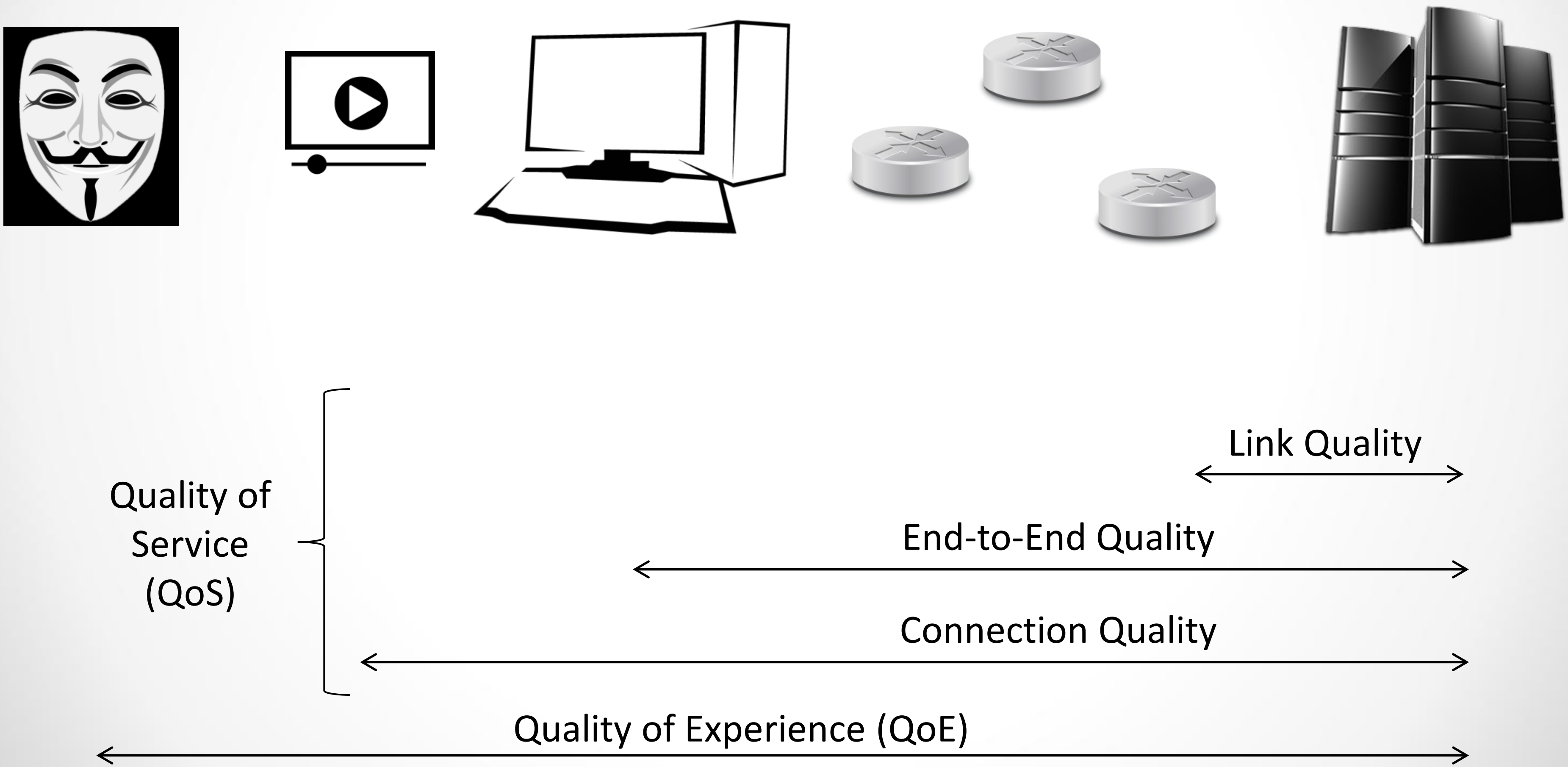
Quality of Service: Congestion Control

E.P. Stepanov

Quality of Service in Computer Networking



Types of applications and their quality requirements

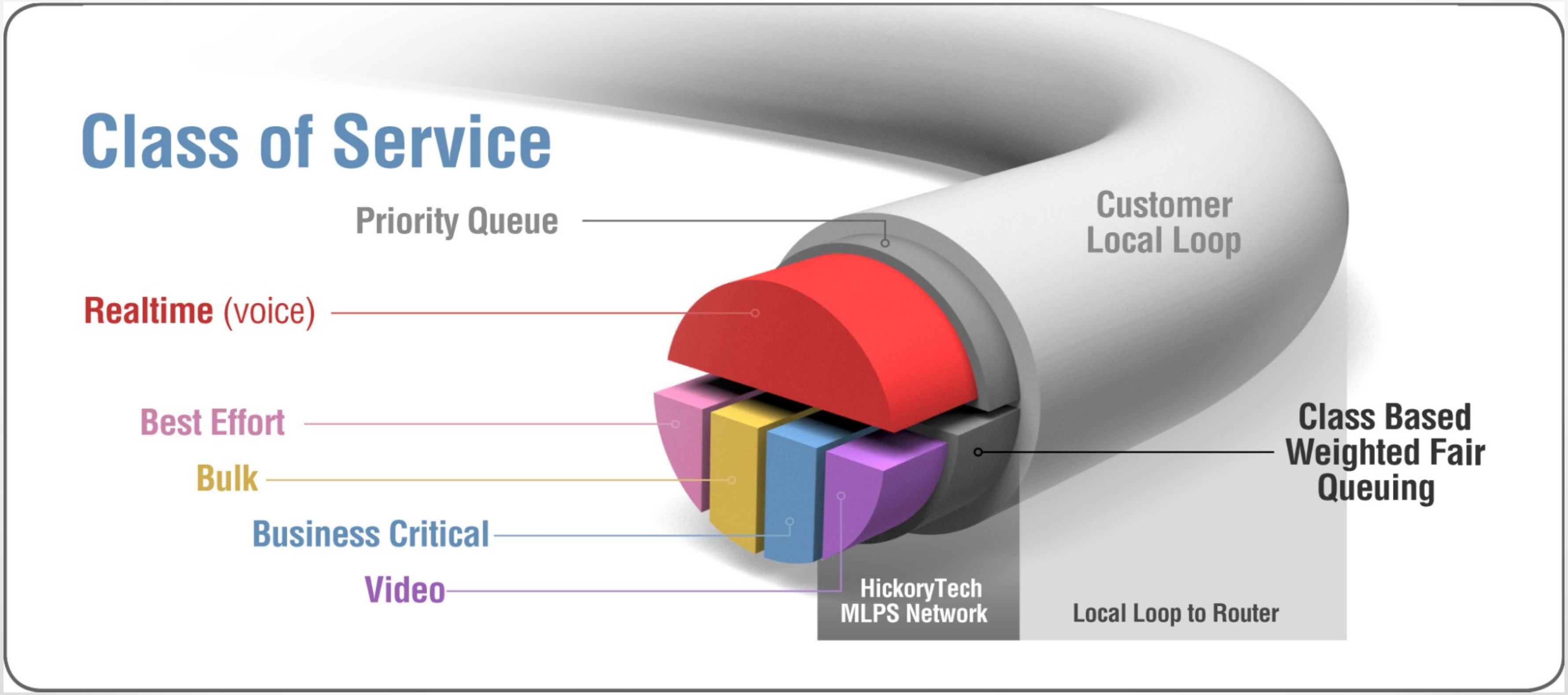
- Real time:
  - Voice, video, stock exchanges, …
  - Hard real time: the level of losses and the maximum data transfer time (delay & bandwidth) are always less than the specified values
  - Soft real time: quality metrics may violate requirements with some probability
- No real time requirements :
  - File Transfer, Mail, Consoles, …

What is QoS management?

- Methods allowing to serve different data flows with different quality
- Methods for distributing network resources between different data flows
- Methods for ensuring predictable and consistent network behavior in an ever-changing configuration
- Methods to improve the efficiency and utilization of network equipment

What is QoS?

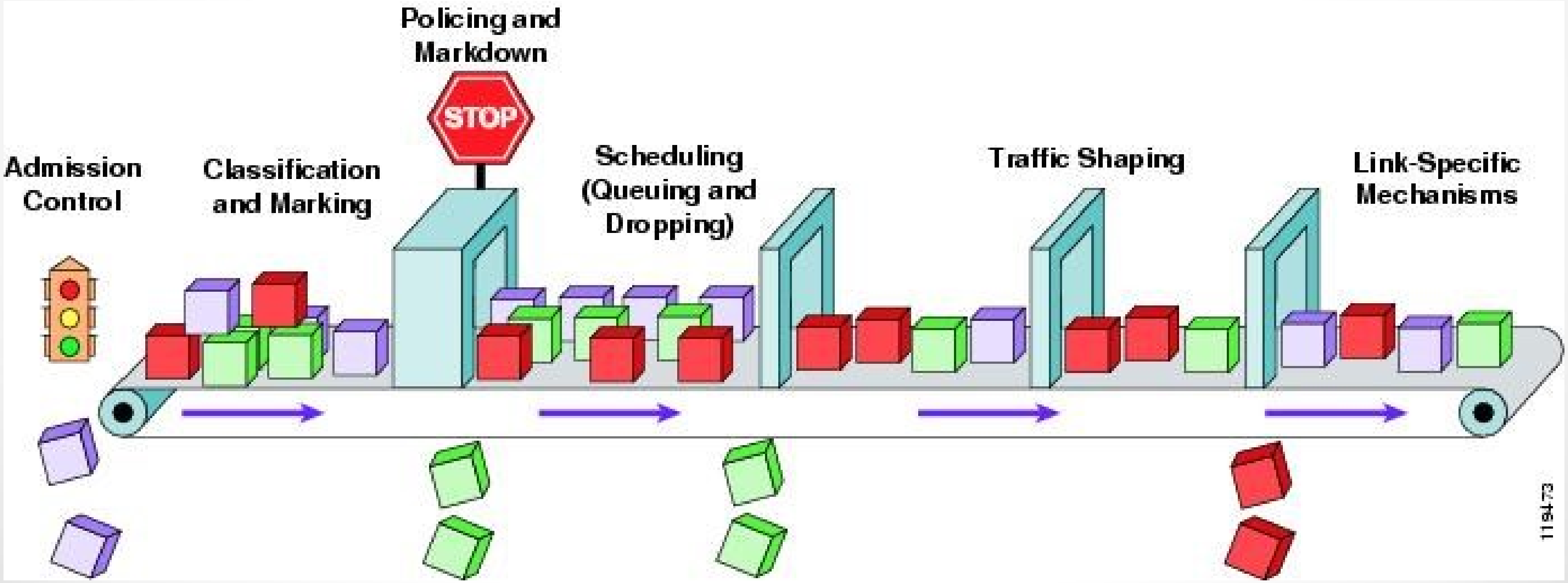
- Methods allowing to serve different data flows with different quality



http://blogs.salleurl.edu/raising-a-data-center/the-importance-of-the-qos/

What is QoS?

