# High Performance Switching and Routing

*Telecom Center Workshop: Sept 4, 1997.*


**Nick McKeown**

**Assistant Professor of Electrical Engineering and Computer Science**

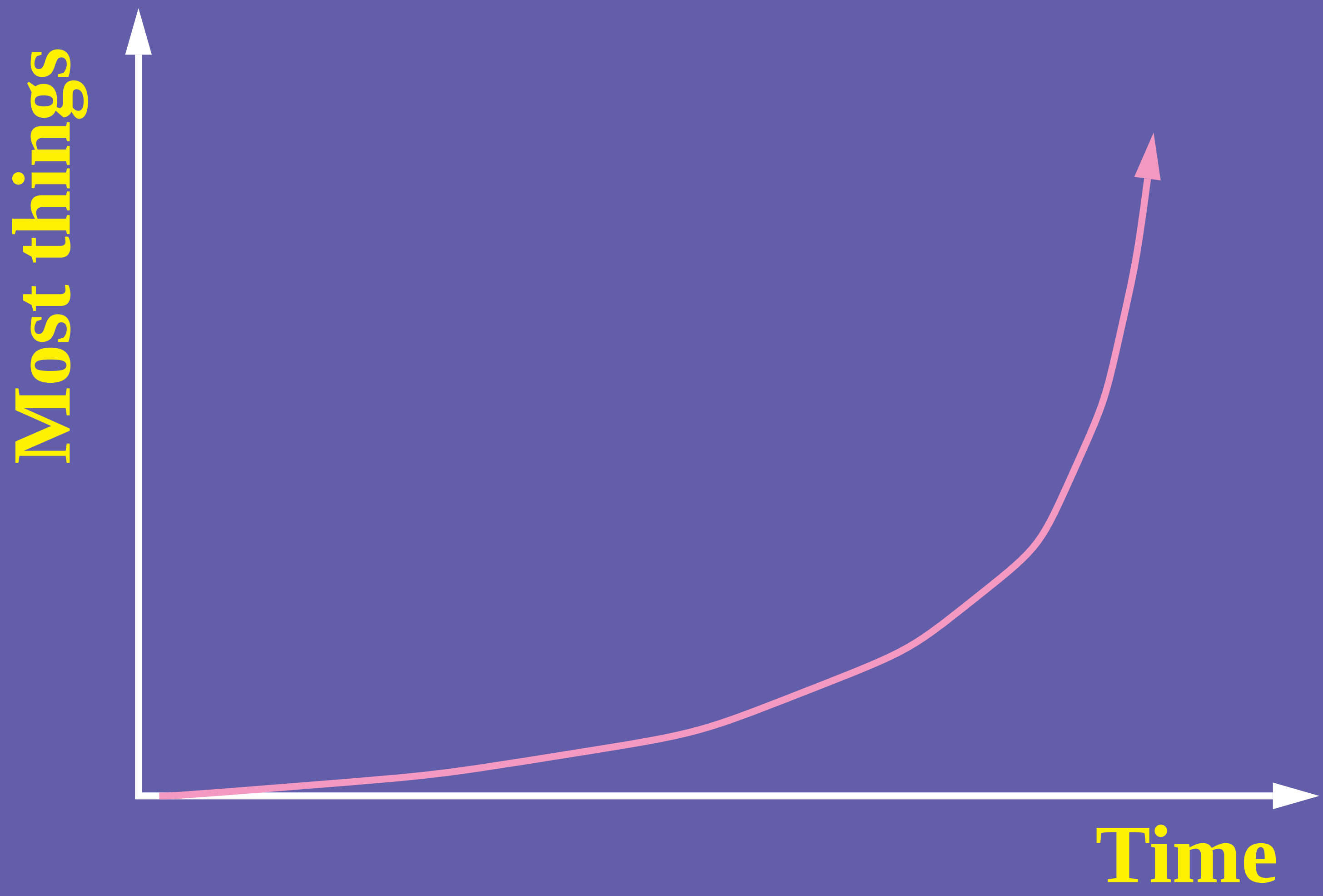
**nickm@ee.stanford.edu**
**http://ee.stanford.edu/~nickm**

# Our Group

*Shang-tse "Da" Chuang, Ken Chang, Pankaj Gupta, Youngmi Joo, Steve Lin, Adisak Mekkittikul, Nick McKeown, Rolf Muralt, Kanta Yamamoto (visitor).*

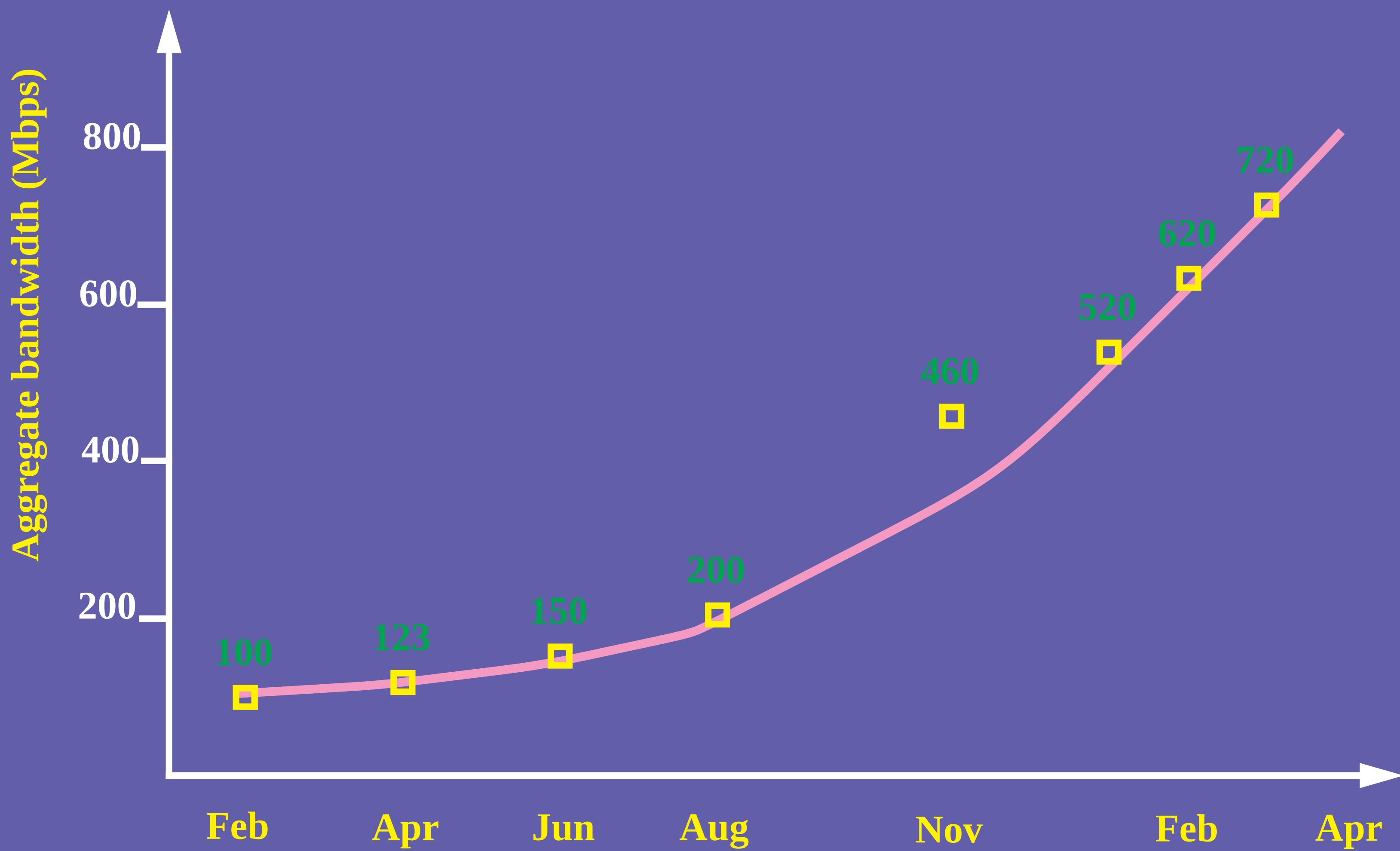
1. The Demand for Bandwidth
2. The Shortage of Switching/Routing Capacity
3. The Architecture of Switches and Routers
4. Some (of our) solutions

