CS 622
Distributed Networks

Three-location Data Network Design

Dr. Xiaobo Zhou
Department of Computer Science

Review: A Two-Location Problem

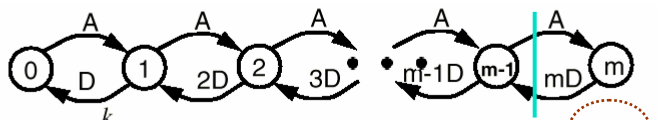
- Design a network connecting two locations, Anagon and Bregen, 200km apart
- Anagon: 5 employees, Bregen: 10 employees
- Each employee
  - call other site 4 times/day, avg. 5 minutes each

    4*5*15=300 min/day
  - call others in the same office 10 times/day about joint work, each last avg. 3 minutes

    10*3*15=450 min/day

    Note here we are not using C(10,2)+C(5,2) for the # of calls
- How can we best provide the communications between the 2 cities?

Review: Loss with m Lines (m servers, no queue)



A: Arrive Rate; D: Departure Rate; E=A/D

$AP_{k-1} = kDP_k \rightarrow P_k = E/k * P_{k-1}$

- $P_k = \frac{E^k}{k!} P_0$
- $\sum_{k=0}^{m} p_k = \sum_{k=0}^{m} \frac{E^k}{k!} P_0 = 1 \rightarrow 1/P_0 = \sum_{k=0}^{m} \frac{E^k}{k!}$
- Probability of loss = $P_m = \frac{E^m}{m!} P_0 = \frac{\frac{E^m}{m!}}{\sum_{k=0}^{m} \frac{E^k}{k!}} = B(E, m)$
- $B(E, k) = \frac{EB(E, k-1)}{EB(E, k-1) + k} \forall k = 1, 2, \ldots, m$ $\quad B(E, 0) = 1$

Erlang-B Function Recursion

3-location Data Network Design

° **3 locations separated by 200 km among pairs.**

° **Give the new populations below 296 users, design the data network.**

Site	*Population*
Anagon	96
Bregen	128
Charmes	72

Voice traffic	*Data traffic*
Fixed bandwidth	Varying bandwidth
Short duration calls	Long duration calls
One connection/person	Many connections/person
Extreme delay sensitivity	Varying sensitivity
Tolerates some loss	Varies

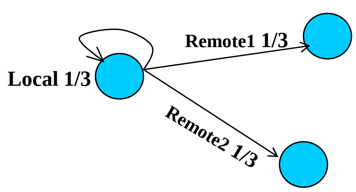
Voice Traffic vs. Data Traffic

Data Traffic Statistics

- Three applications: email, external Web access, distributed DataBase
  - 20% of internal email, www, DB traffic occurs in the busy hour
  - External email arrives evenly during the day.
- Data Traffic Statistics:
  - Average internal email size 60 KB (= 60 * 8 Kb)
  - External email size 12 KB
  - Each URL request generates 6 datagrams to server, 6 datagrams back to client for setup connections, a datagram average 128 B.
    - Its http response is 2KB datagram.

Traffic	Volume
Internal email	10 pieces sent and received per employee/day
External email	4000 pieces/day arriving at a steady rate
WWW	40 fetches/day/user
Database	50 queries + 5 updates per employee/day

DataBase Traffic



- Data distributed in 3 servers, one at each site (one copy only)
  - Probability of data in a server is 1/3. Evenly spread
- Each employee makes 50 queries and 5 updates
- Query:
  - Each query first goes to the local server, then goes some remote server -- *Does it need to go to third server*?
  - Total # queries *actually* made by one employee is
    - 50 * (1/3 * 1 + 1/3 * 2 + 1/3 *3) = 100 if the file distribution is unknown
    - 50 * (1/3 + 1/3 + 1/3) = 50 if the file distribution is known
  - Query packet avg. 800B, response packet average 3500B
- Update:
  - Update packet avg. 6000B, response packet 500B

Cost of Services and Components

How can we best provide the communications between the 3 sites?

multiple LANs but 1 WAN link → Terminal router

multiple WANs → Transit router

Item	Cost
Terminal router	$2000 purchase price
Transit router	$3700 purchase price
WAN adapter	$500 purchase price
64,000 bps internode link	$700/month
256,000 bps internode link	$1400/month
2,048,000 bps internode link	$3800/month
64,000 bps internet link	$1400/month
256,000 bps internet link	$2800/month
2,048,000 bps internet link	$7600/month

- **Cost of PC's, workstations, servers not considered**
- **Routers can handle 2000 datagrams/sec >> the traffic**
  - **→ Delay can be neglected.**

Data Network Design Principle 2.2

- **Blocking in not important; delay is the issue**
- **Highly utilized links are not desirable because of large delay)**

Design Principle 2.2

- *In a voice network, highly utilized links can be cost-effective, since they exploit the available bandwidth to the fullest extent, and when the link is given to a connection it receives a high grade of service (a highly utilized link is desirable for exclusive use in a circuit-switched voice network).*
- *In a (packet-switched) data network, highly utilized links are terrible since all call traffic using that link suffers inordinate (queueing) delay.*

Queueing theory in Data Network!

Traffic Burstiness – it is necessary to share

- In a data network, traffic is more bursty than a voice network
  - Burstiness = peak rate / avg rate
  - If you and I both have data calls in progress that are busy for 10% of the time, why we should have an exclusive access to a line, instead of sharing !
  - Problem: what happens if we both want to use the line at the same time?

