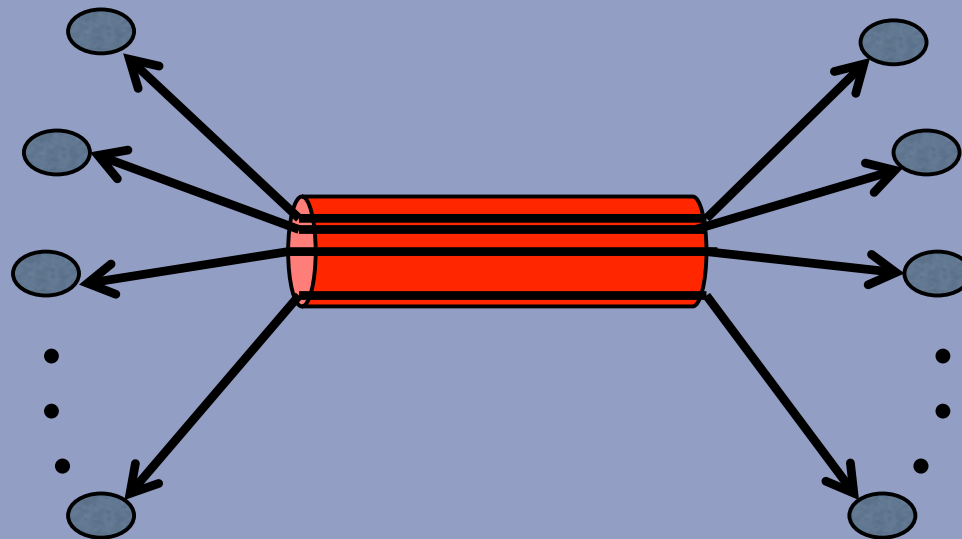


Multiplexing

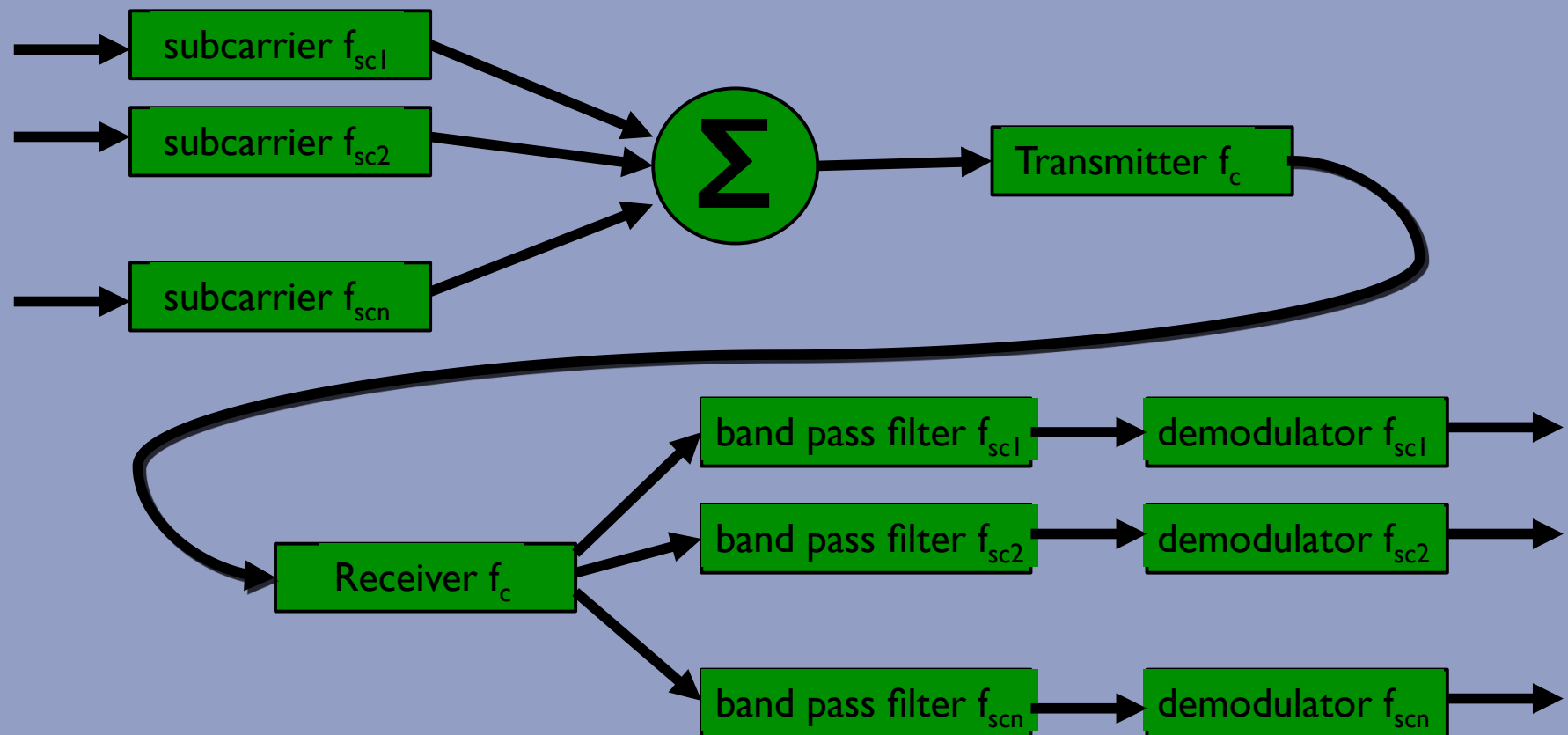
”From one channel to multiple channels”

How to share one medium while facilitating multiple channels of communication:



Frequency Division and Time Division Multiplexing

Frequency Division Multiplexing (FDM alla FSK)



Standards

# voice channel	bandwidth	spectrum	US/AT&T	ITU_T
12	48 kHz	60-108 kHz	Group	Group
60	240 kHz	312-552 kHz	Super Group	Super Group
300	1,23 MHz	812-2044 kHz	Super Group	Master Group
600	2,52 MHz	564-3084 kHz	Master Group	Master Group
900	3,87 MHz	8,52-12.39 MHz	Master Group	Super Master Group
3600	16,98 MHz	0.56-17,55 MHz	Jumbo Group	Jumbo Group
10800	57,44 MHz	3,12-60,57 MHz	Jumbo Group Multiplexed	Jumbo Group Multiplexed