# What's the Problem?



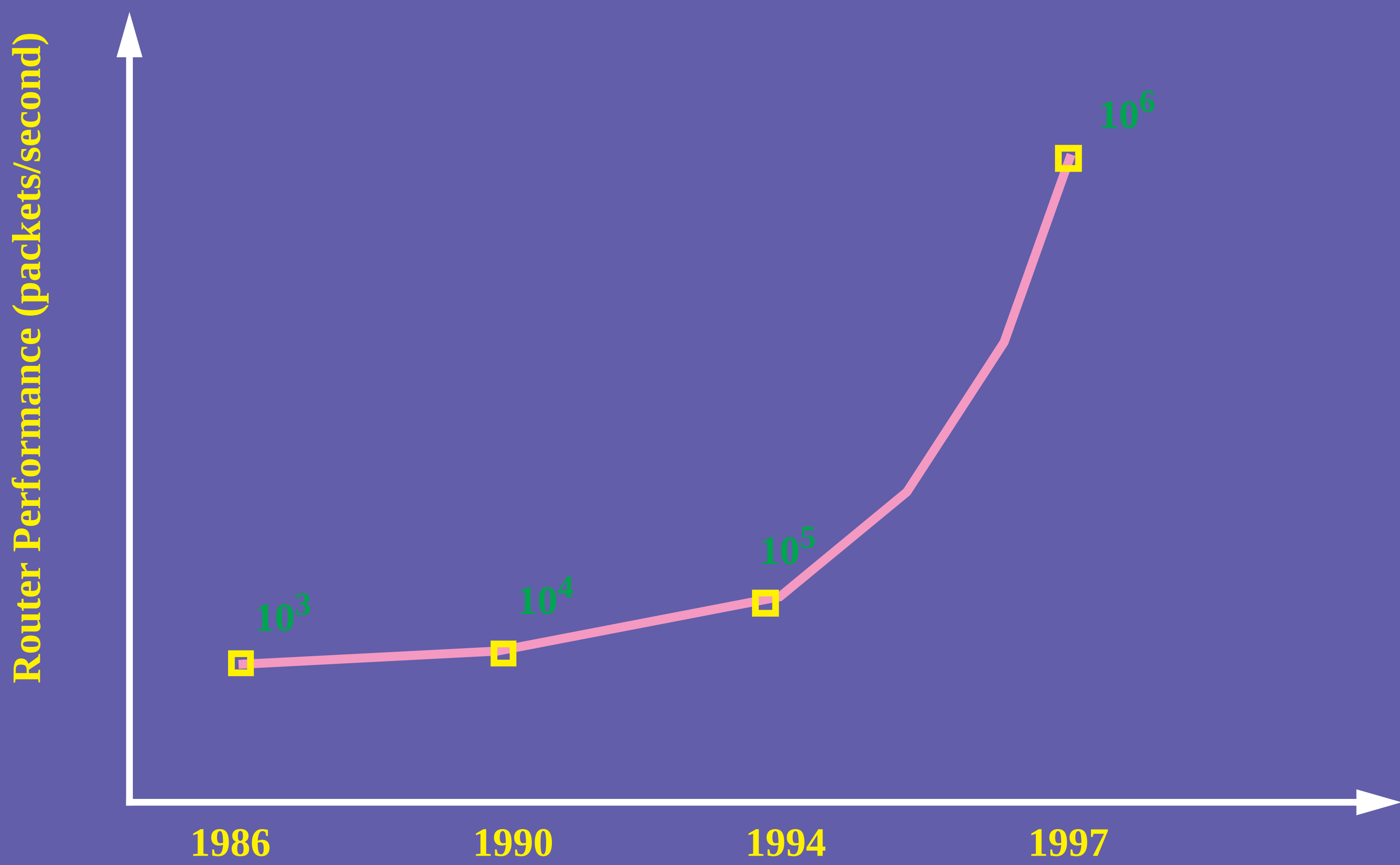
# The demand

*The San Jose NAP*

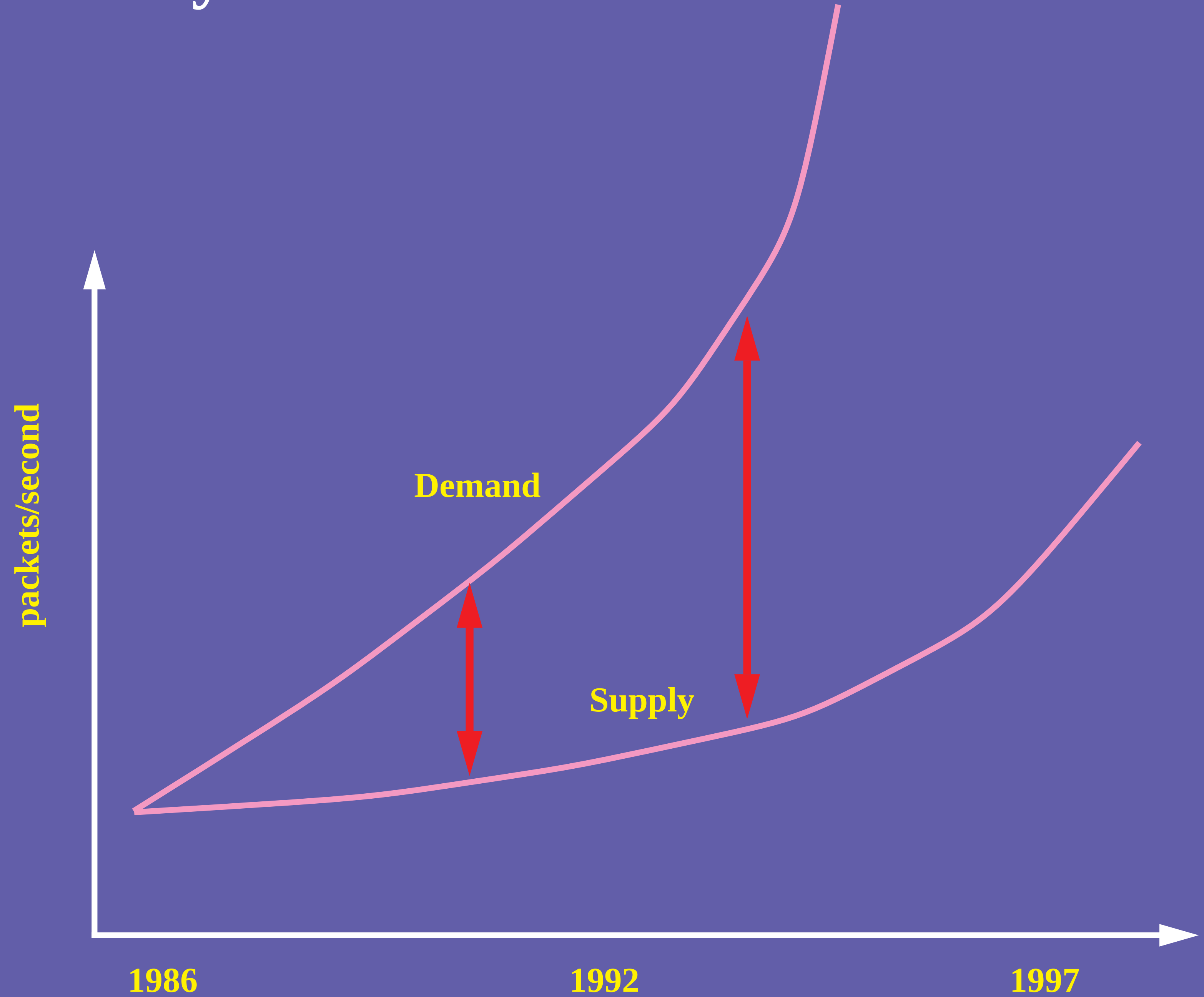



***Source: http://www.mfsdatanet.com/MAE/west.stats.html***

# The supply


Router Performance (packets/second)
$10^3$
$10^4$
$10^5$
$10^6$
1986
1990
1994
1997

# Why we need faster switches/routers

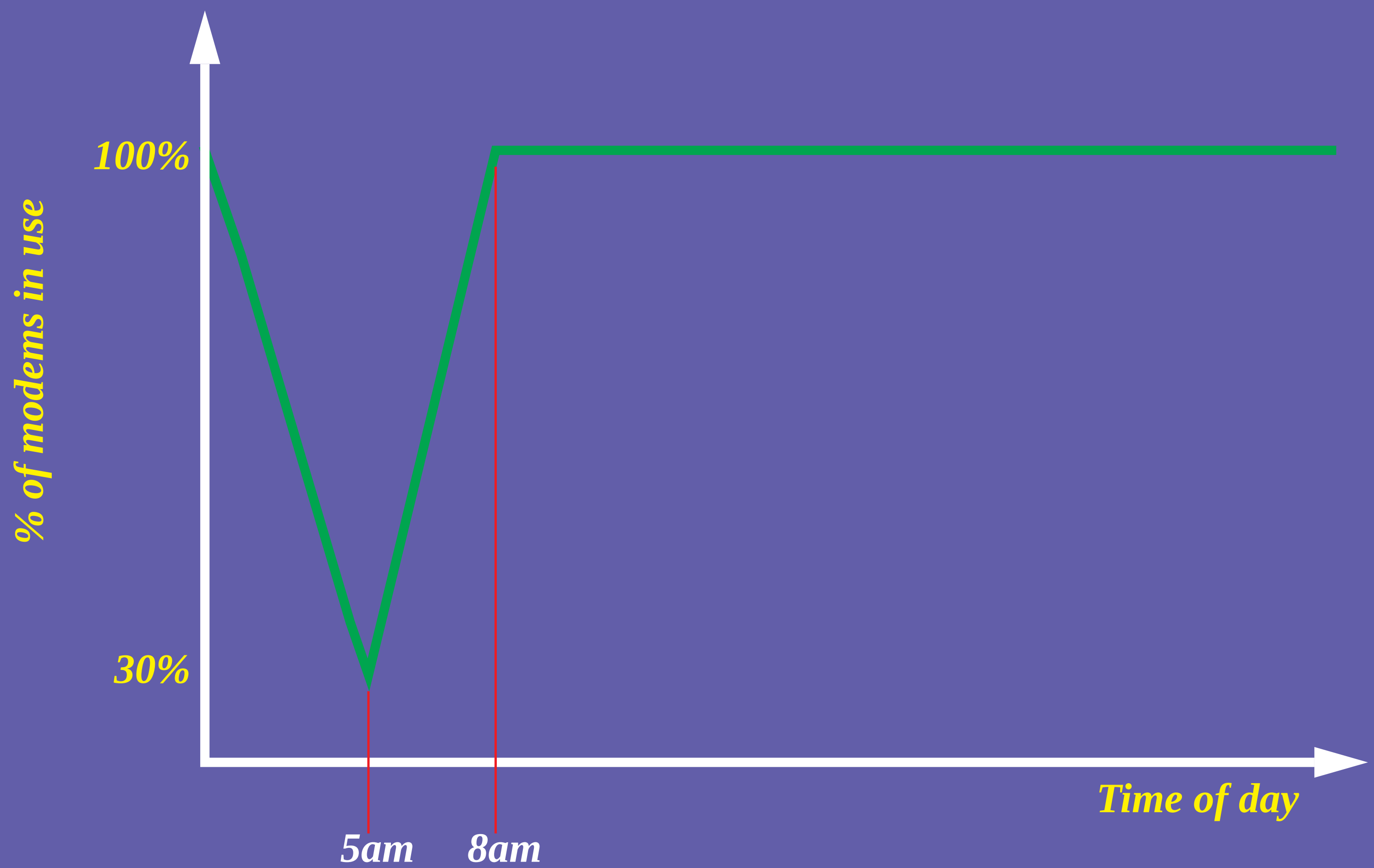


# Why the growth?

- **Exponential growth in the number of users.**
- **Exponential growth in traffic per user per hour.**
- **Linear growth in hours per user per day.**

