CS 5224
High Speed Networks and Multimedia Networking

Traffic Model and Engineering

Dr. Chan Mun Choon
School of Computing, National University of Singapore

Acknowledgement/Reference

- Slides are taken from the following source:
  - S. Keshav, "An Engineering Approach to Computer Networking", Chapter 14: Traffic Management

Motivation for Traffic Models

- In order to predict the performance of a network system, we need to be able to "describe" the "behavior" of the input traffic
  - Often, in order to reduce the complexity, we classify the user behavior into classes, depending on the applications
  - Sometimes, we may be even able to "restrict" or shape the users' behavior so that they conform to some specifications
- Only when there is a traffic model is traffic engineering possible

An example

- Executive 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*

Traffic management

- Set of policies and mechanisms that allow a network to *efficiently* satisfy a *diverse* range of service requests
- Tension is between **diversity** and **efficiency**
- Traffic management is necessary for providing *Quality of Service (QoS)*
  - Subsumes congestion control (congestion == loss of efficiency)

Time Scale of Traffic Management

- Less than one round-trip-time (cell-level)
  - Perform by the end-points and switching nodes
  - Scheduling and buffer management
  - Regulation and policing
  - Policy routing (datagram networks)
- One or more round-trip-times (burst-level)
  - Perform by the end-points
  - Feedback flow control
  - Retransmission
  - Renegotiation

Time Scale (cont.)

- Session (call-level)
  - End-points interact with network elements
  - Signaling
  - Admission control
  - Service pricing
  - Routing (connection-oriented networks)
- Day
  - Human intervention
  - Peak load pricing
- Weeks or months
  - Human intervention
  - Capacity planning

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 increased welfare
  - can increase welfare by matching service menu to user requirements
  - BUT need to know what users want (signaling)

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% loaded10 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

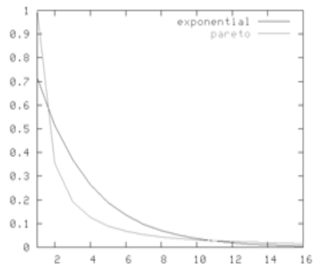
- Can increase welfare either by
  - matching services to user requirements *or*
  - increasing capacity blindly
- Which is cheaper?
  - depends on technology advancement
  - User behavior/expectation/tolerance
  - small and smart vs. big and dumb
- It seems that smarter ought to be better
  - otherwise, 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!

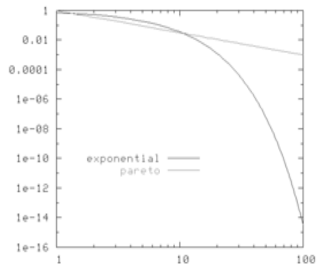
Telephone traffic models (Call)

- 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 (in the mid-90s) show that it is *heavy tailed*
    - A small number of calls last a very long time
    - Why?

Exponential/Heavy Tail Distribution

- Exponential Distribution: $P(X>x) = e^{-x/3}$
- Pareto Distribution: $P(X>x) = x^{-1.5}$
- Means of both distributions are 3





Packet Traffic Model for Voice

- A single voice source is well represented by a two state process: an alternating sequence of active or talk spurt, follow by silence period
  - Talk spurts typically average 0.4 – 1.2s
  - Silence periods average 0.6 – 1.8s
  - Talk spurt intervals are well approximated by exponential distribution, but mot true for silence period
  - Silence periods allow voice packets to be multiplexed
  - For more detail description, take a look at Chapter 3 of “Broadband Integrated Networks”, by Mischa Schwartz, 1996.

Internet traffic modeling

- A few apps account for most of the traffic
  - WWW, FTP, E-mail
  - P2P
- A common approach is to model apps (this ignores distribution of destination!)
  - time between app invocations
  - connection duration
  - # 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 usage (more bytes/call)
  - longer holding times
- Many parameters are heavy-tailed
  - examples
    - # bytes in call (e.g. file size of a web download)
    - 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*

Traffic classes

- Networks should match offered service to source requirements (corresponds to utility functions)
  - Telephone network offers one single traffic class
  - The Internet offers little restriction on traffic behavior
- Example: telnet requires low bandwidth and low delay
  - utility increases with decrease in delay
  - network should provide a low-delay service
  - or, telnet belongs to the low-delay *traffic class*

Traffic classes - details

- A basic division: guaranteed service and best effort
  - like flying with reservation or standby
- Guaranteed-service
  - utility is zero unless app gets a minimum level of service quality: bandwidth, delay, loss
  - open-loop flow control (e.g. do not send more than x Mbps) with admission control
  - e.g. telephony, remote sensing, interactive multiplayer games
- Best-effort
  - send and pray
  - closed-loop flow control (e.g. TCP)
  - e.g. email, ftp

GS vs. BE (cont.)

