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
    \[4 \times 5 \times 15 = 300 \text{ min/day}\]
  - call others in the same office 10 times/day about joint work, each
    last avg. 3 minutes
    \[10 \times 3 \times 15 = 450 \text{ min/day}\]

  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)

- \( P_k = \frac{E^k}{k!} P_0 \)
- \( A_P = kD_P \rightarrow P_k = \frac{E}{k} \cdot P_{k-1} \)
- \( P_k = \frac{E}{k} \cdot P_{k-1} \)
- Probability of loss = \( P_m = \frac{E^m}{m!} P_0 = \frac{E^m}{m!} \cdot \sum_{k=0}^{m} \frac{E^k}{k!} \) = \( B(E, m) \)
- \( B(E, k) = \frac{EB(E, k-1)}{EB(E, k-1) + k} \) for \( k = 1, 2, \ldots, m \)

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.

<table>
<thead>
<tr>
<th>Site</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anagon</td>
<td>96</td>
</tr>
<tr>
<td>Bregen</td>
<td>128</td>
</tr>
<tr>
<td>Charmes</td>
<td>72</td>
</tr>
</tbody>
</table>

Voice Traffic vs. Data Traffic

<table>
<thead>
<tr>
<th></th>
<th>Voice traffic</th>
<th>Data traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed bandwidth</td>
<td>Varying bandwidth</td>
<td></td>
</tr>
<tr>
<td>Short duration calls</td>
<td>Long duration calls</td>
<td></td>
</tr>
<tr>
<td>One connection/person</td>
<td>Many connections/person</td>
<td></td>
</tr>
<tr>
<td>Extreme delay sensitivity</td>
<td>Varying sensitivity</td>
<td></td>
</tr>
<tr>
<td>Tolerates some loss</td>
<td>Varies</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Traffic</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal email</td>
<td>10 pieces sent and received per employee/day</td>
</tr>
<tr>
<td>External email</td>
<td>4000 pieces/day arriving at a steady rate</td>
</tr>
<tr>
<td>WWW</td>
<td>40 fetches/day/user</td>
</tr>
<tr>
<td>Database</td>
<td>50 queries + 5 updates per employee/day</td>
</tr>
</tbody>
</table>

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?

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal router</td>
<td>$2000 purchase price</td>
</tr>
<tr>
<td>Transit router</td>
<td>$3700 purchase price</td>
</tr>
<tr>
<td>WAN adapter</td>
<td>$500 purchase price</td>
</tr>
<tr>
<td>64,000 bps internode link</td>
<td>$700/month</td>
</tr>
<tr>
<td>256,000 bps internode link</td>
<td>$1400/month</td>
</tr>
<tr>
<td>2,048,000 bps internode link</td>
<td>$3800/month</td>
</tr>
<tr>
<td>64,000 bps internet link</td>
<td>$1400/month</td>
</tr>
<tr>
<td>256,000 bps internet link</td>
<td>$2800/month</td>
</tr>
<tr>
<td>2,048,000 bps internet link</td>
<td>$7600/month</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>14,400 bps</td>
<td>A cheap V.32 bis modem.</td>
</tr>
<tr>
<td>28,800 bps</td>
<td>A more expensive V.34 modem.</td>
</tr>
<tr>
<td>56,000 bps</td>
<td>A digital circuit in the United States. Denoted D56.</td>
</tr>
<tr>
<td>64,000 bps</td>
<td>A digital circuit in Europe and Asia. Denoted D64.</td>
</tr>
</tbody>
</table>

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:** Propagation delay for 1000 mile ring = \( \frac{1000}{186,000} = 5.376 \text{ ms} \)
- **B:** Transmission delay for 1000 bit packet at 16Mbps
  - \( \frac{1000}{16,000,000} = 0.0625\text{ms} \ll 5.376 \text{ ms} \)
- 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 \( \lambda \) (in voice network, denoted by \( A \)):
  - The number of packets coming from the client (incoming link) in a unit time
- Service rate \( \mu \) (in voice network, denoted by \( D \)):
  - The number of packets forwarded by the server (outgoing link) in unit time
- Utilization \( \rho = \frac{\lambda}{\mu} \) (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 \( \neq \) 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!
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 ($t_0, t_1-t_0, t_2-t_1, ..., t_9-t_8$)
  - Packet length / Service time distributions ($l_0, l_1, l_2, l_3, ..., l_9$)
  - 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

\[ E(T_w) = \frac{1}{2\lambda} \frac{(\lambda b)^2}{(1 - \lambda b)} = \frac{1}{2\lambda} \frac{\rho^2}{(1 - \rho)} \]

\[ E(T) = T_w + T_s = \frac{1}{2\lambda} \frac{(\lambda b)^2}{(1 - \lambda b)} + b = \frac{1}{2\lambda} \frac{\rho^2}{(1 - \rho)} + b \]
**M/M/1 Queue – A General Queue**

Poisson arrival process, exponential service time, single server  

$p_k = \text{Prob}[k \text{ customers in the system}]$  

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.