Example: ADSL

ADSL Asymmetric Digital Subscriber Line

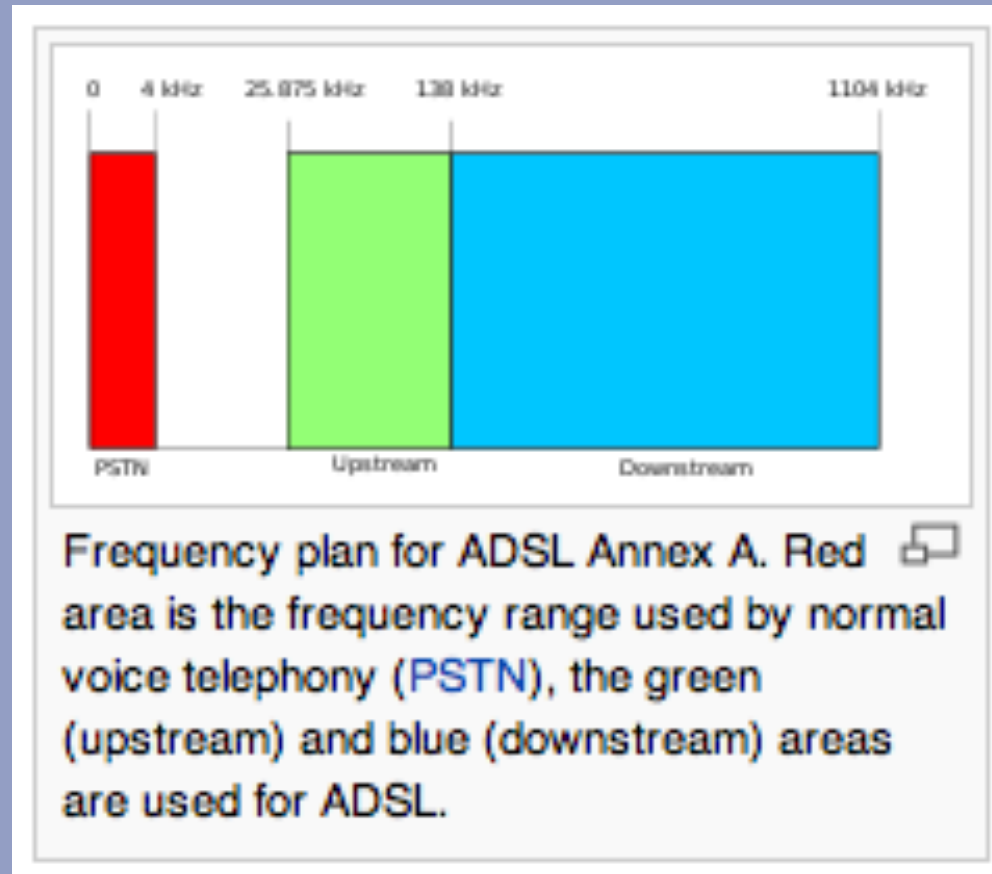
Originally for Video-on-Demand: less control
going up – lots of image going down

➔ Very similar to internet usage !!!!!!!

**Multiple “regular” phone connections at
the same time on which QAM (Quadratic
Amplitude Modulation) is implemented**

- Reserve lowest 25 kHz for Voice (POTS, Plain Old Telephone Service) **25 instead of 5 to prevent cross talk between voice&data**
- Facilitate two bands: small upstream / big downstream
- Use FDM within upstream and downstream band

ADSL 4 kHz channels

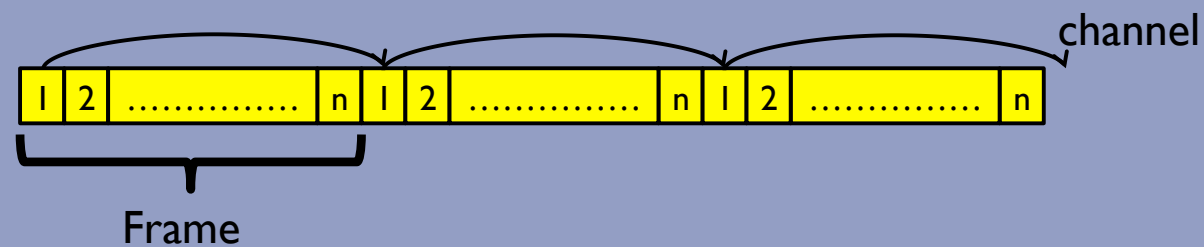
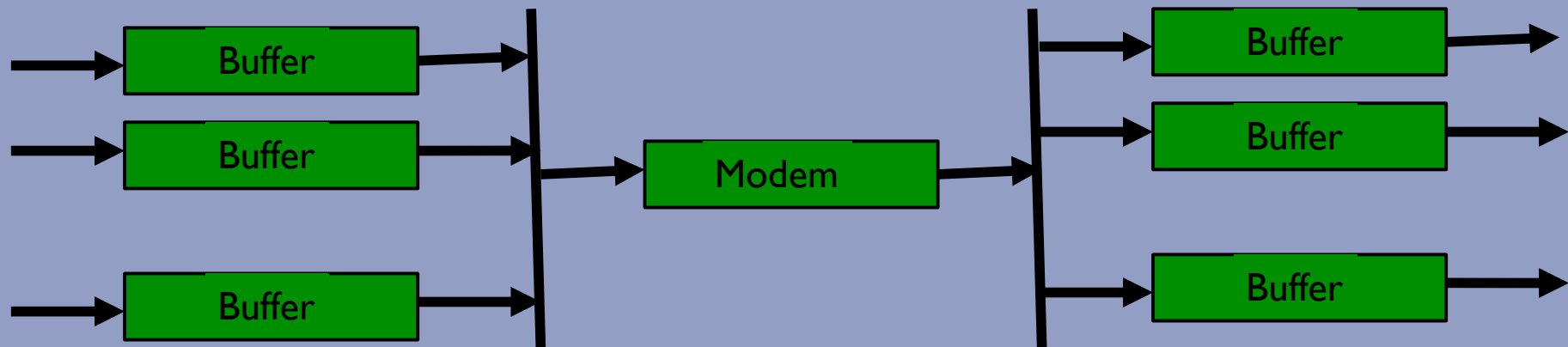