# Dialup Demand

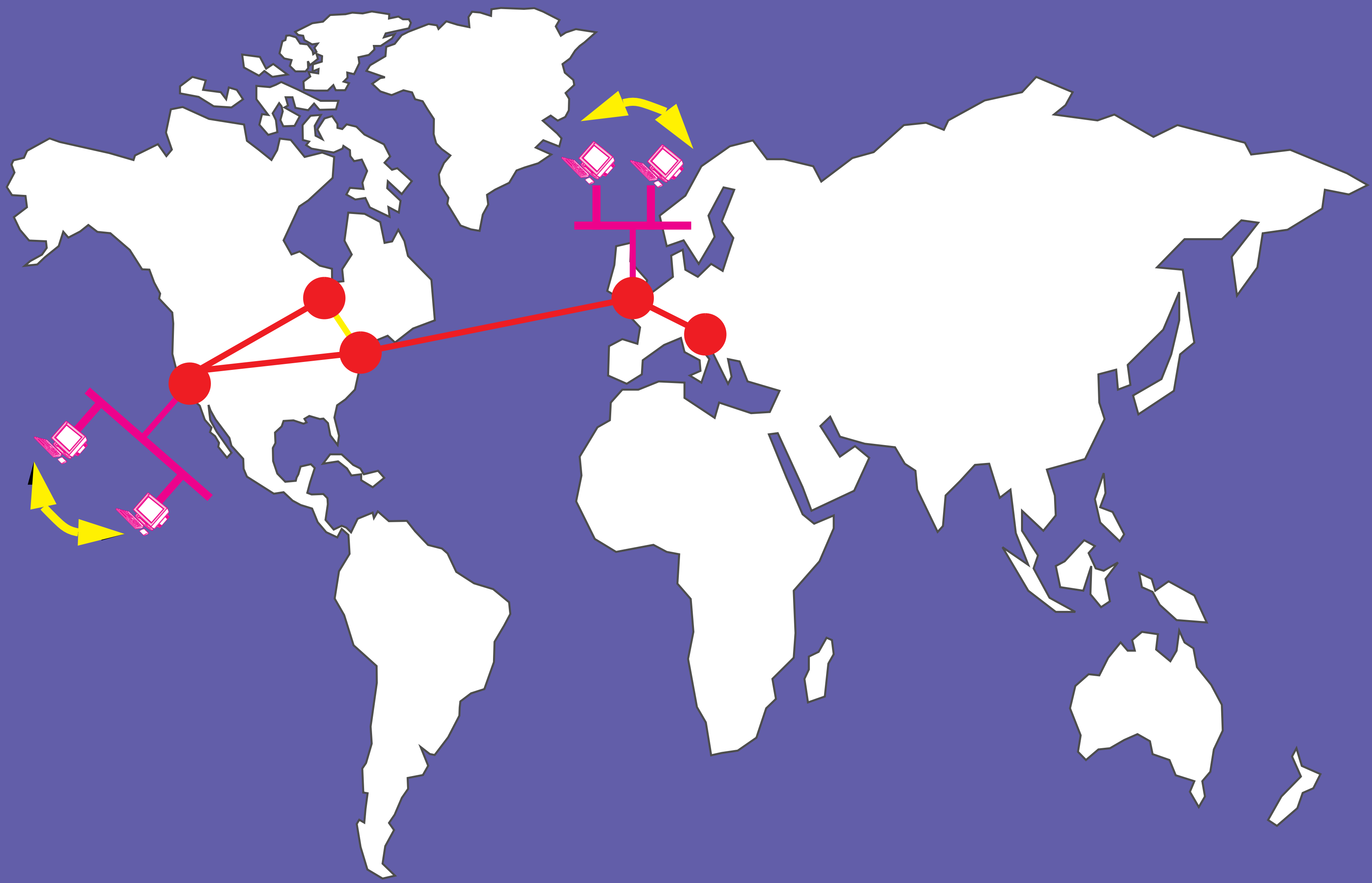
***Modem usage at U.C. Berkeley***




# *"America on Hold"*

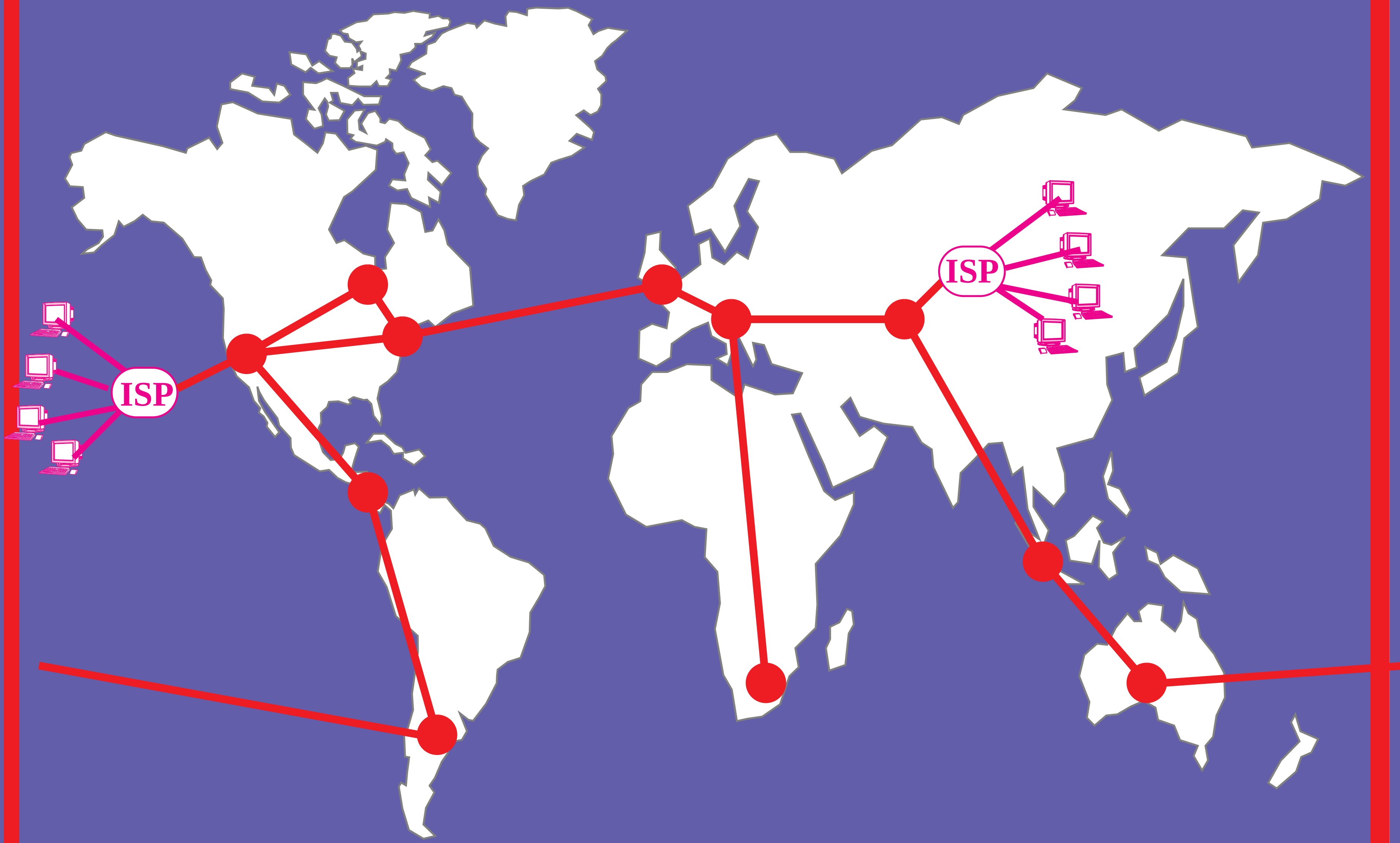
# Traffic Inversion

*10 years ago*

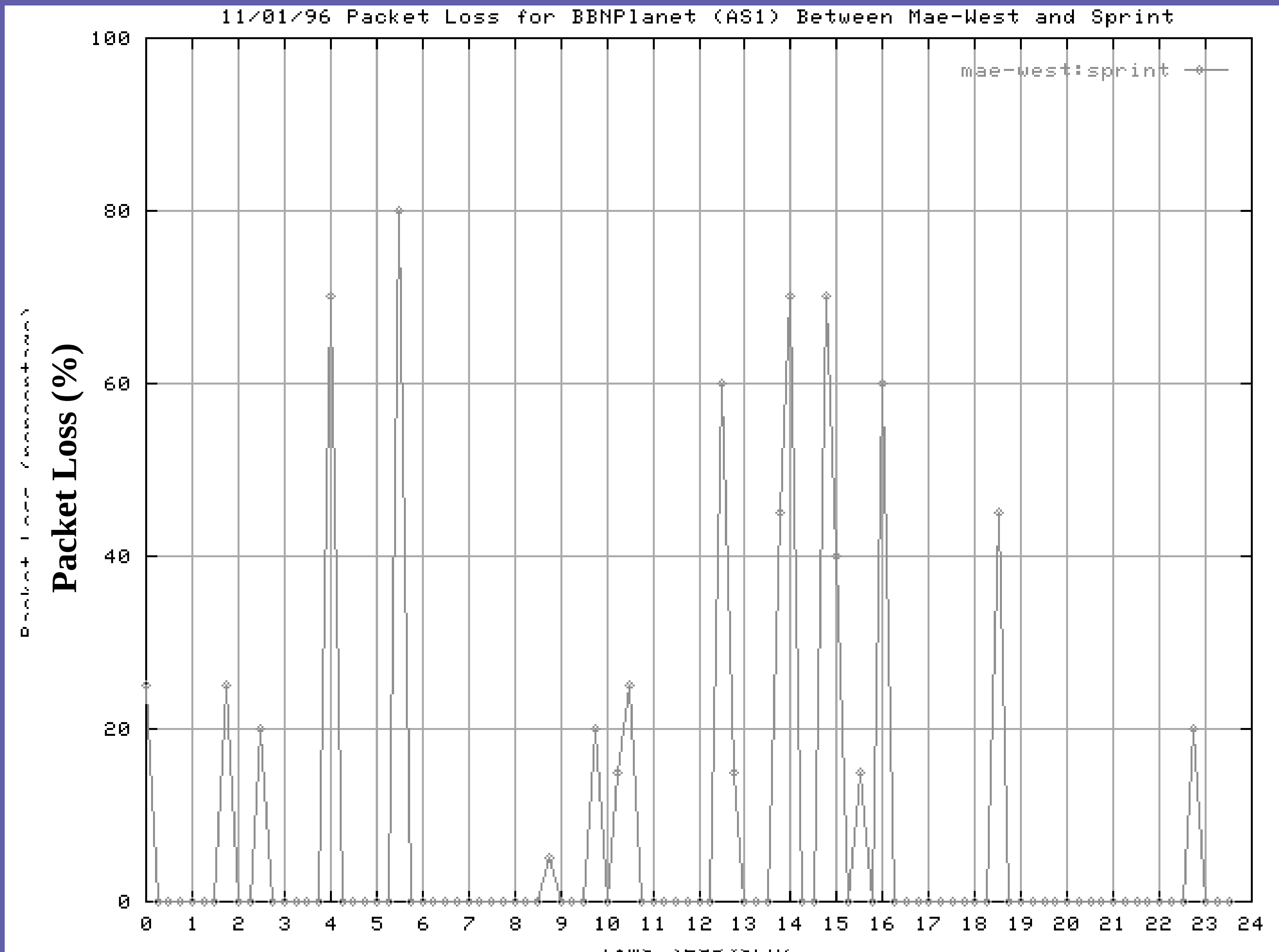


# Traffic Inversion

*Today*

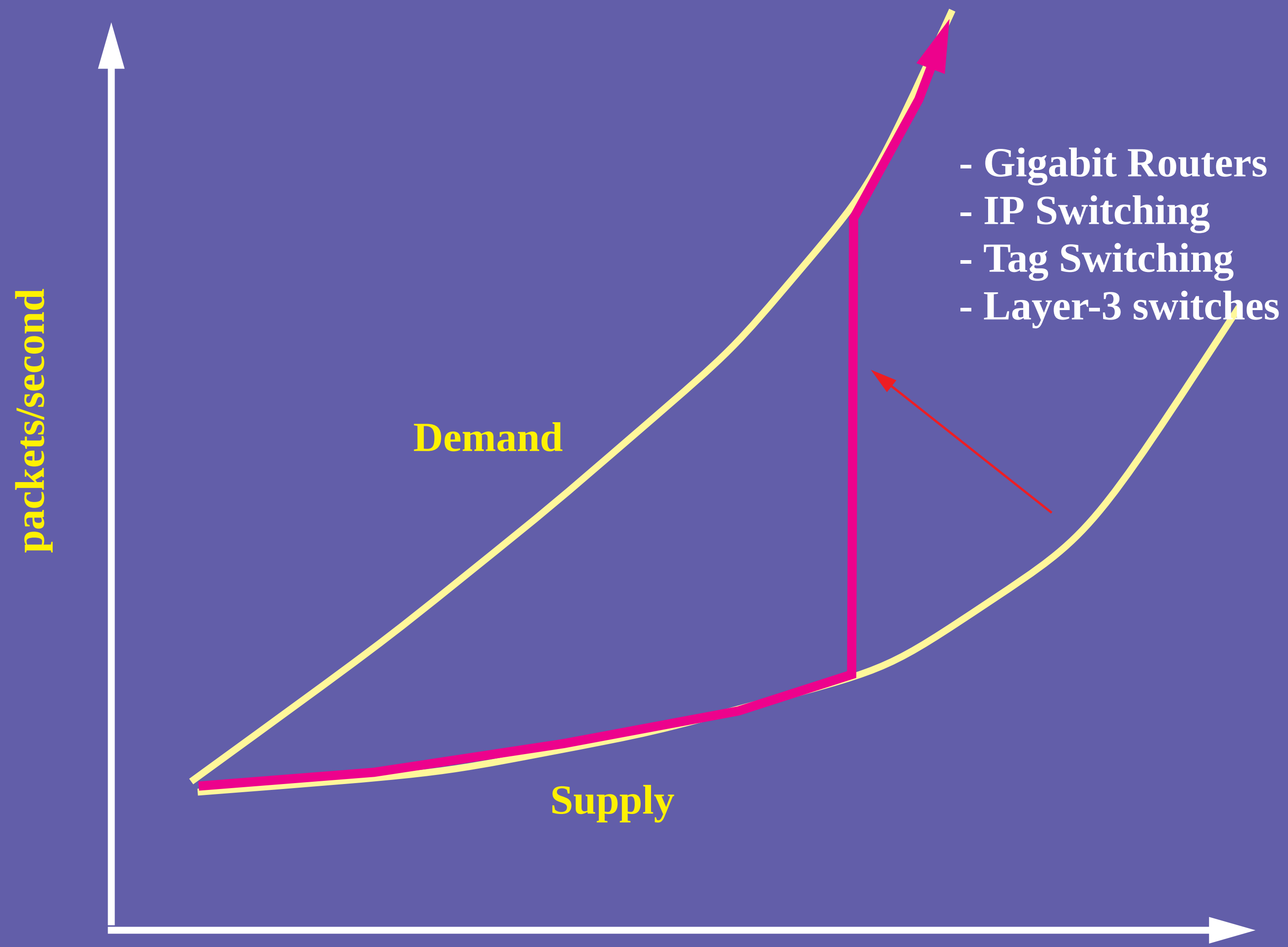


# Why is this a problem?




**November 1st, 1996**

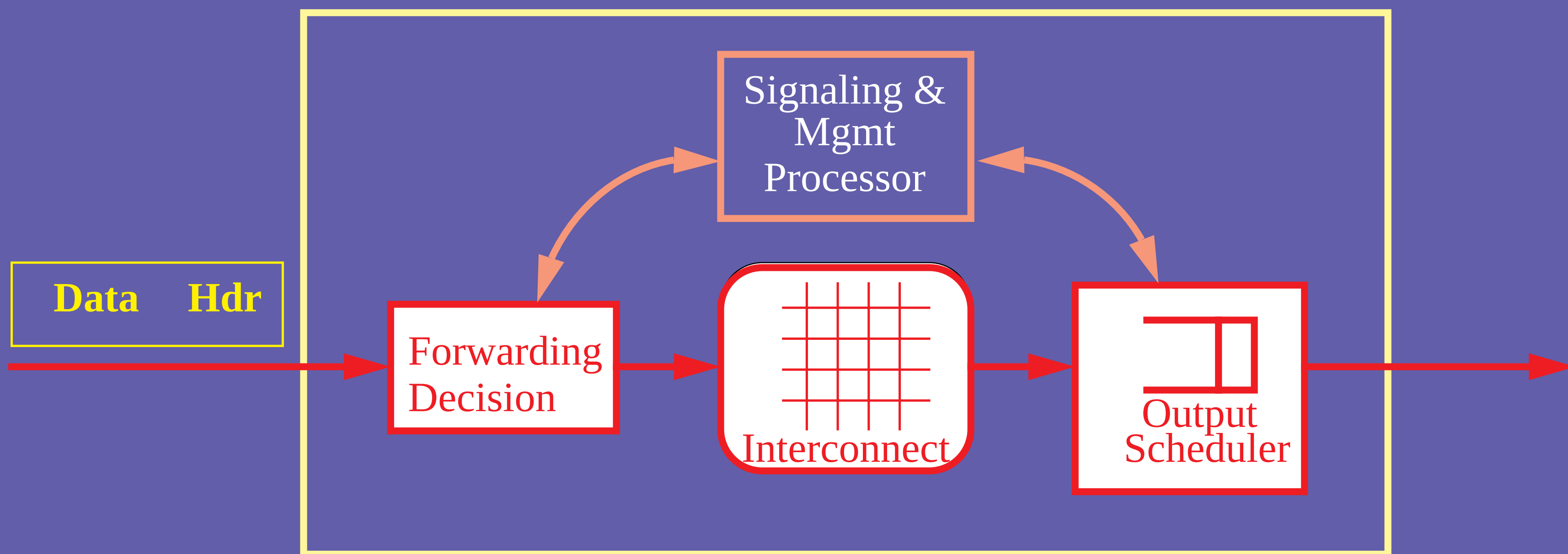
# The race is on...



