Internet2 Quality of Service

Stanislav Shalunov ⟨shalunov@internet2.edu⟩

Campus-Focused Workshop, Houston, 2002-04-10

Overview

- Past: Premium Service
- Present: Scavenger Service
- Future: More Than One Best-Effort Class of Service
- Total of 36 slides

Past: Premium Service

- Motivation
- Candidate Applications
- Does *your* application really need it?
- Practical Uses
- Transient Deployment Problems
- Supply and Demand Circular Dependency
- Non-Transient Deployment Problems
- Paradigm Shift for Networks
- Economic Considerations
- Other Elevated Priority Services

Premium Service: Motivation

- The Internet is susceptible to denial-of-service attacks and in general will give its users unpredictable service
- Statistical provisioning works fairly well for voice traffic, data networks are harder to predict
- Especially true in the situation when there is no per-bit charge for Internet traffic (as it is the case with Internet2)
- Even individual users can very significantly affect network utilization; statistics won't work unless impact of individuals is small
- Protect at least important traffic from sudden changes in network utilization (which might be due to deliberate malice or to natural usage pattern changes)

Premium Service: Candidate Applications

- Streaming video/audio
- Voice over IP
- Video-conferencing
- Tele-immersion
- Tele-surgery
- High-throughput transfers of important data

Premium Service: Does your application really need it?

- Streaming video/audio: Can buffer as much as you want; why any QoS?
- Voice over IP: codecs can be made tolerant to losses and jitter; what really matters is delay
- Video-conferencing: requirements very similar to VoIP (except for utilization)
- Tele-immersion: fancy video-conferencing
- Tele-surgery: suppose it's your child to be operated by remote control; circuits or packets?
- High-throughput transfers of important data: TCP is not really suitable for running under Premium service; Assured-style forwarding would be better. Also, just how important is the data? See item above.

Premium Service: Practical Uses

- So, *what* are the applications for Premium service?
- Interactive video/audio (voice over IP, video-conferencing, games, tele-immersion) seems somewhat suitable
- Premium service on a well-provisioned network doesn't do much in terms of changes to packet forwarding under normal conditions
- If we were to implement some ideal form of Premium service on Internet2 today, packet loss rates and jitter wouldn't normally change
- Premium service is about *guaranteeing* certain service quality
- In a good network, this quality will often be the same as best-effort quality under normal circumstances

Premium Service: Transient Deployment Problems

- Today's high-speed routers have limited QoS functionality
- Achieving cutting-edge line rates detracts from the features
- Enabling features that are necessary for Premium service deployments can result in packet-per-second rate drop of 50%
- Viable Premium service implementations require hooks in policing and shaping into routing (to be able to do things like "rate-limit DSCP 46 traffic with next-hop AS of X to Y bits/second")
- Without such hooks, NOC work becomes absurdly hard

Premium Service: Transient Deployment Problems (Cont.)

- No high-speed router today really provides these hooks
- Forwarding, as done in line cards, doesn't require real routing information, therefore, to reduce the price of line cards, forwarding tables usually only contain next-interface data
- Including the AS path (or a portion of it), etc., in the forwarding tables would make routers significantly more expensive
- Are we prepared to pay the difference?
- If so, is there enough of a customer base for these feature-rich routers to make their development worthwhile for router vendors?

Premium Service: Supply and Demand Circular Dependency

- Users don't get Premium service today and their applications have learned to adapt to best-effort service
- Users potentially *could* derive benefits from Premium service, but they have no reason to demand guaranteed IP service (as they don't even know whether it's technically feasible)
- Since users don't ask for Premium service, network operators don't have a reason to offer it
- It's expensive to offer this new service; it'll require new routers in many places, new line cards in others, different conduct of network operations everywhere, and you can't start small

Premium Service: Non-Transient Deployment Problems I

- Premium service has to be deployed on a granularity of a diffserv region (in most cases, on a per-network basis)
- No incremental deployment strategy: You can't deploy priority queuing inside the network without policing at *all* edges
- Further, once the service is deployed, how do you verify that it's working properly? Active measurements and network monitoring can only show that it's *not* working correctly
- Since Premium service is really about guaranteeing what should usually be normal service, that's not enough

