Fundamental Design Issues for Future Internet

Scott Shenker

Try to answer

- Should Internet adopt a new service model?
- How this service model should be invoked?
- Should the service model include admission control?
- Provide a *"framework"* for discussing the various architectural alternatives.

The current Internet

- The Internet offers a single class of "best-effort" service.
  - no admission control
  - no assurance of when, or even if, packets will be delivered
- Current applications (e.g., telnet, ftp, SMTP) are *"elastic"* applications. Elastic in the sense that they can adopt and reduce their transmission rate if there is congestion.
- Emerging real-time multimedia applications are *"non-elastic"*, problems:
  - High variable delay in the current Internet implies high packet lost rate.
  - Applications do not *back off* in the presence of congestion.

Implications:

- grap resources from traditional applications, and
- may have poor performance.

- Partial solutions:
  - Fair Queueing in routers: we can *protect* flow but requires change in the Internet architecture.
  - Adaptive real-time applications: make application adaptive to delay, e.g., buffering.
- Still, problems remain:
  - No class differentiation
  - No assurance of QoS
  - Limitation of adaptability

What is the goal of network design?

- Users-centric rather than network-centric
- Measure performance *not* in utilization, throughput,...etc. But in degree in satisfying user's requirement.
- Define:
  - vector s_i: describe the service delivered to the i^{th} application or user, e.g., delay, throughput, packet drop, ...etc.
  - U_i: utility function of the i^{th} application.
- **Network Design Goal**: to maximize the *"efficacy"* (V):

$$V = \sum_i U_i(s_i)$$

Why we need to extend service model?

Ways to maximize V:

1. Buy more resources (e.g., bandwidth)
2. Fix the amount of resource, introduce *wider variety of services.* (**Illustration**)

Intuition: closer the services aligned with the application needs, the higher is the efficacy.

Model

Compare a *single class* FCFS with *two priority classes* FCFS.

For an M/M/1 system, arrival rate is λ, service rate is μ, average response time T is

$$T = \frac{1/\mu}{1 - \lambda/\mu}$$

Let there be two classes with $\lambda_1 = \lambda_2 = 0.25$. $U_1 = 4 - 2T_1$; $U_2 = 4 - T_2$. So class 1 "cares" more about delay. For *best effort* (or no class differentiation), we have:

$$T_1 = T_2 = \frac{1}{1 - 0.5} = 2$$

Therefore:

$$V^{FCFS} = U_1 + U_2 = 2$$

Consider a strict priority, preemptive system (note: *the computation in the paper is wrong*). We have:

$$T_1 = \frac{1}{1-0.25} = 4/3 \quad ; \quad U_1 = 4/3$$
$$T_2 = 13/6 \quad ; \quad U_2 = 11/6$$

So $V^{priority} = U_1 + U_2 = 19/6$.

Conclusion:

- without changing the capacity of the server, we have a *better* efficacy.
- We have to set $\mu = 65/58$ (increase the bandwidth) to have the same efficacy.

Engineering/economic argument: sometimes it is *easier* to simply add more bandwidth.

Other important comment:

- Time scales of the QoS requirements of real-time applications are very small, we can't operate a network with reasonable utilization and provide the QoS.
- Not every client gains directly from the increase of efficacy (e.g., client 2 above). Intuition: we shift the resources from applications that are less performance sensitive to those that are.
- To improve efficacy, we need to **MAP** applications to service classes. For example, flow 1 to class 1 and flow 2 to class 2. This implies we need **differentiation, marking**, and **heterogeneous treatment** of classes.

WHO chooses the service for a flow?

Two options:

1. the flow can pick the service
2. the network can pick the service (IMPLICIT)

What are their tradeoffs?

Let the Network decides

Network divides into categories (e.g., asynchronous bulks, interactive bulk, real-time,...etc).

Advantages are:

- Applications do not have to be changed.
- Mapping from applications to services need not be uniform across all routers.
- immediately deployable and then modified incrementally.

Disadvantages are:

- Entail a fixed set of application classes and it may not be able to properly service some applications
- Cannot accommodate individual or situational variations within a single application. For example, same video software for different applications.
- May need to over-allocation of some resources.

Let application decides

The disadvantages of this "implicit" approach are:

- **Incentive:** network needs to provide incentives so users can choose the *proper* classes. E.g., low QoS requirement to lower class. This implies **PRICING**.

  Pricing implies the we need to consider *authentication, accounting, billing,...etc.* Which are NOT available in the current Internet architecture.

  Deployment issues.

- **Stability:** the offered service model must be stable, though extensible.

  This implies the advertised service model must be uniform across all routers.

  Deployment issues.

Conclusion: author suggests to use the *"explicit"* model.

Do we need admission control (AC)?

Some new service models may require resource reservation so to have a QoS guarantee. Whether we need AC depends on which choice maximizes the efficacy.

Admission control is to prevent overloading. One can prevent overloading by *"over-provisioning"*. But what is overloading?

Overloading: a network is overloaded if the value of V goes up when one remove a flow.