Two solutions to simultaneous arrive of data calls:

- Coordination- e.g., token ring
  - Allow one holding the only token to be served
  - Good in LAN, but not good for WAN
    - Token Propagation Delay (P.D.) for 1000 mile ring
      - 1000(mile) / 186000 (mile/s) = 5 ms
    - Transmission delay for 1000 bit packet at 16Mbps
      - 1000 (bit) / 16000000 Mbps << P.D.
- Using store-and-forward for packet forwarding (switching)
  - Packets that arrive to find the link busy queue up at intermediate buffers, for how long?

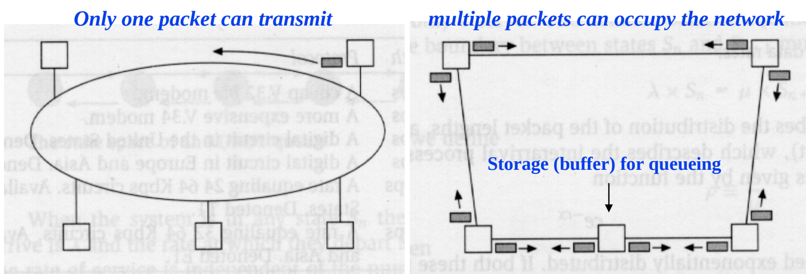
Common Data Transfer Rate

Bandwidth	*Protocol*
14,400 bps	A cheap V.32 *bis* modem.
28,800 bps	A more expensive V.34 modem.
56,000 bps	A digital circuit in the United States. Denoted D56.
64,000 bps	A digital circuit in Europe and Asia. Denoted D64.
1.544 Mbps	A rate equaling 24 64 Kbps circuits. Available in the United States. Denoted T1.
2.048 Mbps	A rate equaling 32 64 Kbps circuits. Available in Europe and Asia. Denoted E1.

The service time for a packet of n bits on a link of speed S bps is n/S

Example: transfer a 1Kb packet by a T1 link takes 1KB/1.544Mbps

Token Ring vs. Packet Switching



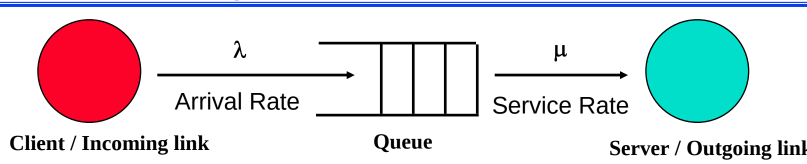
A. Token Ring **B: Packet Switch**

- A: Propagation delay for 1000 mile ring = 1000/186,000 = 5.376 ms
- B: Transmission delay for 1000 bit packet at 16Mbps
  - 1000/16,000,000 = 0.0625ms << 5.376 ms

- For WAN, token ring protocol is not suitable
- A packet switching network where each link segment operates independently and simultaneously is a more efficient design

- A packet switching network can be modeled as a set of queues.

Client-Server Queue Model in Data Network

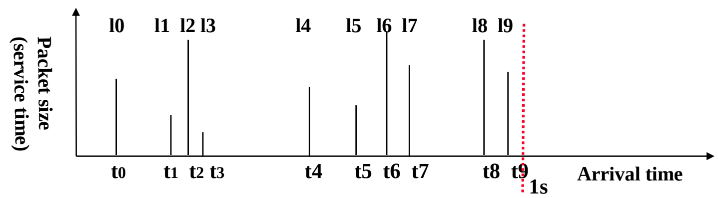


° Arrival rate λ (in voice network, denoted by A):
  - The number of packets coming from the client (incoming link) in a unit time

° Service rate μ (in voice network, denoted by D):
  - The number of packets forwarded by the server (outgoing link) in unit time

° Utilization $\rho = \lambda / \mu$ (in voice network, denoted by E=A/D)
  - If $\rho > 1$, the queue's length is to be infinite, so is the queueing delay
  - If $\rho = 1$, the queue's length is to be infinite, so is the queueing delay if the inter-arrival distributions != the inter-leaving distributions; otherwise, finite queue's length, no queueing-delay (D/D/1 queue)
  - If $\rho < 1$, finite queue's length, finite queueing delay!
    - Most scenarios belong to this case !

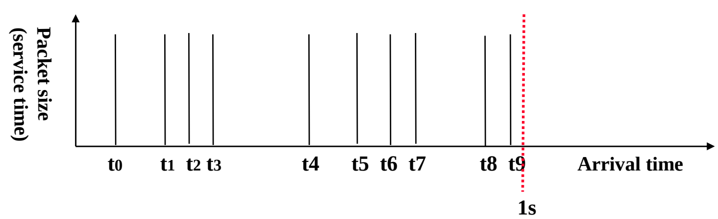
Preliminary of Queueing Theory



Arrival rate $\lambda = 10$ packets / requests per second, the 10 packets/requests come to the link usually in an un-uniform way, and each packet/request generally has different size (requiring different forwarding/service time)

- **Three Components**
  - Inter-arrival time distributions (t0, t1-t0, t2-t1, t3-t2, …, t9-t8)
  - Packet length / Service time distributions (l0, l1, l2, l3, …, l9)
  - The number of servers (FCFS queue discipline)
- **Distributions**
  - Deterministic distribution (D)
  - Exponential distribution (M): (ce^{-cx}) – Poisson distribution
  - General distribution (like pareto…)
- **Response time (total time) = queueing delay + service time**