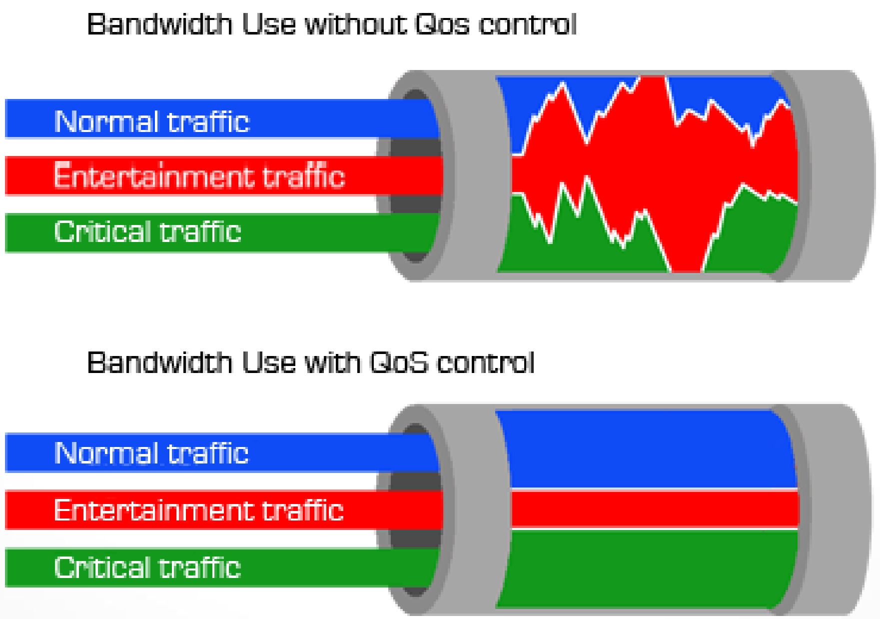
- Methods for distributing network resources between different data flows



http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND/QoS-SRND-Book/QoSIntro.html

What is QoS?

- Methods for ensuring predictable and consistent network behavior in an ever-changing configuration



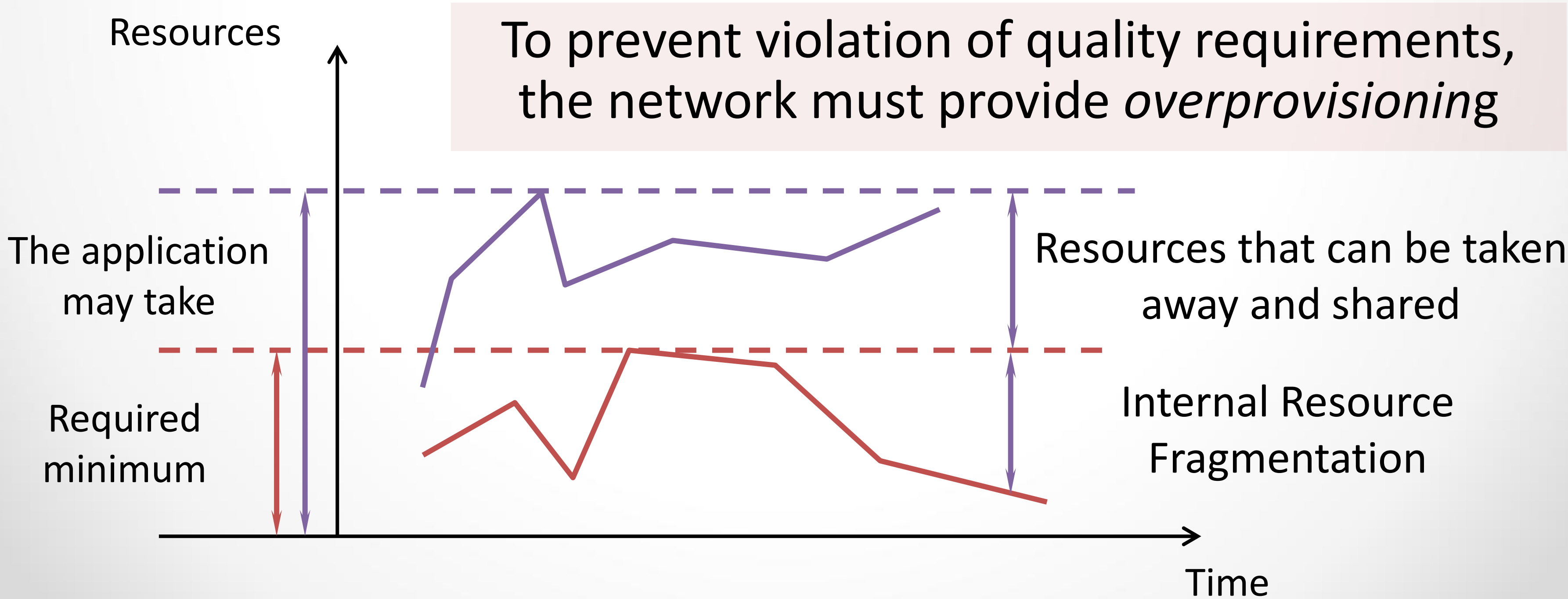
https://prohardver.hu/tudastar/qos_quality_of_service.html

What is QoS?

- Methods to improve the efficiency and utilization of network equipment

The network works correctly if each application is provided with the required amount of resources

To prevent violation of quality requirements, the network must provide *overprovisioning*

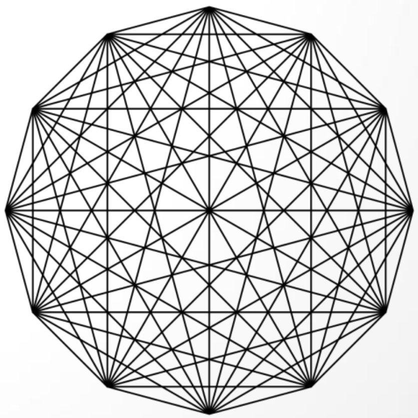


Why is QoS becoming an increasingly important issue?

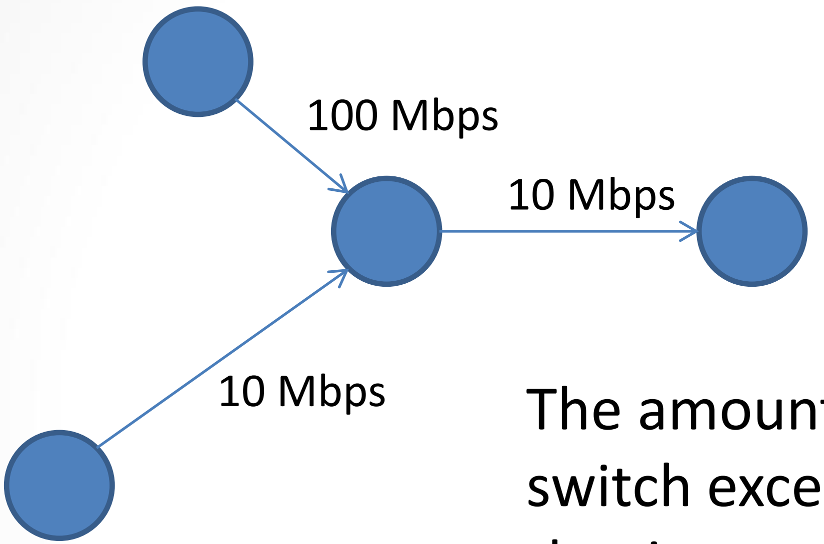
- It has become important for providers not only to ensure connectivity, but also to ensure the quality of connections
  - Subscription to IP Services (live streaming, VOD, DVR)
- Network traffic is becoming less predictable - it's hard to design networks
  - 80/20 distribution is no longer relevant
- Increasing equipment capacity reduces network efficiency
- QoS implementation reduces network maintenance costs
  - More rare hardware updates

Oversubscription

- *The effect of oversubscription* is manifested if the network infrastructure is not able to provide the proper quality of service to all its users at the same time.



Congestion



The amount of data entering the switch exceeds the amount of data that it can transmit.

The switches buffer the traffic passing through them:

- Short-term congestion increases delay
- With a sufficiently long congestion, packets are discarded

TCP Congestion Control: goals

- Connections should adapt to the quality of the provided communication link and strive to use the provided resources as *efficiently* as possible
- Connections should automatically allocate the bandwidth of the shared link in a *fair* manner

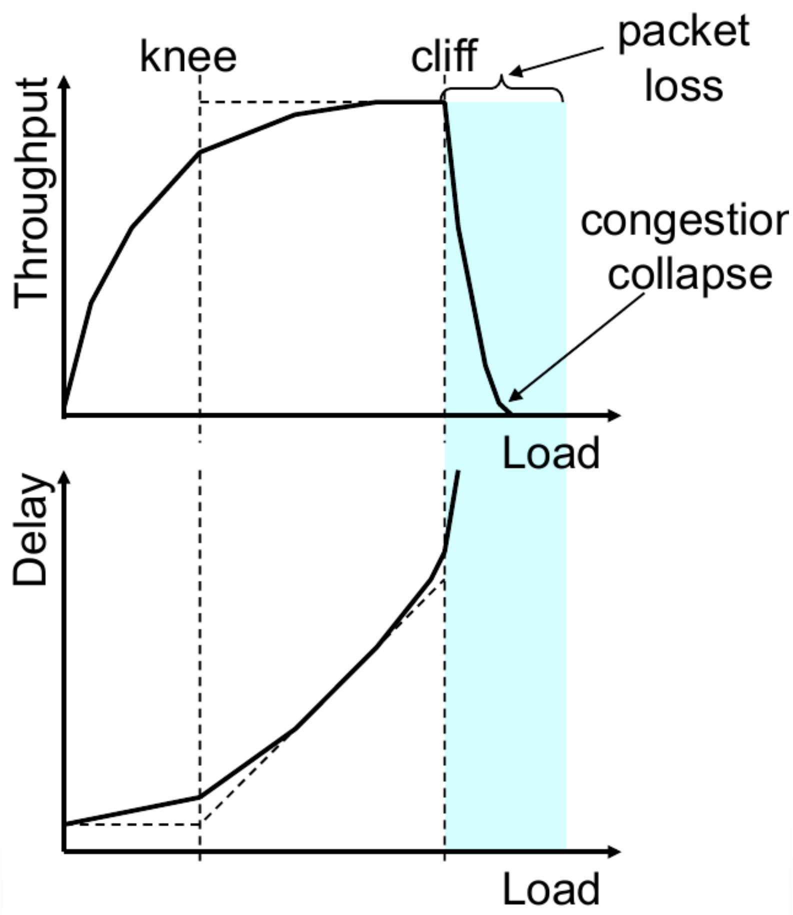
TCP Congestion Control: work principles

- Network Interacting
  - Network devices indicate about congestion (TCP/ECN)
- No interaction with switches
  - Congestion is determined indirectly (with packet loss, delay increase etc.)
- Reactive (generally, loss based)
  - Detect congestion occurence
- Proactive (generally, delay based)
  - Limit the connection bandwidth, foreseeing an early congestion

