Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.892 Fall 1999 **Debating the Future Internet Service Model** November 8, 1999

Overview. Best-effort vs. reservations. Uniform vs. priority drop in the context of layered video.

1 Best-effort vs. multiple service classes

1.1 Motivation

- The Internet today offers a single class of "best-effort" service. No assurance of when or if packets will be delivered, no admission control.
- But the Internet isn't just ftp, email, and telnet any more.
- The above applications are *elastic* applications. Q: what does this term mean? (A: they can derive additional "utility" from every marginal increase in bandwidth starting from 0.)
- Starting to see more and more real-time streaming applications such as voice and video.
- Claim: these apps are not as elastic or adaptive. Delay variations and losses pose severe problems. And they don't *typically* react to congestion.

  This is a questionable assumption, but seems to be true to some extent for real-time Internet telephony.
- **Question: Should we change the underlying Internet architecture to better accomodate these applications?**
- I.e., should the fundamental best-effort Internet architecture be changed to accomodate multiple service classes? And should the architecture be changed to perform *admission control*?
- Goal of paper: To provide a concrete framework in which this debate can rage on.
- At the highest level, the "right" answers depend on the nature of future network applications, the cost of bandwidth, the cost of more complex mechanisms, etc. Hard to quantify!

1.2 What should network architects optimize?

- Throughput? Utilization? Packet drops? Delays?
- Claim: No! Really want to optimize "user happiness." More precisely, if s_i is the network service (in terms of throughput, drops, etc.) delivered to user i, there is a *utility function* $U_i(s_i)$ which measures the performance of the application and the degree of user satisfaction. The goal of network design is to maximize $V = \sum_i U(s_i)$.
- But what about other optimization criteria? For example, a service provider might choose to optimize revenue.

- Anyway, this formulation makes it apparent that it's worth rethinking the basic service model the architecture provides. Simple examples show that pure FIFO without favor toward particular flows doesn't always optimize V.
- But should we add multiple service classes, or just add more bandwidth?
- Note: while the goal is to optimize V, it won't necessarily optimize each of the $U_i's$. I.e., the nature of this optimization is to take bandwidth away from the elastic applications and satisfy the more sensitive, inelastic applications.

1.3 How is service for a flow chosen?

- Option #1: Implicitly supplied
  - Here, network automatically chooses service class based on packet/flow.
  - Advantage: No changes to end-hosts or applications.
  - Disadvantage: New applications cause problems, since routers don't know what to do about them.
  - In general, this approach embeds application information in the IP layer, which has deficiencies.
  - Also suffers from stability problems, because of a changing service model, changing unknown to applications.
- Option #2: Explicitly requested
  - Here, applications explicitly ask for particular levels of service.
  - Problem: incentives. Why will some applications volunteer to ask for lower levels of service?
  - Possible solution: pricing.
  - Social implications of usage-based pricing. This will discourage the browsing mentality of users (which information providers find attractive).
  - Key point: even in a single best-effort class, the notion of incentives is important. Addresses the issue of "whether to send data."
  - Usage-based pricing at a higher granularity than an individual user might be more appropriate.
  - In the explicit model, the network service model is known to the application.
- Link-sharing as an implicit model. Works well when dealing with aggregates of flows.

1.4 Do we need admission control?

- Some services might need explicit resource reservation, so the network may need to turn away flows, which if admitted, will violate its current quantitative commitments.
- One approach to answering this question: If there are values of population size n and n' such that $V(n) > V(n')$ for $n < n'$, then $V()$ has a local maximum, and it's better to turn flows away.

- Question amounts to monotonicity of $V = \sum_i U(s_i)$.
- What is the shape of $U(s_i)$ for various classes of applications?
  1. Elastic applications: concave. For this, $V()$ is monotonically increasing with n.
  2. Hard real-time apps: step function. Clearly admission control is important here (e.g., telephone networks).
  3. Delay-adaptive real-time apps: convex first then concave after inflection point.
  4. Rate-adaptive real-time apps: earlier inflection point than above case.
- If U is convex in some region, then $V()$ has a local maximum. This argues for admission control.
- Q: What does this argument miss? Dissatisfaction derived from denied flows!
- Overprovisioning as an alternative: how much you have to overprovision depends on variance of bandwidth use. Furthermore, while backbone bandwidths will continue to improve dramatically with time, it's unclear that the same trend is true for last-mile access links.
- Need to overprovision even if you have admission control!
- Shenker concludes that admission control is more cost-effective than the amount of overprovisioning you otherwise have to do.
- Bottom line: Paper sets wonderful framework for this debate, which is still very open. Next week, we will look at integrated and differentiated services and RSVP.