M/D/1 Queue – the simplest queue



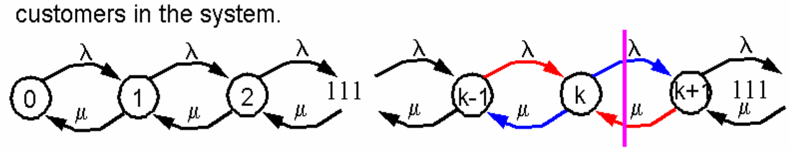
- **Inter-arrival time distribution is exponential** $f(x) = \lambda e^{-\lambda x} \quad (0 < x < \infty)$
  - **The mean is** $\int_0^{\infty} f(x) = x\lambda e^{-\lambda x} \mathrm{dx} = \frac{1}{\lambda}$
- **Service time distribution is deterministic (fixed and uniform)**
- **Giving arrival rate λ, service rate μ (1/*b*, where *b* is the fixed uniform service time), according to Queueing Theory (Kleinrock75a&b)**
  - **Average queueing-delay: E(*Tw*) =** $\frac{1}{2\lambda} \cdot \frac{(\lambda b)^2}{(1-\lambda b)} = \frac{1}{2\lambda} \cdot \frac{\rho^2}{(1-\rho)}$
  - **Average response time (*T=Tw+Ts*) =** $\frac{1}{2\lambda} \cdot \frac{(\lambda b)^2}{(1-\lambda b)} + b = \frac{1}{2\lambda} \cdot \frac{\rho^2}{(1-\rho)} + b$

M/M/1 Queue – A General Queue

Poisson arrival process, exponential service time, single server

p_k=Prob[k customers in the system]

To analyze M/M/1 Queue, let us examine its system behavior described by the following state transition diagram. State number represent the number of customers in the system.



At equilibrium state the following equations hold

$\lambda p_0 = \mu p_1$

$(\lambda + \mu) p_k = \lambda p_{k-1} + \mu p_{k+1}$ $\quad k \geq 1$ // the prob. flow comes in = p.f. goes out

Alternatively, $\lambda p_k = \mu p_{k+1}$ $\quad k \geq 1$ // equilibrium on the links between states.

Solving $p_k = \frac{\lambda}{\mu} p_{k-1} = \rho p_{k-1}$ where $\rho = \frac{\lambda}{\mu}$ by definition

$p_k = (1-\rho)\rho^k \quad k \geq 0$ ← $\sum_{i=0}^{\infty} p_i = p_0 \sum_{i=0}^{\infty} \rho^i = \frac{\rho_0}{1-\rho} = 1 \Rightarrow p_0 = 1-\rho$

Mean number of customers in the system $N = \sum_{k=0}^{\infty} k p_k = \frac{\rho}{1-\rho}$

M/M/1 Average Waiting Time

With probability 1-ρ, a message arrives to an empty system and no wait.

With probability (1-ρ)ρ, a message arrives with 1 message in service and wait for it to complete.

- Since the service time is exponential distributed, the length of the time that message has been in service does not affect the time it now takes to complete.
- The average wait is the average service time, Ts.

With probability $P_k=(1-\rho)\rho^k$, a message waits for k other messages to complete service and the waiting time = kTs.

Therefore the average waiting time $T_w = \sum_{k=0}^{\infty} p_k(kT_s) = T_s \sum_{k=0}^{\infty} k(1-\rho)\rho^k = Ts\frac{\rho}{1-\rho}$.

The total time in the system $T = T_s + T_w = T_s\left(1 + \frac{\rho}{1-\rho}\right) = \frac{T_s}{1-\rho}$

Mean number of customers in the system $N = \lambda T = \lambda \frac{T_s}{1-\rho} = \lambda \frac{1/\mu}{1-\rho} = \sum_{k=0}^{\infty} k p_k = \frac{\rho}{1-\rho}$

N=λT is called Little's Law, hold true for a lot of queues.

Average queueing-delay: E(*Tw*) = $\frac{\rho/\mu}{1-\rho}$

Average response time (*T=Tw+Ts*) = $\frac{1}{\mu(1-\rho)}$

M/G_p/1 Queue (paper 4.4)

° **M/G/1 queue is a single-server with Poisson arrivals and arbitrary service-time distribution**

- **Expected queueing-delay (Pollaczek-Khinchin formula):**

$$E[W] = \frac{\lambda E[X^2]}{2(1-\lambda E[X])}$$

° **Recent Internet workload measurements indicate that for many Web applications, a heavy-tailed distribution is more accurate model for service time distribution than the exponential distribution.**

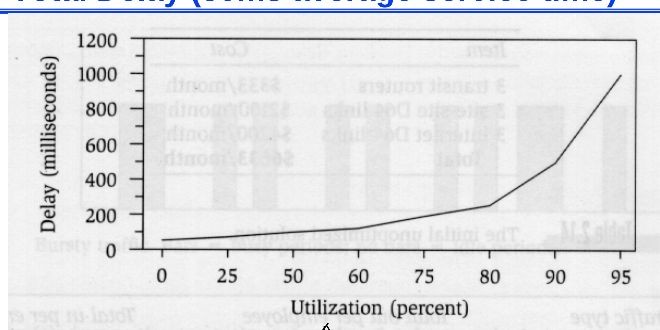
° **M/G_p/1 queue is a single-server with Poisson arrivals and Pareto service-time distribution**

- **The Pareto distribution is a typical heavy-tailed distribution, with probability function** $f(x) = \alpha k^{\alpha} x^{-\alpha-1} \qquad \alpha, k > 0, x \ge k$

- **In practice, there is some upper bound on the maximum size of a job (p) -- Bounded Pareto distribution**

$$f(x) = \frac{1}{1-(k/p)^{\alpha}} \alpha k^{\alpha} x^{-\alpha-1} \qquad \alpha, k > 0, p \ge x \ge k$$