TCP Congestion control: reaction rate

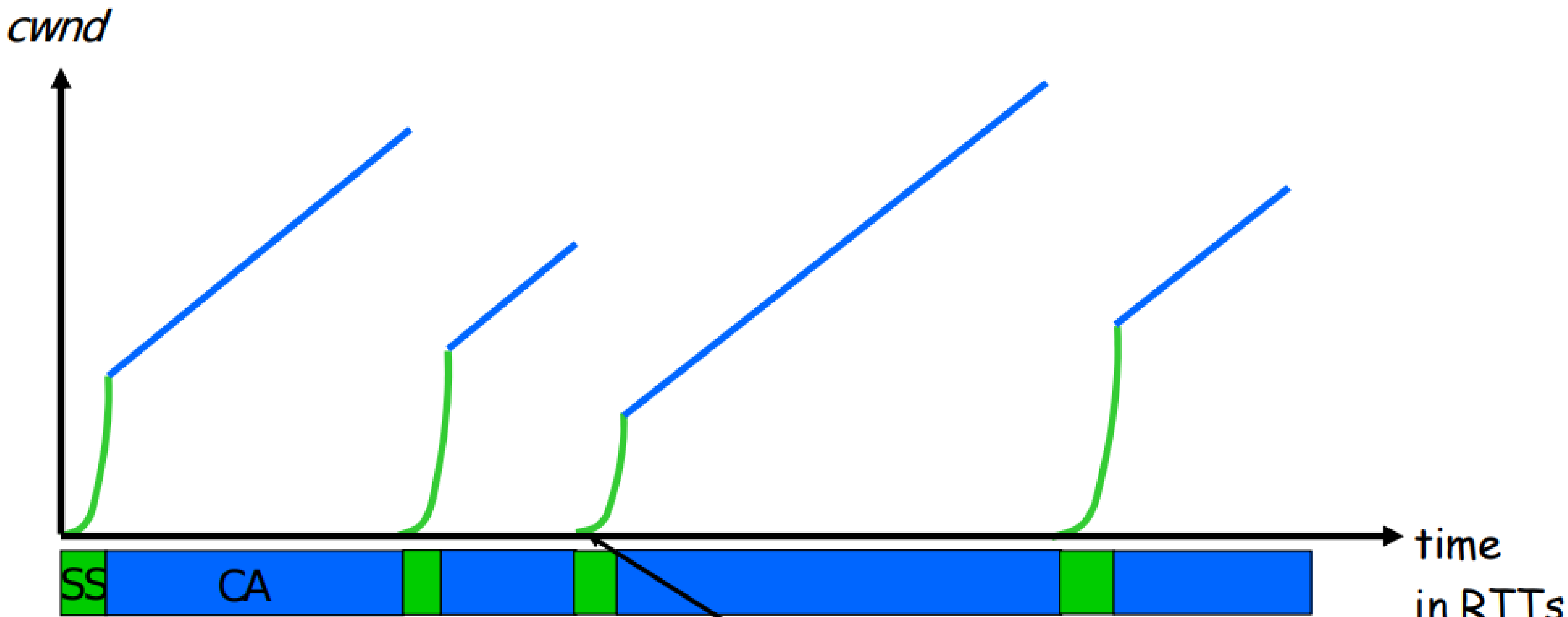
- In the absence of additional service packets, the sender receives information from incoming ACK messages
- The ACK, corresponding to the some sent packet, arrives after one Round Trip Time (RTT)
- Hosts are able to adapt to the state of the network no faster than RTT

TCP congestion control algorithms



Van Jacobson ***Congestion avoidance and control***
Proceedings of SIGCOMM '88 -- Stanford, CA

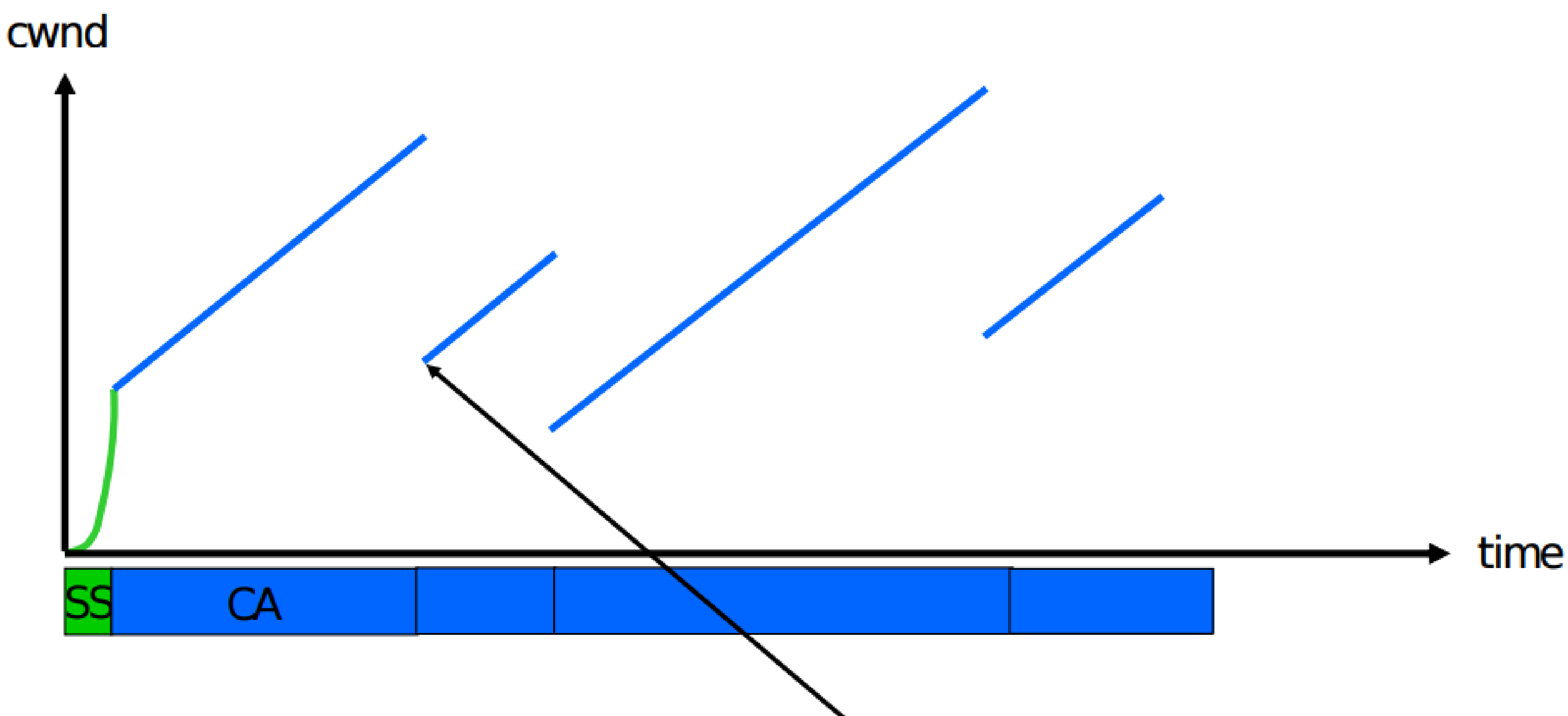
TCP Tahoe (Jacobson 1988)



SS: Slow Start
CA: Congestion Avoidance

- Decrease to 1 for either timeout or 3 dupACKs
- Fast retransmit on 3 dupACKs

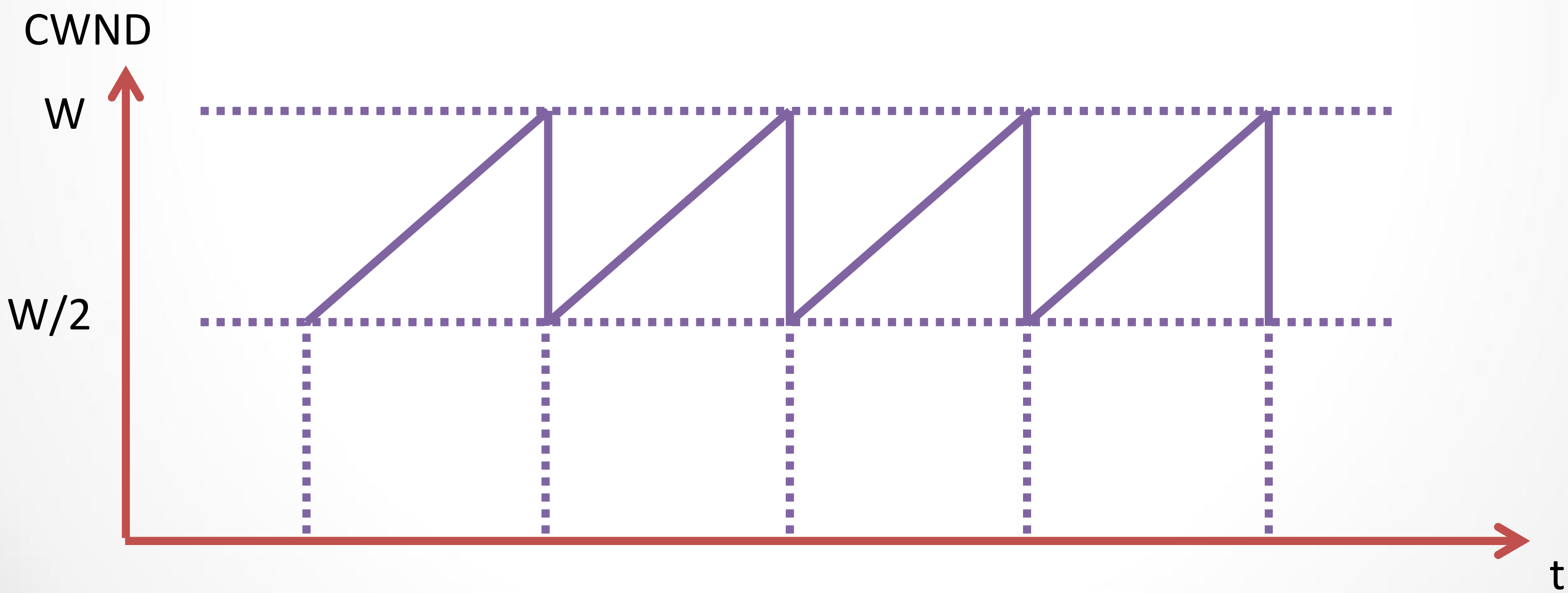
TCP Reno (Jacobson 1990)



SS: Slow Start
CA: Congestion Avoidance

Additive Increase Multiple Decrease (AIMD)

In steady state, the graph of the congestion window (CWND) of time (t) for the TCP Reno algorithm looks like a saw



The dependence CWND of packet loss p

- Between consecutive losses :
  - goes through W / 2 rounds of the congestion algorithm, which enlarges the window from W / 2 to W
  - 1/p packets transmitted
- $(W/2 + \ W)/2 * W/2 = 1/p$
- $W \ = sqrt(8/3p)$

TCP Reno issues:
1) networks with large BDP

- ***BDP = Bandwidth Delay Product***
- TCP Reno grows too slowly
- When using a 10 Gb x 100 ms channel, 50,000 RTT (more than an hour) will be required to increase CWND from W / 2 to W
  – With frequent packet loss, the algorithm can never reach the maximum channel bandwidth

TCP Reno issues:
2) RTT fairness

- Flows with lower RTT adapt to bandwidth faster than streams with higher RTT
- «Faster» connections take advantage and use more network resources

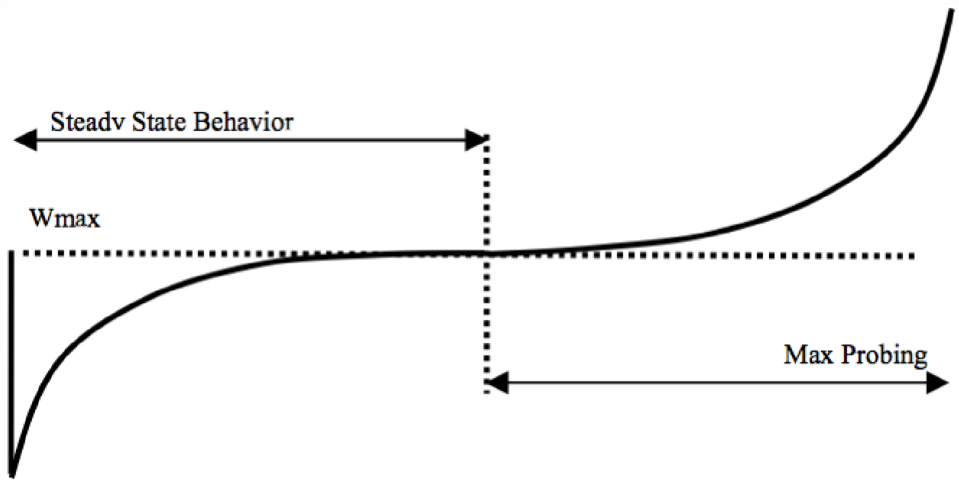
TCP Reno issues: 3) TCP friendliness

- In theory, delay-based congestion control algorithms can usually work more efficiently.
  - Algorithms can avoid packet loss
  - The delay value contains more information about the network environment.
- In practice, delay-based connections have poor performance because they are transmitted along with loss-based connections
  - Switch buffers are overloaded regardless of delay-based connection strategies