Premium Service: Non-Transient Deployment Problems II

- If a new interface is deployed for a new connector or peer and policing doesn't work, when would this be detected?
- Not until the particular connector starts injecting more Premium traffic than was agreed
- Even at this point, the failure might not be detected if other users are underusing their capacity; it's only *post factum* that you will learn the reason of a network failure

Premium Service: Non-Transient Deployment Problems III

- The Internet was designed to degrade gracefully
- The upside is that in an event of something in the network not working quite correctly the application, as percieved by the user, has good chances of continuing to operate
- The downside is that users usually don't notice network failures until they're catastrophic
- The problem is exaggerated by finger-pointing and blame-shifting at network peering points

Premium Service: Non-Transient Deployment Problems IV

- In a world of guaranteed services, the applications will either rely on the guarantees provided by the network, or they will continue to include code to adapt
- In the latter case, graceful deterioration until catastrophic failure can happen
- In the former case, applications lose advantage of being able to work over "normal" best-effort networks

Premium Service: Non-Transient Deployment Problems V

- Further, these difficulties of verification of guarantee described above are from a network operator's perspective
- For a user it's much harder to determine whether the network operator is indeed providing guarantees or merely has a well-provisioned network that at the moment is working fine but could be brought down by a denial-of-service attack
- Therefore, stiff economic penalties (much worse than "your money back") would be required for failures of services

Premium Service: Paradigm Shift for Networks

- An ISP's technical interface to the world is a combination of unicast IPv4 and BGP
- Economic interface is per-line and/or per-bit charges
- A typical SLA could negotiate a few parameters, but the number of them would be very limited
- In a QoS world, dramatic changes to both the technical and to the economic interface are required
- On the technical side, either complex prior agreements about reservations or some additional (with respect to IPv4 and BGP) dynamic signaling and reservations mechanism would be required
- This alone would suddenly make the Internet much more complex than it is today
- We're going to discuss the ecomonic side now

Premium Service: Economic Considerations I

- Traditionally identified problems: billing; settling of disputes; demands for customer service; explaining the new service model to confused users; increased litigation
- New to this talk: Why would a network provide high-quality best-effort service for transit traffic in a QoS world?
- To answer this question, we could ask it about today's world
- Today, it's hard for a network to differentiate between traffic from direct customers and traffic from peers
- If QoS mechanisms that allow routers to look at the AS path of packets in transit are deployed, that reason goes away

Premium Service: Economic Considerations II

- Today, providing poor quality best-effort service for transit traffic today can help conserve resources, but won't translate into immediate monetary revenues
- In a Premium service world, making a customer who otherwise doesn't pay you directly switch to Premium service and pay seems too obvious a trick not to play
- Providers could start treating "somebody else's" traffic worse than "their own"
- Erosion of best-effort service will lead to a completely different world where all serious work gets done over Premium service and users are generally expected to make virtual circuit reservations for most of what they do
- Do we want to go there?
- Do we want to *supplement* best-effort service or to *replace* it?

Other Elevated Priority Services

- All of the points given above would be equally applicable to assured-forwarding-based services and, in fact, to most elevated priority services
- If one were to overcome these difficulties, going after only insignificantly more complex goal of deploying Premium service would seem natural
- While we only discuss Premium service, the conclusions are believed to be valid for any elevated priority IP network service

Questions about Premium?

Present: Scavenger Service

- Definition
- Who could use this service?
- Why would anybody possibly want to mark their precious traffic for degraded service?
- Current State
- Success Metrics
- Usage
- Network operator's perspective
- Power user's perspective
- Resources

Scavenger Service: Definition (Informal)

- Additional class of best-effort service
- Overall goal: let one saturate the link without affecting "normal" traffic
- Parallel virtual network with very scarce capacity—but elastic: can expand into what the default best-effort class doesn't use
- Special global DSCP value: binary 001000
- Formal definition can be found at `http://qbone.internet2.edu/qbss/qbss-definition.txt`

Scavenger Service: Who could use this service?

- Bulk data transfers (lasting for hours and days), network backups, stable content (such as software) distribution and mirroring
- New kinds of distributed applications that can take advantage of idle network capacity in the same way distributed computations take advantage of idle CPU cycles
- QBSS can be used to implement local policy (but beware of TCP/UDP-port-number-based marking)

Scavenger Service: Why would anybody possibly want to mark their precious traffic for degraded service?

