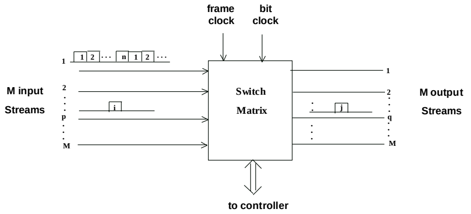
Telecom Networks and Switching:

Switch Architecture

Dr. Bhaskar Ramamurthi
Professor,
Department of Electrical Engineering,
IIT Madras, Chennai.

Multiplexed PCM Streams

- In modern digital switches, digitised, 64 kbps, PCM-encoded, voice signals to/from line cards are multiplexed into 2.048/8.192/…. Mbps (32/128/… channels) streams
  - ⇒ a PCM bus carries each such stream from/to one or more line cards to/from switch matrix



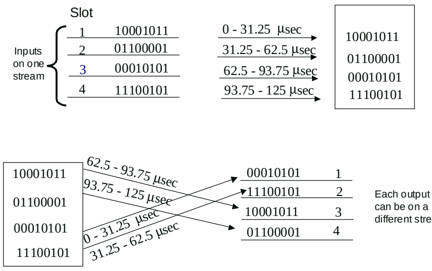
Multiplexed PCM Streams (contd.)

- each voice signal takes up a one-byte *time slot* every 125 µsec (1/8000 sec)
  - ⇒n slots per *frame* (n = 32 for 2.048 Mbps stream)
- a switch matrix with M multiplexed input/output streams supports a total of Mn *simplex* connections
  - — each voice call takes up two simplex connections
  - — each tone generator or announcement takes up one simplex connection, if switch can perform one-to-many switching
  - ⇒e.g., feed dial tone simultaneously to several lines
  - — each DTMF detector likewise taken up one simplex connection
- a pair of input and output slots (same slot on some line) is a *port*
  - — each telephone line/trunk line is connected to a port

Time Switching

- Switch matrix can put a PCM signal occuring in slot ion any *input stream* p into slot j on any *output* stream q)
  - — to complete a duplex connection, slot j on *input* stream q must also be fed into slot i on *output* stream p
    - ⇒ this establishes a duplex connection between port (i,p) and port (j,q)
- since PCM sample appearing in one time slot is placed into another time slot, this is called ***time switching***
  - — a time switch fundamentally requires *memory*
    - ⇒ store input samples of one frame in memory and play them out in next frame at different times (and on different streams)

Time Switching: an Example with n=4



Each output slot can be on a different stream

Size of Time Switch

- A time switch has to read Mn PCM samples into memory and write Mn samples from memory, every 125 µsec
  - — speed limit of current technology plays a role in deciding Mn
- Choice of M decides number of pins on IC, and the speed of each input / output line (64 n kHz)
  - — M = 16, n = 32 commonly available
  - — M = 32, n = 128 also used (sometimes the switch IC is custom built)
    - ⇒ to increase n, at the cost of pinout, sometimes each multiplexed stream is fed as 8-bit parallel bus

Combining Switching Modules

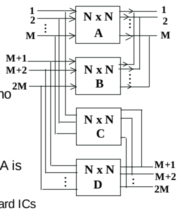
- to build a 2N x 2N matrix using N x N modules

  ⇒ 4 N x N modules

- output streams must be *tri-stated* during a slot if no connection is made to it that slot

  e.g, if in slot 1, input stream M+1 is connected to output stream 1, then output stream 1 of module A is tri-stated during slot 1

  ⇒ can build switch matrix of desired size using standard ICs



Space Switching

PCM input in *slot i* on stream p can be *can* be switched to any stream q but only in slot i
i.e., only port (i.p) → port (i,q) is possible

⇒ ***no memory used*** in switch matrix

- because switching occurs from one physical line to another, it is called ***space switching***
- with n =1, can switch any input port (≡ line) to any output port (≡ line)

  ⇒ physical connection from port (i,p) to port (i,q) exists for duration of call

- for n > 1, can ***change*** the connections between input streams and output streams ***every slot***