ADSL standards

Version	Standard name	Common name	Downstream rate	Upstream rate	Approved in
ADSL	ANSI T1.413-1998 Issue 2	ADSL	8.0 Mbit/s	1.0 Mbit/s	1998
ADSL	ITU G.992.1	ADSL (G.DMT)	12.0 Mbit/s	1.3 Mbit/s	1999-07
ADSL	ITU G.992.1 Annex A	ADSL over POTS	12.0 Mbit/s	1.3 Mbit/s	2001
ADSL	ITU G.992.1 Annex B	ADSL over ISDN	12.0 Mbit/s	1.8 Mbit/s	2005
ADSL	ITU G.992.2	ADSL Lite (G.Lite)	1.5 Mbit/s	0.5 Mbit/s	1999-07
ADSL2	ITU G.992.3	ADSL2	12.0 Mbit/s	1.3 Mbit/s	2002-07
ADSL2	ITU G.992.3 Annex J	ADSL2	12.0 Mbit/s	3.5 Mbit/s	
ADSL2	ITU G.992.3 Annex L	RE-ADSL2	5.0 Mbit/s	0.8 Mbit/s	
ADSL2	ITU G.992.4	splitterless ADSL2	1.5 Mbit/s	0.5 Mbit/s	2002-07
ADSL2+	ITU G.992.5	ADSL2+	24.0 Mbit/s	1.1 Mbit/s	2003-05
ADSL2+	ITU G.992.5 Annex M	ADSL2+M	24.0 Mbit/s	3.3 Mbit/s	2008

VDSL (Very-high-bitrate Digital Subscriber Line) is een opvolger van ADSL, waarbij een veel hogere download- en uploadsnelheid mogelijk is. Bij VDSL zijn snelheden tot in de praktijk 52 Mbps download en 5 Mbps upload mogelijk is.

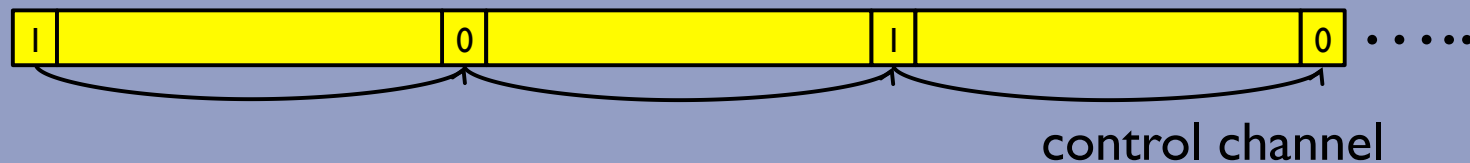
Time Division Multiplexing(TDM)



Synchronous TDM: not synchronous but frames are fixed and slots are always filled

How is framing implemented

Added digit framing



Pulse Stuffing

Frequency $> \sum$ (freq. of the sources)

additional bits are added at fixed position in the frame

Relationship with data link framing

←
 $F_1 F_2 A_1 F_2 C_1 A_2 d_1 C_2 d_1 d_2 \dots$

for

→
 $F_1 f_1 f_1 d_1 d_1 d_1 C_1 A_1 F_1$ in characters
↓ ↓ ↓ ↓ ↓
flag FCS information control address

Example: Telephony

DS-1 transmission format

Voice → PCM (8000 samples per second, 8-bit)

TDM-frame = 24 (channels) × 8 bits + 1 (frame bit)
= 193 bits

Data rate: $8000 \times 193 = \mathbf{1.544 \text{ Mbps}}$

DATA only 23 out of 24 channels used,
24th channel has special SYNC BYTE
per channel 1 bit for user/system data

→ $7 \times 23 \times 8000 = 56 \times 23 \text{ kbps} = \mathbf{56 \text{ kbps}}$ p. channel

Standards Telephony

US/JAPAN			ITU-T		
	# channels	Mbps	Level	#channels	Mbps
DS-1	24	1.544	1	30	2.048
DS-1C	48	3.152	2	120	8.448
DS-2	96	6.312	3	480	34.368
DS-3	672	44.736	4	1920	139.264
DS-4	4032	274.176	5	7680	565.148

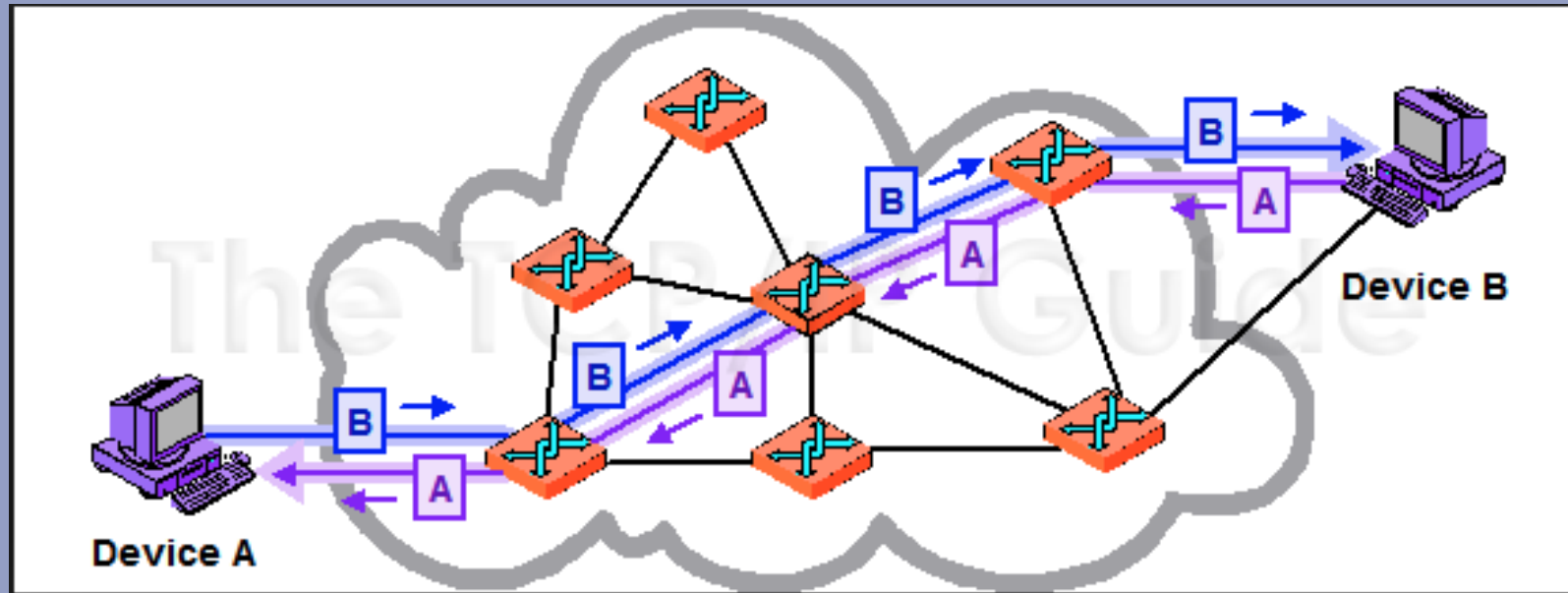
Network **SWITCHING**

Next to multiplexing: **switching** is required to realize multi to multi connections

Especially needed in Wide Area Networks (WAN)

Also present in Local Area Networks (LAN) or in multi processors architectures.