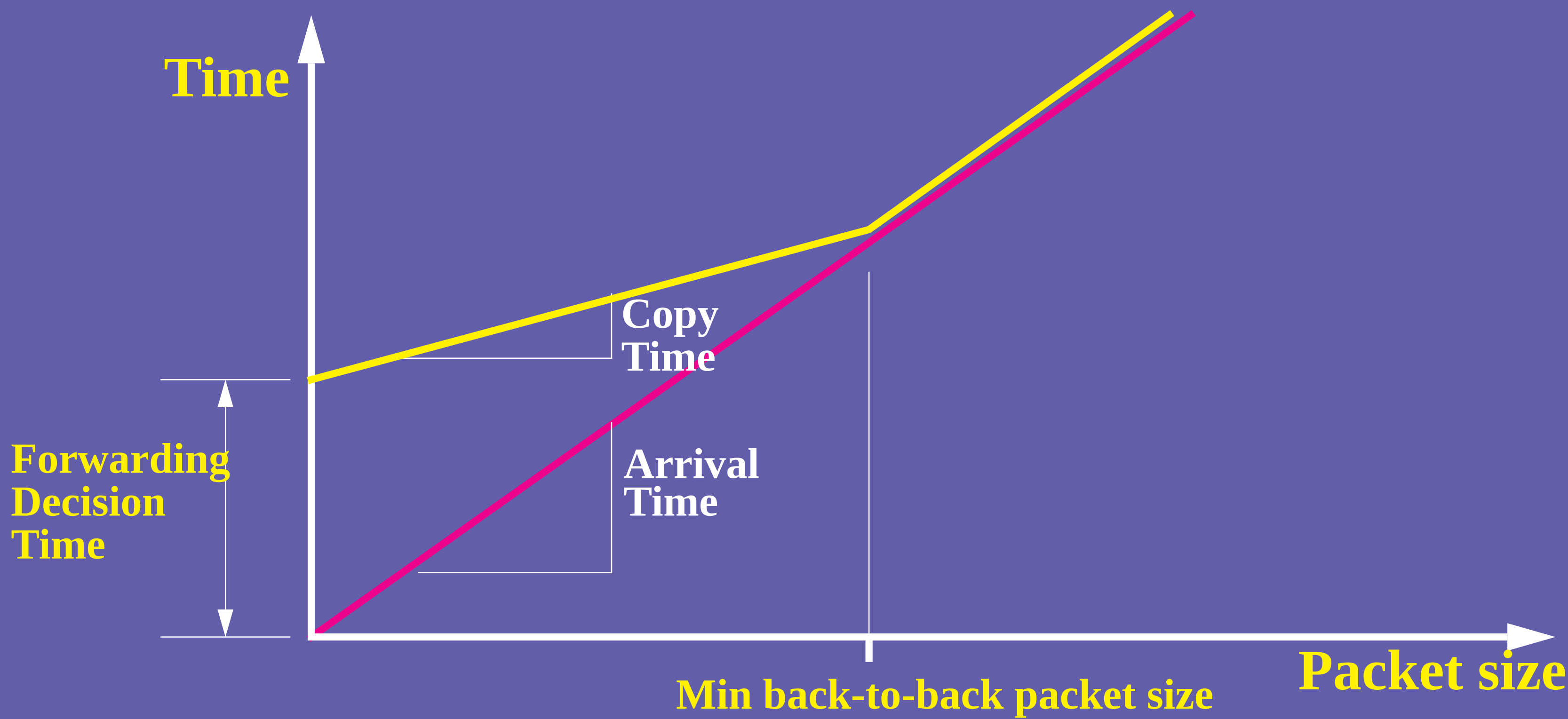
1. The Demand for Bandwidth
2. The Shortage of Switching/Routing Capacity
3. The Architecture of Switches and Routers
4. Some (of our) solutions

# The Architecture of Switches and Routers

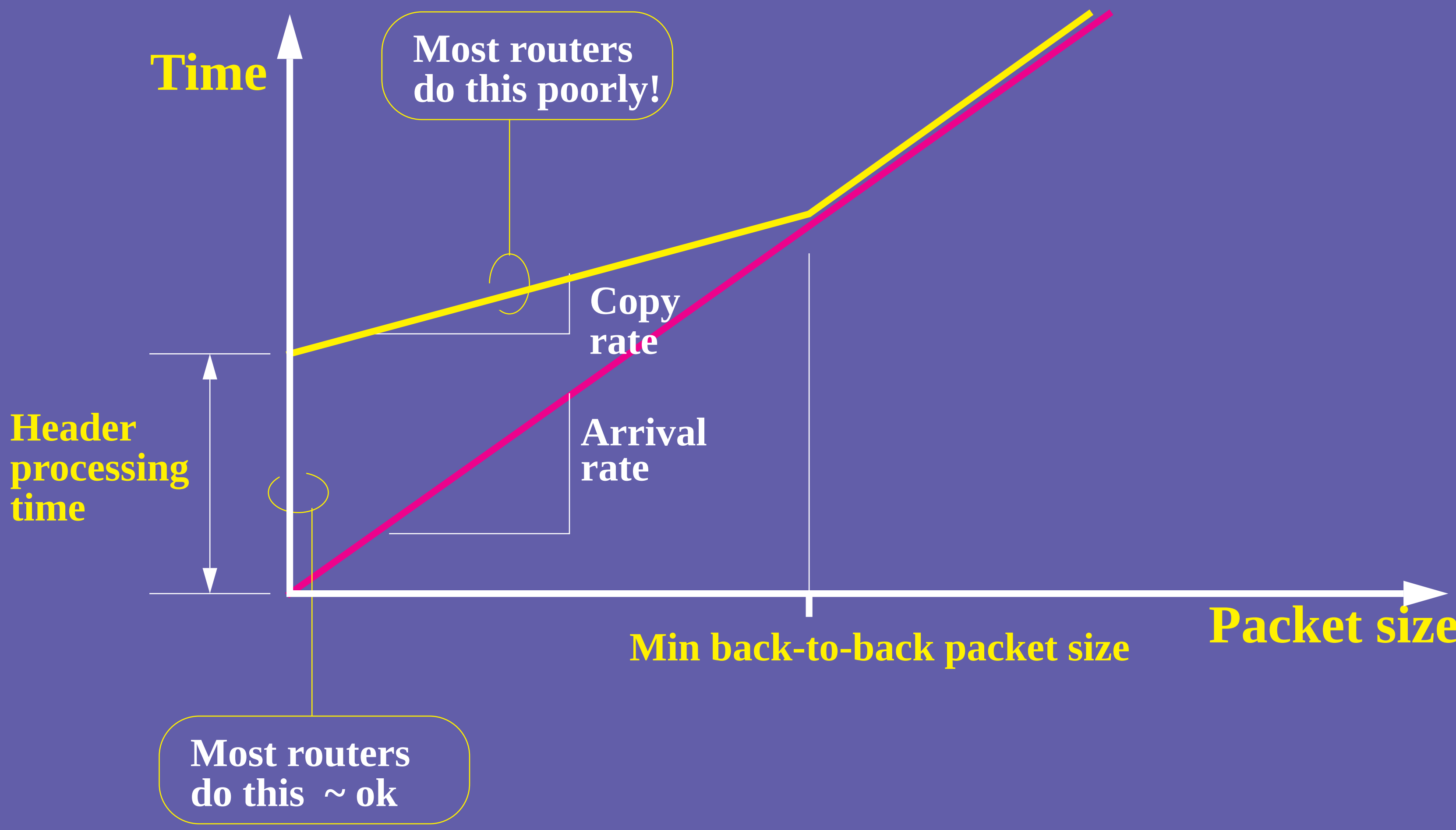
Generic Packet Processor:
(e.g. IP Router, ATM Switch, LAN Switch)