Total Delay (50ms average service time)



Throughput / Capacity

° **In order to get the highest utilization (or throughput):**

° The server/forwarding link should never be idle

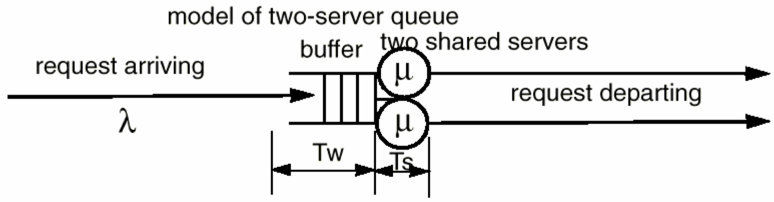
° The queue should never be empty

° **In order to minimize the queueing-delay (or response time):**

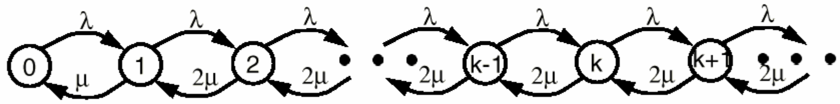
° The queue should be empty

° The server/forwarding link will be idle

M/M/2 Queue



Its state transition diagram is as follows:

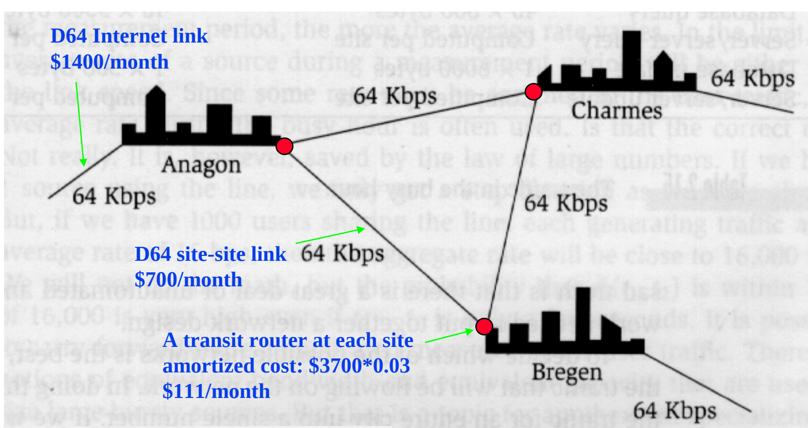


At equilibrium state the following equations hold

$\lambda p_0 = \mu p_1 \rightarrow p_1 = \rho p_0$

Alternatively, $\lambda p_k = 2\mu p_{k+1} \rightarrow p_{k+1} = \frac{\rho}{2} p_k \quad k \geq 1$

Initial Data Network Design & Cost



Item	*Cost*
3 transit routers	$333/month
3 site-site D64 links	$2100/month
3 internet D64 links	$4200/month
Total	**$6633/month**

Traffic in Busy Hour

- 20%=0.2 traffic in busy hour.

Traffic type	*Total out per employee*	*Total in per employee*
Internal email	2 × 60,000 bytes	2 × 60,000 bytes
External email	~~4000 × 0.2 × 12,000/296 bytes~~	~~4000 × 0.2 × 12,000/296 bytes~~
WWW	8 × 6 × 128 bytes	8 × (6 × 128 + 2000) bytes
Database query	10 × 800 bytes	10 × 3500 bytes
Server/server query	Computed per site	Computed per site
Database update	1 × 6000 bytes	1 × 500 bytes
Server/server update	Computed per site	Computed per site

External evenly distributed:
4000 * 12000 / (8hr * 296)

Design Principles 2.3 & 2.4

- **Design Principle 2.3**
  - ***Seek to make a network where all the links have a 50% utilization (50% threshold is a good tradeoff between the queueing delay and cost)***
- **Design Principle 2.4**
  - ***Seek to make a network where all the links have about 50% utilization and as few links as possible are underutilized***
- **A fuzzy example:**

Name	*Cost*	*Average delay*	*Maximum utilization*
Design-48	$160,000/month	0.096	48%
Design-54	$131,000/month	0.108	54%

Design-54 may be picked due to low cost though it violates the 50% principle!

Question1: How we calculate the delay?

Question2: For high speed link, can we have high utilization?

Apply M/M/1 Formula

- **In M/M/1 queue, average response-time E(*T*) is** $\frac{1}{\mu(1-\rho)}$
- **Assuming a 1000B packet (8000 bits)**
  - Case1: T1 link = 1.536 Mbps, 50% utilization
  - Case2: OC-3 link = 135 Mbps, 80% utilization
  - Which one has lower delay?

Case1: T1 link = 1,536,000 bps, ρ = 0.5 (50%utilization)

- $1/\mu$ = service time = packetsize / transmission speed =8000 / 1536000
- $T = (1/\mu) / (1-\rho) = (1/(1-\rho)) * (1/\mu) = (1/(1-0.5)) * 8000/1.536M = 10.4$ ms.