TCP Cubic

- The idea is to assume that losses occur at regular intervals
- The window size is scaled so that it reaches its maximum at the time of the next loss
  - Window size is independent of RTT!
- The current window size is determined by the cubic function
  - fast growth after congestion

TCP Cubic: CWND

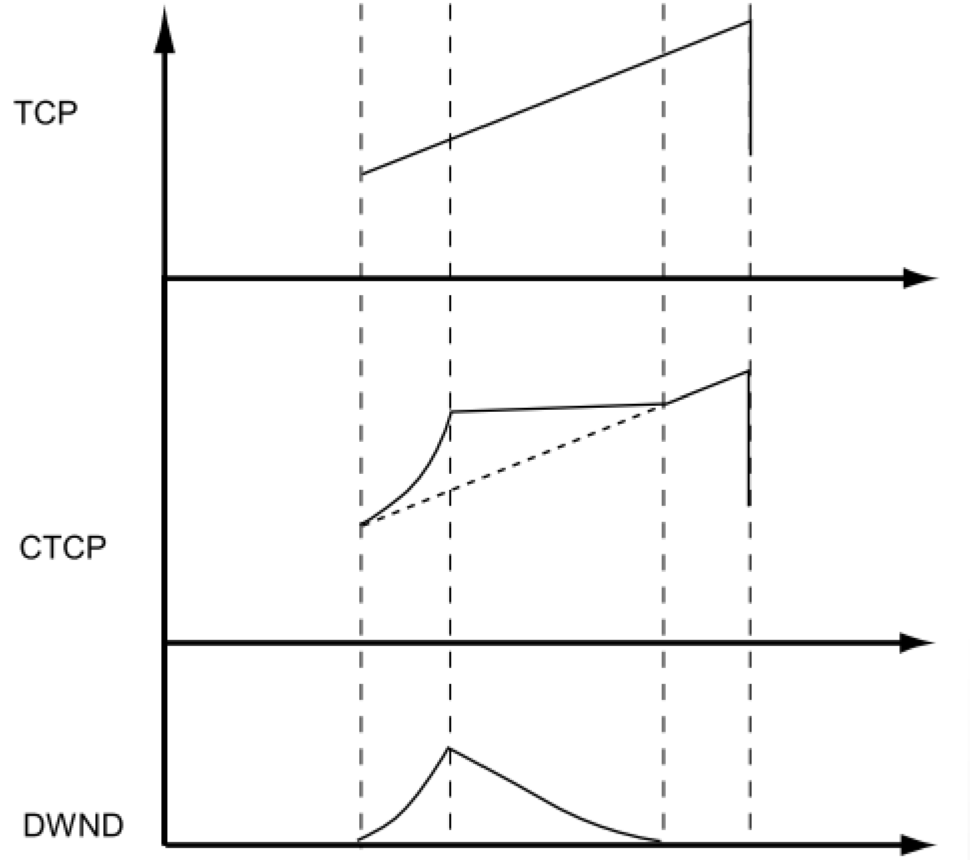


- CWND initially grows faster than TCP Reno
- As the probability of loss increases, growth slows down
- CWND is equal to the expected optimum for as long as possible
- If loss does not occur, the ceiling of the connection is incorrect
- The algorithm begins to grow rapidly to find the correct bandwidth

TCP Compound

- Idea is an effective combination of delay-based and loss-based algorithms
  - Delay-based connection rate
  - Equal fight with loss-based connections
- The congestion window consists of loss-based and delay based components:
  - Loss-based changes similar to TCP Reno
  - Delay-based changes exponentially depending on the current delay: it can both increase or decrease

TCP Compound



Data Center TCP:

specialized congestion control algorithm for data centers

- Horizontal service scaling: map-reduce
  - Requests to the server are sent to auxiliary servers
  - Each of the servers processes the request in its own area and returns the result to the central server
  - The central server combines the results and forms the final response to the request

Data Center TCP:

specialized congestion control algorithm for data centers

- Many “short” connections compete with long-living connections
- Frequent drop of short connection packets increases system latency:
  - Switch buffers do not fit multiple simultaneous responses
  - Switch buffers filled with packets of long-lived connections

Data Center TCP:

specialized congestion control algorithm for data centers

- The idea is to label connection packets transmitted through high-load buffers
- With sufficiently frequent receipt of labeled packets, the sender reduces the size of the CWND
- With rare receipt of labeled packets, the CWND size increases

Quality of Service: switch device

E.P. Stepanov

How do switches work?

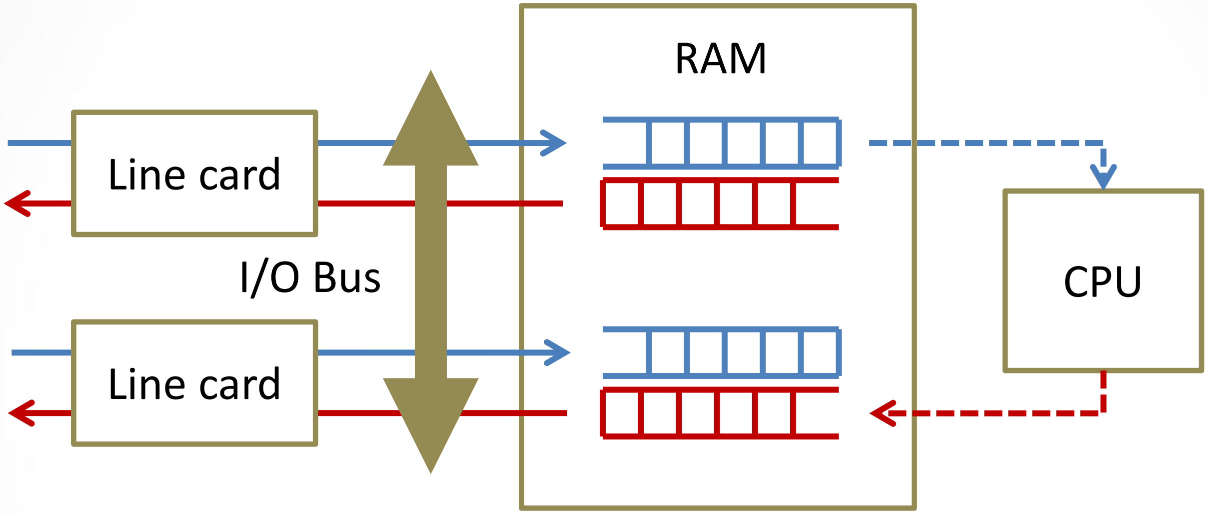
Classification of switches with generations (1)

- Generations reflect achieved performance characteristics, not a fundamental change in technology
- Evolution is achieved by changing the balance between cost and complexity of the switching device
- Each generation has taken its place and continues to be used today

Classification of switches with generations (2)

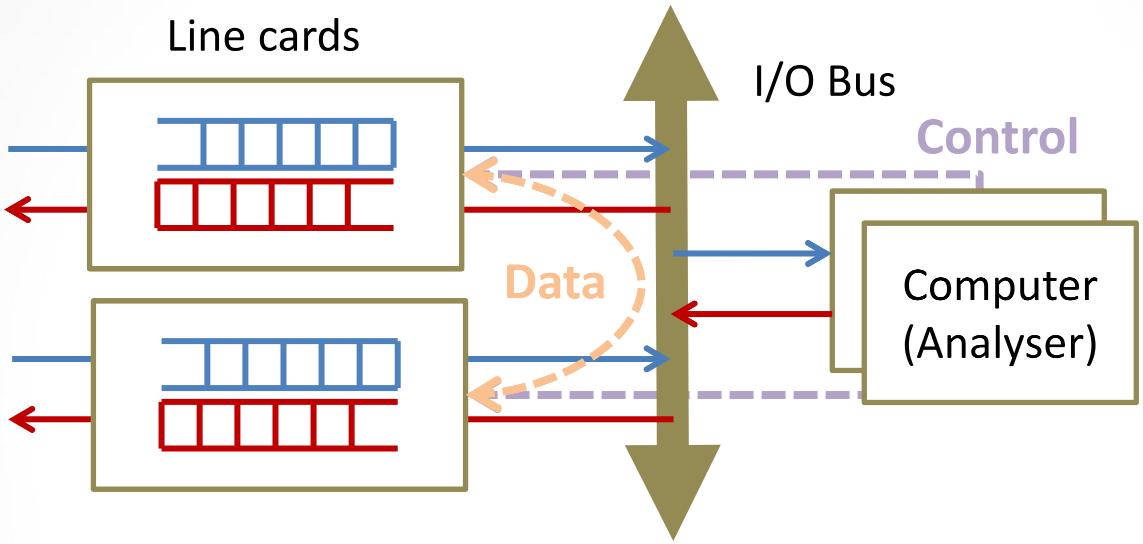
1. Interface Message Processors - SISD computers with multiple network interfaces *[см. William Yeager]*
2. Distributed MIMD architecture with own controllers on interfaces
3. Migrating from a single bus to a switch fabric architecture

The first generation of switches



- Most home Ethernet switches and routers
- Bottleneck can be a data bus or processor - depending on the type of traffic and the performance of these components

