Traffic Management
&
Traffic Engineering

An example

- Executives participating in a worldwide videoconference
- Proceedings are videotaped and stored in an archive
- Edited and placed on a Web site
- Accessed later by others
- During conference
  - Sends email to an assistant
  - Breaks off to answer a voice call

What this requires

- For video
  - *sustained bandwidth of at least 64 kbps*
  - *low loss rate*
- For voice
  - *sustained bandwidth of at least 8 kbps*
  - *low loss rate*
- For interactive communication
  - *low delay (< 100 ms one-way)*
- For playback
  - *low delay jitter*
- For email and archiving
  - *reliable bulk transport*

What if…

- A million executives were simultaneously accessing the network?
  - What *capacity* should each trunk have?
  - How should packets be *routed*? (Can we spread load over alternate paths?)
  - How can different traffic types get different *services* from the network?
  - How should each endpoint *regulate* its load?
  - How should we *price* the network?

- These types of questions lie at the heart of network design and operation, and form the basis for **traffic management.**

Traffic management

- Set of policies and mechanisms that allow a network to *efficiently* satisfy a *diverse* range of service requests
  - The mechanisms and policies have to be deployed at both node level as well as network level
- Tension is between **diversity** and **efficiency**
- Traffic management is necessary for providing *Quality of Service (QoS)*
  - Subsumes congestion control (congestion == loss of efficiency)

Traffic Engineering

- Engineering of a given network so that the underlying network can support the services with requested quality
- Encompasses
  - Network Design
    - Capacity Design (How many nodes, where)
    - Link Dimensioning (How many links, what capacity)
    - Path Provisioning (How much bandwidth end-to-end)
    - Multi-homing (Reliability for customer)
    - Protection for Reliability (Reliability in Network)
  - Resource Allocation
  - Congestion Control
    - routing around failures
    - adding more capacity

Why is it important?

- One of the most challenging open problems in networking
- Commercially important
  - AOL ‘burnout’
  - Perceived reliability (necessary for infrastructure)
  - Capacity sizing directly affects the bottom line
- At the heart of the next generation of data networks
- Traffic management = Connectivity + Quality of Service

Outline

- Economic principles
- Traffic classes
- Time scales
- Mechanisms
  - Queueing
  - Scheduling
  - Congestion Control
  - Admission Control
- Some open problems

Let’s order Pizza for home delivery

- Customer
  - calls a closest pizza outlet (what is selection based on??)
  - orders a pizza
    - Requirement specification
      - type, toppings (measurable quantities)
  - order arrives at home
    - Service Quality
      - How fast it arrived
      - Is the right pizza? Anything missing (quality measurements)
  - Customer Satisfaction (based on feeling!!, all parameters not measurable)
    - How was the service?
    - Is Pizza cold or hot? Is it fresh?
- Pizza company
  - How many customers and how fast to serve
  - Customer Satisfaction – Only through complaints (cannot really measure)
  - What they know – only what customer ordered (Requirement!!)

Economics Basics: utility function

- Users are assumed to have a *utility function* that maps from a given quality of service to a level of satisfaction, or utility
  - Utility functions are private information
  - Cannot compare utility functions between users
- *Rational* users take actions that maximize their utility
- Can determine utility function by observing preferences
- Generally networks do not support signaling of utility
  - They only support signaling of requirements (bandwidth, delay)
  - Networks use resource allocation to make sure requirements are satisfied
  - Measurements and Service Level Agreements (SLAs) determine customer satisfaction!!

Example: File Transfer

- Let $u(t) = S - \alpha t$
  - $u(t)$ = utility from file transfer
  - S = satisfaction when transfer infinitely fast
  - t = transfer time
  - α = rate at which satisfaction decreases with time
- As transfer time increases, utility decreases
- If $t > S/\alpha$, user is worse off! (reflects time wasted)
- Assumes linear decrease in utility
- S and α can be experimentally determined

Example: Video Conference

- Every packet must receive before a deadline
- Otherwise, the packet is too late and cannot be used
- Model:

  u(t)* = if *(t < D)* then *S

  else (-β)

  t is the end to end delay experienced by a packet

  D is the delay deadline

  S is the satisfaction

  -β is the cost (penalty) for missing deadline

  - causes performance degradation

- Sophisticated Utility measures for delay and packet loss
  - ***u(ε) = S(1- ε)*** where ε is the packet loss probability

Social welfare

- Suppose network manager knew the utility function of every user
- *Social Welfare* is maximized when some combination of the utility functions (such as sum) is maximized while minimizing the infrastructure cost
- An economy (network) is *efficient* when increasing the utility of one user must necessarily decrease the utility of another
- An economy (network) is *envy-free* if no user would trade places with another (better performance also costs more)
- Goal: maximize social welfare
  - subject to efficiency, envy-freeness, and making a profit