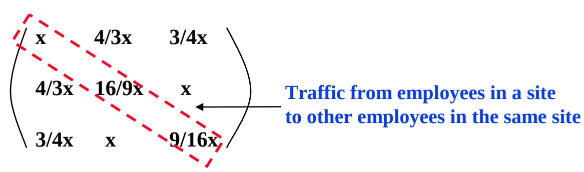
Case2: OC-3 link = 135Mbps, ρ=0.8 (80% utilization)

- $T = (1/(1-0.8)) * (8000/135M) = 5 * (8000/135M) = 2.96$ms

- ***We may be willing tolerate a higher utilization on the high-speed links.***

Calculating Internal Email Traffic

- **Internal Email: related to the populations of source and destination sites. The ration of populations among Anagon, Bregen, and Charmes is (96, 128, 72) = (1, 4/3, ¾)**
- **Total busy-hour internal email: 10*0.2*60,000*8*296/3600(s)=78933 bps**
- **Let x be the volume of internal email from Analog to itself**
  - **Then the traffic from Anagon to Bregen is 4/3 x**
  - **The traffic from Anagon to Charmes is ¾ x**



- **Counting all directional internal email traffic**
  - **9.507x = 78933bps → x=8303bps**

Tabular Representation of Internal Email Traffic

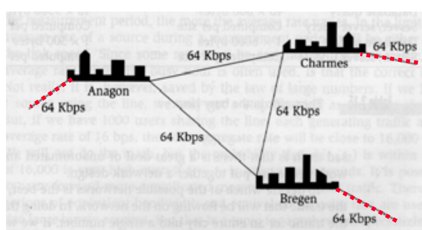
```
%TABLE TRAFFIC
 SOURCE++++++ DEST++++++ BANDWIDTH  COMMENT++++++++
 Anagon       Bregen     11070       internal email
 Bregen       Anagon     11070       internal email
 Anagon       Charmes     6227       internal email
 Charmes      Anagon      6227       internal email
 Bregen       Charmes     8303       internal email
 Charmes      Bregen      8303       internal email
```

Inter-site internal email

Anagon	Anagon	8303	in-site internal email
Bregen	Bregen	14760	in-site internal email
Charmes	Charmes	4670	in-site internal email

Intra-site internal email

Calculating External Email Traffic



- **In the initial design, each site has its own Internet connection. Therefore, the external emails do not go through the inter-site internal network.**
- **Internet links are expensive: first targets to remove; then external emails could go over the inter-site network.**
- **External email traffic:**
  - **With 4000 emails/day, 12000 B/email each employee receives 4000 * 12000 * 8 / (3600 * 8hr * 296) = 45.045bps, and sends same 45.045bps external emails**
- **Multiply the population in each site we get the following external traffic table.**

Tabular Representation of External Email Traffic

- Anagon: 96 * 45.045 = 4324.32 bps external email traffic
- Anagon: 128 * 45.045 = 5765 bps external email traffic
- Anagon: 72 * 45.045 = 3243 bps external email traffic

%TABLE TRAFFIC

SOURCE++++++	DEST++++++	BANDWIDTH	COMMENT++++++++
Anagon	GateA	4324	external email
GateA	Anagon	4324	external email
Bregen	GateB	5765	external email
GateB	Bregen	5765	external email
Charmes	GateC	3243	external email
GateC	Charmes	3243	external email

Busy Hour WWW Traffic

- Outbound small requests traffic in the busy hour:
  - 40fetch/day * 0.2 * 6req/fetch * 128B/req *8b/B / (3600s) = 13.653bps
- Inbound big WWW document and response traffic:
  - 40response/day *0.2 * (6x128 + 2000) * 8b/B / (3600s) = 49.209bps
- For Anagon,
  - Outbound WWW traffic: 13.653bps * 96 = 1310.72bps
  - Inbound WWW traffic: 49.209bps * 96 = 4724.05bps

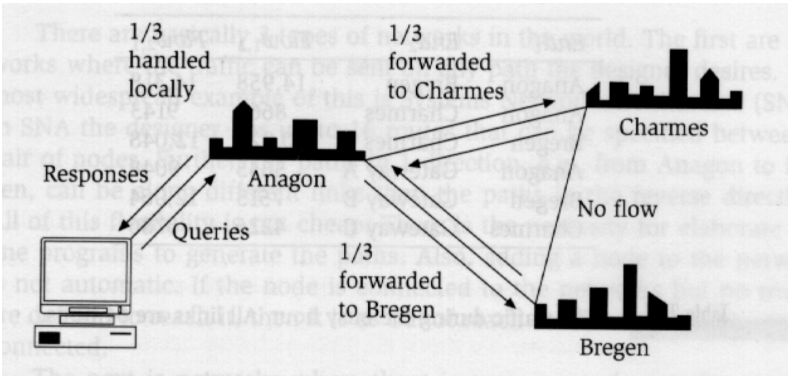
%TABLE TRAFFIC

SOURCE++++++	DEST++++++	BANDWIDTH	COMMENT++++++++	
Anagon	GateA	1311	WWW	Outbound traffic
GateA	Anagon	4702	WWW	Inbound traffic
Bregen	GateB	1748	WWW	Outbound traffic
GateB	Bregen	6298	WWW	Inbound traffic
Charmes	GateC	983	WWW	Outbound traffic
GateC	Charmes	3543	WWW	Inbound traffic

Tabular Representation of WWW Traffic

DB Query Flow

- Assume a query can be answered by a single remote server
  - The distribution of files on 3 sites is known *in priori*
  - On average, 50 queries from each employee evenly go to 3 sites



DB Traffic in Busy Hour (20%)

DB Query Traffic (800B outbound and 3500B inbound):

- 1/3 queries to each remote server:
  50 * 0.2 * 800B * 8b/B * (1/3) / 3600s = 5.930 bps
- Their requests come back from each remote server:
  50 * 0.2 * 3500B * 8b/B * (1/3) / 3600 = 25.926bps

DB Update Traffic (6000B outbound and 500B inbound):