  — but, only some ports (those that occur in same time slot) can be connected

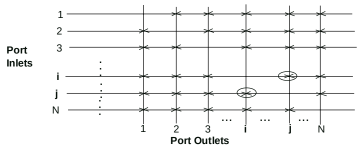
---

Analog Space Switching

- Space switching works for analog signals too

  — in fact, it is the only way to switch analog signals

- an N x N space switch needs N(N-1) ≈ $\mathbf{N^2}$ *cross points*



- cross point array has been implemented in many technologies

  — $\underbrace{\textbf{Strowger, crossbar}}_{\text{electromechanical}}$**, electronic**

Blocking Vs Non-Blocking Switches

- Treat cross point array as an abstract representation of any switch matrix (time, or space, switch)
- if any N connections can be made simultaneously in an NxN switch matrix

  ⇒ **non-blocking** switch

- if implementation is such that < N simultaneous connections can be made in some or all cases

  ⇒ **blocking** switch

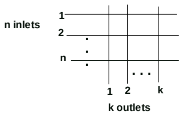
- blocking switches are implemented using multiple switching stages — **why?**

Rectangular Cross Point Array

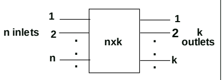
- no notion of input and output *ports*

  — only *inlets* and *outlets*

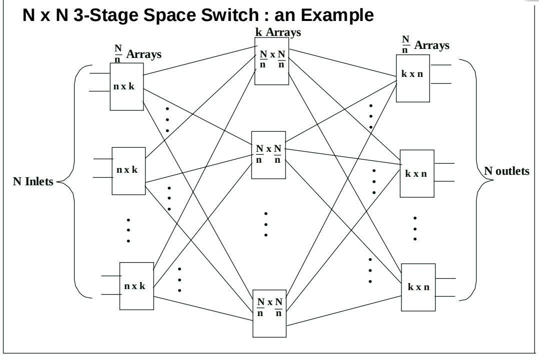
- Number of cross points = nk



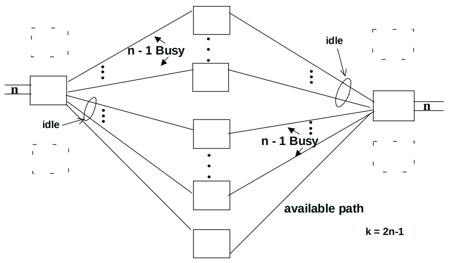
- represented as shown alongside
- this is **not** a switch matrix with n input ports and k output ports



N x N 3-Stage Space Switch : an Example



Non-Blocking 3-Stage Switch



Number of Cross-Points

- $N_c = 2N(2n-1) + (2n-1)N^2/n^2$
- Optimum choice of n for large $N = \sqrt{\frac{N}{2}}$

$\Rightarrow N^*_c \approx 4N(\sqrt{2N} - 1)$

— instead of N^2 !

Example : for N = 8000 ⇒ reduction by 16 times

Blocking Switches

- k << 2n-1
- objective : blocking probability should be **low**, i.e.,
  - ⇒ compared to probability that called party is using the phone, this should be negligible
- if line utilisation is, say, 10%, blocking prob. can be 1%
- for 8K switch, k is reduced 8 times compared to 2n-1
  - ⇒ N_c reduces by a further factor of 9 compared to non-blocking design

---

Pros and Cons of Blocking Switches

- for 8K switch,

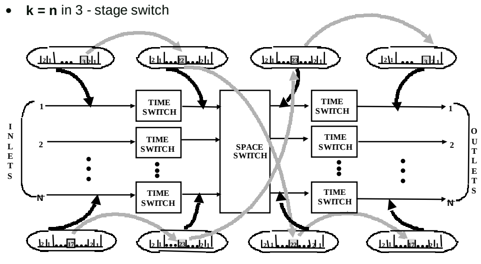
Single-stage non-blocking $\xrightarrow{/16}$ 3-stage non-blocking $\xrightarrow{/9}$ 3-stage blocking (10% util., 1% blocking)

- however, if link utilisation increases, blocking prob. goes up **many times** over

e.g., Util.1% → 1.6% ⇒ blocking prob. 1% → 16%!