![State Transition Diagram](image)

At equilibrium state the following equations hold:

\[ \lambda p_k = \mu p_{k+1} \quad k = 0, 1, 2, \ldots \]  
\[ \lambda + \mu = \lambda + \mu \]  
\[ \rho = \frac{\lambda}{\mu} \]

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

\[ \sum_{k=0}^{\infty} p_k = \rho \sum_{k=0}^{\infty} \rho^k = 1 \Rightarrow p_0 = 1 - \rho \]

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-$\rho$, a message arrives to an empty system and no wait.  

With probability $(1-\rho)\rho$, 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, $T_s$.

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

Therefore the average waiting time

\[ T_w = \sum_{k=0}^{\infty} p_k (k T_s) = T_s \sum_{k=0}^{\infty} k (1-\rho)^k = T_s \frac{\rho}{1-\rho} \]

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 $N = \lambda T = \lambda T_s \left(1 + \frac{1}{1-\rho}\right) = \frac{\lambda}{1-\rho}$

$N=\lambda T$ is called Little’s Law, hold true for a lot of queues.

Average queueing-delay: $E(T_w) = \frac{\rho T_s}{1-\rho}$  

Average response time ($T=T_w+T_s$) = $\frac{1}{\mu(1-\rho)}$
**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
    \[ f(x) = k\alpha x^{-\alpha - 1}, \quad x \geq k, \quad k > 0, \quad \alpha > 0 \]
  - In practice, there is some upper bound on the maximum size of a job (p) -- **Bounded Pareto distribution**
    \[ f(x) = \frac{1}{1 - (k/p)\alpha} x^{-\alpha - 1}, \quad x \geq k, \quad p \geq x \geq k, \quad k > 0, \quad \alpha > 0 \]

---

**Total Delay (50ms average service time)**

- 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

The M/M/2 queue model is a two-server queue model where customers arrive according to a Poisson process with rate $\lambda$, and are served by one of two servers with service rates $\mu$. The buffer capacity $T_b$ and the service time $T_s$ are constant.

![Diagram of M/M/2 queue](image)

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,

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

### Initial Data Network Design & Cost

- **D64 Internet link**
  - $1400/month

- **D64 site-site link**
  - $700/month

- A transit router at each site
  - Amortized cost: $3700*0.03 = $111/month

- **Cost Summary**

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 transit routers</td>
<td>$333/month</td>
</tr>
<tr>
<td>3 site-site D64 links</td>
<td>$2100/month</td>
</tr>
<tr>
<td>3 internet D64 links</td>
<td>$4200/month</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$6633/month</strong></td>
</tr>
</tbody>
</table>
Traffic in Busy Hour

- 20% = 0.2 traffic in busy hour.

<table>
<thead>
<tr>
<th>Traffic type</th>
<th>Total out per employee</th>
<th>Total in per employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal email</td>
<td>2 × 60,000 bytes</td>
<td>2 × 60,000 bytes</td>
</tr>
<tr>
<td>External email</td>
<td>4000 × 0.2 × 12,000/296</td>
<td>4000 × 0.2 × 12,000/296</td>
</tr>
<tr>
<td>WWW</td>
<td>8 × 6 × 128 bytes</td>
<td>8 × (6 × 128 + 2000) bytes</td>
</tr>
<tr>
<td>Database query</td>
<td>10 × 500 bytes</td>
<td>10 × 3500 bytes</td>
</tr>
<tr>
<td>Server/server query</td>
<td>Computed per site</td>
<td>Computed per site</td>
</tr>
<tr>
<td>Database update</td>
<td>1 × 6000 bytes</td>
<td>1 × 500 bytes</td>
</tr>
<tr>
<td>Server/server update</td>
<td>Computed per site</td>
<td>Computed per site</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Cost</th>
<th>Average delay</th>
<th>Maximum utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design-48</td>
<td>$160,000/month</td>
<td>0.096</td>
<td>48%</td>
</tr>
<tr>
<td>Design-54</td>
<td>$131,000/month</td>
<td>0.108</td>
<td>54%</td>
</tr>
</tbody>
</table>

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

Question 1: How do we calculate the delay?
Question 2: For high-speed links, 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)
  - Case 1: T1 link = 1.536 Mbps, 50% utilization
  - Case 2: OC-3 link = 135 Mbps, 80% utilization
  - Which one has lower delay?