- Degree of synchrony
  - time scale at which peer endpoints interact
  - GS are typically *synchronous* or *interactive*
    - interact on the timescale of a round trip time
    - e.g. telephone conversation or telnet
  - BE are typically *asynchronous* or *non-interactive*
    - interact on longer time scales
    - e.g. Email
- Sensitivity to time and delay
  - GS apps are *real-time*
    - performance depends on wall clock
  - BE apps are typically indifferent to real time
    - automatically scale back during overload

Example of Traffic Classes

- ATM Forum
  - based on sensitivity to bandwidth
  - GS
    - CBR, VBR
  - BE
    - ABR, UBR
- IETF
  - based on sensitivity to delay
  - GS
    - intolerant
    - tolerant
  - BE
    - interactive burst
    - interactive bulk
    - asynchronous bulk

ATM Forum GS subclasses

- Constant Bit Rate (CBR)
  - constant, cell-smooth traffic
  - mean and peak rate are the same
  - e.g. telephone call evenly sampled and uncompressed
  - constant bandwidth, variable quality
- Variable Bit Rate (VBR)
  - long term average with occasional bursts
  - try to minimize delay
  - can tolerate loss and higher delays than CBR
  - e.g. compressed video or audio with constant quality, variable bandwidth

ATM Forum BE subclasses

- Available Bit Rate (ABR)
  - users get whatever is available
  - zero loss if network signals (in RM cells) are obeyed
  - no guarantee on delay or bandwidth
- Unspecified Bit Rate (UBR)
  - like ABR, but no feedback
  - no guarantee on loss
  - presumably cheaper

IETF GS subclasses

- Tolerant GS
  - nominal mean delay, but can tolerate "occasional" variation
  - not specified what this means exactly
  - uses *controlled-load* service
    - book uses older terminology (predictive)
  - even at "high loads", admission control assures a source that its service "does not suffer"
  - it really is this imprecise!
- Intolerant GS
  - need a worst case delay bound
  - equivalent to CBR+VBR in ATM Forum model

IETF BE subclasses

- Interactive burst
  - bounded asynchronous service, where bound is qualitative, but pretty tight
    - e.g. paging, messaging, email
- Interactive bulk
  - bulk, but a human is waiting for the result
  - e.g. FTP
- Asynchronous bulk
  - bulk traffic
  - e.g P2P

Some points to ponder

- The only thing out there is CBR (example?) and asynchronous bulk (example?)!
- These are application requirements. There are also organizational requirements (how to provision QoS end-to-end)
- Users needs QoS for other things too!
  - billing
  - reliability and availability

Reading

- Reference
  - Bertsekas and Gallager, "Data Networks", 2nd Edition, Chapter 3: Delay Models in Data Network, Prentice Hall

Motivation for Traffic Engineering

- Traffic engineering for a wide-range of traffic models and classes is difficult even for a single networking node
- However, if we restrict ourselves to a small set of traffic model, one can get some good intuition
  - For example, traffic engineering in the telephone network has been effective
  - The M/M/* queuing analysis is a simple and elegant way to perform basic traffic engineering

A Question …

- Waiting time at two fast-food stores MD and BK
  - In MD, a queue is formed at each of the m servers (assume a customer chooses queue independently and does not change queue once he/she joins the queue)
  - In BK, all customers wait at a single queue and served by m servers
  - Which one is better?

Multiplexing of Traffic

- Traffic engineering involves the sharing of resource/link by several traffic streams
- Time-Division Multiplexing (TDM)
  - Divide transmission into time slots
- Frequency Division Multiplexing (FDM)
  - Divide transmission into divide frequency channels
- For TDM/FDM, if there is no traffic in a data stream, bandwidth is wasted
- In statistical multiplexing, data from all traffic streams are merged into a single queue and transmitted in a FIFO manner
- Statistical multiplexing
  - has smaller delay per packet than TDM/FDM
  - can have larger delay variance
  - Results can be shown using queuing analysis