Circuit Switching



1. A dedicated path between two end stations is realized or channel (TDM/FDM)
2. Data is being transmitted (Switches don't inspect data)
3. Path is broken up

Circuit switching

IMPORTANT CHARACTERISTIC:

BLOCKING VS NON-BLOCKING

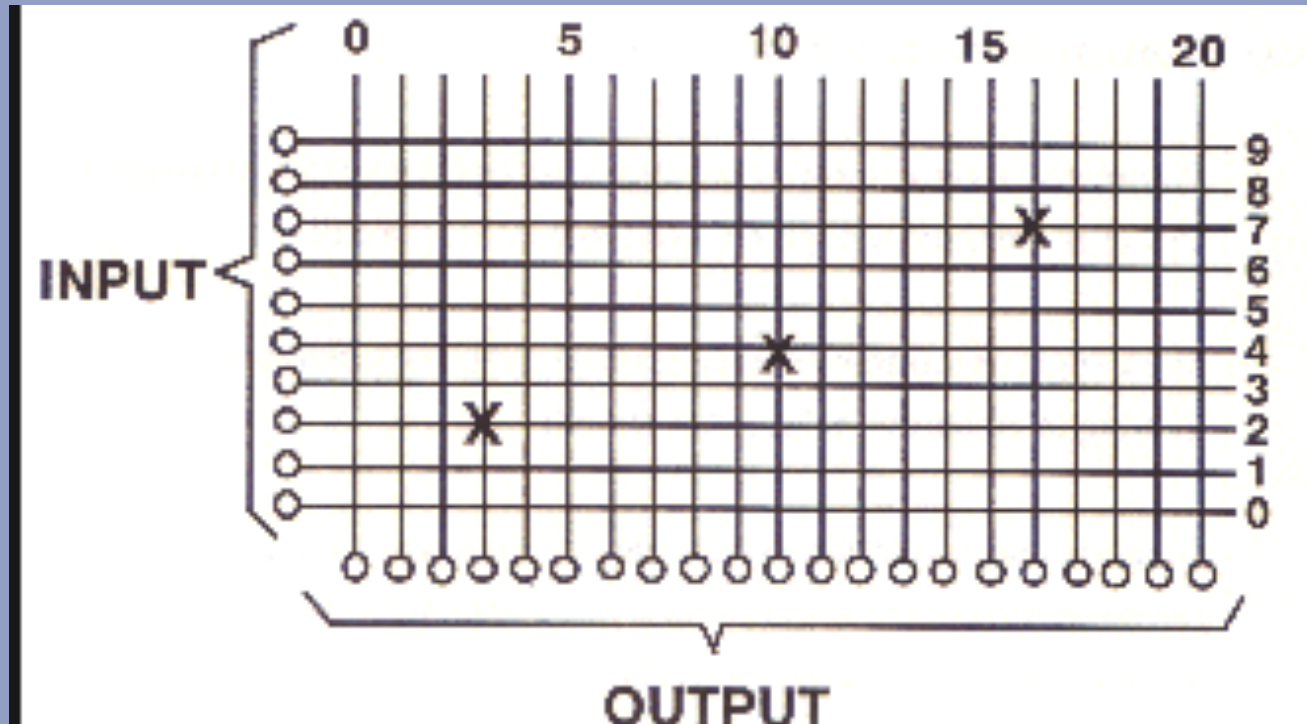


Connection cannot be realized
because all paths are occupied



Connection can always be realized and at any time

Space Division Switching: non-blocking **Crossbar Switch**

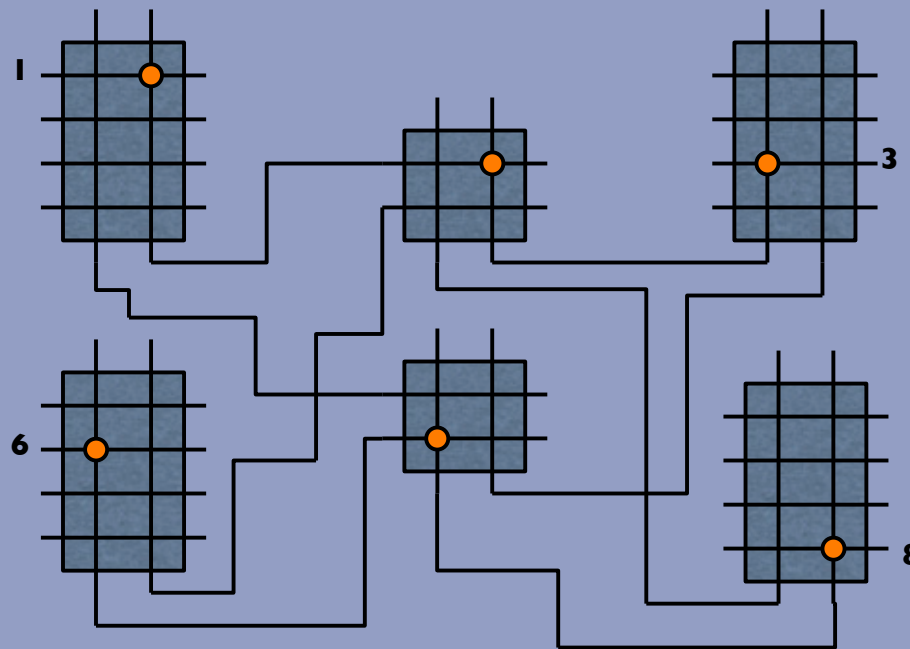


Very costly: N^2 switches

Multistage Networks

Use many small crossbar switches and connect them wisely.