# Performance of IP Routers

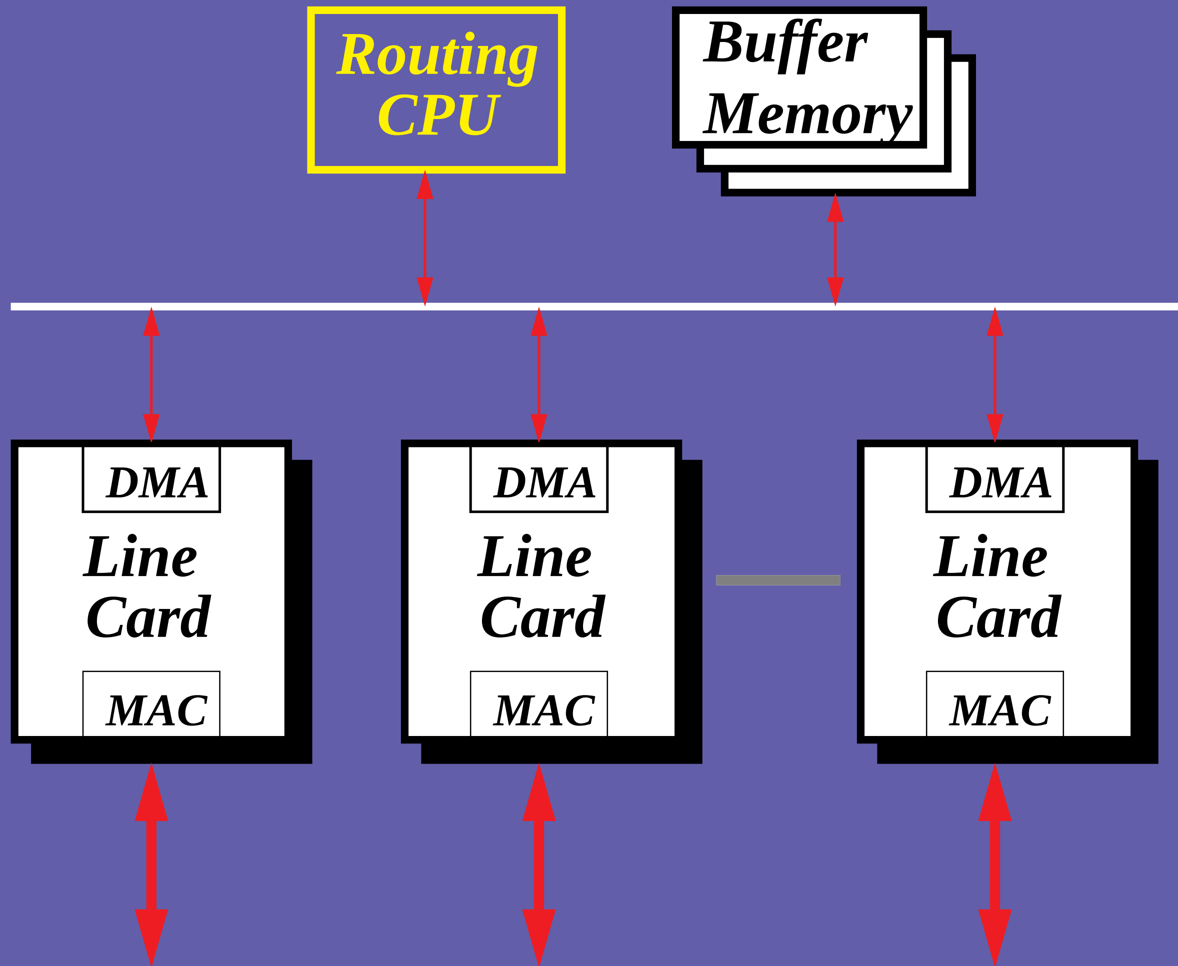


# Performance of IP Routers



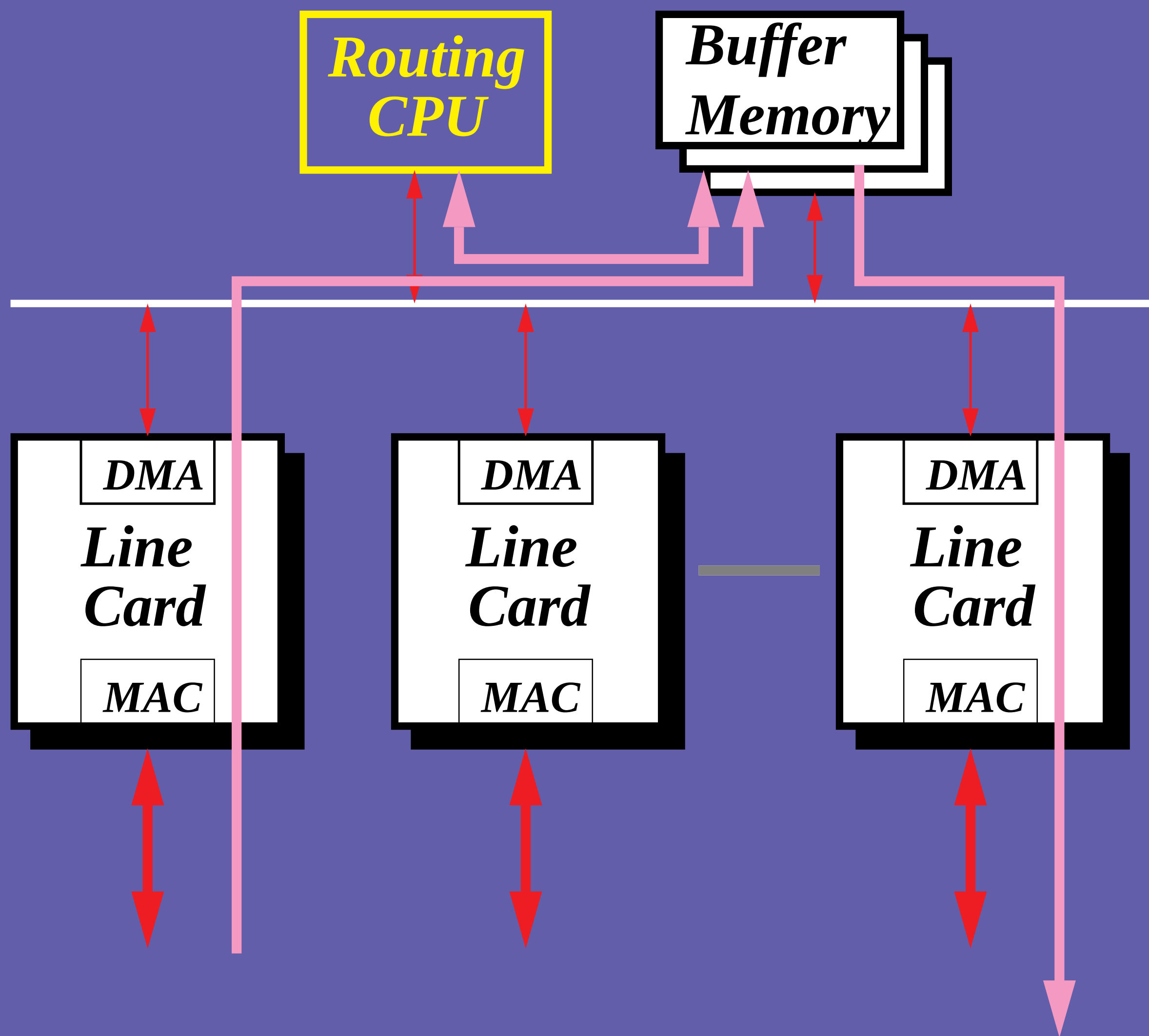
# The Evolution of Routers

*The first shared memory routers*



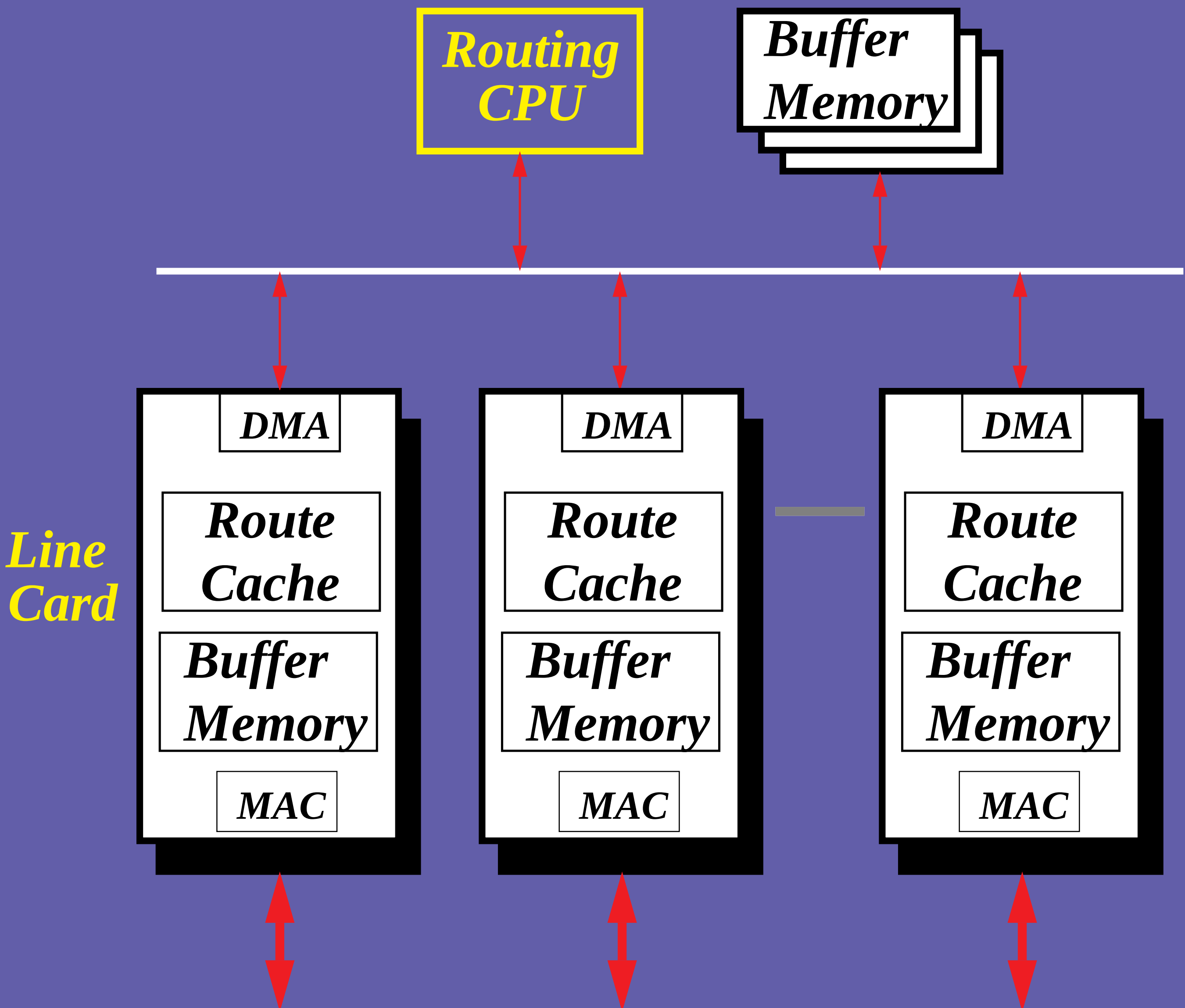
# The Evolution of Routers

*The first shared memory routers*



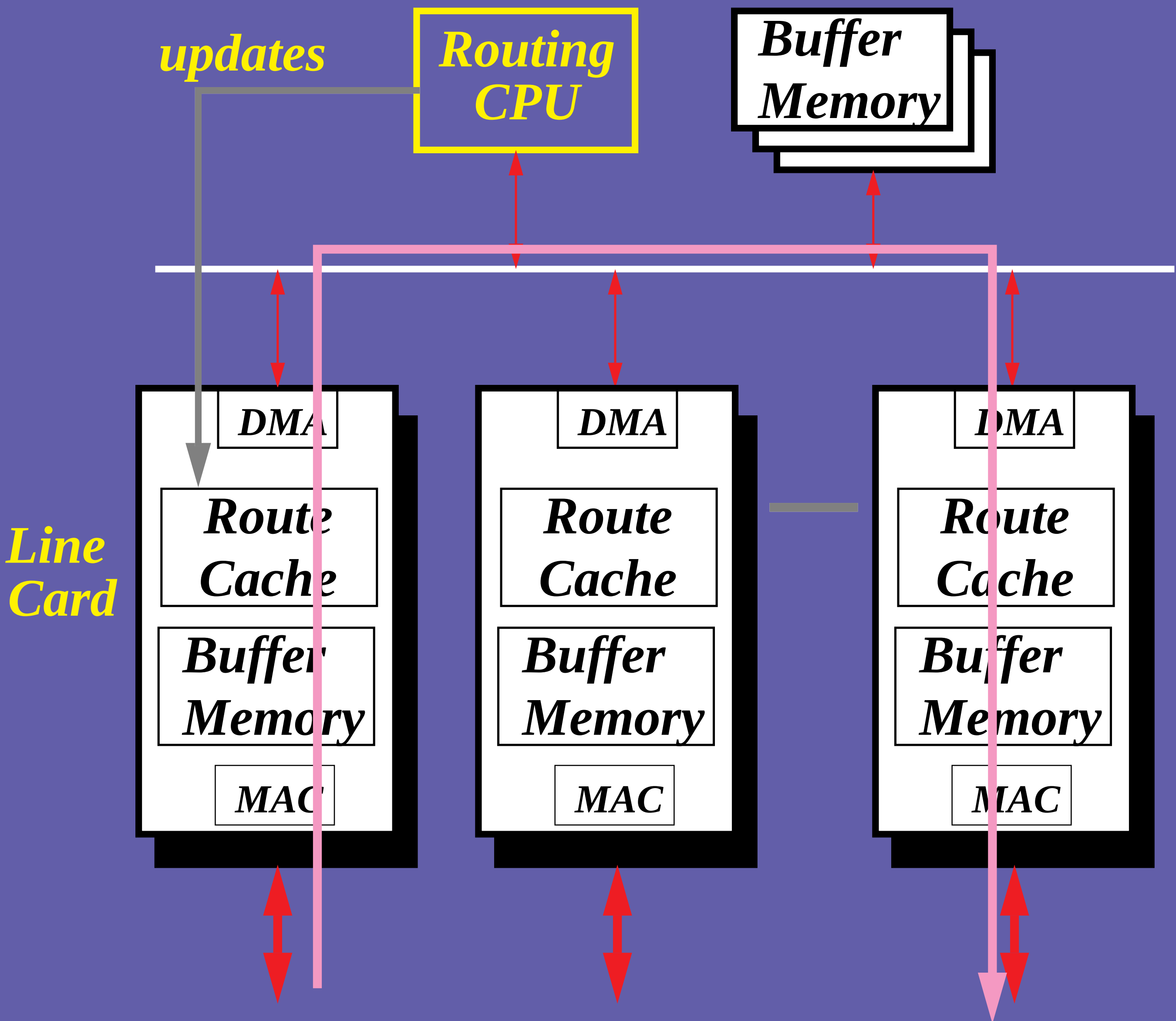
# The Evolution of Routers

## *Reducing the number of bus copies*



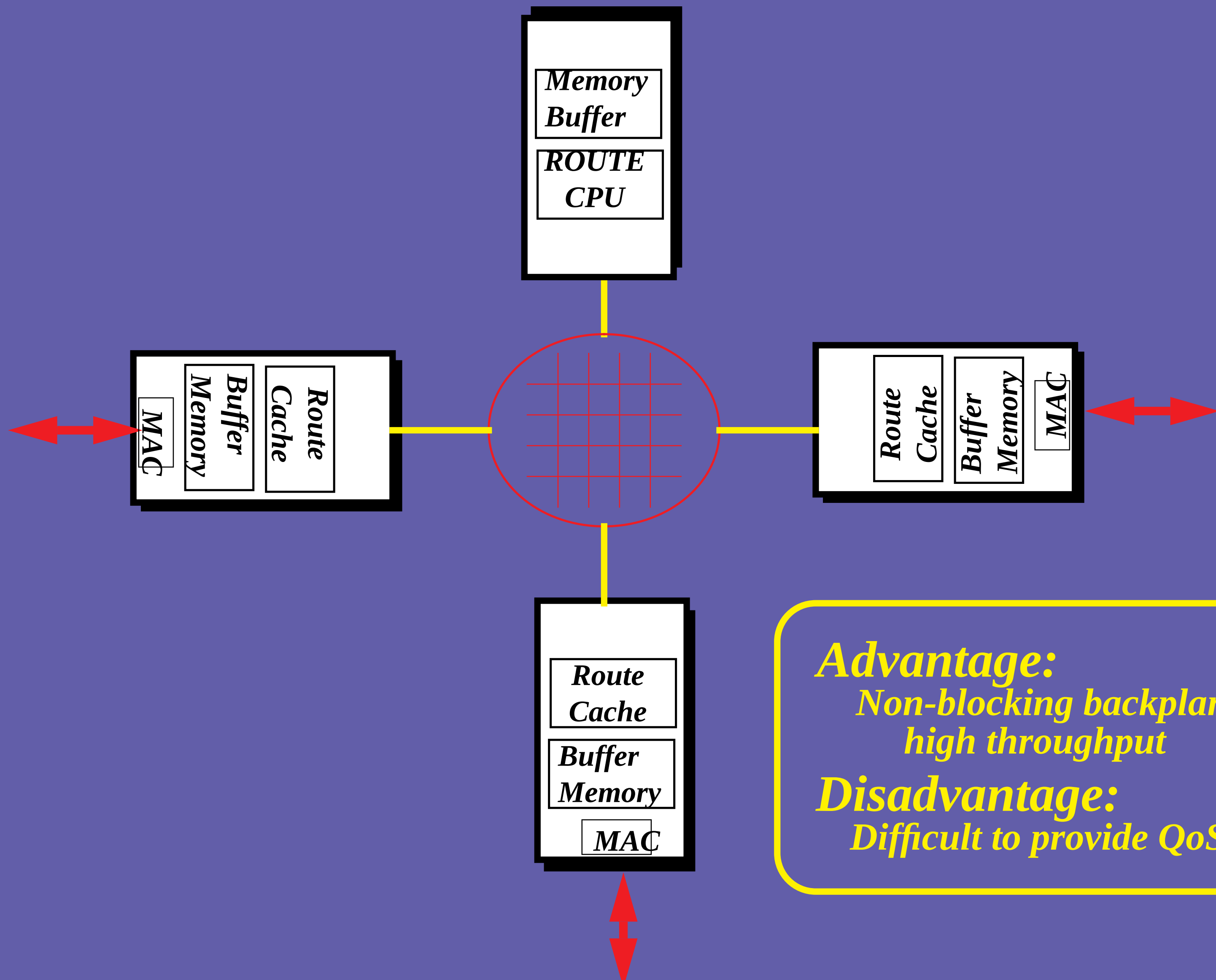
# The Evolution of Routers

*Reducing the number of bus copies*



# The Evolution of Routers

*Avoiding bus contention*