Example

- Assume
  - Single switch, each user imposes load (ρ=0.4)
  - A's utility: 4 - d
  - B's utility : 8 - 2d
  - Same delay (d) to both users
- Conservation law [$\sum(\rho_i d_i)$ = Constant]
  - 0.4d + 0.4d = C => d = 1.25 C => Sum of utilities = 12-3.75 C
- If B wants lower delay say to 0.5C, then A's delay = 2C
  - Sum of utilities = 12 - 3C (Larger than before)
  - By giving high priority to users that want lower delay, network can increase its utility
- *Increase in social welfare need not benefit everyone*
  - A loses utility, but may pay less for service

Some economic principles

- A single network that provides heterogeneous QoS is better than separate networks for each QoS
  - unused capacity is available to others
- Lowering delay of delay-sensitive traffic increases welfare
  - can increase welfare by matching service menu to user requirements
  - BUT need to know what users want (signaling)
- For typical utility functions, welfare increases more than linearly with increase in capacity
  - individual users see smaller overall fluctuations
  - can increase welfare by increasing capacity

Principles applied

- A single wire that carries both voice and data is more efficient than separate wires for voice and data
  - ADSL
  - IP Phone
- Moving from a 20% loaded 10 Mbps Ethernet to a 20% loaded 100 Mbps Ethernet will still improve social welfare
  - increase capacity whenever possible
- Better to give 5% of the traffic lower delay than all traffic low delay
  - should somehow mark and isolate low-delay traffic

The two camps

- Can increase welfare either by
  - matching services to user requirements *or*
  - increasing capacity blindly
- Which is cheaper?
  - no one is really sure!
  - small and smart vs. big and dumb
- It seems that smarter ought to be better
  - otherwise, to get low delays for some traffic, we need to give *all traffic* low delay, even if it doesn’t need it
- But, perhaps, we can use the money spent on traffic management to increase capacity
- We will study traffic management, assuming that it matters!

How useful are utility functions and economic framework?

- Do users really have such functions that can be expressed mathematically?
  - Practically no or less clear
  - Even if users cannot come up with a mathematical formula, they can express preference of one set of resources over other
    - These preferences can be codified as utility function
  - Best way to think about utility functions is that they may allow us to come up with a mathematical formulation of the traffic management problem that gives some insight
- Practical economic algorithms may never be feasible
- But policies and mechanisms based on these are still relevant

Network Types

- Single-Service Networks
  - Provide services for single type of traffic
  - e.g., Telephone Networks (Voice), Cable Networks (Video), Internet (Best effort Data)
- Multi-Service Networks
  - Provide services for multiple traffic types on the same network
  - e.g., Asynchronous Transfer Mode (CBR, VBR, ABR, UBR), Frame Relay, Differentiated Services (Diff-Serv), Integrated Services (Int-Serv), MPLS with Traffic Engineering
- Application types need to match the service provided
- Traffic models are used for the applications in order to match services, design, deploy the equipment and links.

Application Types

- Elastic applications (Adjust bandwidth and take what they get)
  - Wide range of acceptable rates, although faster is better
  - E.g., data transfers such as FTP
- Continuous media applications.
  - Lower and upper limit on acceptable performance
  - Sometimes called “tolerant real-time” since they can adapt to the performance of the network
    - E.g., changing frame rate of video stream
    - “Network-aware” applications
- Hard real-time applications.
  - Require hard limits on performance – “intolerant real-time”
  - E.g., control applications

Traffic models

- To align services, need to have some idea of how applications, users or aggregates of users behave = traffic model
  - e.g. how long a user uses a modem
  - e.g. average size of a file transfer
- Models change with network usage
- We can only guess about the future
- Two types of models
  - measurements
  - educated guesses

Telephone traffic models

- How are calls placed?
  - call arrival model
  - studies show that time between calls is drawn from an exponential distribution
  - call arrival process is therefore *Poisson*
  - memoryless: the fact that a certain amount of time has passed since the last call gives no information of time to next call
- How long are calls held?
  - usually modeled as exponential
  - however, measurement studies show it to be *heavy tailed*
  - means that a significant number of calls last a very long time
  - specially after usage of modems!!

Traffic Engineering for Voice Networks

- For a switch with N trunks, and with large population of users ($M \rightarrow \infty$), the probability of blocking (i.e., a call is lost) is given by Erlang-B formula

$$P_B = p_N = \frac{A^N / N!}{\sum_{n=0}^{N} A^n / n!}, \qquad \text{where} \quad A = \lambda / \mu$$

- λ is the call arrival rate (calls /sec)
- $1/\mu$ is the call holding time (3 minutes)
- Example: (For A = 12 Erlangs)
  - P_B = 1% for *N = 20; A/N = 0.6*
  - P_B = 8% for *N = 18; A/N = 0.8*
  - P_B = 30% for *N = 7; A/N = 1.7*