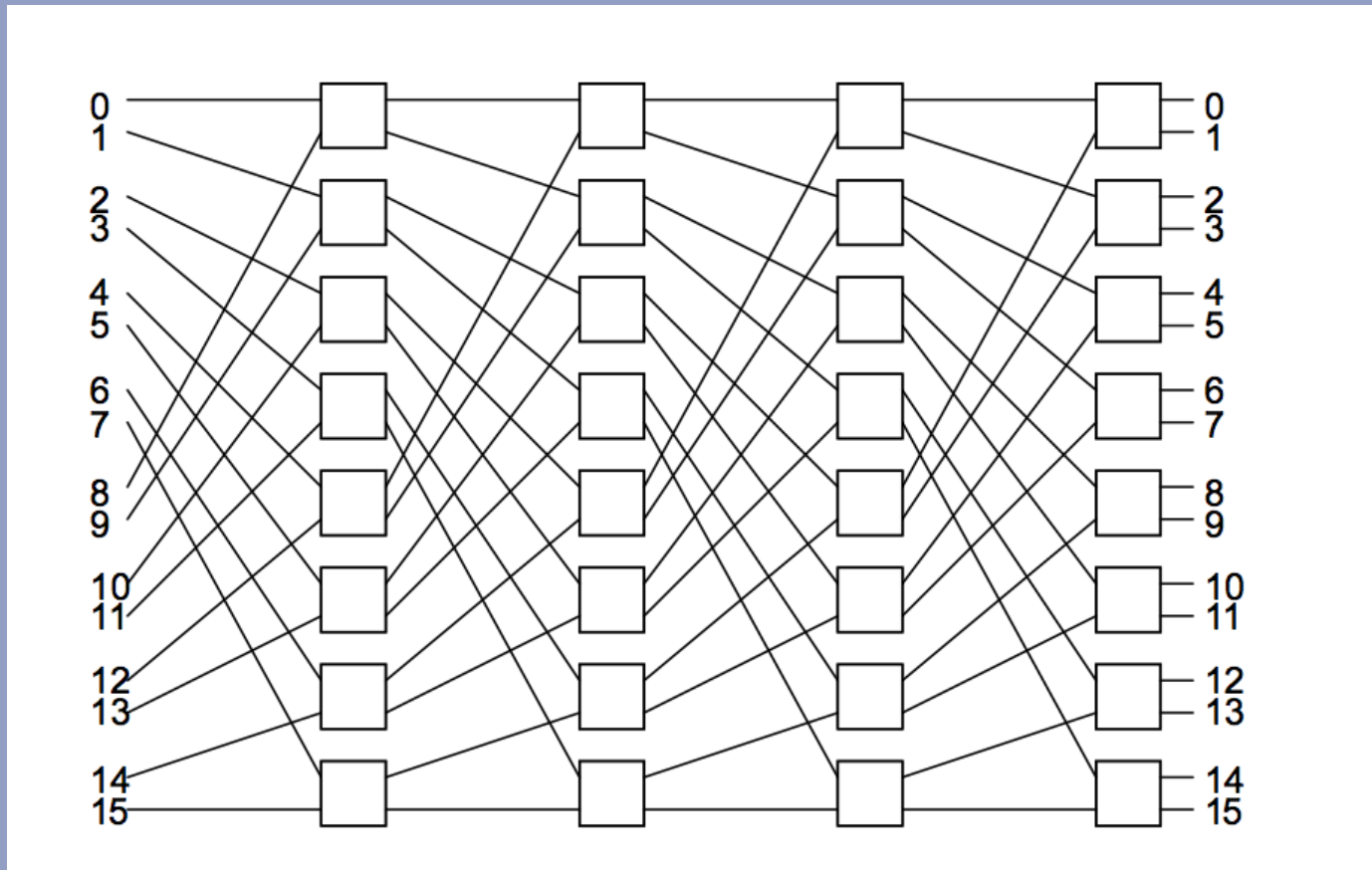
Blocking can occur!!!!



5 cannot be connected to 2

Omega Networks

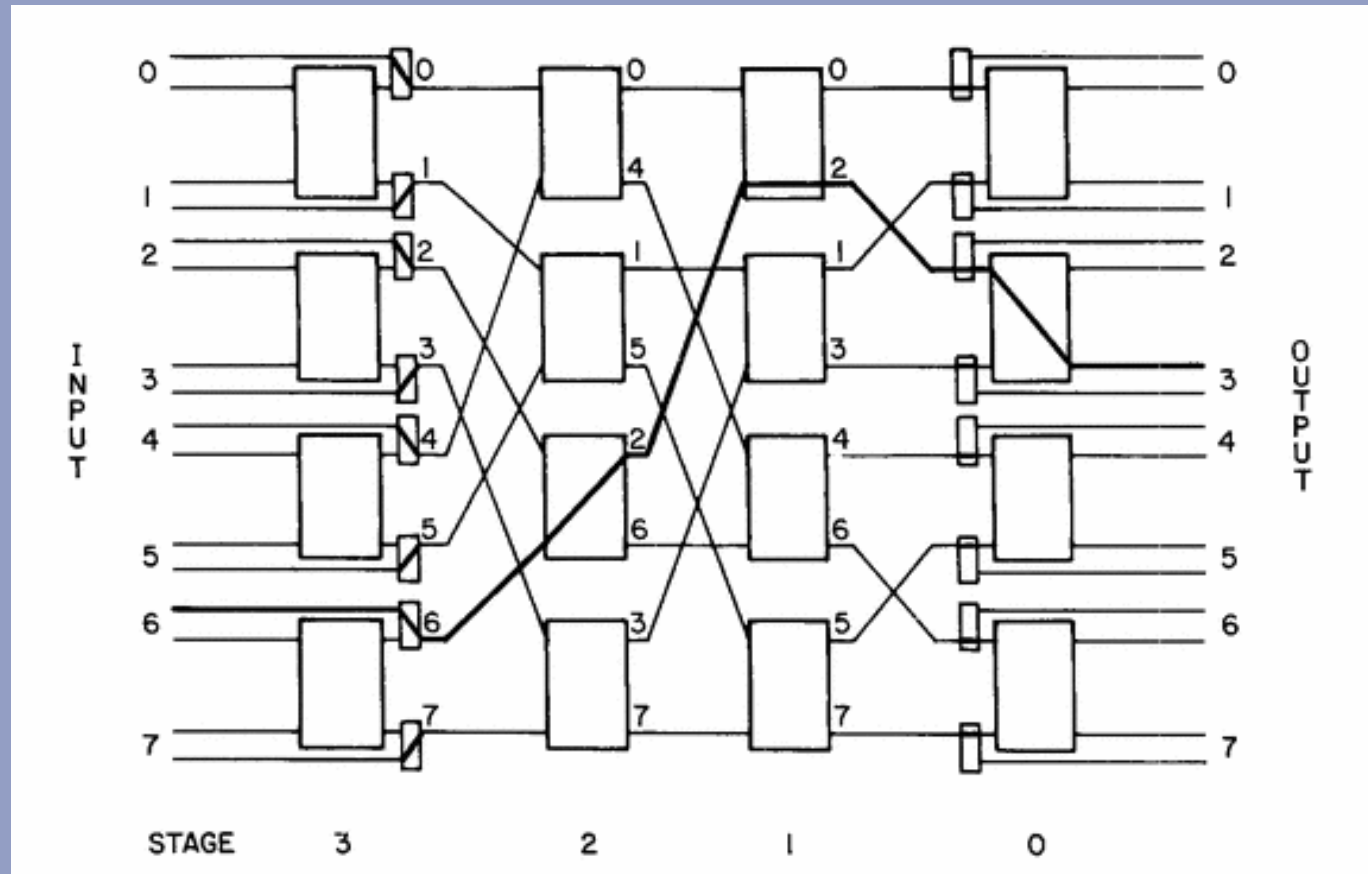
(based on Perfect Shuffles)