Time-Space-Time Switch

- **k = n** in 3 - stage switch



- Space Switch **changes** connections in **every slot**
  - ⇒ equivalent to k (=n) space switches

Large Essentially Non - Blocking Switches

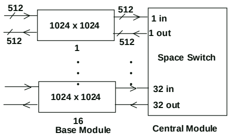
- since TST switch is equivalent to 3 - stage switch with k = n
  - ⇒ blocking switch

However,

- blocking probability very, very small for k = n ($\sim 10^{-6}$!)
  - ⇒ even if utilization doubles or triples, blocking probability may go up100 times, still performance is good
  - ⇒ large exchanges consist of TS modules interconnected in a **scalable** fashion to a space switch

Essentially Non-Blocking 3-stage Switch : an Example

- CDOT's 16,000 line exchange
- TST configuration (slightly modified)
- Time Switch with n = 512, k = 512 in first and third stages
- 32 x 32 space switch in middle stage
  - — re-configured in every slot as usual
- Base Module consists of **first and third** stages put together



Essentially Non-Blocking 3-stage Switch : an Example (contd.)

- Base module is **not** two independent 512 x 512 TSs, but one 1024 x 1024 TS
  - ⇒ calls between ports on same BM can be switched locally in BM
    - — in conventional TST switch, every call has to go through the space switch
  - ⇒ reduces blocking probability (in any case, insignificant)
- no separate path for control messages between processors in BMs and CM
  - — use some of the 64 kbps channels ⇒ slightly less than 512 slots carry voice signals to CM
- DTMF detectors, tone generators, etc., in each BM
  - ⇒ number of ports per BM ≃ 480

Telecom Networks and Switching:

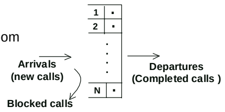
Teletraffic Modelling

Dr. Bhaskar Ramamurthi
Professor,
Department of Electrical Engineering,
IIT Madras, Chennai.

Telephone Network as a Queueing System

- Occurrence of telephone calls is **random**
- Call "arrivals" at an exchange, are akin to "customers" requesting service in a **queue**
- An exchange capable of supporting N simultaneous calls or a trunk group with N channels, is akin to N "servers" handling customers
- If a call cannot be completed, busy tone is fed
  - ⇒ no "waiting room" for "customers", i.e., **blocked calls cleared**

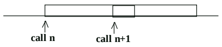
⇒ **a queue** with **N servers** and no waiting room



Call Statistics

- Model call arrivals on each line as a **Poisson process**
  - ⇒ only one parameter λ_u = avg. no. of calls / unit time
- Poisson process ≡ inter-arrival time τ_i exponentially distributed

  $(p(\tau_i) \; \lambda_u e^{-\lambda_u \tau_i})$
- Property of Poisson arrivals : total arrivals on M lines is also Poisson, with rate $M\lambda_u$
- Call holding-time also modelled as exponentially distributed
  - ⇒ only one parameter, τ_h : average holding time
- An assumption **:** λ_u , τ_h such that, in the model, probability of next call beginning on a line while previous call is still being held, is negligible (this is impossible in reality)



Teletraffic

- Unit of teletraffic is **Erlang** (in honour of a Swedish mathematician)
- traffic = $\lambda_u \tau_h$ Erlangs (dimensionless, like radians)
  - ⇒ average utilisation of a line (as per a specific statistical model)
- An important property of queues with Poisson arrivals and exponential holding times (so called M/M queues) is that **queue behaviour depends only on product $\lambda_u\tau_h$**
  - ⇒ traffic from M similar lines = M λ_u τ_h Erlangs
  - ⇒ total traffic from M_1 lines of one type (say, business), and M_2 lines of another (say, residential) = $M_1 \lambda_{u1} \tau_{h1} + M_2 \lambda_{u2} \tau_{h2}$

Blocking Probability

- For N servers and A Erlangs traffic,

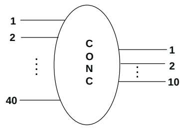
  $P_B = A^N/N! \;\; \sum_{i=0}^{N}\left(A^i/i!\right)$ **Erlang B** formula

- Tables / curves are a good tool for **dimensioning**
- Typically, we want P_B small (say 0.01)
  - — for small N, permissible traffic << N Erlangs
  - — as N becomes large (~100), permissible traffic increases and saturates slowly thereafter to a reasonably large fraction of N
- P_B very sensitive to traffic level A for given N (except for N=1)
  - ⇒ increases many times over compared to increase in A