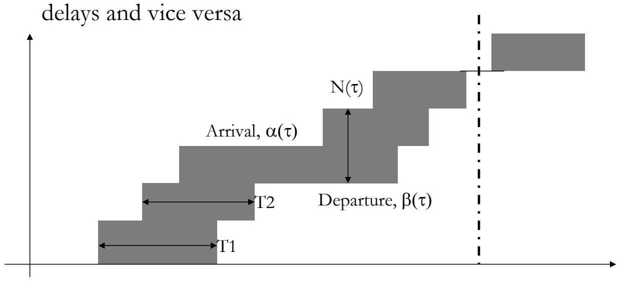
Little's Theorem

- Given customer arrival rate (λ), service rate (μ)
  - What is the average number of customers (N) in the system and what is the average delay per customer (T) ?
- Let
  - N(t) = # of customers at time t
  - α(t) = # of customers arrived in the interval [0,t]
  - T_i = time spent in system by i^{th} customer
- N_t, "typical" # of customers up to time t is $\frac{1}{t}\int_0^t N(\tau)d\tau$

$$N = \lim_{t \to \infty} N_t \qquad \lambda = \lim_{t \to \infty} \lambda_t \qquad T = \lim_{t \to \infty} T_t$$

Little's Theorem

- Little's Theorem: N = λT
  - Average # of customers = average arrival rate * average delay time of a customer
  - Crowded system (large N) are associated with long customer delays and vice versa



Derivation of Little's Theorem

Little's Theorem (cont'd)

- Little's Theorem is very general and holds for almost every queuing system that reaches statistics equilibrium in the limit

Example

- BG, Example 3.1
  - L is the arrival rate in a transmission line
  - N_Q is the average # of packets in queue (not under transmission)
  - W is the average time spent by a waiting packet (exclude packet being transmitted)
  - From LT, $N_Q = \lambda W$
  - Furthermore, if X is the average transmission time,
    - $\rho = \lambda X$
    - where ρ is the line's utilization factor (proportion of time line is busy)

Example

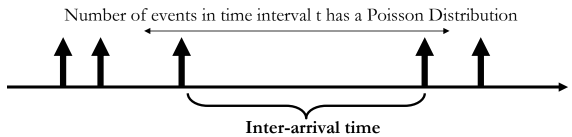
- BG, Example 3.2
  - A network of transmission lines where packets arrived at n different nodes with rate $\lambda_1 \lambda_{2,\ldots,}\lambda_n$
  - N is total number of packets in network
  - Average delay per packet is $T = \dfrac{N}{\sum_{i=1}^{n} \lambda_i}$
  - independent of packet length distribution (service rate) and routing

What is a Poisson Process?

- A Poisson Process A(t)
  1. A(t) is a **counting process** that represents the total number of arrivals that have occurred from 0 to t, A(t) – A(s) equals the number of arrivals in the interval (s,t]
  2. Number of arrivals that occur in disjoint intervals are **independent**
  3. Number of arrivals in any interval τ is Poisson distributed with parameter $\lambda\tau$

$$P\{A(t+\tau) - A(t) = n\} = e^{-\lambda\tau} \frac{(\lambda\tau)^n}{n!}$$

Inter-arrival Time



- Based on the definition of Poisson process, what is the inter-arrival time between arrivals?
- The distribution of inter-arrival time, t, can be computed as P{A(t) = 0}
- Using only Property 2, it can be shown that inter-arrival times are independent and exponentially distributed with parameter λ

Exponential Distribution

- Probability Density Distribution $p\{\tau\} = \lambda e^{-\lambda\tau}$
- Cumulative Density Distribution $P\{\tau \leq s\} = 1 - e^{-\lambda s}$
- Mean $E\{\tau\} = \dfrac{1}{\lambda}$
- Variance $Var\{\tau\} = \dfrac{1}{\lambda^2}$

Poisson Process

- **Merging:** if two or more independent Poisson process are merged into a single process, the merged process is a Poisson process with a rate equal to the sum of the rates
- **Splitting**: if a Poisson process is split probabilistically into two processes, the two processes are obtained are also Poisson

Memoryless Property

- For service time with exponential distribution, the additional time needed to complete a customer's service in progress is independent of when the service started

$$P\{\tau_n > r + t \mid \tau_n > t\} = P\{\tau_n > r\}$$

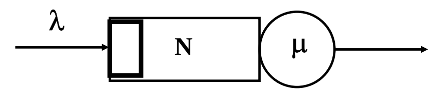
- Inter-arrival time of bus arriving at a bus stop has an exponential distribution. A random observer arrives at the bus stop and a bus just leave t seconds ago. How long should the observer expects to wait?

Applications of Poisson Process

- Poisson Process has a number of "nice" properties that make it very useful for analytical and probabilistic analysis
- Has been used to model a large number of physical occurrences [KLE75]
  - Number of soldiers killed by their horse (1928)
  - Sequence of gamma rays emitting from a radioactive particle
  - Call holding time of telephone calls
  - **In many cases, the sum of large number of independent stationary renewal process will tend to be a Poisson process**

[KLE75] L. Kleinrock, "Queuing Systems," Vol I, 1975.

