Differentiated Services

- Identification of which parallel best effort network packet is destined, is coded in each packet
  - IPv4 TOS field is reformatted
    - No routing nor precedence
    - Generic class identifier

DiffServ Router

- Packets are forwarded based on the destination address and **class information**
- Scheduling and queueing is done based on the class information
**DiffServ Router**

- DiffServ router has one additional element in datapath compared to basic Best Effort router:
  - Conditioner
- Control plane of a DiffServ router has one extra element ie policy controller, which is responsible of internal management and configuration of conditioner and scheduler

**DiffServ Conditioner**

- Traffic Conditioner is constructed a set of:
  - Classifiers
    - Responsible of logical separation of packet streams
  - Meters
    - Responsible of rate metering of logical streams
  - Markers
    - Responsible of actions based on metering results and predefined thresholds

**DiffServ PHB**

- Per hop behavior is block which contains queue management methods required to implement desired service
  - Queues
  - Queue space management algorithms
  - Schedulers

**DiffServ terminology**

- Workload in DiffServ is divided between two inherently different types of routers
  - Edge routes
  - Core routers
- Edge routers are on the domain edge interfacing
  - Customer
  - Other ISP
DiffServ terminology

- Logical network is concatenation of PHBs which interact together.
- These logical networks have target service called per domain behavior (PDB).
- Target service is loose definition for the goal of the logical network when it is provisioned and configured in a predefined way.
- Edge router chooses PDB for each packet which comes from the customer
  - Marks packet with DSCP of PHB used to implement PDB

DiffServ

- Service decision in edge router can be based on:
  - Metering result
    - Rate based
  - Predefined set of filters
    - IP address is customer
    - TCP/UDP port is application
  - User request
    - Precoded DSCP
    - RSVP signaling

Core routers do nothing but forwarding of packets based on the extra information in DSCP field of packets
- Requires
  - Classifier to detect DSCP fields
  - PHB to implement forwarding behaviors

Service classes

- Differentiated Services is aligned between Best Effort and IntServ
- There is counterpart for each IntServ service class in DiffServ
  - Guaranteed Service ↔ Expedited Service
  - Controlled Load ↔ Assured Forwarding

Expedited Forwarding (EF) [RFC2598]

- Leased line emulation
  - From destined ingress point to destined egress point
  - End-to-end service with
    - Low loss
    - Low latency
    - Low jitter
    - Assured bandwidth
**EF**

- Service commitment is **only** assured
  - Resources inside EF class are shared
    - Amount of other EF traffic influences to the value of delay, jitter and loss
  - Path is freely chosen
    - Delay constraint can not be held as the delay of paths are inherently different
  - No reservation is done
    - **Provisioning is in the key role**

**Leased Line**
- Dedicated resources
  - Full isolation
  - No room for overflow

**Virtual Leased Line**
- Shared resources
  - Partial isolation
    - From other than leased line traffic
  - Can accommodate overflow
    - Vague service guarantee

**Control of service guarantee**
- Access control
  - Rate control
  - User control
- Provisioning
  - At least sum of contracted rates is allocated to EF traffic
  - High priority in the network
    - Scheduled ahead of other traffic
    - Starvation of lower priorities?
      - Only small fraction of total link capacity (10–30%)

**Assured Forwarding (AF)** [RFC2597]

- **Four** independent service classes
  - All packets of a flow are destined to one of the classes
  - No association of service level between the classes
- **Three** precedences in each class
  - Flow can have packets with different precedences
  - Order of packets in all flow is not allowed to change
    - Precedence can not be used to scheduling decissions inside the class