Distributions

- Long/heavy-tailed distributions
  - power law

$$P[X > x] \approx cx^{-\alpha} \; x \to \infty, \; \alpha, c > 0$$

  - Pareto

$$P[X > x] = c^{\alpha} x^{-\alpha}, \;\; x \geq \; b$$

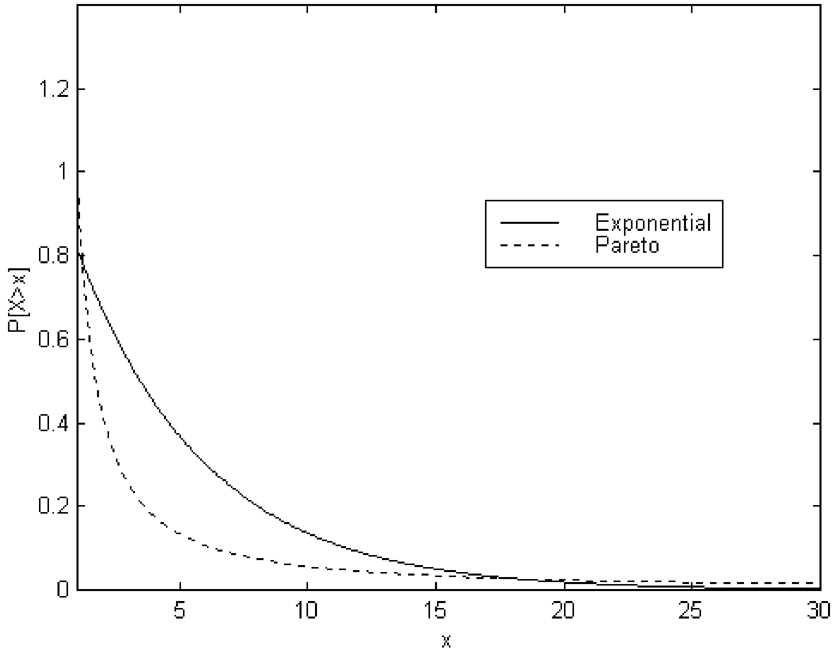
- Exponential Distribution

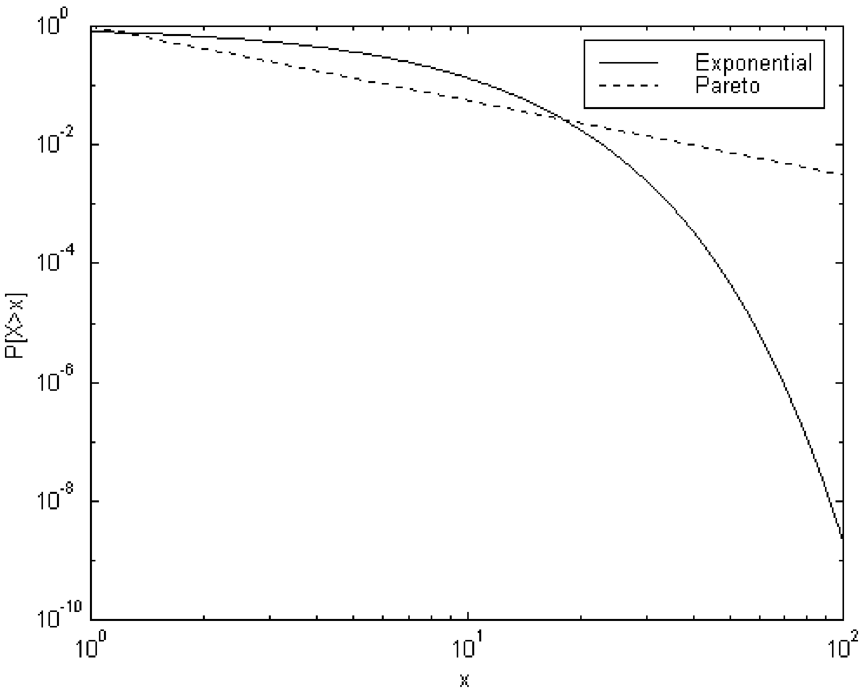
$$P[X > x] = e^{-ax}$$

Pareto distribution

- 1<α<2 => infinite variance

Power law decays more slowly than exponential ⇒ heavy tail





Internet traffic modeling

- A few apps account for most of the traffic
  - WWW
  - FTP
  - telnet
- A common approach is to model apps (this ignores distribution of destination!)
  - time between app invocations
  - connection duration
  - # of bytes transferred
  - packet inter-arrival distribution
- Little consensus on models
- But two important features

Internet traffic models: features

- LAN connections differ from WAN connections
  - Higher bandwidth (more bytes/call)
  - longer holding times
- Many parameters are heavy-tailed
  - examples
    - # of bytes in call
    - call duration
  - means that a *few* calls are responsible for most of the traffic
  - these calls must be well-managed
  - also means that *even aggregates with many calls not be smooth*
  - can have long bursts
- New models appear all the time, to account for rapidly changing traffic mix