- Self-policing users exist today—no incentive necessary for them
- If network charges per bit, can charge less for QBSS
- Administrative policies and social pressure
- Non-voluntary marking: everything out of a particular port

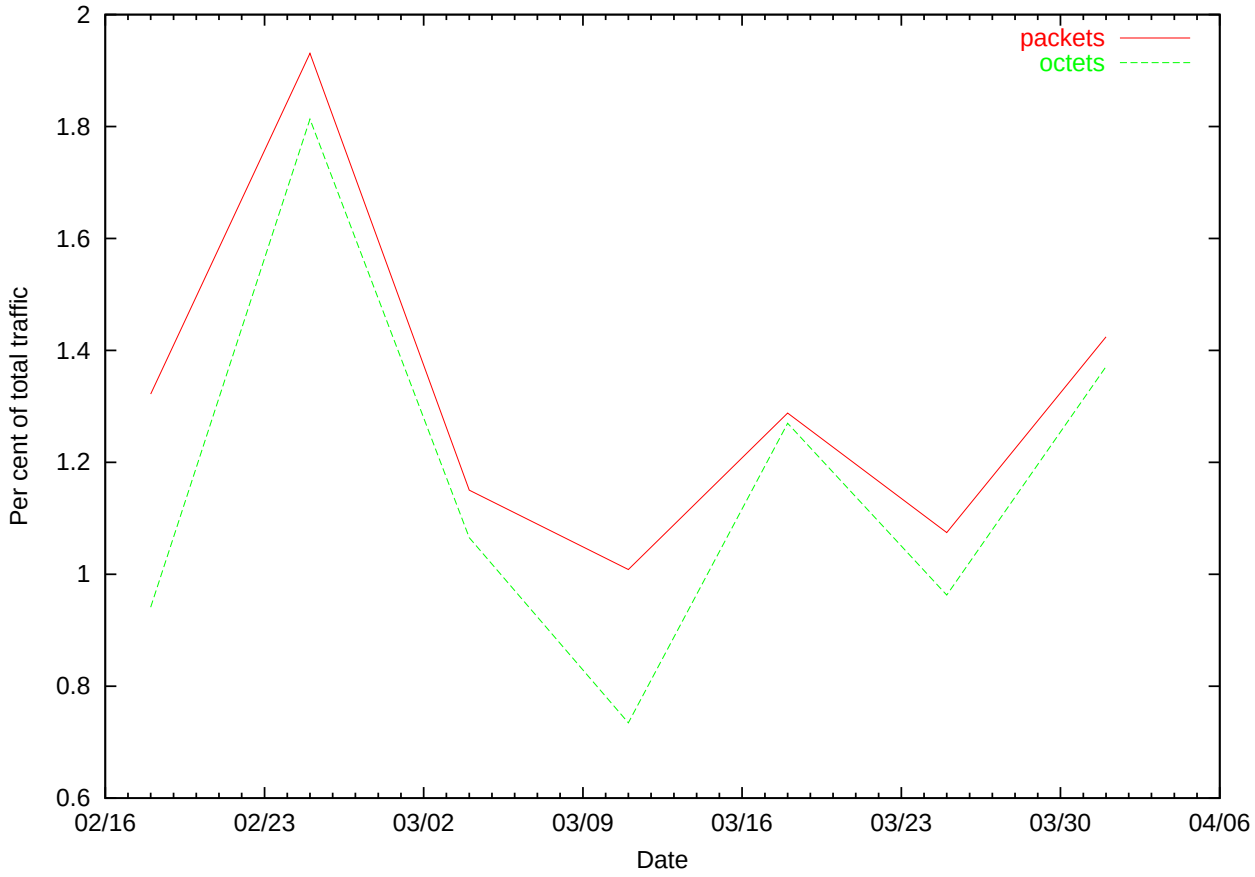
Scavenger Service: Current State

- It's one-year-old and it can walk
- Service announced during Internet2 Spring member meeting in February 2001
- Service definition produced by QoS WG approved as "Internet2 Technical Report, Proposed Service Definition"
- Interest expressed by people who need to move around large amounts of data (SLAC, TransPAC, CERN, UKERNA)
- Gear tested (Cisco 7200, 7500, GSR; Juniper)
- Some operational traction