The second generation of switches

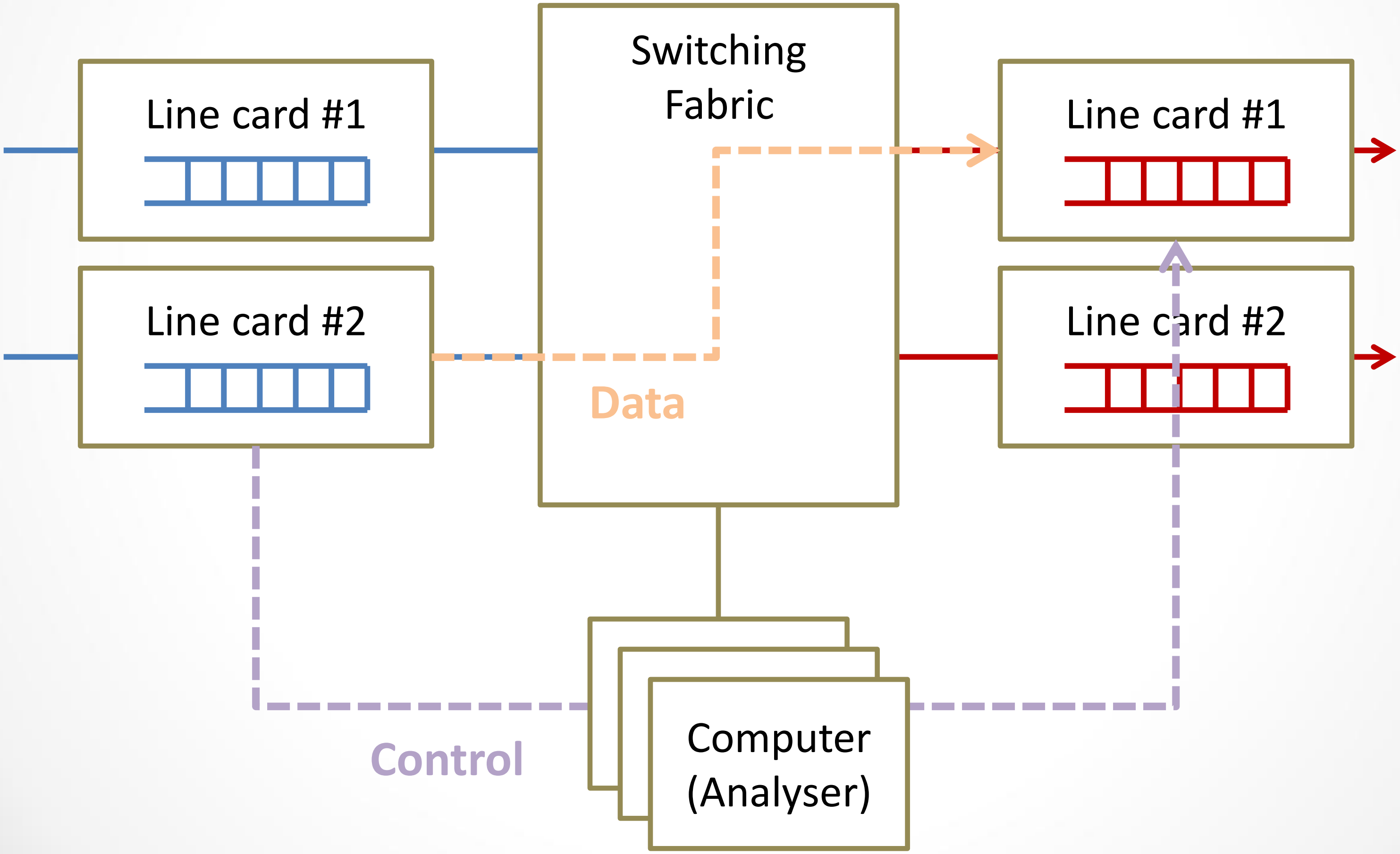


- Smart network interfaces with own controller and built-in memory
- Data is sent between cards directly, bypassing RAM
- Weak spot - shared data bus

Which generation are NFV solutions?

- The most modern implementations of software switches and some **virtual network functions** (VNF) are second-generation switches
- Packet analysis takes longer than passing packets between interfaces
- Example: NVWare - CGNAT with bandwidth up to 80 Gbit/sec

The third generation of switches



The third generation of switches

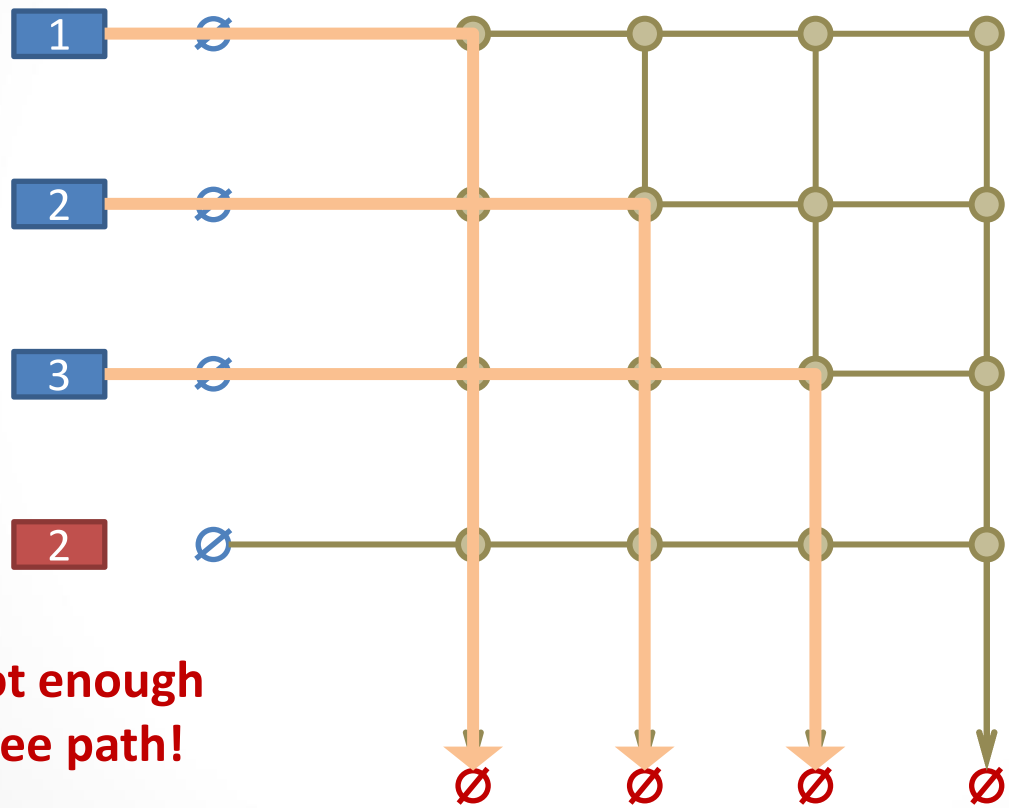
If in the second generation a common bus is used for packet transmission, in the third - **the switching matrix**

The switching matrix is capable of transmitting several packets between interfaces simultaneously (it has K transmission planes):

- For a data bus: $K = 1$
- Switching matrix $N \times N$:
  $1 \leq k \leq N$ (depends of load)
- Switching matrix $N \times N^2$:
  $\mathrm{k} = N$ (depends of load)

Crossbar Switching Fabric

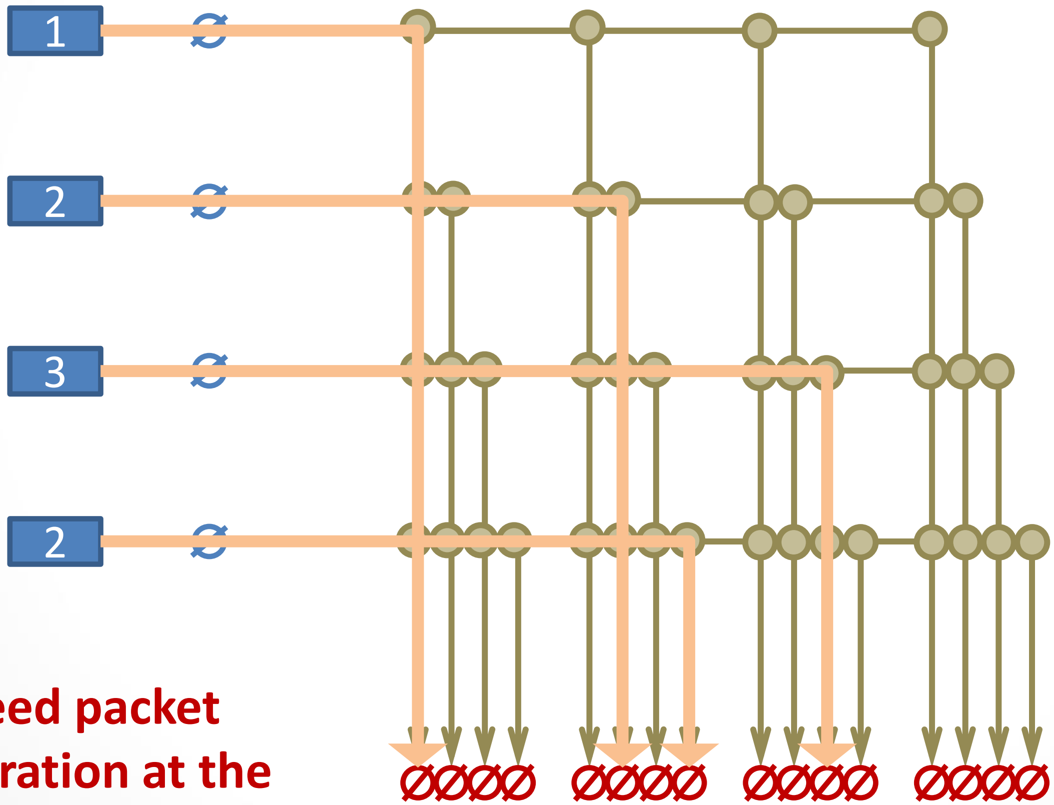
N^2 switches



Not enough free path!

Crossbar Switching Fabric

N^3 switches



Need packet arbitration at the output!

Switching modes

Δ_f -- packet switching delay (FIFO)

Δ_s -- packet serialization delay

Δ_h -- packet processing delay (FIFO)

Store-and-Forward: $\Delta_f = \Delta_s + \Delta_h$

Processing begins after receiving the entire packet

Works in terms of packets

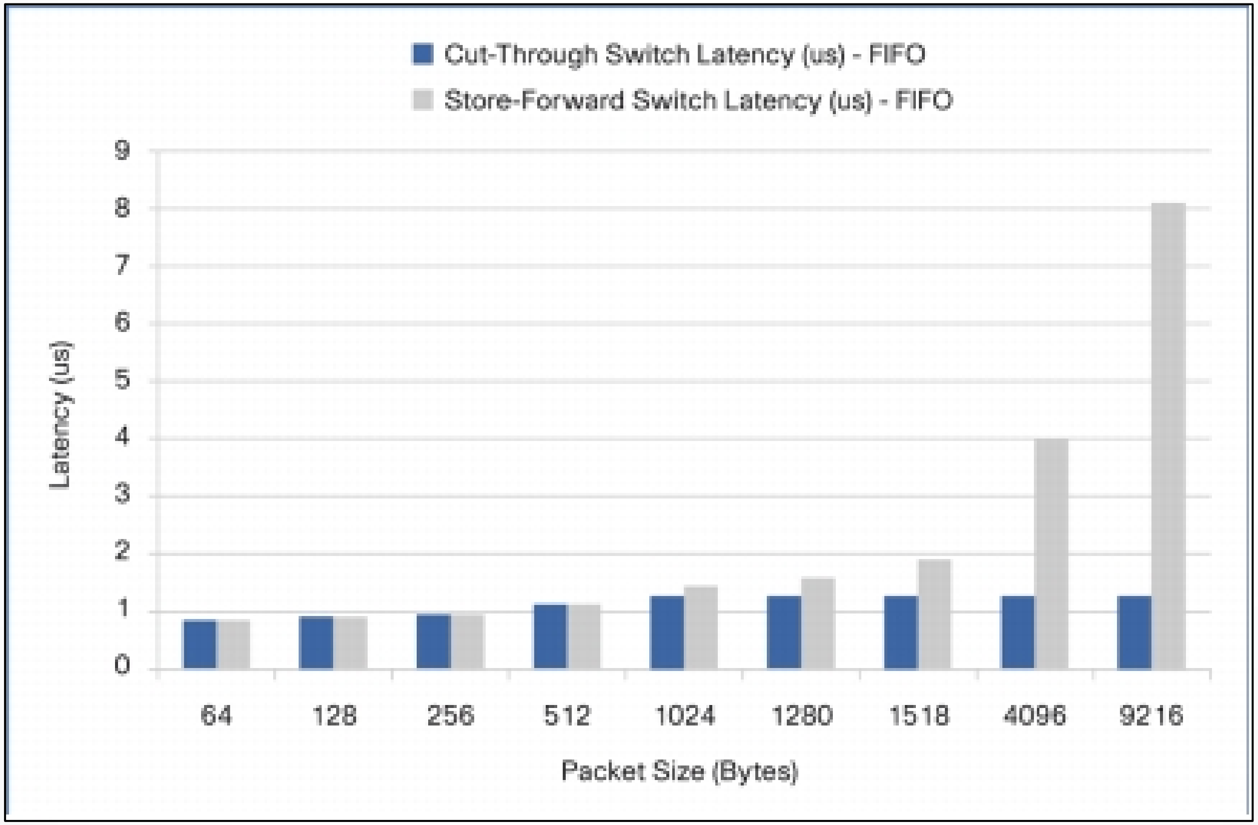
Cut-Trough [& Fragment free]: $\Delta_f = d + \Delta_h$

d – header bit serialization time

Processing begins immediately after receiving a *sufficient number* of bits of the packet header

Works in terms of fixed-length **cells**

Switching modes



Store-and-forward vs. Cut-Trough using 10 Gb channel

Why do we need in buffers?