Basic Queuing Model



M/M/1

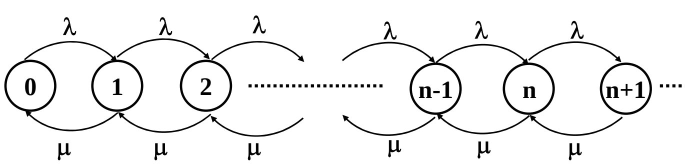
Departure Process
Exponential with mean $1/\mu$

Number of servers

Arrival Process
Memoryless (or Poisson process with rate λ)

- **Default N is infinite**
- **D - deterministic, G - General**

Birth-Death Process



- **Model queue as a discrete time Markov chain**
- **Let P_n be the steady state probability that there are n customers in the queue**
- **Balance equation: at equilibrium, the probability a transition out of a state is equal to the probability of a transition into the same state**

Derivation of M/M/1 Model

- **Balance Equations:**
  - $\lambda P_0 = \mu P_1, \lambda P_1 = \mu P_2, \ldots, \lambda P_{n-1} = \mu P_n$
- **Let $\rho = \lambda/\mu$**
- $\rho P_0 = P_1, \rho P_1 = P_2, \ldots, \rho P_{n-1} = P_n$

$$P_n = \rho^n P_0$$

Derivation of M/M/1 Model

$$P_n = \rho^n P_0$$

$$\Sigma_n P_n = \Sigma_n \rho^n P_0 = P_0 / (1 - \rho) = 1 \ (\rho < 1)$$

$$P_0 = (1 - \rho)$$

$$P_n = \rho^n (1 - \rho)$$

Average Number of Customers in System, N

$$N = \Sigma_n nP_n = \rho / (1 - \rho) = \lambda / (\mu - \lambda)$$

Properties of M/M/1 Queue

- $N = \rho / (1 - \rho) = \lambda / (\mu - \lambda)$
  - ρ can be interpreted as the utilization of the queue
  - System is unstable if $\rho > 1$ or $\lambda > \mu$ as N is not bounded
- In M/M/1 queue, there is no blocking/dropping, so waiting time can increase without any limit
  - Buffer space is infinite, so customers are not rejected
  - But there are "infinite number" of customers in front

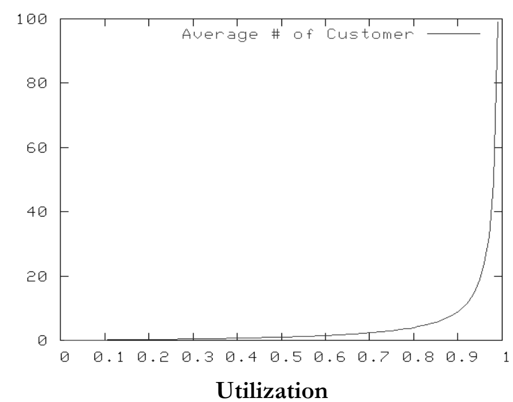
M/M/1

- From Little's Theorem,

$$T = \frac{N}{\lambda} = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\mu - \lambda}$$

$$W = \frac{1}{\mu - \lambda} - \frac{1}{\mu} = \frac{\rho}{\mu - \lambda}$$

More properties of M/M/1



Example

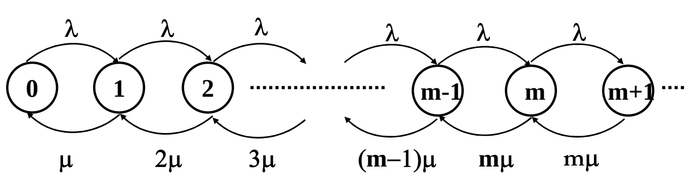
- BG, Example 3.8 (Statistical Multiplexing vs. TDM)
  - Allocate each Poisson stream its own queue (λ,μ) or shared a single faster queue (kλ,kμ)?
  - Increase λ and μ or a queue by a constant k > 1
  - $\rho = k\lambda/k\mu = \lambda/\mu$ (no change in utilization)
  - $N = \rho / 1-\rho = \lambda / \mu - \lambda$ (no change)
  - What changes?
    - $T = 1/k(\mu - \lambda)$
    - Average transmission delay decreases by a factor k
  - **Why?**

Example

- BG, Example 3.9
  - Consider k TDM/FDM channels
  - From previous example, merging k channels into a single (k times faster) will keep the same N but reduces average delay by k
  - So why use TDM/FDM ?
    - Some traffic are not Poisson. For example, voice traffic are "regular" with one voice packet every 20ms
    - Merging multiplexing traffic streams into a single channel incurs buffering, "queuing delay" and jitter