Scavenger Service: Success Metrics

1. Rough measure of initial acceptance: $\geq 1\%$ QBSS on Abilene (we're there)
2. Evidence of scavenger service being used on bottleneck links to solve bulk data transfer needs without affecting best-effort traffic (getting close)
3. Close to 100% utilization on some circuits without adverse effects on best-effort traffic (the Holy Grail of scavenger)

Scavenger Service: Usage



Scavenger Service: Network operator's perspective

- Gain:
  - Extend life of uncongested best-effort service within your network
  - Get a negotiating tool for price of packet delivery outside of your network
- Do:
  - Pass QBSS codepoint unchanged;
  - Configure QBSS queue (start with your bottlenecks—this might be your *commodity* connection, too)
  - Monitor your network for QBSS traffic
  - Suggest use of QBSS to power users

Scavenger Service: Power user's perspective

- Gain:
  - Ability to self-police easily
  - Get everything from a network nobody else wants
- Do: Mark your bulk data transfers for QBSS

```
int qbss = 0x20;
setsockopt(sock, IPPROTO_IP, IP_TOS,
           (char*) &qbss, sizeof qbss);
```

Scavenger Service: Design Team

`http://qbone.internet2.edu/qbss/`

- Configuration examples
- Testing results
- Mailing list archive
- Milestones
- Service definition document

Questions about scavenger?

Future: More Than One Best-Effort Class of Service

- Low Delay—the Real Need
- Building a Low-Delay Self-Regulating Service
- Possible Implementation
- Provisioning

Future: Low Delay—the Real Need

- Capacity compensates for loss (error-correcting codes)
- Delay compensates for jitter and reordering (buffering)
- Important network characteristics: capacity and delay
- Capacity can be provisioned
- Nothing can buy you lower delay
- Interactive applications such as VoIP can be made to work fine despite loss, jitter, and reordering (with ECCs and buffering) if adequate capacity is provisioned and *delay is low*

Future: Building a Low-Delay Self-Regulating Service

- Without a loss of generality, forget about scavenger
- Our network will have two different best-effort classes of service: low-loss and low-delay
- They would be serviced in one of two ways: either a line card doesn't pay attention to DSCP, or it uses WFQ/WRR to give classes pre-defined (by the operator) fractions of overall capacity
- The low-delay class has a hard and low limit on queue length
- Configure the low-delay queue on every interface that could be congested

Future: Building a Low-Delay Self-Regulating Service (Cont.)

- Result: the low-delay class always provides *guaranteed* low delay; due to lack of accomodation of burstiness TCP performance will be bad in low-delay class
- The low-loss class is equivalent to what we have today

Future: Possible Implementation

- Pick two otherwise unused (in particular, non-zero and non-scavenger) DSCP values: one for the low-loss class, one for the low-delay class; conservatively treat DSCP 0 as low-loss
- Close to the edges, re-mark the traffic as follows:
  - If DSCP $\neq 0$, do not change DSCP
  - If DSCP $= 0$, set DSCP according to local heuristics
- Note: several re-marking routers on the path is OK because as soon as one of them sets DSCP, the rest leave its value alone

Future: Possible Implementation (Cont.)

- Examples of local heuristics:
  - TCP or DNS or multicast or ICMP $\Rightarrow$ low-loss; otherwise $\Rightarrow$ low-delay
  - Traffic to or from local VoIP gateway $\Rightarrow$ low-delay; otherwise $\Rightarrow$ low-loss
  - TCP or DNS or ICMP or multicast $\Rightarrow$ low-loss; traffic to or from local VoIP gateway $\Rightarrow$ low-delay; otherwise $\Rightarrow$ leave DSCP $= 0$
- Lets the network guess and allows the users to override
- If a user puts TCP traffic into the low-loss class, he hurts himself

Future: Provisioning

- How do you choose how much to give to the low-delay class?
- Provisioning in general is a hard problem
- Most developed aspect of network provisioning is voice traffic
- We have an advantage: if we overprovision, we aren't wasting money as would be the case with a circuit-based network

Questions?

- Stanislav Shalunov ⟨shalunov@internet2.edu⟩
- QBone Web Site: `http://qbone.internet2.edu/`