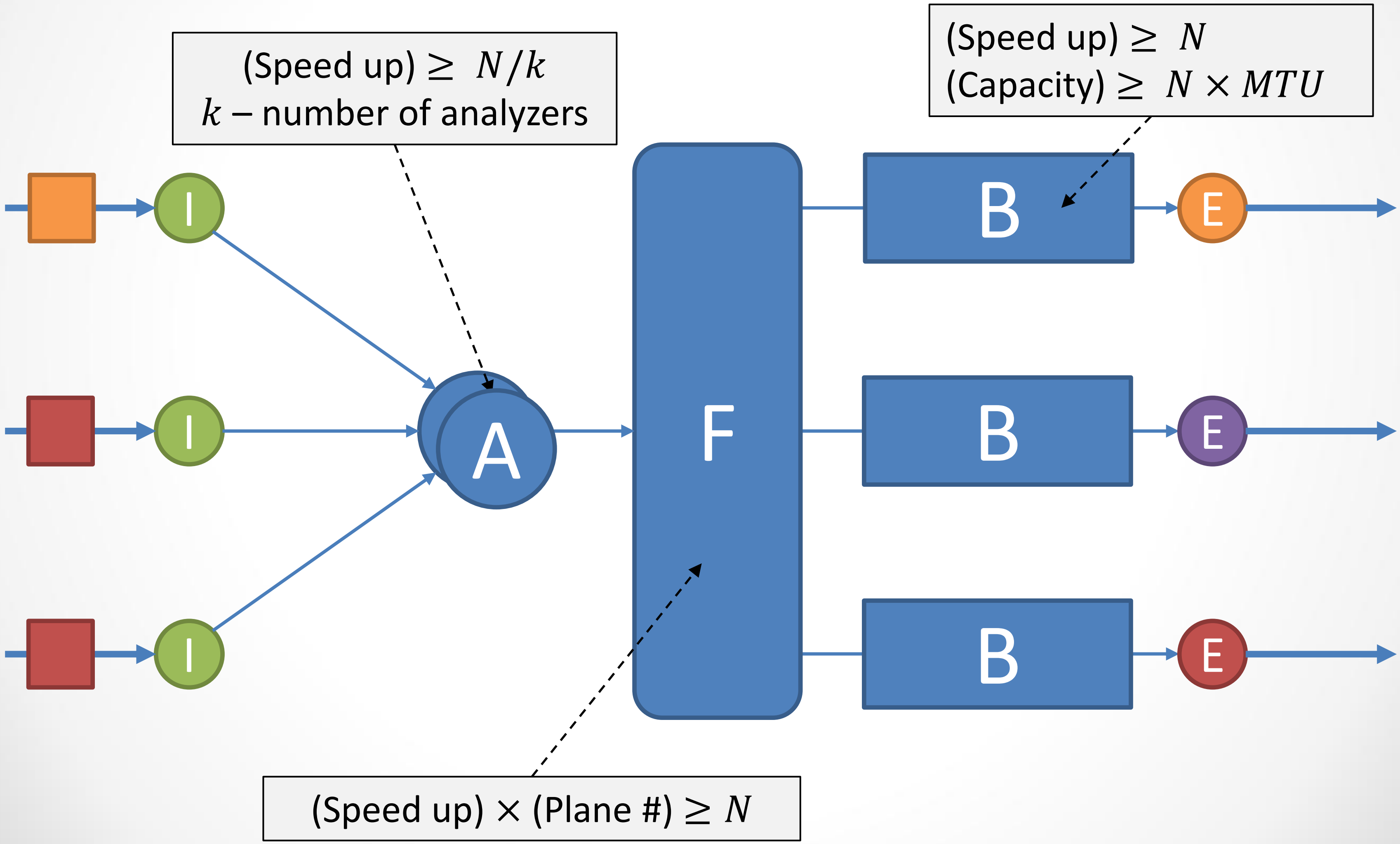
- Communication links cannot transmit multiple packets at once
- What should be done if several packets need to be sent through the same link?
  - Discard one of the packets
  - Store the packet in the buffer
- A classic switch design task is where to place buffers to assemble the best device:
  - Maximum performance
  - Good scalability
  - Minimum cost

Switch Performance Measurement

RFCs 2544 & 6815

- If the distribution of traffic is such that for each output port the total speed of data that needs to be transmitted through it does not exceed the speed of the communication link connected to it, then the distribution is called **admissible** for the switch
- If the switch does not drop packets when traffic arrives with an admissible distribution for it, then they say that this switch meets the requirements of a ***full backplane***

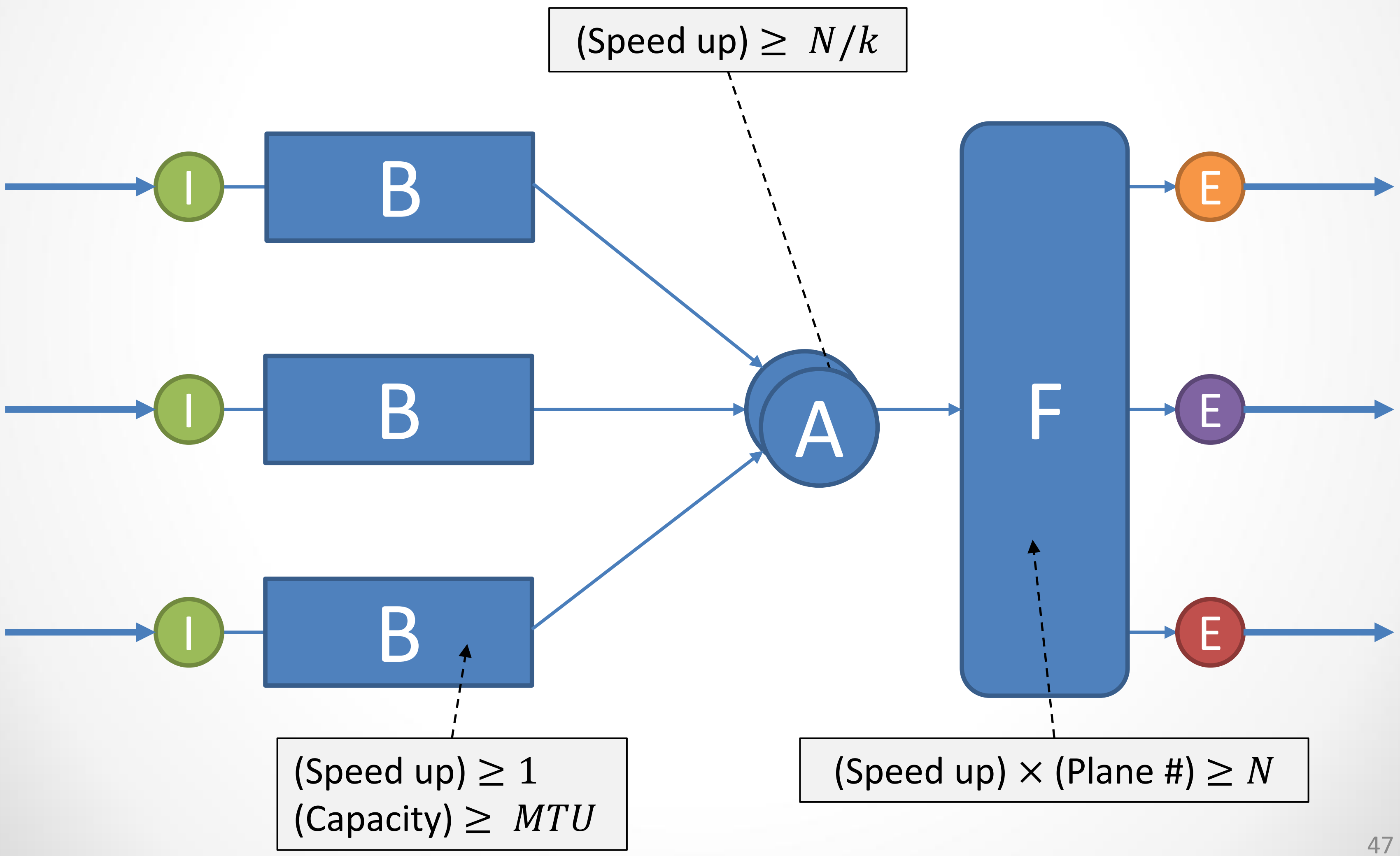
Output Queuing



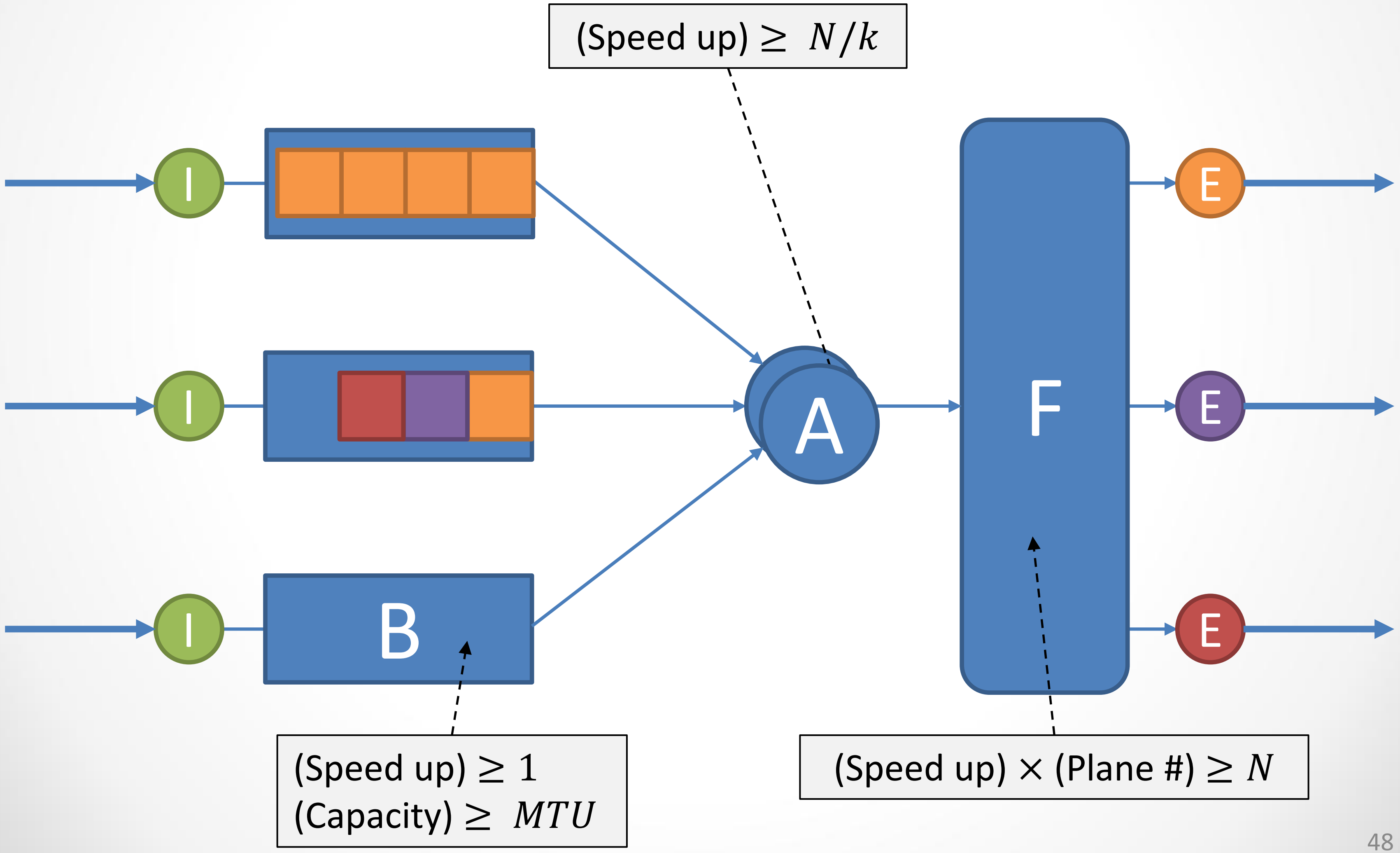
Switch performance

- Analyzers should work with acceleration N/k, where k is the number of analyzers
- The speed of the matrix should exceed the speed of communication links N times
- Buffer blocks must operate with an acceleration of at least N, and their volume must be at least N MTU
  – What should be the frequency and width of the memory access bus in order to support the operation of modern switches?