- 1/3 updates to each remote server
  - 5 * 0.2 * 6000B * 8b/B * (1/3) / 3600 = 4.444 bps
- 1/3 updates responses back from each remote server
  - 5 * 0.2 * 500B * 8b/B * (1/3) / 3600 = 0.370 bps

DB Traffic From Anagon to Bregen:

- Consider just DB Queries (in the textbook) !
  - 96 * 5.930 + 128 * 25.926 = 3887.8
- Consider all DB traffic: The update traffic should not be ignored
  - 96 * (5.930 + 4.444) + 128 * (25.926+0.370) = 4357.568

DB Traffic Table

```
%TABLE TRAFFIC
 SOURCE++++++ DEST++++++ BANDWIDTH  COMMENT++++++++
 Anagon       Bregen      3887       DB
 Bregen       Anagon      3402       DB
 Anagon       Charmes     2436       DB
 Charmes      Anagon      2915       DB
 Bregen       Charmes     2625       DB
 Charmes      Bregen      3745       DB
```

- **DB Traffic From Anagon to Bregen: 96 * 5.930 + 128 * 25.926 = 3887**
- **DB Traffic From Bregen to Anagon: 128 * 5.930 + 96 * 25.926 = 3248?**
- **DB Traffic From Anagon to Charmes: 96 * 5.930 + 72 * 25.926 = 2436**
- **DB Traffic From Charmes to Anagon: 72 * 5.930 + 96 * 25.926 = 2915**
- **DB Traffic From Bregen to Charmes: 128 * 5.930 + 72 * 25.926 = 2625**
- **DB Traffic From Charmes to Bregen: 72 * 5.930 + 128 * 25.926 = 3745**

Busy Hour Traffic (64kbps links)

End_1	End_2	$Flow_{1,2}$	$Flow_{2,1}$
Anagon	Bregen	14,958	14,318
Anagon	Charmes	8663	9143
Bregen	Charmes	10,928	12,048
Anagon	Gateway A	5635	9048
Bregen	Gateway B	7513	12,064
Charmes	Gateway C	4226	6786

The traffic during the busy hour. All links are 64 Kbps.

Link Utilization in Busy Hour

TABLE TRAFFIC							
SOURCE	DEST	BW	COMMENT	BW	COMMEN	SUM	UTILIZATION
Anagon	***Bregen***	***11070***	***internal emai***	***3887***	***DB***	***14957***	***0.2337***
Bregen	Anagon	11070	internal email	3248	DB	***14318***	0.2237
Anagon	Charmes	6227	internal email	2436	DB	8663	0.1354
Charmes	Anagon	6227	internal email	2915	DB	9142	0.1428
Bregen	Charmes	8303	internal email	2625	DB	10928	0.1708
Charmes	Bregen	8303	internal email	3745	DB	12048	0.1883
Anagon	GateA	4324	external email	1311	WWW	5635	0.0880
GateA	Anagon	4324	external email	4702	WWW	***9026***	0.1410
Bregen	GateB	5765	external email	1748	WWW	7513	0.1174
GateB	Bregen	5765	external email	6298	WWW	12063	0.1885
Charmes	***GateC***	***3243***	***external emai***	***983***	***WWW***	***4226***	***0.0660***
GateC	Charmes	3243	external email	3543	WWW	6786	0.1060

- Highest utilization is at the link from Anagon to Bregen: 23.4%
- Lowest utilization is at the link from Charmes to Gateway C: 6.6%

Drop Algorithm for Network Design

- **Drop algorithm:**
  - **Drop the lightest utilized component / link in the network**
  - **Calculate the new routes for all traffic that use the dropped component**
- **But do we really have control over the routing in the network?**
- **We will examine 3 types of routing:**
  - **SNA (IBM System Network Architecture)** **tight control**
  - **OPSF (Open Shortest Path First)** **some control**
  - **RIP (Routing Information Protocol)** **no control**

Routing in SNA

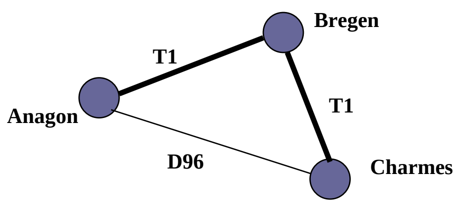
- On IBM SNA (System Network Architecture)
  - designer has up to 16 routes that can be specified between a pair of nodes. The paths are directional. The return/reverse path of a route can go through different links.
- Advantage:
  - Flexible
  - A lot of control
- Disadvantage:
  - adding a node is not automatic, required offline programs to generate the paths

OPSF Routing

- Assign each link a length (or weight) in each direction
- Routes are calculated using shortest-path algorithms
- Traffic are directed to the next link along the shortest path
- Two routes between a pair of nodes.
  - compared to max. of 16 for SNA
- Weight can be measured as delay on the directional link.
- Link weights can be broadcast periodically and routing table recalculated.