- Consider an unlimited set of homogeneous users sharing a link with bandwidth B. We have:

$$V(n) = nU(\frac{B}{n})$$

- If $V(n)$ is always increasing, we don't need AC.
- If $V(n)$ is maximized at some finite $\hat{n}$, then we need to decide whether we need to provide AC or over-provisioning.

  Whether there is overloading depends on the structure of $U(x)$!

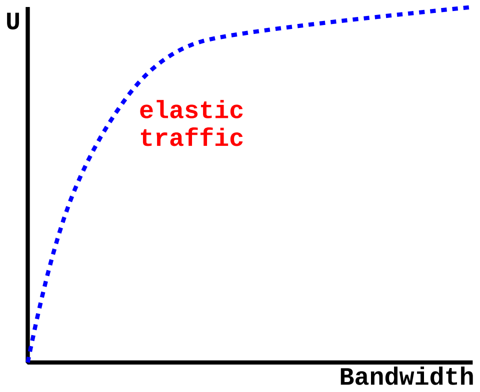
- Interesting observation:
  - If $U(x)$ is non-decreasing and convex but not concave in $[0, \epsilon)$, there exits some $\hat{n}$ such that $V(\hat{n}) > V(n)$ for $n > \hat{n}$.
  - if $U(x)$ is everywhere strictly concave, then $V(n)$ is strictly monotonically increasing function of n. Network is never overloaded.
- If $U(x) = x^p$, $V(n) = nU(B/n) = \frac{B^p}{n^{p-1}}$.
- If $p > 1$, $n = 1$ maximized $V(n)$. If $p < 1$, $V(n)$ is maximized at $n = \infty$.

So overloading depends on the shape of the utility function!

Utility Functions

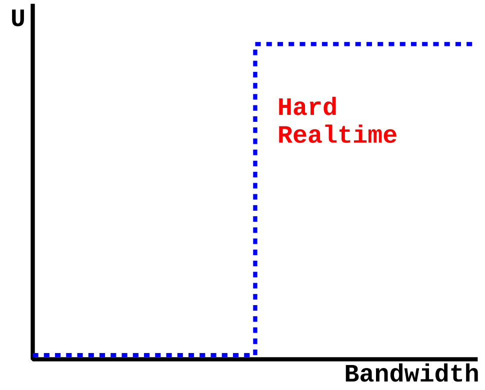
What do utility functions look like?

- Traditional applications like Ftp, emails, telnet:



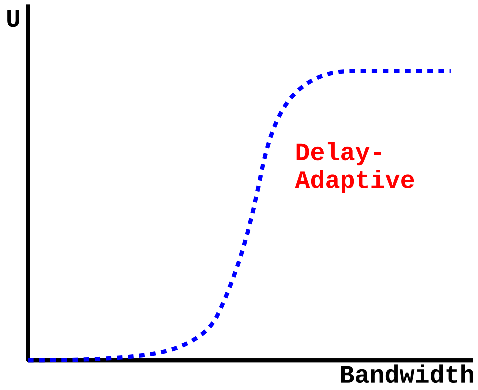
  - They have *decreasing marginal degradation* due to incremental increases in delays.
  - It is concave everywhere.
  - V is maximized at $n = \infty$. NO AC!!

- real-time applications with hard deadline.



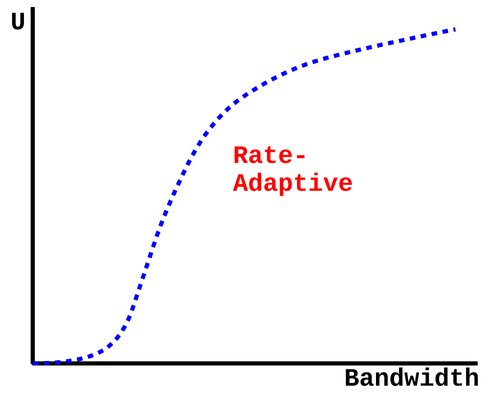
  - if bandwidth drops below the requirement, performance is zero.

- Delay sensitive and adaptive applications (e.g., with delay deadlines or bounds)



  - many video/audio applications have to be designed to be adaptive.
  - utility is convex but not concave in a neighborhood around zero.
  - network can be overloaded, point of overloading is function of the shape of U.

- Rate adaptive applications



  - applications use different rate at different available bandwidth.
  - convex but not concave around zero.
  - network overloading point is smaller than delay-sensitive. (because that is the minimal requirement).

Question: should one build a network that has admission control or by over-provisioning to avoid overloading?

Over-provisioning

Can we simply use over-provisioning to make overloading a *rare* event ?

- In short-term (with next few years), the growth of LAN's bandwidth and high bandwidth application will out weight the access link. We **can't** accomplish over-provisioning in near term.
- For longer term, the main issue is whether people are willing to pay for the cost of over-provisioning.
- Two types of users, *"normal"* and "leading edge users. The normal users only need bandwidth of less than 1 mbps while leading edge users, though their population is small, require *large bandwidth*.

- We only need small increase of bandwidth to satisfy normal users. But for large user, we need to increase bandwidth significantly.
- normal user is willing to pay for their share, but leading edge users will not because it is more expensive.
- In conclusion, author believes that over-provisioning will not be cost-effective and there is a need for AC.