**Case 1: T1 link = 1,536,000 bps, $\rho = 0.5$ (50% utilization)**

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

**Case 2: OC-3 link = 135 Mbps, $\rho = 0.8$ (80% utilization)**

- $T = (1/(1-0.8)) * (8000/135M) = 5 * (8000/135M) = 2.96 \text{ ms.}$

- *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 ratio of populations among Anagon, Bregen, and Charmes is (96, 128, 72) = (1, 4/3, ¾)

- Total busy-hour internal email: $10*0.2*60,000*8*296/3600(s) = 78933 \text{ bps}$

- Let $x$ be the volume of internal email from Analog to itself
  - Then the traffic from Anagon to Bregen is $4/3 \times x$
  - The traffic from Anagon to Charmes is $3/4 \times x$

- Counting all directional internal email traffic
  - $9.507x = 78933 \text{ bps} \rightarrow x = 8303 \text{ bps}$
Tabular Representation of Internal Email Traffic

<table>
<thead>
<tr>
<th>TABLE TRAFFIC</th>
<th>SOURCE</th>
<th>DEST</th>
<th>BANDWIDTH</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-site internal email</td>
<td>Anagon</td>
<td>Bregen</td>
<td>11070</td>
<td>internal email</td>
</tr>
<tr>
<td></td>
<td>Bregen</td>
<td>Anagon</td>
<td>11070</td>
<td>internal email</td>
</tr>
<tr>
<td></td>
<td>Anagon</td>
<td>Charmes</td>
<td>6227</td>
<td>internal email</td>
</tr>
<tr>
<td></td>
<td>Charmes</td>
<td>Anagon</td>
<td>6227</td>
<td>internal email</td>
</tr>
<tr>
<td></td>
<td>Bregen</td>
<td>Charmes</td>
<td>8303</td>
<td>internal email</td>
</tr>
<tr>
<td></td>
<td>Charmes</td>
<td>Bregen</td>
<td>8303</td>
<td>internal email</td>
</tr>
</tbody>
</table>

Inter-site internal email

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>DEST</th>
<th>BANDWIDTH</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anagon</td>
<td>Anagon</td>
<td>8303</td>
<td>in-site internal email</td>
</tr>
<tr>
<td>Bregen</td>
<td>Bregen</td>
<td>14760</td>
<td>in-site internal email</td>
</tr>
<tr>
<td>Charmes</td>
<td>Charmes</td>
<td>4670</td>
<td>in-site internal email</td>
</tr>
</tbody>
</table>

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: $96 \times 45.045 = 4324.32$ bps external email traffic
- Anagon: $128 \times 45.045 = 5765$ bps external email traffic
- Anagon: $72 \times 45.045 = 3243$ bps external email traffic

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>DEST</th>
<th>BANDWIDTH</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anagon</td>
<td>GateA</td>
<td>4324</td>
<td>external email</td>
</tr>
<tr>
<td>GateA</td>
<td>Anagon</td>
<td>4324</td>
<td>external email</td>
</tr>
<tr>
<td>Bregen</td>
<td>GateB</td>
<td>5765</td>
<td>external email</td>
</tr>
<tr>
<td>GateB</td>
<td>Bregen</td>
<td>5765</td>
<td>external email</td>
</tr>
<tr>
<td>Charmes</td>
<td>GateC</td>
<td>3243</td>
<td>external email</td>
</tr>
<tr>
<td>GateC</td>
<td>Charmes</td>
<td>3243</td>
<td>external email</td>
</tr>
</tbody>
</table>