<table>
<thead>
<tr>
<th>Class</th>
<th>Precedence</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF11</td>
<td>AF21 AF31 AF41</td>
</tr>
<tr>
<td>AF12</td>
<td>AF22 AF32 AF42</td>
</tr>
<tr>
<td>AF13</td>
<td>AF23 AF33 AF43</td>
</tr>
<tr>
<td>AF14</td>
<td>AF24 AF34 AF44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Precedence</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF1</td>
<td>AF2 AF3 AF4</td>
</tr>
<tr>
<td>AF12</td>
<td>AF22 AF32 AF42</td>
</tr>
<tr>
<td>AF13</td>
<td>AF23 AF33 AF43</td>
</tr>
<tr>
<td>AF14</td>
<td>AF24 AF34 AF44</td>
</tr>
</tbody>
</table>

**What is the differentiation?**

- **No end-to-end semantics**
  - Service can be deployed as any to any service
    - Like today
  - Uncontrollable resource usage inside the network
    - Very vague QoS
  - Class / precedence in contrast to service guarantee ????
AF

- Class differentiation
  - Associate timing
    • Real-time to Bulk
  - Associate money
    • First class to cattle class
  - Associate user
    • CEO to laundry man
  - Associate protocol
    • TCP / UDP
  - Associate application
    • Clustering of similar application types

- Precede differentiation
  - Associate rate
    • Under/over subscription
  - The rest same as class based except timing can not be used

AF

- Construct services based on previous aspects
  - Many dimensions of freedom
  - How to make sure that system can not be manipulated
    • User control vs Network control

Nasty thoughts

- Can we find justification for DiffServ?
  - No provable service logic
  - No clear structure of service
  - Additional management
  - Let's try it through a chain of thoughts ...

---

<table>
<thead>
<tr>
<th>Best-Effort Service</th>
<th>Differentiated Service</th>
<th>Integrated Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectionless</td>
<td>Connection-oriented</td>
<td></td>
</tr>
<tr>
<td>Aggregated state</td>
<td>End-to-end session state</td>
<td></td>
</tr>
<tr>
<td>Local session state</td>
<td>Session signaling [RSVP]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Admission control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Leaky-bucket traffic control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CoS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Per-class WFQ$^2$</td>
<td></td>
</tr>
</tbody>
</table>

$^1$ Border routers may keep track individual sessions if required by policing or multifield classification

$^2$ Scheduling depends on per hop behavior [PHB]. Minimum requirement is FIFO with multilevel RED.
Goals

- **User standpoint**
  - Get a **good service** (with a extra money get **better service**)

- **Operator standpoint**
  - Get **higher revenues** from the **same infrastructure** than with the best effort service

Quick simulation ...

- How to achieve reasonable gain from differentiation with as little extra management as possible
  - As few classes as possible

- **Lets look some simulations to find out what happens between three different categories of applications**
  - Real–Time UDP (VoIP)
  - Aggressive TCP (HTTP)
  - Friendly TCP (FTP)

Number of classes

- **Best Effort network (1 class) does not have any control over traffic mix**
  - Applications interfere the other

- **DiffServ with two classes (VoIP separated to own class) shows that there is a new division of resources between TCP applications**

- **DiffServ with two classes and 2 priorities reveals the importance of access policing. Rates of the application groups tend to get higher**

- **DiffServ with three classes finally shows the difference between HTTP and FTP**

Number of classes

- **Justification for the DiffServ comes however here**
  - Service provider implementing DiffServ can pack almost 50% traffic to the network that one not using DiffServ
    - Number of packet drops and timeouts is much lower than before
    - Higher revenues from the same infrastructure
Based on previous

- Based on previous
  - Only way the DiffServ brings something new of valuable is that traffic within the network is well engineered i.e. traffic types sharing common buffer needs to be with similar requirements
  - Only way to achieve this is to let the network to do classification and differentiation
    - Users are not, at large, well enough educated to make wise choices for the service classes
    - Or they try to exploit some resource with malicious intent

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 average 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 exception 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

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

Bandwidth Broker

• Outside intelligence which controls the network provisioning
  – Makes possible to offer end-to-end semantics
    • Domain wide
      – That's 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 rapidly 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

• 2Mbps access link is easily 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
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

Conclusion

- Differentiated Services is service architecture which allows to build N logically 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

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