Input Queuing



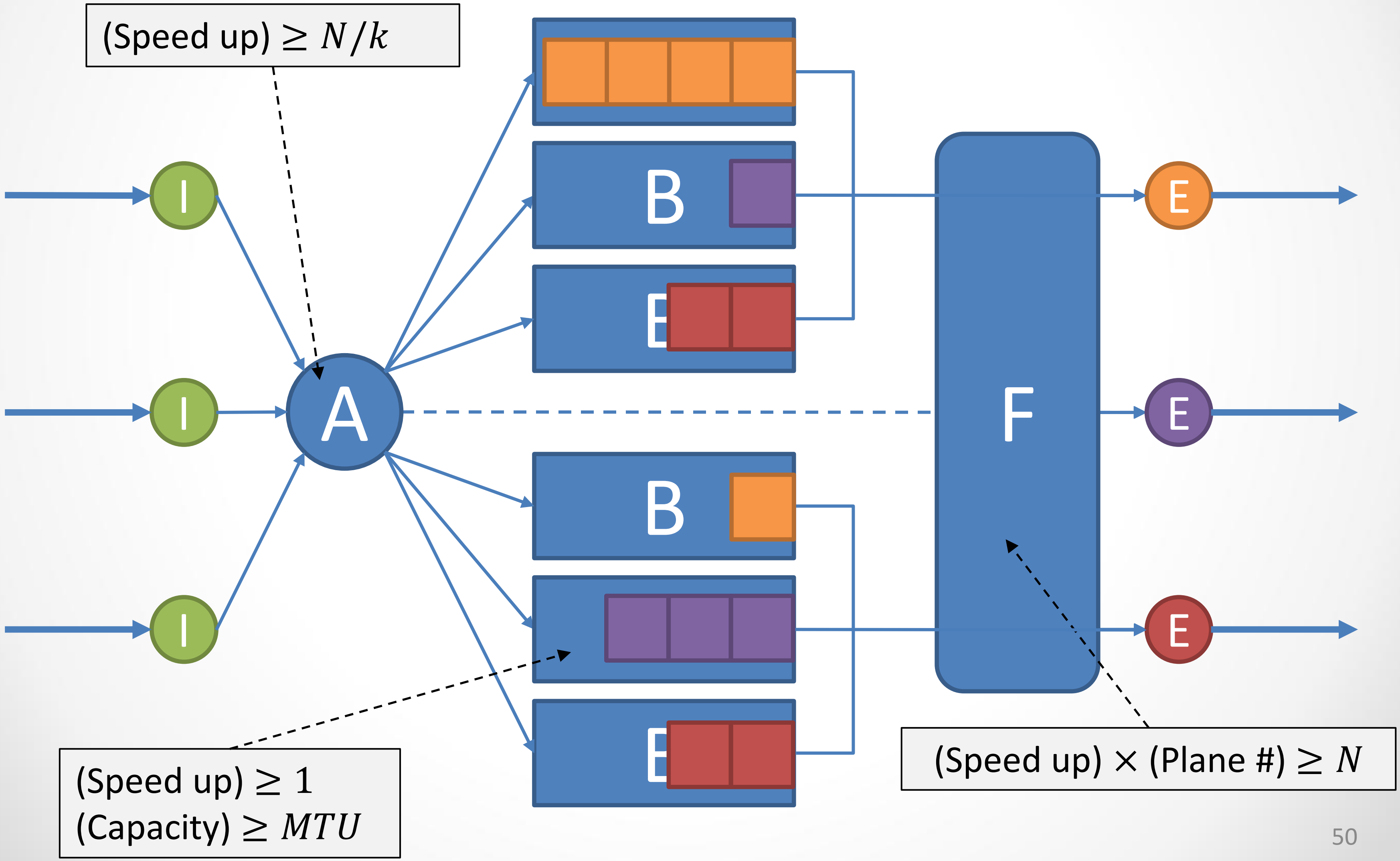
Input Queuing



Input Queuing

- No need for super-fast memory
- If packets from several input ports begin to compete for the same input of the switching fabric, packets that are behind them are blocked – ***Head Of Line (HOL) Blocking***
- With a uniform distribution of packet transmission routes, the IQ switch performance is less than 59% of the output queueing switch
- M. Karo; M. Hluchyj; S. Morgan
  Input Versus Output Queuing on a Space-Division Packet Switch (1987)

Virtual Output Queuing

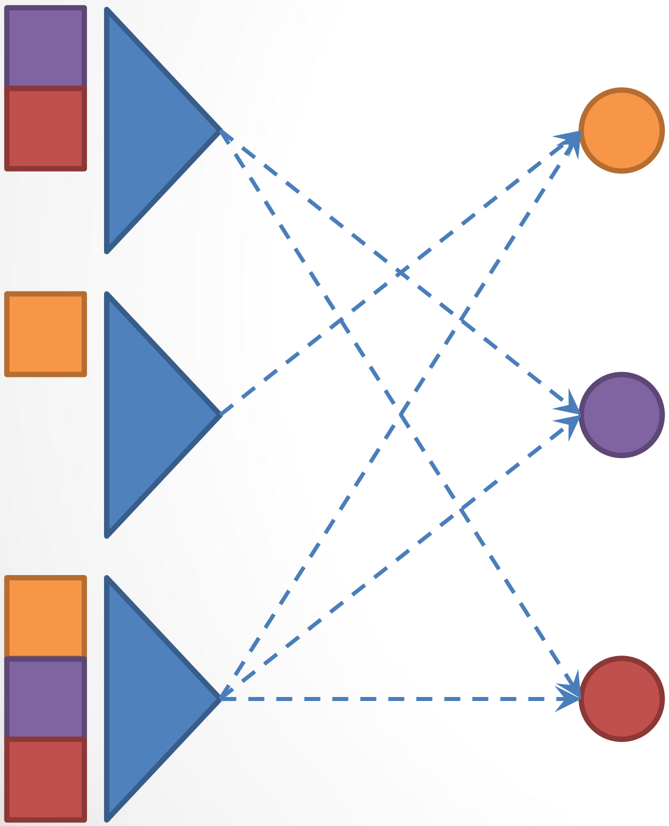


Virtual Output Queuing

N. McKeown A. Mekkittikul V. Anantharam J. Walrand
Achieving 100% Throughput in an Input-Queued Switch

- HOL blocking problem does not occur
- N^2 packet queues appear
- A switching matrix with N^2 inputs is not required, but a fast arbitration algorithm is required to select the desired queue on each of the interfaces
- Complex dynamic arbitration algorithms are difficult to implement in hardware

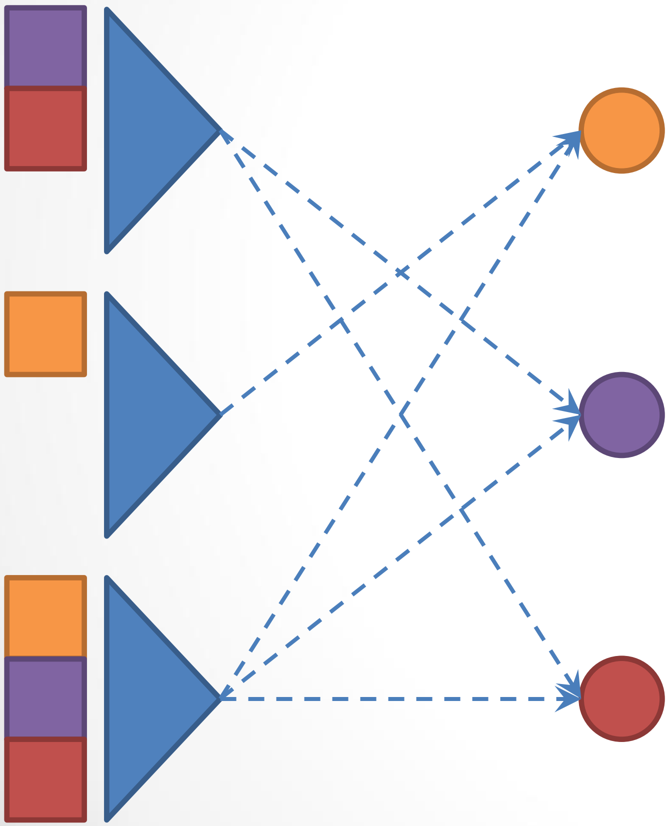
Switch Matrix Arbitration Algorithms



Model Assumptions:

- The commutation matrix has N **planes**
- Packets are divided into **cells** of a fixed length at the entrance and are restored at the exit of the matrix
- The switching matrix works on ticks - at each tick it can have one cell from each input and place one cell on each output

Switch Matrix Arbitration Algorithms



Find a switching matrix arbitration algorithm such that:

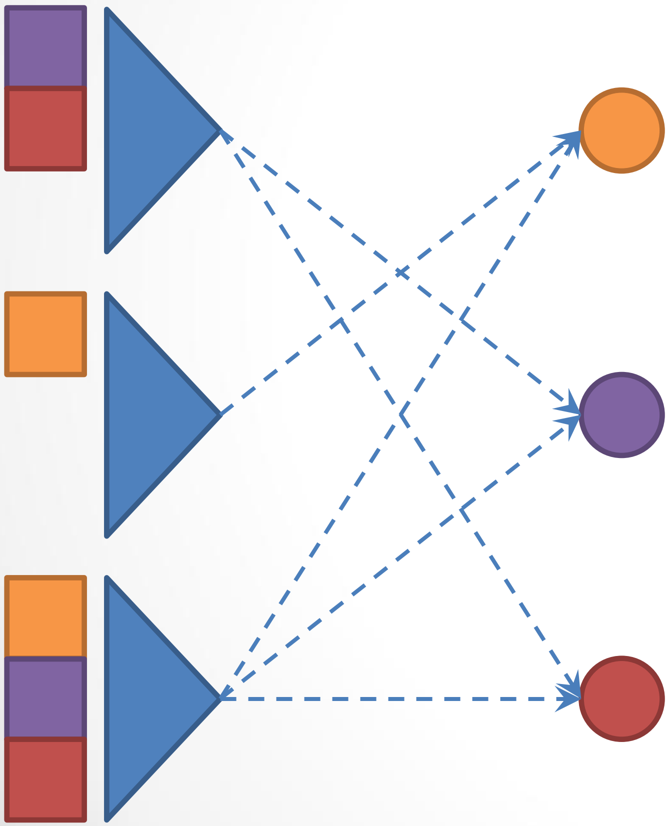
(Performance)