Tabular Representation of WWW Traffic

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

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>DEST</th>
<th>BANDWIDTH</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anagon</td>
<td>GateA</td>
<td>1311</td>
<td>WWW Outbound traffic</td>
</tr>
<tr>
<td>GateA</td>
<td>Anagon</td>
<td>4702</td>
<td>WWW Inbound traffic</td>
</tr>
<tr>
<td>Bregen</td>
<td>GateB</td>
<td>1748</td>
<td>WWW Outbound traffic</td>
</tr>
<tr>
<td>GateB</td>
<td>Bregen</td>
<td>6298</td>
<td>WWW Inbound traffic</td>
</tr>
<tr>
<td>Charmes</td>
<td>GateC</td>
<td>983</td>
<td>WWW Outbound traffic</td>
</tr>
<tr>
<td>GateC</td>
<td>Charmes</td>
<td>3543</td>
<td>WWW Inbound traffic</td>
</tr>
</tbody>
</table>
**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:
  \[ \frac{50 \times 0.2 \times 800 \times 8b/B \times (1/3)}{3600} = 5.930 \text{ bps} \]
- Their requests come back from each remote server:
  \[ \frac{50 \times 0.2 \times 3500 \times 8b/B \times (1/3)}{3600} = 25.926 \text{ bps} \]

**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>
<thead>
<tr>
<th>SOURCE</th>
<th>DEST</th>
<th>BANDWIDTH</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anagon</td>
<td>Bregen</td>
<td>3887</td>
<td>DB</td>
</tr>
<tr>
<td>Bregen</td>
<td>Anagon</td>
<td>3402</td>
<td>DB</td>
</tr>
<tr>
<td>Anagon</td>
<td>Charmes</td>
<td>2436</td>
<td>DB</td>
</tr>
<tr>
<td>Charmes</td>
<td>Anagon</td>
<td>2915</td>
<td>DB</td>
</tr>
<tr>
<td>Bregen</td>
<td>Charmes</td>
<td>2625</td>
<td>DB</td>
</tr>
<tr>
<td>Charmes</td>
<td>Bregen</td>
<td>3745</td>
<td>DB</td>
</tr>
</tbody>
</table>

- DB Traffic From Anagon to Bregen: \(96 \times 5.930 + 128 \times 25.926\) = 3887
- DB Traffic From Bregen to Anagon: \(128 \times 5.930 + 96 \times 25.926\) = 3248
- DB Traffic From Anagon to Charmes: \(96 \times 5.930 + 72 \times 25.926\) = 2436
- DB Traffic From Charmes to Anagon: \(72 \times 5.930 + 96 \times 25.926\) = 2915
- DB Traffic From Bregen to Charmes: \(128 \times 5.930 + 72 \times 25.926\) = 2625
- DB Traffic From Charmes to Bregen: \(72 \times 5.930 + 128 \times 25.926\) = 3745

**Busy Hour Traffic (64kbps links)**

<table>
<thead>
<tr>
<th>End₁</th>
<th>End₂</th>
<th>Flow₁₂</th>
<th>Flow₂₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anagon</td>
<td>Bregen</td>
<td>14,958</td>
<td>14,318</td>
</tr>
<tr>
<td>Anagon</td>
<td>Charmes</td>
<td>8663</td>
<td>9143</td>
</tr>
<tr>
<td>Bregen</td>
<td>Charmes</td>
<td>10,928</td>
<td>12,048</td>
</tr>
<tr>
<td>Anagon</td>
<td>Gateway A</td>
<td>5635</td>
<td>9048</td>
</tr>
<tr>
<td>Bregen</td>
<td>Gateway B</td>
<td>7513</td>
<td>12,064</td>
</tr>
<tr>
<td>Charmes</td>
<td>Gateway C</td>
<td>4226</td>
<td>6786</td>
</tr>
</tbody>
</table>

The traffic during the busy hour. All links are 64 Kbps.
### Link Utilization in Busy Hour

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

- 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 \((1000 \times 8b / 1.535Mbps) \times 2 = 10.42ms\)
  - 2: a single hop path with 9.6kbps link has \(1000 \times 8b / 9600bps = 833ms \gg 10.42ms\)
- 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

```c
1:  drop_algorithm(design_name) {
  2:      Read in the design;
  3:      Mark all Links DELETABLE; /* All links can be candidates */
  4:      while (some link is DELETABLE) {
  5:          link-select_candidate(design);
  6:          Delete the link from the design;
  7:          redistribute the flow on the remaining links;
  8:          if (Resized network is cheaper) {
  9:              Mark all links DELETABLE;
 10:          } continue;
 11:      } else {
 12:          restore the network by adding the link back;
 13:      } /* endif */
 14:      mark the link UNDELETABLE;
 15:  } /* endwhile*/
 16:  Write out the design;
 17:  } /* end drop_algorithm */
```

We can always make a network feasible by possibly upgrading other links to more capacity.
Apply Drop Algorithm on Initial Design