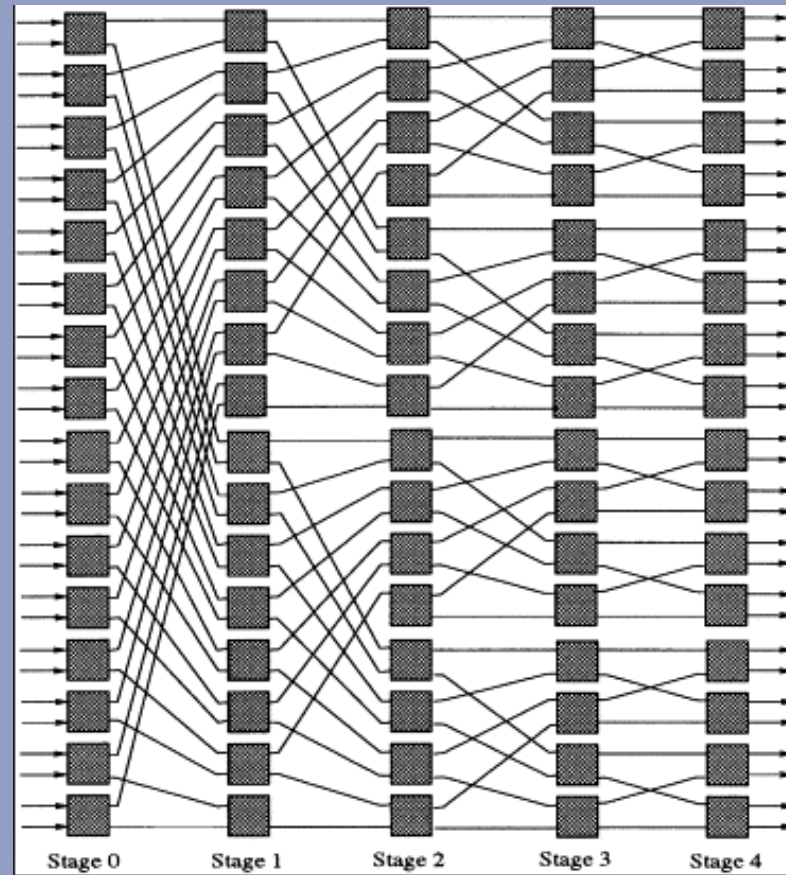
$2\log N + 1$ stages: non blocking with $O(N\log N)$ switches)