Routing Information Protocol (RIP)

- A minimum-hop protocol: use hop count instead of accumulated link weight for compute the route.
- Does not consider the bandwidth of each link.
- For 1000-byte packet,
  - 1: a two hop path with T1 link has (1000 * 8b / 1.535Mbps) * 2 = 10.42ms
  - 2: a single hop path with 9.6kbps link has 1000 * 8b / 9600bps = 833ms >> 10.42ms
  - With RIP, scenario 1 is selected due to the minimum-hop protocol



Assumptions for the Drop Algorithm

- Assume we can use shortest path routing within the corp. domain
- All three inter-site links have a length of 10
- The distance to all external domains is the same through all three gateways.
- *We try to reduce cost by removing links and see if remaining network remain feasible*

The Drop Algorithm

1. **Initially, mark all links as being deletable.**
2. **Find the most expensive deletable link. If there is a tie, take the link with the lowest utilization. We call this the candidate link for deletion**
3. **If such link exists, delete the link and see if the remaining network is feasible (can carry the traffic).**
   - **If it is feasible, go back to step 2.**
   - **If not feasible, mark the link "not being deletable" and loop back to step 2.**
4. **If such link does not exist, terminate.**

Modified Drop Algorithm Code

```
drop_algorithm(design_name) {

  Read in the design;

  Mark all Links DELETABLE;  /* All links can be candidates */

  while (some link is DELETABLE) {
     link=select_candidate(design);
     Delete the link from the design;
     Redistribute the flow on the remaining links;
     Resize the links;
     if (Resized network is cheaper) {
        Mark all links DELETABLE;
        continue;
     } else {
        restore the network by adding the link back;
        mark the link UNDELETABLE;
     }  /* endif */
  }  /*endwhile*/

  Write out the design;
} /* end drop_algorithm */
```

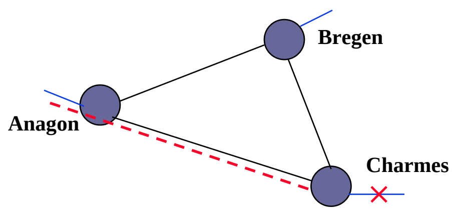
Consider increase other link's capacity

We can always make a network feasible by possibly upgrading other links to more capacity.

Apply Drop Algorithm on Initial Design

Round 1.

- **Step2. Among 3 external links, choose Charmes to gateC.**
- **Step3'. Redirect traffic to Gateway A (with less traffic)**
  by reducing the length btw Anagon and Charmes to 9 (shortest-path)
- **GateC → Charmes traffic go over a path GateA→Anagon→Charmes**
  - Traffic: External Email 3243bps + WWW 3543bps = 6786 bps
- **Charmes→GateC traffic go over Charmes→Anagon→GateA**
  - Traffic: External Email 3243bps + WWW 983bps = 4226 bps
- **The new traffic flow is shown next page**



Traffic Flow After Removing Link to GateC

8663 + 3243 + 3543=15449 9142 + 3243 + 983 = 13368

TABLE TRAFFIC

SOURCE	DEST	BW	COMMENT	BW	COMMENT	SUM	UTILIZATION
Anagon	Bregen	11070	internal email	3887	DB	14957	0.2337
Bregen	Anagon	11070	internal email	3402	DB	***14472***	0.2261
Anagon	Charmes	6227	internal email	2436	DB	15449	0.2414
Charmes	Anagon	6227	internal email	2915	DB	13368	0.2089
Bregen	Charmes	8303	internal email	2625	DB	10928	0.1708
Charmes	Bregen	8303	internal email	3745	DB	12048	0.1883
Anagon	GateA	**7567**	external email	2294	WWW	**9861**	0.1541
GateA	Anagon	**7567**	external email	8245	WWW	***15812***	0.2471
Bregen	GateB	5765	external email	1748	WWW	7513	0.1174
GateB	Bregen	5765	external email	6298	WWW	12063	0.1885
Charmes	GateC	3243	external email	983	WWW	4226	0.0660
GateC	Charmes	3243	external email	3543	WWW	6786	0.1060

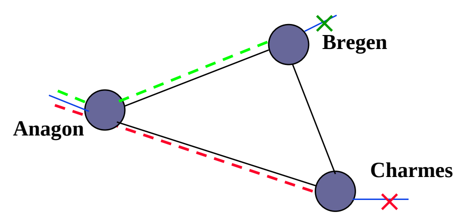
Details of Table 2.18 **9026 + 3243 + 3543 = 15812 5635 + 3243 + 983 =9861**

All link utilizations < 0.5; cost saving=$1400

Apply Drop Algorithm on Initial Design

Round 2.

- **Step2. Among 2 external links, choose Bregen to gateB since it has less traffic now.**
- **Step3'. Redirect traffic to Gateway A (with less traffic)**
- **GateB→Bregen traffic go over GateA→Anagon→Bregen**
  - Traffic: External Email 5765 bps + WWW 6298 bps = 12063 bps
- **Bregen→GateB traffic go over Bregen→Anagon→GateA**
  - Traffic: External Email 5765 bps + WWW 1748 bps = 7513 bps
- **The new traffic flow is shown next page.**



Traffic Flow After Removing Link To GateB

14957 + 5765 + 6298 = 27020   14472 + 5765 + 1748 = 21985

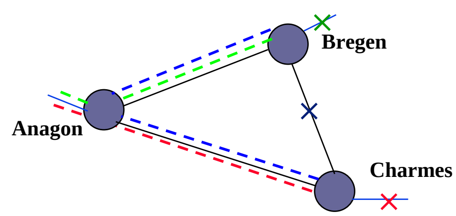
TABLE TRAFFIC							
SOURCE	DEST	BW	COMMENT	BW	COMMENT	SUM	UTILIZATION
Anagon	Bregen	11070	internal email	3887	DB	27020	0.4222
Bregen	Anagon	11070	internal email	3402	DB	*21985*	0.3435
Anagon	Charmes	6227	internal email	2436	DB	15449	0.2414
Charmes	Anagon	6227	internal email	2915	DB	13368	0.2089
Bregen	Charmes	8303	internal email	2625	DB	10928	0.1708
Charmes	Bregen	8303	internal email	3745	DB	12048	0.1883
Anagon	GateA	**13332**	external email	4042	WWW	**17374**	0.2715
GateA	Anagon	**13332**	external email	14543	WWW	*27875*	0.4355
Bregen	GateB	5765	external email	1748	WWW	7513	0.1174
GateB	Bregen	5765	external email	6298	WWW	12063	0.1885
Charmes	GateC	3243	external email	983	WWW	4226	0.0660
GateC	Charmes	3243	external email	3543	WWW	6786	0.1060

