Problems with IntServ

- Scalability: per-flow state, classification, etc.
  - Aggregation/encapsulation techniques can help
  - Can overprovision big links, per-flow ok on small links
  - Scalability can be fixed, but no second chance

- Economic arrangements:
  - Need sophisticated settlements between ISPs
  - Right now, settlements are primitive (barter)

- User charging mechanisms: need QoS pricing
Differentiated Services (DiffServ)

- Some traffic should get better treatment
  - Application requirements: interactive vs. bulk transfer
  - Economic arrangements: first-class versus coach

- What kind of better service could you give?
  - Measured by drops, or delay (and drops)

- How do you know which packets to give better service?
  - Bits in packet header

DiffServ (cont’d)

- Build around the concept of domain
- Domain – a contiguous region of network under the same administrative ownership
- Differentiate between edge and core routers
- Edge routers
  - Perform per aggregate shaping or policing
  - Mark packets with a small number of bits; each bit encoding represents a class (subclass)
- Core routers
  - Process packets based on packet marking
- Far more scalable than Intserv, but provides weaker services
Diffserv Architecture

- **Ingress routers**
  - Police/shape traffic
  - Set Differentiated Service Code Point (DSCP) in Diffserv (DS) field

- **Core routers**
  - Implement Per Hop Behavior (PHB) for each DSCP
  - Process packets based on DSCP

Traffic Limitations

- Can’t give all traffic better service!

- Must limit the amount of traffic that gets better service

- Service Level Agreements (SLA)
  - Source agrees to limit amount of traffic in given class
  - Network agrees to give that traffic “better” service
    - For a price!
  - Economics play an important (fatal?) role in QoS
“Expedited Forwarding”

- Give packet minimal delay and loss service
  - E.g., put EF packets in high priority queue

- To make this a true “absolute” service
  - All SLAs must sum to less than the link speed

Is Delay the Problem?

- With RED, most queues are small

- Packets are dropped when queue starts to grow

- Thus, delays are mostly speed-of-light latency

- Service quality is mostly expressed by drop-rate

- Want to give traffic different levels of dropping
“Assured Forwarding”

- Packets are all serviced in order
  - Makes TCP implementations perform well

- But some packets can be marked as low-drop and others as high-drop
  - Think of it as priority levels for dropping

Example

- 10% premium traffic, 90% ordinary traffic

- Overall drop rate is 5%

- Give premium traffic 0% drops; ordinary traffic a 5.55% drop rate

- Large improvement in service for the small class of traffic without imposing much of a penalty on the other traffic
  - Count on SLAs to control premium traffic
DiffServ “Code Points”

- Use six of the ToS bits in IP packet header

- Define various “code points”
  - Alternative classes of service (drop probability and assured forwarding)

- Each code point defines a desired per-hop behavior
  - Description of the service the packet should get
  - Not a description of the router implementation of that service

Differentiated Service (DS) Field

- DS file reuse the first 6 bits from the former Type of Service (TOS) byte
- The other two bits are proposed to be used by ECN
### Assured Service

[Clark & Wroclawski ‘97]

- Defined in terms of user profile, how much assured traffic is a user allowed to inject into the network

- Network: provides a lower loss rate than best-effort
  - In case of congestion best-effort packets are dropped first

- User: sends no more assured traffic than its profile
  - If it sends more, the excess traffic is converted to best-effort
Assured Service

- Large spatial granularity service
- Theoretically, user profile is defined irrespective of destination
  - All other services we learnt are end-to-end, i.e., we know destination(s) apriori
- This makes service very useful, but hard to provision (why?)