Variants of PS networks

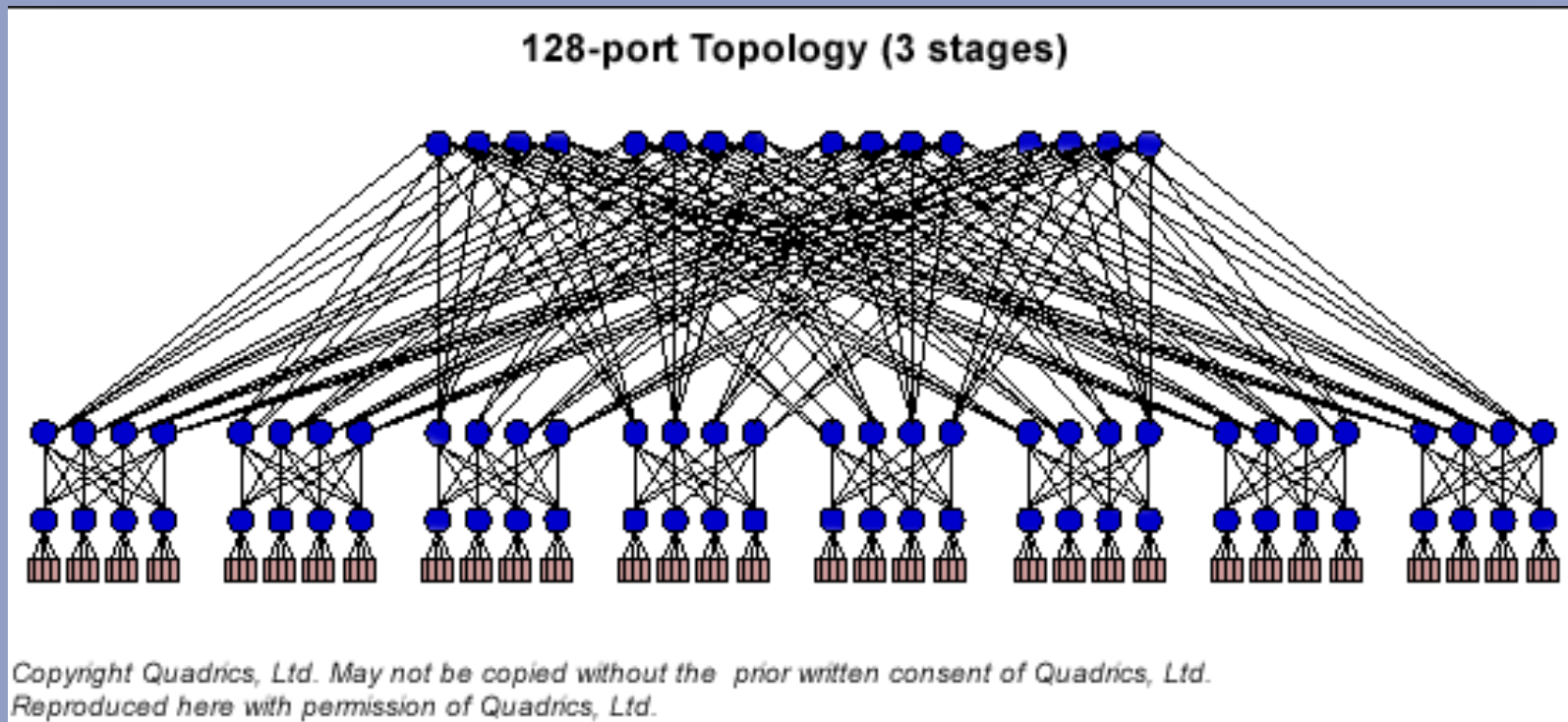
Cube Network



Butterfly Network

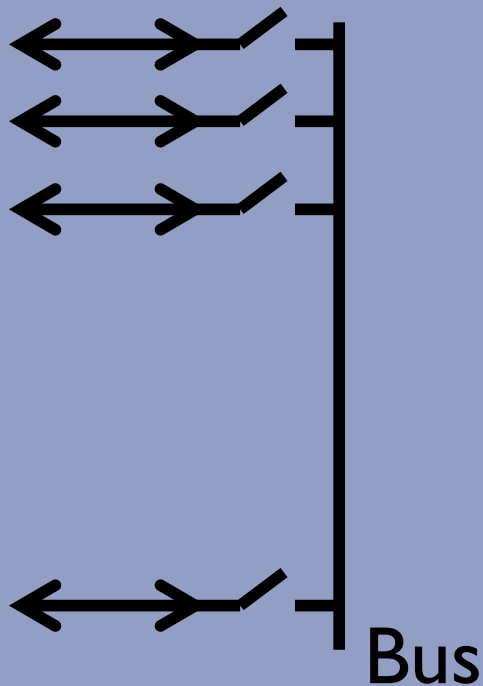


Fat Tree Network



Time Division Switching

Do not confuse with TDM !!!



Bandwidth of Bus $>$
 Σ indiv. bandw.
Then non-blocking!

Routing in Circuit Switched Networks

- Alternate Routing
 - Each switching node has its own routing table

	First choice	Second choice
A to B	Via switch i	Via switch j
A to C	Via switch j	Via switch k

- Fixed Alternate Routing
 - Routing tables do not change
- Dynamic Alternate Routing
 - Depending on time (e.g. time of the day) routing tables will change
- Adaptive Routing
 - Central Controller gets status of all switches and gives routing updates to all switches

Packet Switching

**Data is sent by packets (usually < 1000 octets),
Every switching nodes has buffers**

- **Datagram**

Every packet is routed independently

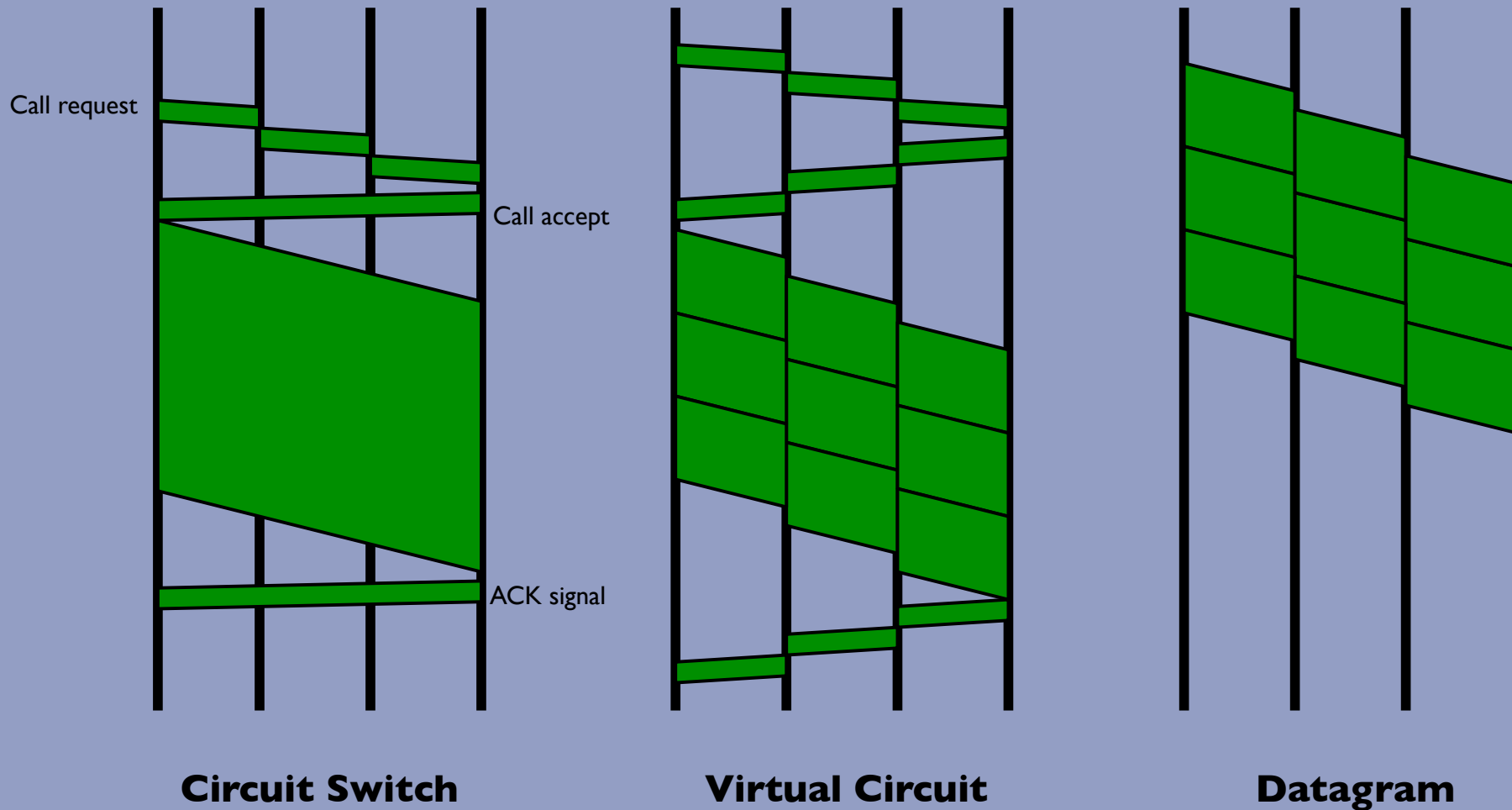
→ As a consequence packets can arrive out of order

- **Virtual Circuit (Wormhole)**

Before communication is initiated a Call-Request packet is sent on the network, which fixates a virtual path between sender and receiver.