Round 1.
- Step 2. Among 3 external links, choose Charmes to gate C.
- Step 3'. Redirect traffic to Gateway A (with less traffic) by reducing the length between Anagon and Charmes to 9 (shortest-path).
- Gate C → Charmes traffic go over a path Gate A → Anagon → Charmes
  - Traffic: External Email 3243bps + WWW 3543bps = 6786 bps
- Charmes → Gate C traffic go over Charmes → Anagon → Gate A
  - Traffic: External Email 3243bps + WWW 983bps = 4226 bps
- The new traffic flow is shown next page

Traffic Flow After Removing Link to Gate C

<table>
<thead>
<tr>
<th>TABLE TRAFFIC</th>
<th>SOURCE</th>
<th>DEST</th>
<th>BW</th>
<th>COMMENT</th>
<th>BW</th>
<th>COMMENT</th>
<th>SUM</th>
<th>UTILIZATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anagon</td>
<td>Bregen</td>
<td>11070</td>
<td>internal email</td>
<td>3887</td>
<td>DB</td>
<td>14957</td>
<td>0.2337</td>
<td></td>
</tr>
<tr>
<td>Bregen</td>
<td>Anagon</td>
<td>11070</td>
<td>internal email</td>
<td>3402</td>
<td>DB</td>
<td>14472</td>
<td>0.2261</td>
<td></td>
</tr>
<tr>
<td>Anagon</td>
<td>Charmes</td>
<td>8227</td>
<td>internal email</td>
<td>2336</td>
<td>DB</td>
<td>13449</td>
<td>0.2173</td>
<td></td>
</tr>
<tr>
<td>Charmes</td>
<td>Anagon</td>
<td>8227</td>
<td>internal email</td>
<td>2915</td>
<td>DB</td>
<td>13362</td>
<td>0.2089</td>
<td></td>
</tr>
<tr>
<td>Bregen</td>
<td>Charmes</td>
<td>8303</td>
<td>internal email</td>
<td>2625</td>
<td>DB</td>
<td>10528</td>
<td>0.1706</td>
<td></td>
</tr>
<tr>
<td>Charmes</td>
<td>Bregen</td>
<td>8303</td>
<td>internal email</td>
<td>3745</td>
<td>DB</td>
<td>12048</td>
<td>0.1893</td>
<td></td>
</tr>
<tr>
<td>Anagon</td>
<td>Gate A</td>
<td>7567</td>
<td>external email</td>
<td>2294</td>
<td>WWW</td>
<td>9861</td>
<td>0.1541</td>
<td></td>
</tr>
<tr>
<td>Gate A</td>
<td>Anagon</td>
<td>7567</td>
<td>external email</td>
<td>8345</td>
<td>WWW</td>
<td>15812</td>
<td>0.2471</td>
<td></td>
</tr>
<tr>
<td>Bregen</td>
<td>Gate B</td>
<td>5765</td>
<td>external email</td>
<td>1748</td>
<td>WWW</td>
<td>7513</td>
<td>0.1174</td>
<td></td>
</tr>
<tr>
<td>Gate B</td>
<td>Bregen</td>
<td>5765</td>
<td>external email</td>
<td>6298</td>
<td>WWW</td>
<td>12063</td>
<td>0.1888</td>
<td></td>
</tr>
<tr>
<td>Charmes</td>
<td>Gate C</td>
<td>3243</td>
<td>external email</td>
<td>983</td>
<td>WWW</td>
<td>4226</td>
<td>0.0660</td>
<td></td>
</tr>
<tr>
<td>Gate C</td>
<td>Charmes</td>
<td>3243</td>
<td>external email</td>
<td>3543</td>
<td>WWW</td>
<td>6786</td>
<td>0.1060</td>
<td></td>
</tr>
</tbody>
</table>

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

Details of Table 2.18

All link utilizations < 0.5; cost saving=$1400
Apply Drop Algorithm on Initial Design

Round 2.

° Step 2. Among 2 external links, choose Bregen to gateB since it has less traffic now.