2 Uniform vs. priority drop for layered video

- Suppose packet drops, rather than delays, are the primary determinant of application performance.
- Should the Internet service model be changed to incorporate priority drops for such applications (e.g., layered video)?
- Paper analyzes this question using application performance and incentive properties. Paper doesn't actually answer this question, but makes some very interesting observations.

2.1 Old Dogma

- The nature of layered video is to send more important packets from the lower layers in preference to higher layers.
- It is therefore completely obvious that priority drop gateways in the network make this easy to do, saving the application from complex end-to-end adaptation algorithms. End of argument.

2.2 New Dogma

- Formulated more recently, best articulated by MJV96.
- Priority drop gateways provide no disincentive against sources sending large amounts of traffic to the receivers, since the quality vs. bandwidth graph doesn't dip after the rate exceeds the bottleneck rate.
- Uniform drop gateways provide a tremendous incentive for "good behavior," since overload causes packet drops across all layers.
- And, if you can find an "optimal" adaptation algorithm to track the ideal quality, then this is the best approach.
- RLM is indeed optimal under some conditions, if the network isn't particularly bursty.
- Uniform drop + RLM (or other such algorithms) is the right approach. Priority-drops are fundamentally evil! No incentives. Interferes with RLM scalability. End of argument.

2.3 BSS intuition (their dogma)

- Traffic is actually quite bursty, and along multiple time scales. When traffic is bursty, priority dropping should in fact greatly outperform uniform+RLM.
- RLM won't do a great job of tracking available bandwidth when you have a network with bursty cross-traffic (and the source is itself bursty).
- Incentive issues: individual optimal points (Nash equilibria) don't necessarily lead to social optimal outcomes. But even so, uniform drops should have better incentive properties.
- Three questions:
  1. Do uniform and priority dropping achieve same optimal performance?
  2. Does RLM achieve optimal performance when traffic is bursty?
  3. What incentives do these dropping mechanisms present apps?
- Some simpyfying assumptions about source traffic model, rate spacing between layers, and per-bit value. Both RED and drop-tail for uniform drop simulations.
- Simple, but interesing source model to capture burstiness. In each time interval t, select rate n_t: $n_t = 1$ with prob. $(P-1)/P$ and $n_t = PA + 1 - P$ with prob. $1/P$. Average is A packets per second, and you can easily show that the standard deviation is an increasing function of $P (\geq 1)$. Assume perfect correlation across layers.

2.4 "The Truth"

- With burstiness, priority dropping does outperform uniform, but not by too much. In fact, they show that the ratio of maximal (best case) utilities for the two cases is at most $e/2$, about a 36% improvement at most for priority over uniform.

- RLM doesn't in fact do a particularly great job of tracking bottleneck bandwidth in the presence of burstiness. But in terms of utilities, this isn't a big problem because even if RLM is off by a factor of 2, it's only one layer! (Still, in some settings priority outperformed RLM by between 50% to 100%, but usually by a lot less in most other situations).
- Incentive issues: even minimal amounts of nonperformance (dis)incentives rectify the poor incentive properties of priority drop gateways! The point is that these disincentives only need to be small to get this. Also, Nash equilibria not socially optimal with droptail gateways. Can be overcome with FQ.
- Bottom line: wonderful intellectual exercise in analyzing a rather hard problem. While some assumptions are necessarily simplifying, the paper does provide preliminary answers to some important questions, raising many others in the process.
- Perhaps the most important conclusion is to never trust one's intuition, especially about answers, in complex systems! *It is therefore important to use your intuition to ask the right questions, more than to answer them!*