→ Packets arrive in order, but it not as flexible as datagram

Summarizing

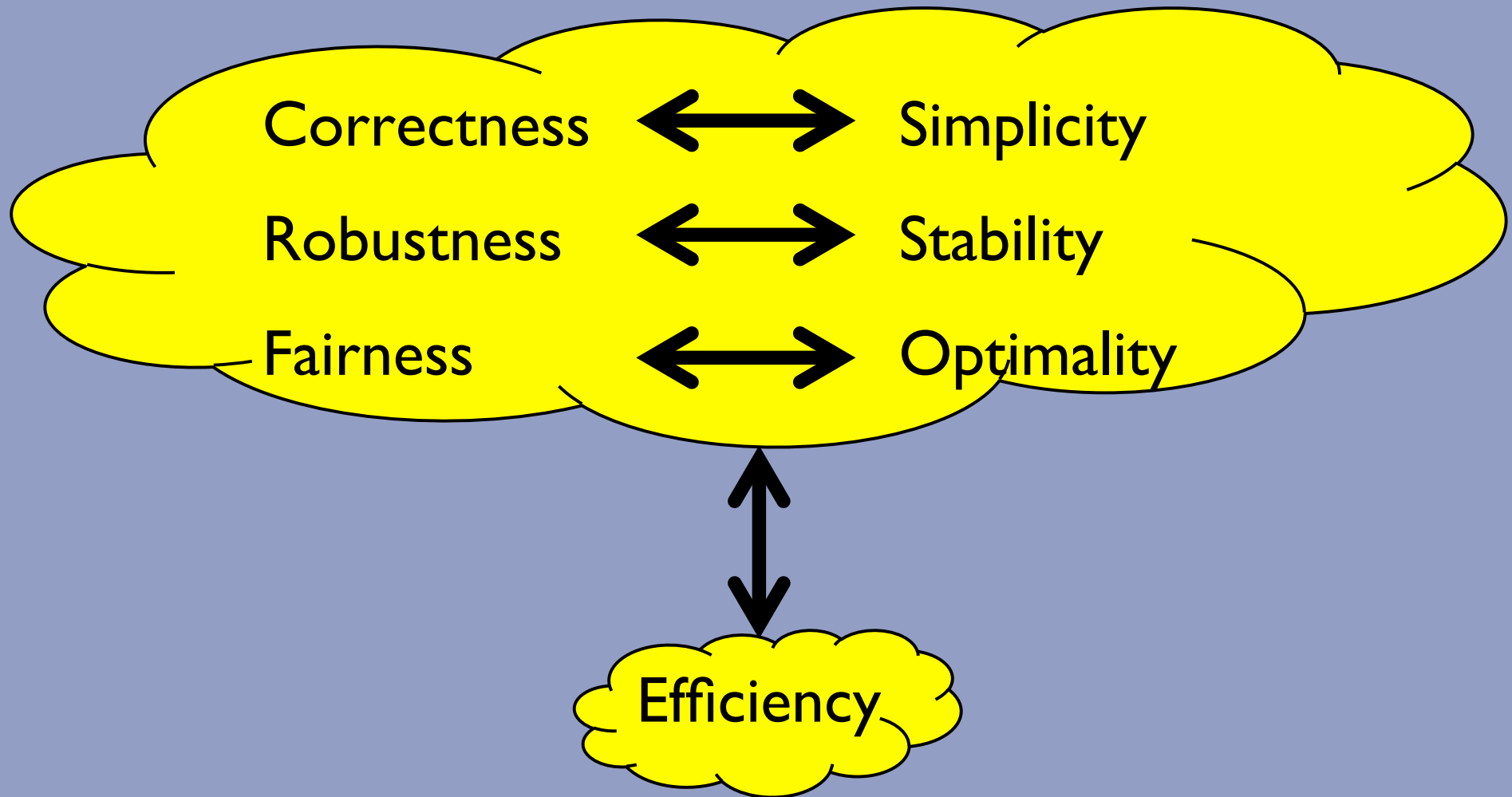


Different Combinations

- External Virtual Circuit & Internal Virtual Circuit
- External Virtual Circuit & Internal Datagram
- External Datagram & Internal Virtual Circuit
- External Datagram & Internal Datagram

Which one makes sense?

Routing Trade Offs



Routing for Packet Switched Networks

Like circuit switching can we differentiate between: **Fixed Routing** and **Alternate Routing**

➔ **No difference between datagram and virtual circuit**

Random Routing

Every node chooses randomly outgoing link, based on some prob. Distribution, e.g.

$$P_i = R_i / \sum R_j$$

with R_i data rate possible on link i .

Flooding (very inefficient, very reliable)

Every node puts incoming packet on every outgoing link, except the incoming link

→ Exponential growth

1. Every node logs all the packets

If packets arrives a second time: **discard**

2. Every packet, contains counter: hop-count

If hop-count > threshold: **discard**

Adaptive Routing (Central vs Distributed)

Every node gets network status information

- Local, e.g. queue length of the outgoing links
- Adjacent nodes
- All nodes

ARPANET

based on Adaptive Routing, Distr. & Adjacent Nodes

First Version: 1969

Based on **Bellman-Ford Algorithm**

Every node i has two vectors:

$$D_i = \begin{pmatrix} d_{i1} \\ d_{i2} \\ \vdots \\ d_{iN} \end{pmatrix}$$

$$S_i = \begin{pmatrix} s_{i1} \\ s_{i2} \\ \vdots \\ s_{iN} \end{pmatrix}$$

$N = \#nodes$

$d_{ij} =$ estimated delay from node i to j

$s_{ij} =$ next node on the route i to j

Every 128 ms every node exchanges delay vector with adjacent nodes.

Then every node k : $d_{kj} = \text{Min}_{i \in A} [d_{ij}^{\text{new}} + d_{ki}]$ and $s_{kj} = i$,

the node i which minimizes d_{kj} . Link delays are the queue length for that link.

Disadvantages: Link delays were not accurate
 Thrashing would occur

2^{de} Generation (1979)

Every node:

- Timestamp on incoming message (arrival time)
- Departure time recorded
- If pos. ACK is received: $\text{delay} = (\text{dept. time} - \text{arrival time})$

Every 10 sec: every node computes the average delay per link

If delay is different: **FLOODING** is used to inform all the other nodes

Every node gets status of the whole network!!!!!!!!!!

Dijkstra's shortest path algorithm is used to compute new routing table

3^{de} Generation (1987)

When load is heavy:

Observed delay under old routing \neq delay under new routing

→ Oscillation effects

→ Instead of BEST route: a “good” route

Smoothing of link costs (delays)

Every 10 seconds:

1. (Queuing theory) $\rho = 2(s-t)/(s-2t)$, with ρ = link utilization
 t = measured delay
 s = service time
2. $U(n+1) = 0.5\rho(n+1) + 0.5U(n)$, $U(n)$ average utilization
3. New delays are computed based on $U(n)$,
terrestrial: 1 Hop for $U(n) < 0.5$, 2 Hops for $U(n) > 0.8$
sattelite: 2 Hops for $U(n) < 0.8$

Otherwise the same as 2-de generation