° Step 3’. 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

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>DEST</th>
<th>BW</th>
<th>COMMENT</th>
<th>BW</th>
<th>COMMENT</th>
<th>SUM</th>
<th>UTILIZATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anagon</td>
<td>Bregen</td>
<td>11070</td>
<td>internal email</td>
<td>4042</td>
<td>WWW</td>
<td>15112</td>
<td>0.2715</td>
</tr>
<tr>
<td>Bregen</td>
<td>Anagon</td>
<td>11070</td>
<td>internal email</td>
<td>3402</td>
<td>DB</td>
<td>14472</td>
<td>0.3435</td>
</tr>
<tr>
<td>Bregen</td>
<td>Charmes</td>
<td>8222</td>
<td>internal email</td>
<td>2915</td>
<td>DB</td>
<td>11137</td>
<td>0.2985</td>
</tr>
<tr>
<td>Charmes</td>
<td>Anagon</td>
<td>6227</td>
<td>internal email</td>
<td>2915</td>
<td>DB</td>
<td>9142</td>
<td>0.2085</td>
</tr>
<tr>
<td>Charmes</td>
<td>Bregen</td>
<td>6227</td>
<td>internal email</td>
<td>2915</td>
<td>DB</td>
<td>9142</td>
<td>0.2085</td>
</tr>
<tr>
<td>Bregen</td>
<td>GateA</td>
<td>5765</td>
<td>external email</td>
<td>1748</td>
<td>WWW</td>
<td>7513</td>
<td>0.1174</td>
</tr>
<tr>
<td>GateB</td>
<td>Bregen</td>
<td>5765</td>
<td>external email</td>
<td>6298</td>
<td>WWW</td>
<td>12063</td>
<td>0.1885</td>
</tr>
<tr>
<td>Charmes</td>
<td>GateC</td>
<td>3243</td>
<td>external email</td>
<td>863</td>
<td>WWW</td>
<td>4106</td>
<td>0.0860</td>
</tr>
<tr>
<td>GateC</td>
<td>Charmes</td>
<td>3243</td>
<td>external email</td>
<td>3543</td>
<td>WWW</td>
<td>6786</td>
<td>0.1060</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>SOURCE DEST</th>
<th>BW (KB)</th>
<th>COMMENT</th>
<th>BW (KB)</th>
<th>COMMENT</th>
<th>SUM (KB)</th>
<th>UTILIZATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anagon Bregen</td>
<td>11070</td>
<td>internal email</td>
<td>3887</td>
<td>DB</td>
<td>39068</td>
<td>0.6104</td>
</tr>
<tr>
<td>Bregen Anagon</td>
<td>11070</td>
<td>internal email</td>
<td>3402</td>
<td>DB</td>
<td>37435</td>
<td>0.5143</td>
</tr>
<tr>
<td>Anagon Charmes</td>
<td>6227</td>
<td>internal email</td>
<td>2436</td>
<td>DB</td>
<td>26763</td>
<td>0.4121</td>
</tr>
<tr>
<td>Charmes Anagon</td>
<td>6227</td>
<td>internal email</td>
<td>2915</td>
<td>DB</td>
<td>29155</td>
<td>0.3971</td>
</tr>
<tr>
<td>Bregen Charmes</td>
<td>8303</td>
<td>internal email</td>
<td>2625</td>
<td>DB</td>
<td>10928</td>
<td>0.1708</td>
</tr>
<tr>
<td>Charmes Bregen</td>
<td>8303</td>
<td>internal email</td>
<td>3745</td>
<td>DB</td>
<td>12048</td>
<td>0.1883</td>
</tr>
<tr>
<td>Anagon GateA</td>
<td>13332</td>
<td>external email</td>
<td>4042</td>
<td>WWW</td>
<td>17374</td>
<td>0.2715</td>
</tr>
<tr>
<td>GateA Anagon</td>
<td>13332</td>
<td>external email</td>
<td>14543</td>
<td>WWW</td>
<td>27875</td>
<td>0.4355</td>
</tr>
<tr>
<td>Bregen GateB</td>
<td>5765</td>
<td>external email</td>
<td>1748</td>
<td>WWW</td>
<td>7513</td>
<td>0.1174</td>
</tr>
<tr>
<td>GateB Bregen</td>
<td>5765</td>
<td>external email</td>
<td>6298</td>
<td>WWW</td>
<td>12063</td>
<td>0.1885</td>
</tr>
<tr>
<td>Charmes GateC</td>
<td>3243</td>
<td>external email</td>
<td>983</td>
<td>WWW</td>
<td>4226</td>
<td>0.0660</td>
</tr>
<tr>
<td>GateC Charmes</td>
<td>3243</td>
<td>external email</td>
<td>3543</td>
<td>WWW</td>
<td>6786</td>
<td>0.1060</td>
</tr>
</tbody>
</table>

° 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 forces more traffic onto longer paths.

- Lesson: Heuristic algorithms often make mistakes.

- If we choose to locate gateway at Bregen, we could remove link between 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 improves it or leaves it alone.

The drop algorithm is not optimal but is it benign?