Full backplane requirements would be met without matrix acceleration

(Fairness)

Throttling traffic flows would never occur

Switch Matrix Arbitration Algorithms



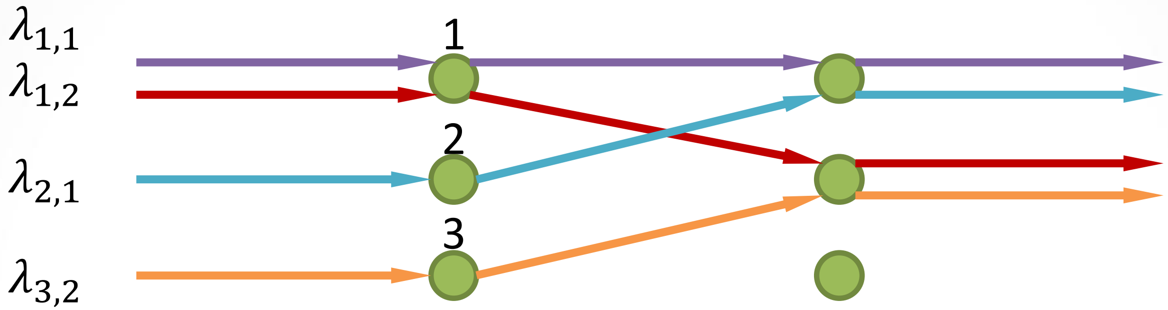
Hypothesis:

- A suitable algorithm should transmit as many cells as possible on each clock cycle of the matrix - the task of finding **the maximum matching**

The hypothesis is wrong - the algorithm has neither high performance nor fair planning

Inapplicability of the algorithm for the maximum matching

Performance



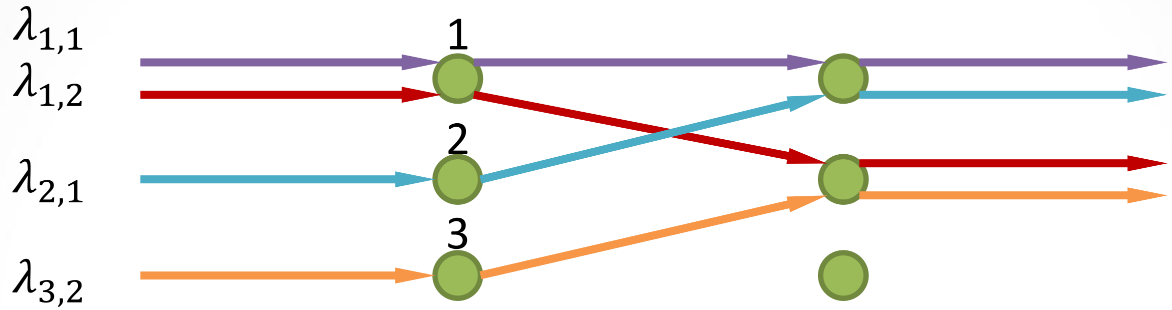
(A1) Let the throughput of each communication link be 1, then the flow rates are equal to:

$$\lambda_{1,1} = \lambda_{1,2} = \lambda_{2,1} = \lambda_{3,2} = \frac{1}{2} - \delta$$

(A2) Let in situations where there are several maximum matching the algorithm selects one of them with equal probability

Inapplicability of the algorithm for the maximum matching

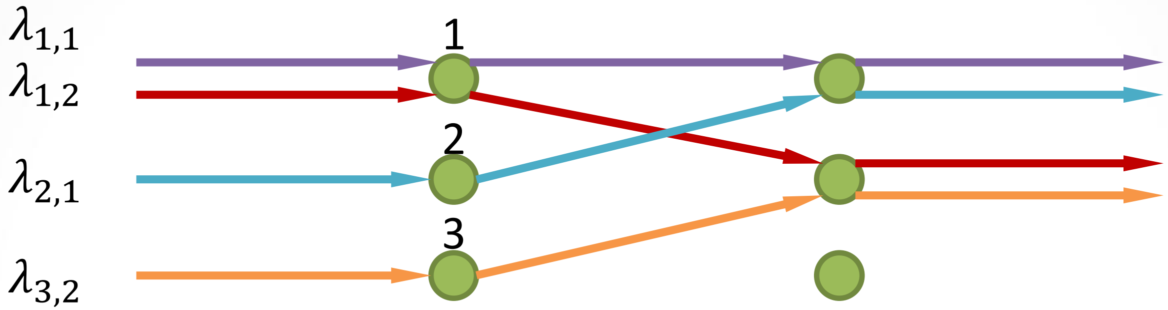
Performance



- Calculate the maximum transfer rate for input 1:
- Let flows $\lambda_{2,1}$ and $\lambda_{3,2}$ are ready to transmit:
  - From **(A1)** the probability of such an event is $(1/2-\delta)$^2
  - Then there are the three maximum matching : $\lambda_{1,1}\&\lambda_{3,2}$; $\lambda_{1,2}\&\lambda_{2,1}$; $\lambda_{2,1}\&\lambda_{3,2}$
  - From **(A2)** probability of choosing the first input is 2/3
- If at least one of $\lambda_{2,1}$ and $\lambda_{3,2}$ is not ready to transmit:
  - The algorithm selects input 1 with probability 1

Inapplicability of the algorithm for the maximum matching

Performance



- According to the full probability formula, the highest data transmission rate for data, passing through the input 1:

$$\frac{2}{3}\left(\frac{1}{2}-\delta\right)^2+\left(1-\left(\frac{1}{2}-\delta\right)^2\right)=1-\frac{1}{3}\left(\frac{1}{2}-\delta\right)^2$$

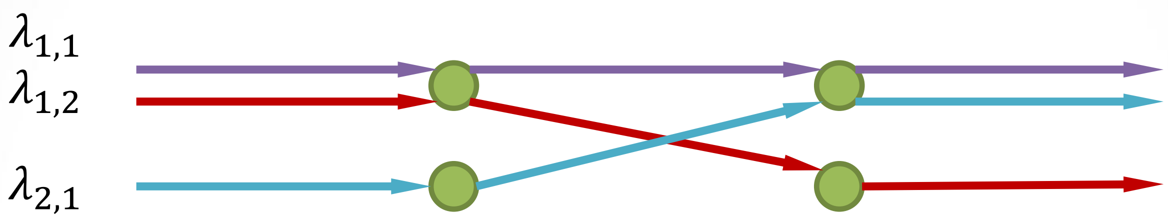
- Full backplane requirements are violated if the input rate of input 1 exceeds the highest transmission rate *:*

$$1-2\delta>1-\frac{1}{3}\left(\frac{1}{2}-\delta\right)^2$$

- Achieved providing $\delta<0.0358$

Inapplicability of the algorithm for the maximum matching

Fairness



- Let the flow rate to be equal to the channel bandwidth $\lambda_{1,1} = \lambda_{1,2} = \lambda_{2,1} = 1$
- The algorithm for finding the maximum matching will always select the flows $\lambda_{1,2}$ and $\lambda_{2,1}$
- Flow $\lambda_{1,1}$ will experience **the starvation**

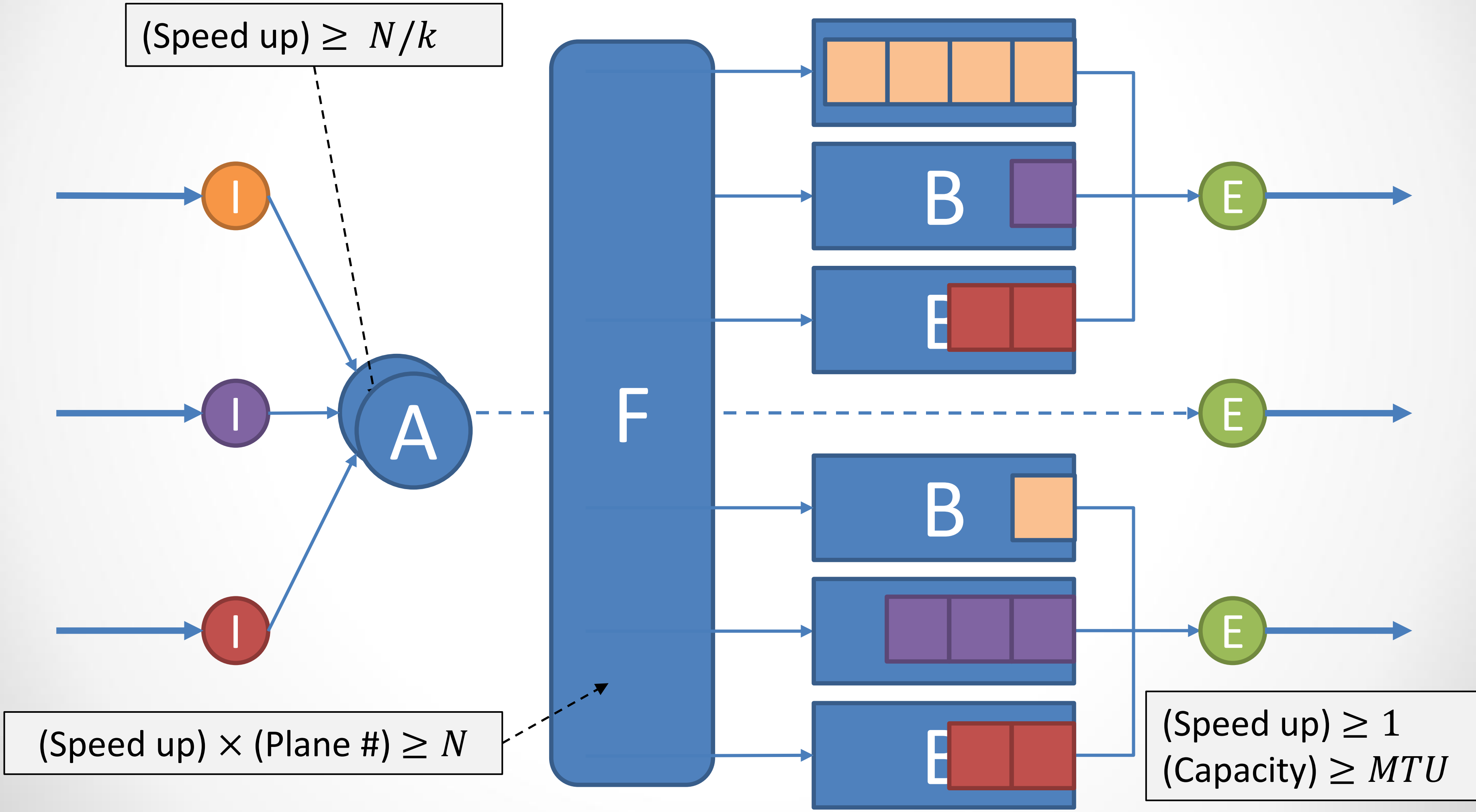
Arbitration algorithm Oldest Cell First

- Each queue is assigned its own weight coefficient - the number of ticks when the cell in the leading position was not selected
- The arbitration algorithm chooses matching in such a way as to maximize the sum of the weights of the selected queues

Disadvantages:

- High implementation complexity

Multiple Output Queuing



Multiple Output Queuing

- No need for switching matrix arbitration algorithm
- There is a need for more sophisticated logic for distributing packets into queues at the output of the switching matrix and an additional packet scheduler to select data from these queues

The practice of switching device design

When the rubber meets the road…

- Models rarely used in pure form
- Manufacturers are trying to find trade-offs between cost and utilization of channels under different loads
- Today, the most common model are Combined Input-Output Queuing (CIOQ)
- Instead of a full-fledged switching matrix, its simplifications are often used - knockout switches or networks of switches