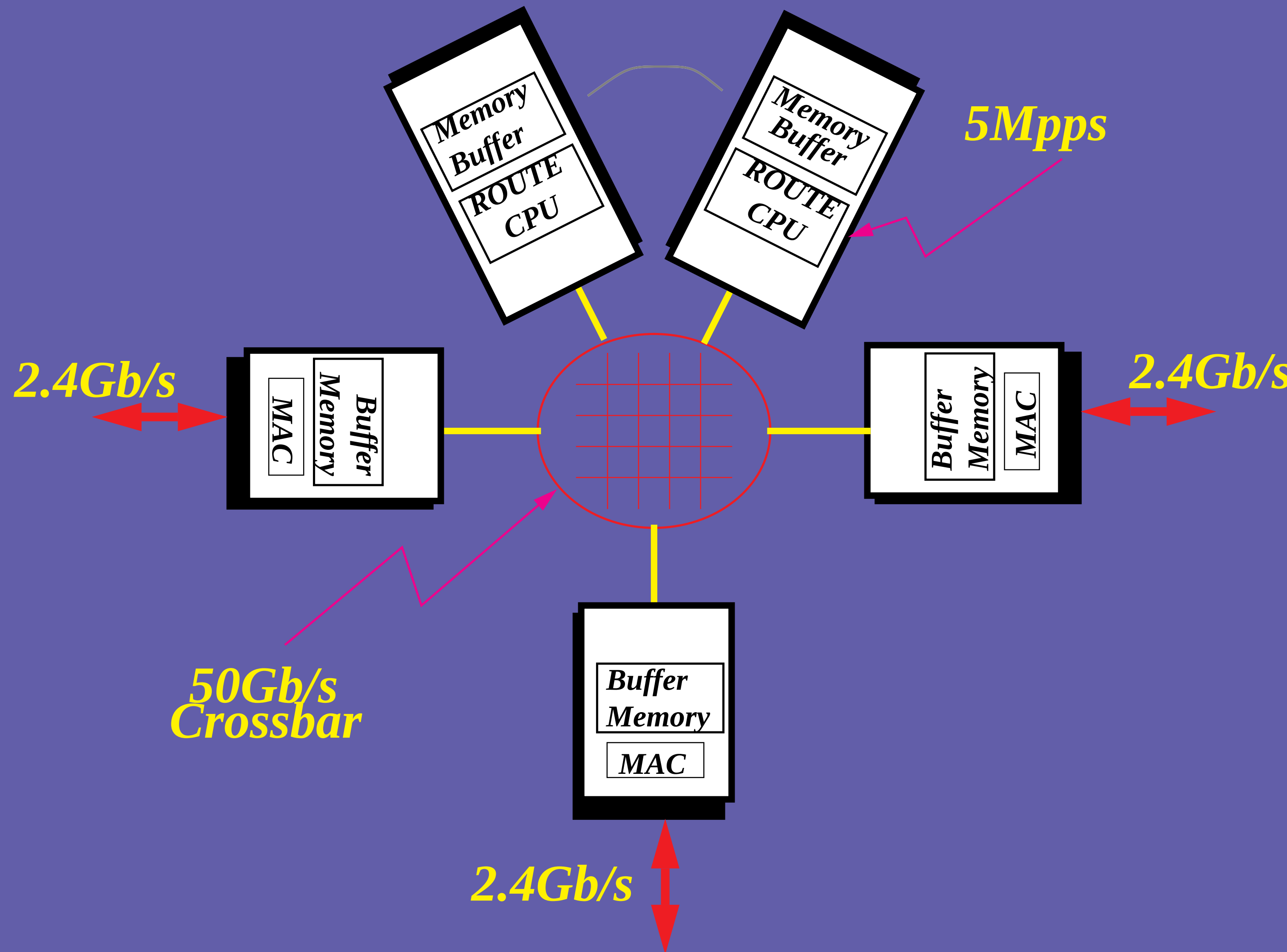



***Advantage:***
*Non-blocking backplane— high throughput*

***Disadvantage:***
*Difficult to provide QoS*

# Multigigabit Routing

## *BBN's Multigigabit Router*



1. The Demand for Bandwidth
2. The Shortage of Switching/Routing Capacity
3. The Architecture of Switches and Routers
4. Some (of our) solutions

# Some (of our) Solutions

## 1. Accelerating Lookups:

- Label-Swapping
- Longest-matching prefixes

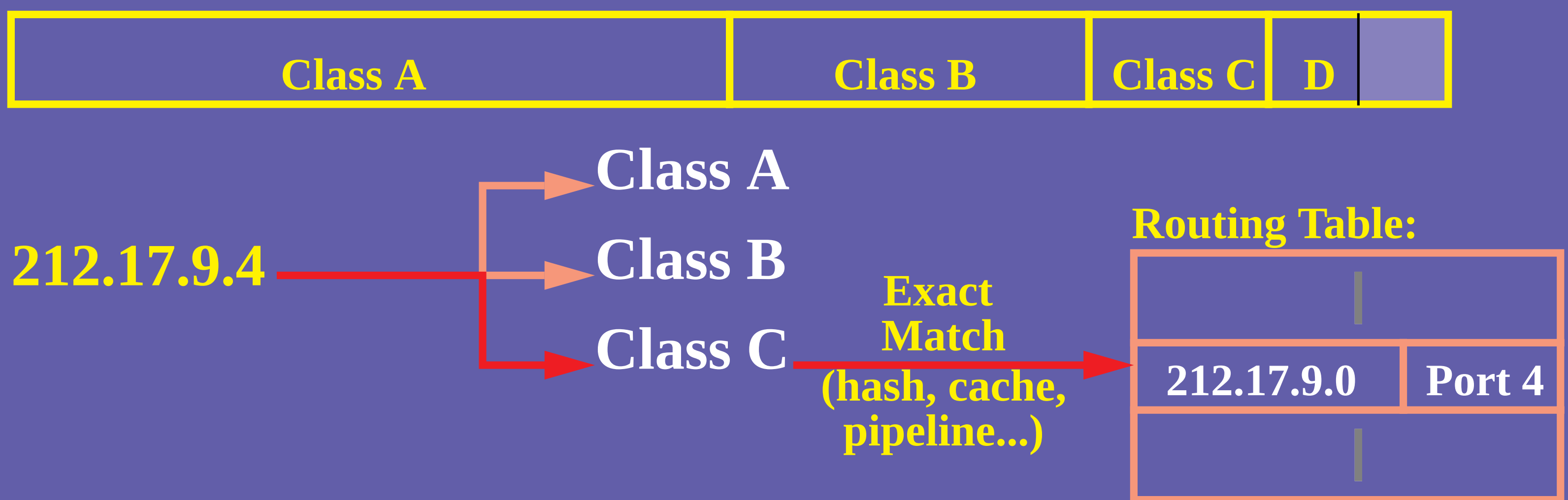
## 2. Switched Backplanes

- Input Queueing
  - — Theory
  - — Unicast
  - — Multicast
- Fast Buffering
- Speedup