Premium Service

[Jacobson ’97]

- Provides the abstraction of a virtual pipe between an ingress and an egress router
- Network: guarantees that premium packets are not dropped and they experience low delay
- User: does not send more than the size of the pipe
  - If it sends more, excess traffic is delayed, and dropped when buffer overflows
Assumptions

- Assume two bits
  - P-bit denotes premium traffic
  - A-bit denotes assured traffic

- Traffic conditioner (TC) implement
  - Metering
  - Marking
  - Shaping
TC Performing Metering/Marking

- Used to implement Assured Service
- In-profile traffic is marked:
  - A-bit is set in every packet
- Out-of-profile (excess) traffic is unmarked
  - A-bit is cleared (if it was previously set) in every packet; this traffic treated as best-effort

User profile (token bucket)

assured traffic

Metering

r bps

Metering

Set A-bit

Clear A-bit

in-profile traffic

out-of-profile traffic

TC Performing Metering/Marking/Shaping

- Used to implement Premium Service
- In-profile traffic marked:
  - Set P-bit in each packet
- Out-of-profile traffic is delayed, and when buffer overflows it is dropped

User profile (token bucket)

premium traffic

out-of-profile traffic (delayed and dropped)

in-profile traffic
Scheduler

- Employed by both edge and core routers
- For premium service – use strict priority, or weighted fair queuing (WFQ)
- For assured service – use RIO (RED with In and Out)
  - Always drop OUT packets first
    - For OUT measure entire queue
    - For IN measure only in-profile queue

Scheduler Example

- Premium traffic sent at high priority
- Assured and best-effort traffic pass through RIO and then sent at low priority
Control Path

- Each domain is assigned a Bandwidth Broker (BB)
  - Usually, used to perform ingress-egress bandwidth allocation
- BB is responsible to perform admission control in the entire domain
- BB not easy to implement
  - Require complete knowledge about domain
  - Single point of failure, may be performance bottleneck
  - Designing BB still a research problem

Example

- Achieve end-to-end bandwidth guarantee
### Comparison to Best-Effort and Intserv

<table>
<thead>
<tr>
<th></th>
<th>Best-Effort</th>
<th>Diffserv</th>
<th>Intserv</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Service</strong></td>
<td>Connectivity</td>
<td>Per aggregate isolation</td>
<td>Per flow isolation</td>
</tr>
<tr>
<td></td>
<td>No isolation</td>
<td>Per aggregate guarantee</td>
<td>Per flow guarantee</td>
</tr>
<tr>
<td><strong>Service scope</strong></td>
<td>End-to-end</td>
<td>Domain</td>
<td>End-to-end</td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td>No setup</td>
<td>Long term setup</td>
<td>Per flow setup</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>Highly scalable (nodes maintain only routing state)</td>
<td>Scalable (edge routers maintain per aggregate state; core routers per class state)</td>
<td>Not scalable (each router maintains per flow state)</td>
</tr>
</tbody>
</table>

### Summary

- **Diffserv** more scalable than **Intserv**
  - Edge routers maintain per aggregate state
  - Core routers maintain state only for a few traffic classes
- **But, provides weaker services than Intserv**, e.g.,
  - Per aggregate bandwidth guarantees (premium service) vs. per flow bandwidth and delay guarantees
- **BB** is not an entirely solved problem
  - Single point of failure
  - Handle only long term reservations (hours, days)
Discussion: Factors Limiting QoS Deployment

- Prevalence of overprovisioning
  - If all links are only at 40% utilization, why do you need QoS?
  - The inter-ISP links are not over provisioned

- Primitive inter-ISP financial arrangements
  - QoS requires financial incentives to enforce tradeoffs
  - Current peering arrangements are not able to carry these incentives through in a meaningful way
    - Must agree on pricing and service
    - Currently agree on neither!

- End-users not used to pricing/performance options

QoS Debates

- Is overprovisioning enough?
  - If so, is this only because access links are slow?
  - What about Korea, Japan, and other countries with fast access links?
  - Disconnect: ISPs overprovision, users get bad service

- Is differentiated services enough?
  - Can one really deliver reliable service just using relative priorities?
  - Is EF service a viable option?

- It all depends on adaptability of applications