Details of Table 2.19   15812 + 5765 + 6298 = 27875   9861 + 5765 + 1748 = 17374

All link utilizations < 0.5; cost saving another $1400

Round 3 & Round 4

- Round 3: Try to delete link to gateA and find it undeletable.
- Round 4: Among the remaining 3 inter-site links, Bregen←→Charmes has less utilization (add both directional traffic)
- Redirect Bregen←→Charmes traffic around Anagon
- The new traffic flow is shown next page



Traffic After Removing Link btw Bregen and Charmes

TABLE TRAFFIC							
SOURCE	DEST	BW	COMMENT	BW	COMMENT	SUM	UTILIZATION
Anagon	Bregen	11070	internal email	3887	DB	39068	0.6104
Bregen	Anagon	11070	internal email	3402	DB	32913	0.5143
Anagon	Charmes	6227	internal email	2436	DB	26377	0.4121
Charmes	Anagon	6227	internal email	2915	DB	25416	0.3971
Bregen	Charmes	8303	internal email	2625	DB	10928	0.1708
Charmes	Bregen	8303	internal email	3745	DB	12048	0.1883
Anagon	GateA	**13332**	external email	4042	WWW	**17374**	0.2715
GateA	Anagon	**13332**	external email	14543	WWW	***27875***	0.4355
Bregen	GateB	5765	external email	1748	WWW	7513	0.1174
GateB	Bregen	5765	external email	6298	WWW	12063	0.1885
Charmes	GateC	3243	external email	983	WWW	4226	0.0660
GateC	Charmes	3243	external email	3543	WWW	6786	0.1060

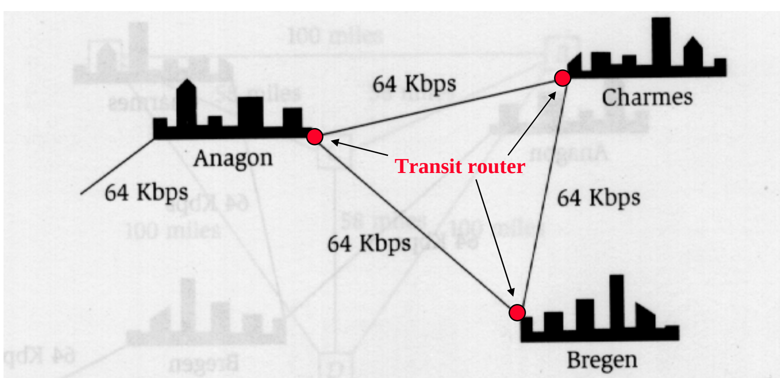
- Utilization between Anagon and Bregen high, need add link?

Rounds 4, 5, 6

- After removing link btw Charmes and Bregen, we need to add capacity to Anagon and Bregen
  - $700 saving by deleting the link
  - $700 by promoting the capacity
  - → no cost saving.
- We also lose *alternative route* (less reliability)
- Decide not to remove.
- Same results for link btw Anagon and Charmes, and link between Anagon to Bregen.
- Algorithm terminates.

Drop Algorithm Result

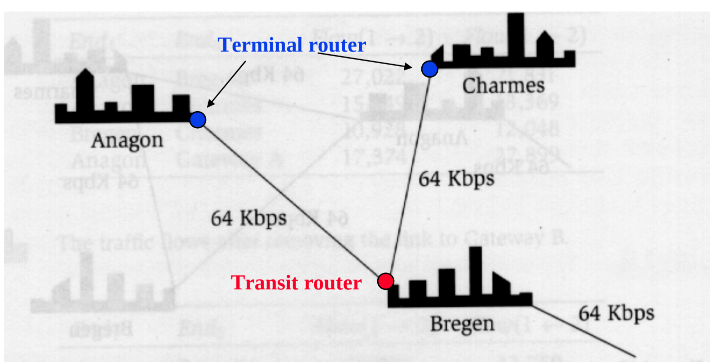
- Utilize the internal network to move the external traffic to the Internet gateway.
- 2 Internet links removed→ cost saving $2800/month
  - Final cost: $6633 – $2800 = $3833



Where the Drop Algorithm Went Wrong?

- It chooses Anagon as the Internet gateway instead of Bregen, which originally has most traffic and largest population (Anagon was selected because its traffic was accumulated because of the deletion of Gateway C, and therefore greater than Bregen's traffic later)
  - This force more traffic onto longer paths
- Lesson: Heuristic algorithms often make mistakes.
- If we choose to locate gateway at Bregen, we could remove link btw Anagon and Charmes:
  - Save $700/month
  - Save $102/month by placing terminal routers (instead of transit routers) at Anagon and Charmes
    - 2 * ($ 3700 - $2000) * 3% = $102 saving/month
    - *This saving could be realized by the previous algorithm too?*
  - Final cost: $3833 - $700 - $102 = 3031/month.

The Optimal Design



Definition 2.3: A benign algorithm is one that does no damage to a design. It only improve it or leave it alone.

The drop algorithm is not optimal but is it benign?