---

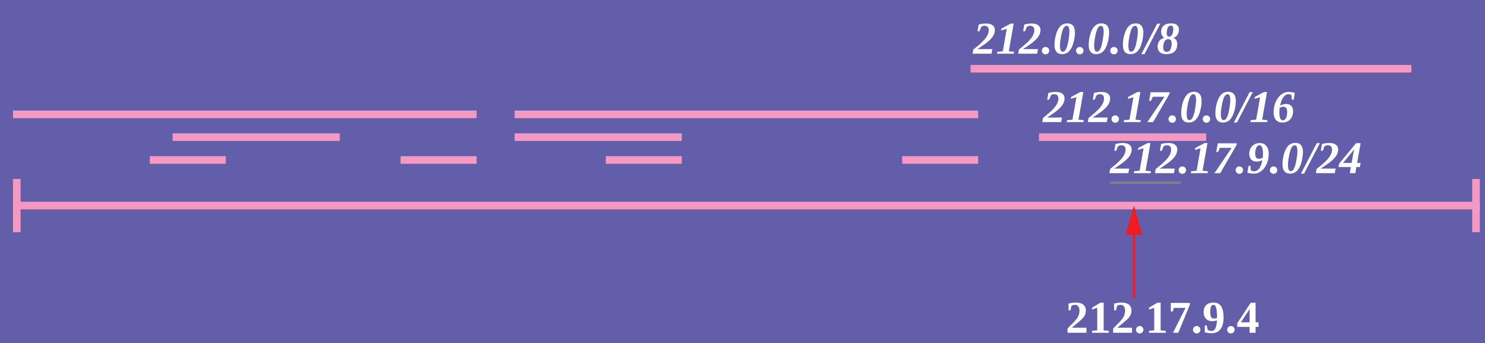
## 3.Our main project: *The Tiny Tera*

# Routing Lookups



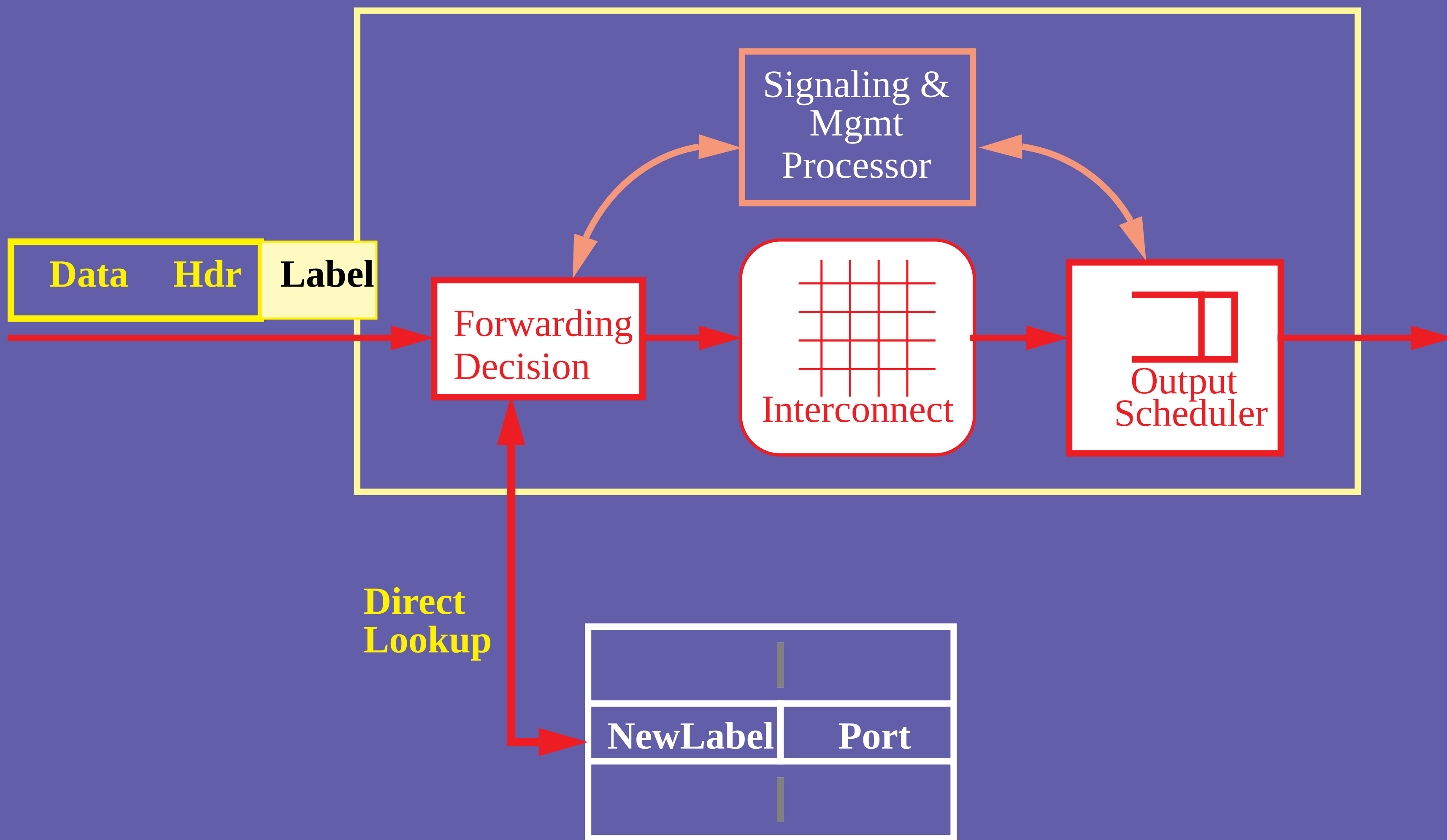
# Routing Lookups with CIDR ("supernetting")

**CIDR uses "longest matching prefix" routing:**




***Hashing, caching and pipelining are hard!***

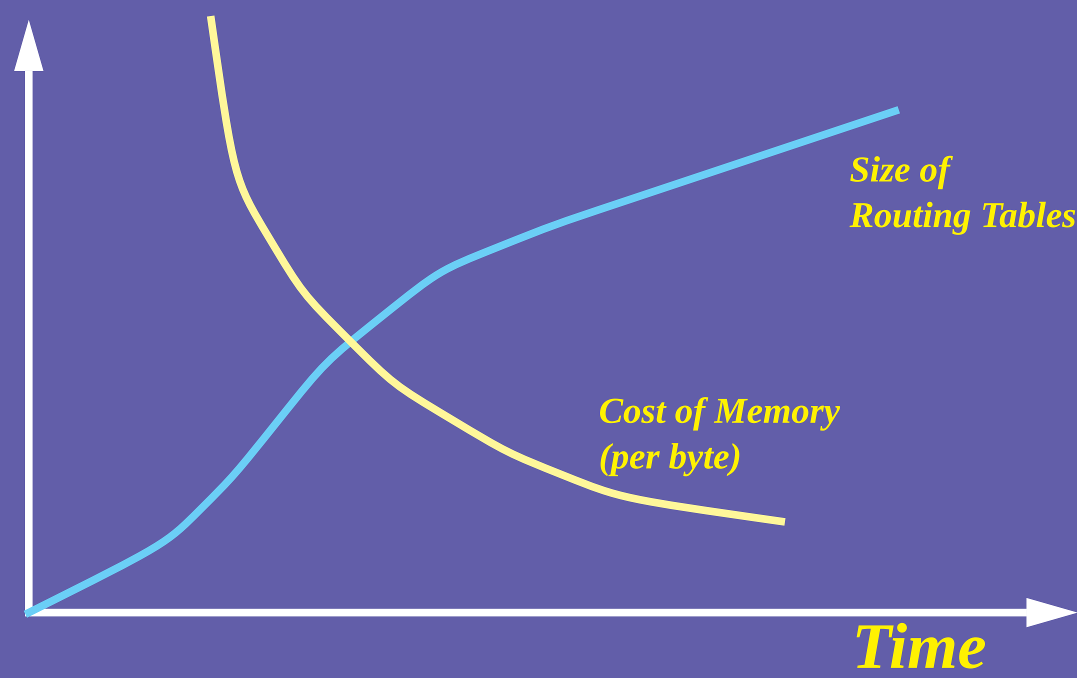
# Solution 1:
# *Label Swapping*




***IP Switching, Tag Switching, ARIS, Cell-switched Router,....***

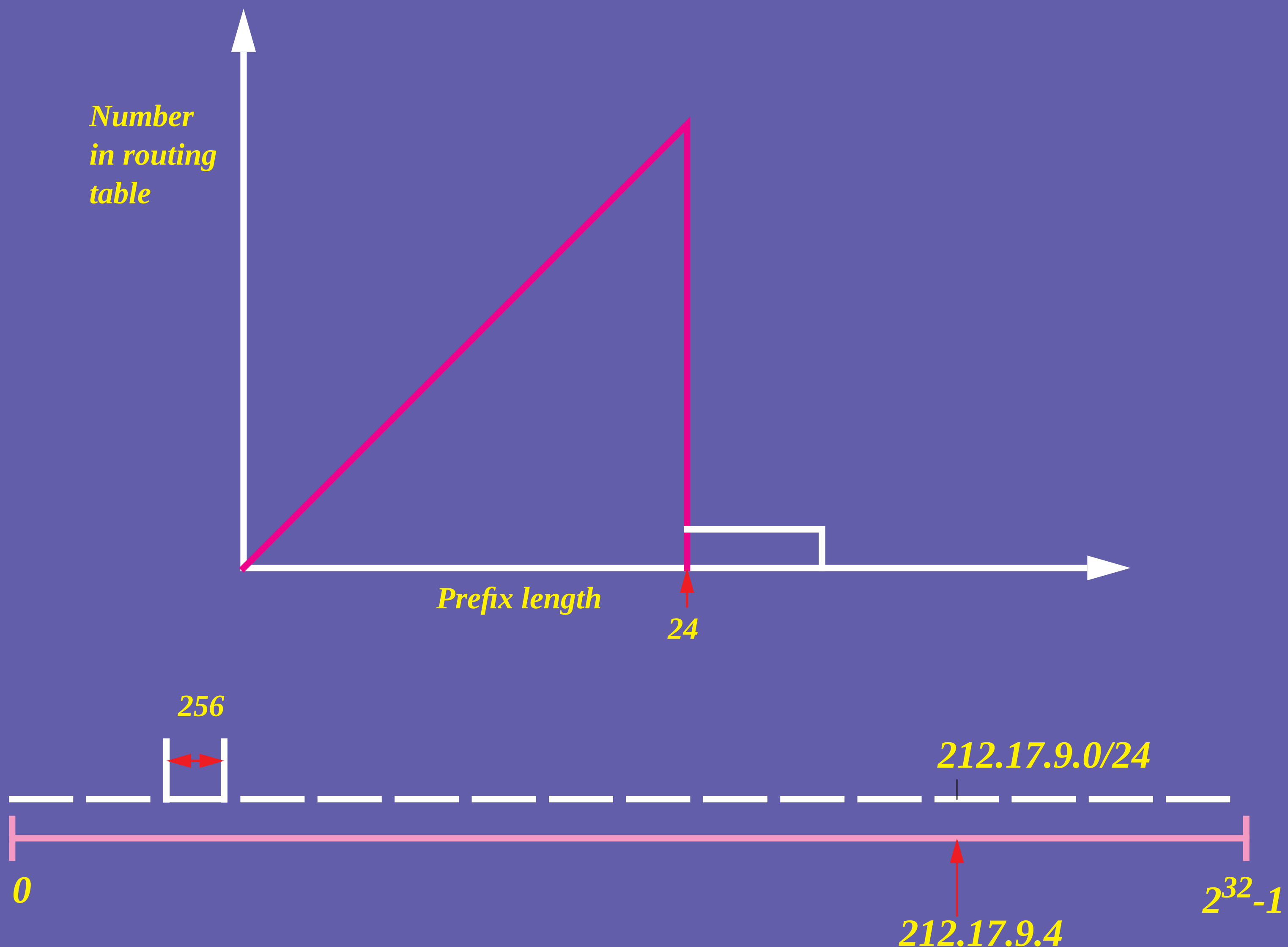
# Solution 2:
# *Perform Lookups Faster!*