Extension to M/M/m Queue

- **There are m servers, a customer is served by one of the servers**
- **$\lambda p_{n-1} = n\mu p_n$ (n <= m)**
- **$\lambda p_{n-1} = m\mu p_n$ (n > m)**



Derivation of M/M/m Model

- **Balance Equations:**
  - **$\lambda P_0 = \mu P_1$, $\lambda P_1 = 2\mu P_2$, … , $\lambda P_{n-1} = n\mu P_n$**
- **Let $\rho = \lambda/m\mu$**

$$p_n = p_0 \frac{(m\rho)^n}{n!}, n \le m$$

$$p_n = p_0 \frac{m^m \rho^n}{m!}, n > m$$

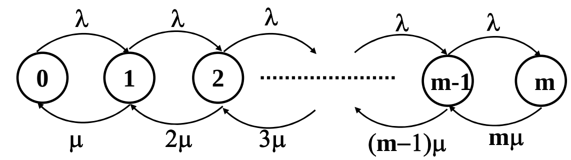
Derivation of M/M/m Model

$$\sum_{n=0}^{\infty} p_n = 1$$

In order to compute P_n, $\mathbf{P_0}$ must be computed first.

Extension to M/M/m/m Queue

- **There are m servers and m buffer size**
- **This is no buffering**
- **Calls are either served or rejected, calls rejected are lost**
- **Common model for telephone switching**



M/M/m/m Queue

Balanced Equations:

$\lambda P_0 = \mu P_1, \lambda P_1 = 2\mu P_2, \ldots, \lambda P_{n-1} = n\mu P_n$

$P_n = P_0 (\rho^n) / n!$

$\Sigma^m{}_n P_n = \Sigma^m{}_n P_0 (\rho^n) / n! = 1$

$P_0 = (\Sigma^m{}_n (\rho^n) / n!)^{-1}$

When does loss happens?

Loss happens when a customer arrives and see m customers in the system

M/M/m/m Queue

- PASTA: Poisson Arrival see times averages
  - P_m is time average
  - Use time averages to compute loss rate
- Loss for M/M/m/m queue is computed as the probability that there are m customers in the system:

$$(\rho^m/m!) \left(\Sigma^m{}_{n=0} (\rho^n/n!) \right)^{-1}$$

- The above equation is known as Erlang B formula and widely used to evaluate blocking probability

What is an Erlang?

- An *Erlang* is a unit of telecommunications traffic measurement and represents the continuous use of one voice path
  - Average number of calls in progress
- Computing Erlang
  - Call arrival rate: λ
  - Call Holding time is: $1/\mu$, call departure rate = μ
  - System load in Erlang is λ/μ
- Example:
  - λ = 1 calls/sec, $1/\mu$ = 100sec, load = 1/0.01 = 100 Erlangs
  - λ = 10 calls/sec, $1/\mu$ = 10sec, load = 10/0.1 = 100 Erlangs
- Load is function of the ratio of arrival rate to departure rate, independent of the specific rates

Erlang B Table

Capacity (Erlangs) for grade of service of				
# of Servers (N)	P=0.02	P=0.01	P=0.005	P=0.001
1	0.02	0.01	0.005	0.001
5	1.66	1.36	1.13	0.76
10	5.08	4.46	3.96	3.09
20	13.19	12.03	11.1	9.41
40	31.0	29.0	27.3	24.5
100	87.97	84.1	80.9	75.2

- **For a given grade of service, a larger capacity system is more efficient (statistical multiplexing)**
- **A larger system incurs a larger changes in blocking probability when the system load changes**

Example

- If there are 40 servers and target blocking rate is 2%, what is largest load supported?
  - P=0.02, N = 40
  - Load supported = 31 Erlang
- Calls arrived at a rate of 1calls/sec and the average holding time is 12 sec. How many trunk is needed to maintain call blocking of less than 1%?
  - Load = 1*12 = 12 Erlang
  - From Erlang B table, if P=0.01, N >= 20

Multi-Class Queue

- We can extend the Markov Chain for M/M/m/n to multi-class queues
- Such queues can be useful, for example, in cases where there is preferential treatment for one class over another

Network of Queues

- In a network, departing traffic from a queue is strongly correlated with packet lengths beyond the first queue. This traffic is the input to the next queue.
  - Analysis using M/G/1 is affected
- Kleinrock Independence Approximation
  - Poisson arrivals at entry points
  - Densely connected network
  - Moderate to heavy traffic load
- Network with Product Form Solutions