Erlang B Model - Blocked Calls Cleared

	A in Erlangs						
	P_B (Blocking Probability)						
N	**0.1%**	**0.2%**	**0.5%**	**1.0%**	**2%**	**5%**	**10%**
5	0.762	0.900	1.13	1.36	1.66	2.22	2.88
10	3.09	3.43	3.96	4.46	5.08	6.22	7.51
15	6.08	6.58	7.38	8.11	9.01	10.6	12.5
20	9.41	10.1	11.1	12.0	13.2	15.2	17.6
25	13.0	13.8	15.0	16.1	17.5	20.0	22.8
30	16.7	17.6	19.0	20.3	21.9	24.8	28.1
40	24.4	25.6	27.4	29.0	31.0	34.6	38.8
50	32.5	33.9	36.0	37.9	40.3	44.5	49.6
60	40.8	42.4	44.8	46.9	49.6	54.6	60.4
70	49.2	51.0	53.7	56.1	59.1	64.7	71.3
80	57.8	59.7	62.7	65.4	68.7	74.8	82.2
90	66.5	68.6	71.8	74.7	78.3	85.0	93.1
100	75.2	77.5	80.9	84.1	88.0	95.2	104.1

Use of Erlang B Table : an Example



$\lambda_u \tau_h = 0.1 \quad \Rightarrow \quad A = 40 \times 0.1 = 4E$

$\Rightarrow P_B = 0.55\%$

Suppose $\lambda_u \tau_h = 0.15 \quad \Rightarrow A = 6E$

$\Rightarrow P_B \simeq 4.4\%!$

- P_B increases 8 times for 50% increase in A

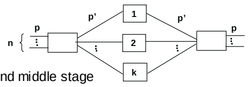
Dimensioning the PSTN Trunk Network

- Blocking in Switches solved by digital, essentially non-blocking, switches
- Choosing N correctly for each route in the trunk network is critical for controlling end-to-end call blocking probability
  - ⇒ **dimensioning** the trunk network
- N for each route chosen based on traffic statistics collected over time
  - ⇒ traffic monitoring at exchanges is an important function
- To account for rapid changes in traffic pattern, trunks usually are provided with scope for expansion
  - ⇒ dark fibre, or use of Wavelength Division Multiplexing on existing fibre, to multiply capacity

Analysis of Blocking in Multi-Stage Switches

- Difficult to do queueing analysis with Poisson call arrival model
  - — a middle stage can set up some calls but not others
    - ⇒ may not be possible to set up - connection for one call arrival on one line but may be possible for an arrival on another line
- queueing analysis for case when server is “idle” when some customers arrive (and sends them away), but not for some others, is very difficult
  - ⇒ simplified approximate analysis

Lee Graph Approximation for 3-Stage Switch



- Probability of line utilisation = p (≡ traffic in Erlangs)
- assume each link between first stage and middle stage is utilised *independently* with probability p' = np/k
- utilisation of links from middle stage to third stage, and of third stage port outputs, are likewise p' and p respectively
- probability of a *particular path* from first to third stage being *free*

  = $(1- p')^2$: *both* links *in series* must be free
- probability of *no path* from first to third stage being *free*

  = $[1- (1- p')^2]^k$: all k *parallel link-pair*s must be busy

  = P_B

Lee Graph Approximation for 3-Stage Switch (contd.)

- Lee approximation under-estimates P_B for $k < n$, but
  over-estimates P_B for $k > 2n$
  (for $k = 2n-1$, it gives $P_B > 0$!)

$\Rightarrow$ useful as a guide to narrowing design choices for blocking switches

— finally, computer simulation will give more accurate picture

- Optimise n and k to minimise N_c while meeting requirement on P_B

e.g. $p = 0.1$, $N = 8192$

$n = 32$, $k = 10$ gives minimum $N_c \simeq 500{,}000$

$n = 64$ $(= \sqrt{N/2})$, $k = 127$ $(= 2n - 1)$ is non-blocking

$\Rightarrow N_c \simeq 4.2$ million