***Observation #1:***



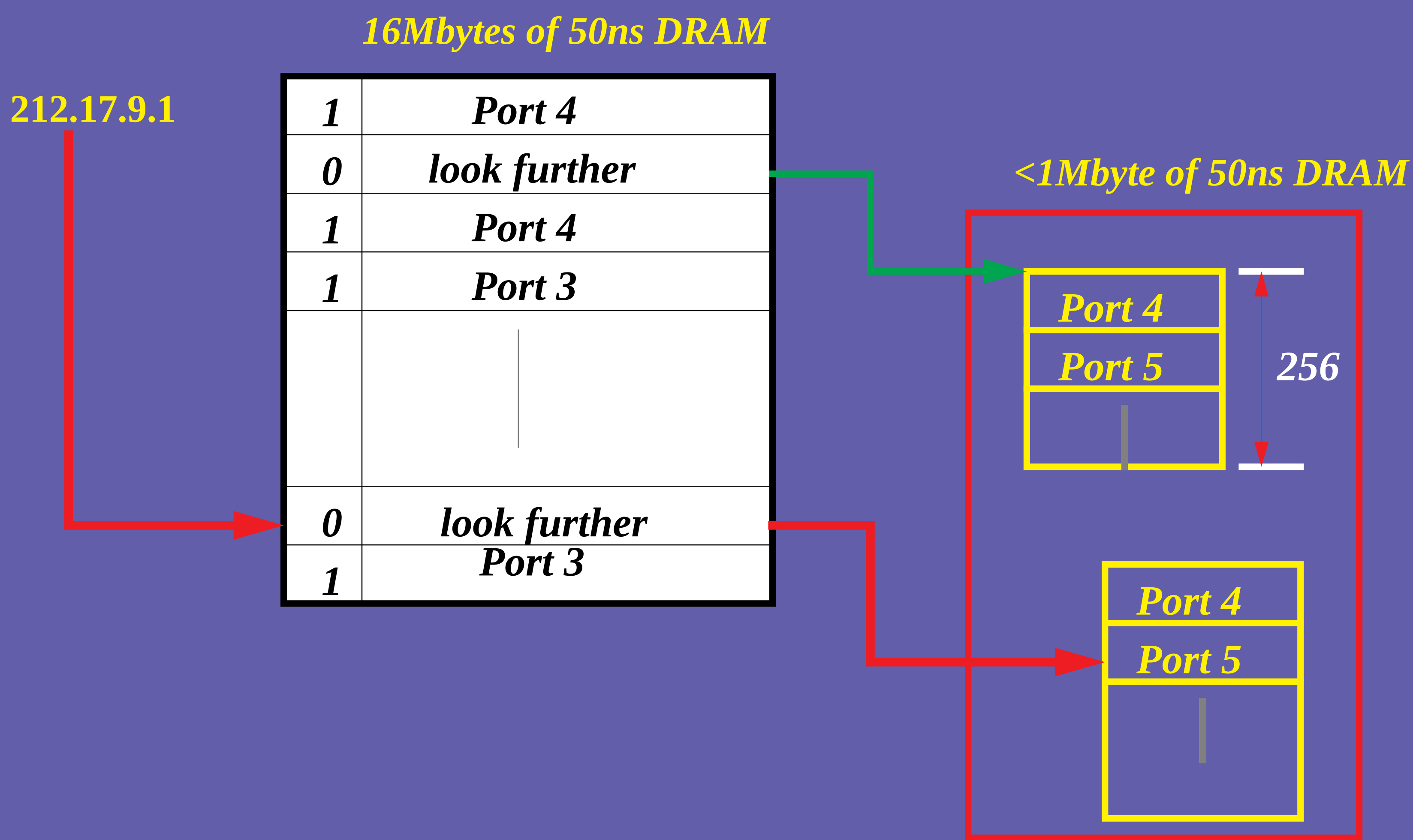
# *Performing Lookups Faster*

## *Observation #2:*



# Solution 2 (cont):

## *20 million lookups per second*



## 1. Accelerating Lookups:

- Label-Swapping
- Longest-matching prefixes

## 2. Switched Backplanes

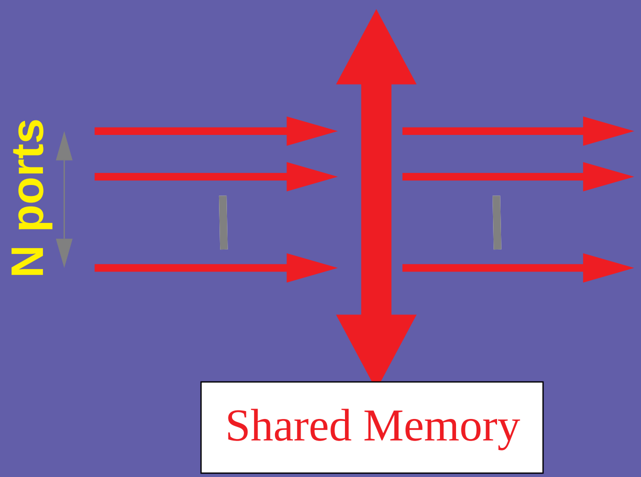
- Input Queueing
  - — Theory
  - — Unicast
  - — Multicast
- Fast Buffering
- Speedup

---

## 3.Our main project: *The Tiny Tera*

# Should we use shared memory or input-queueing?

## Shared Memory:



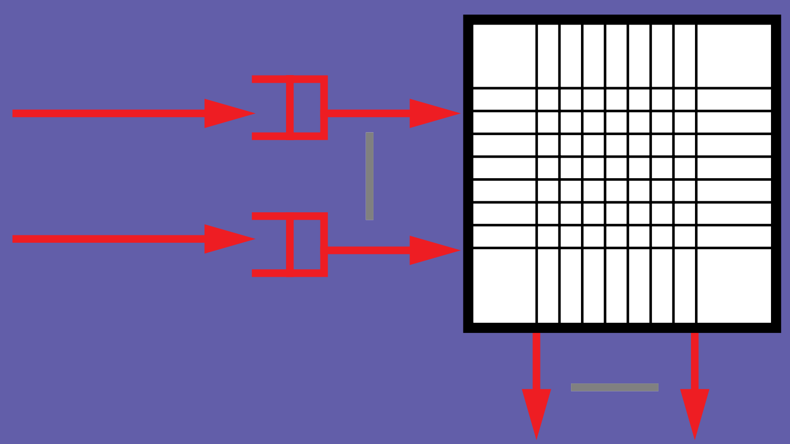

### Advantages:

Highest Throughput.
Possible to control packet delay.

### Disadvantages:

N-fold internal speed-up

## Input Queueing:



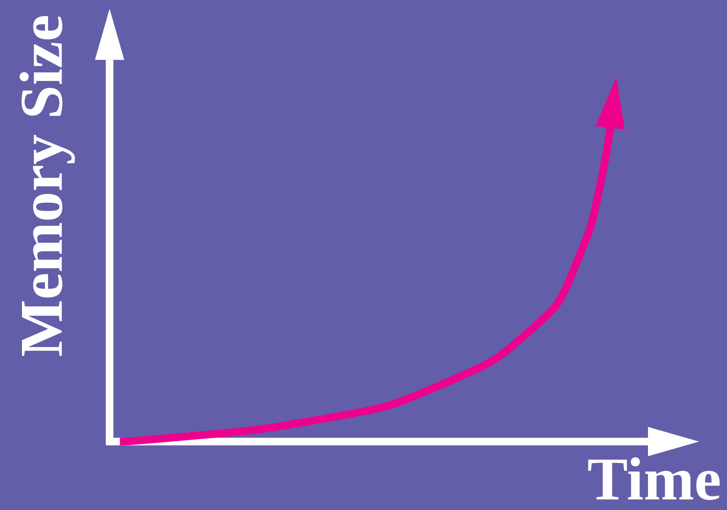
### Advantages:

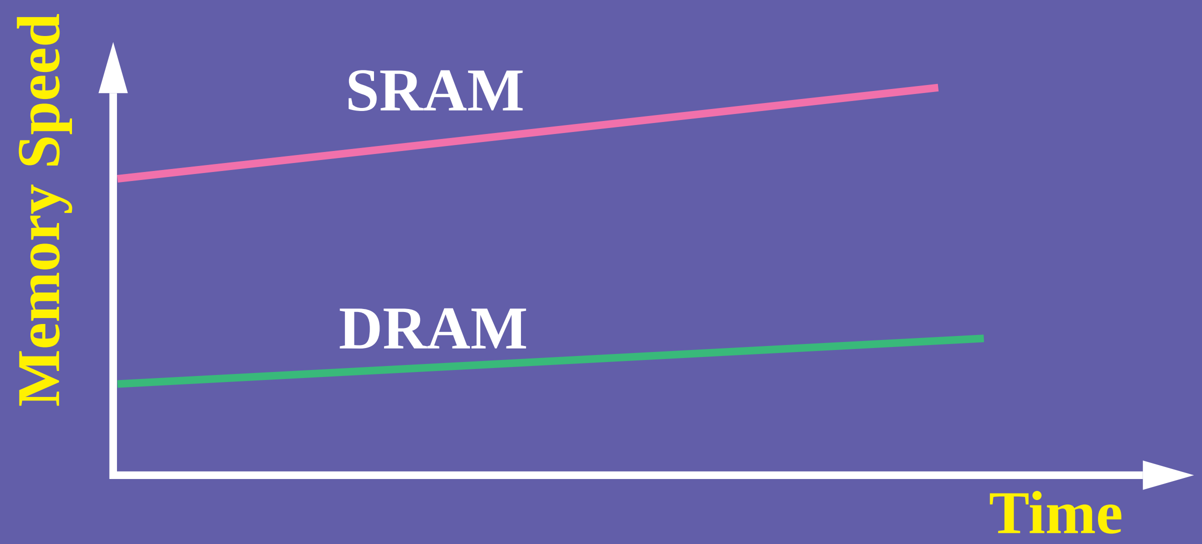
Simplicity
High Bandwidth

### Disadvantages:

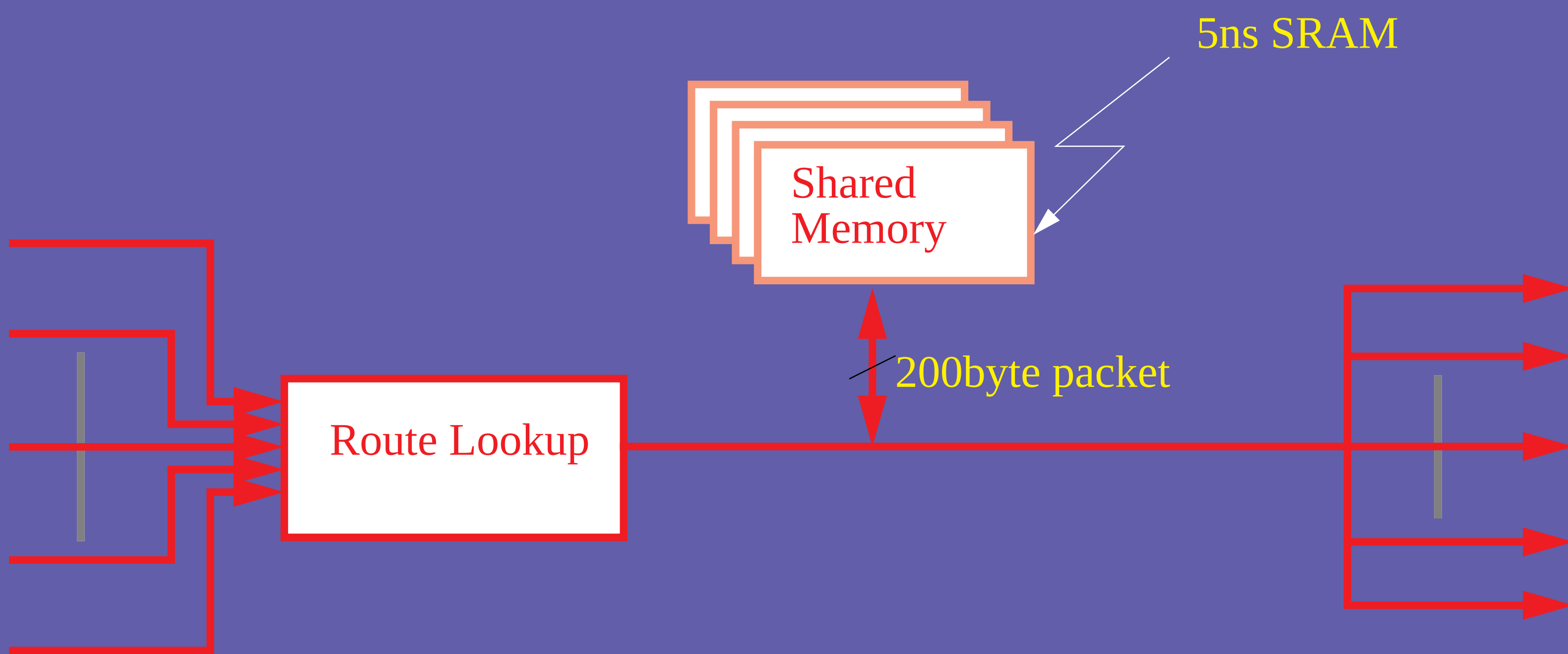
HOL Blocking
Less efficient
Difficult to control packet delay.

# Memory Bandwidth



# *An aside:* How fast can shared memory operate?




***How fast can a 16 port switch run with this architecture?***

$$5ns \text{ per packet} \times 2 \text{ memory operations per cell time}$$
$$\Rightarrow \text{aggregate bandwidth is } 160Gb/s$$

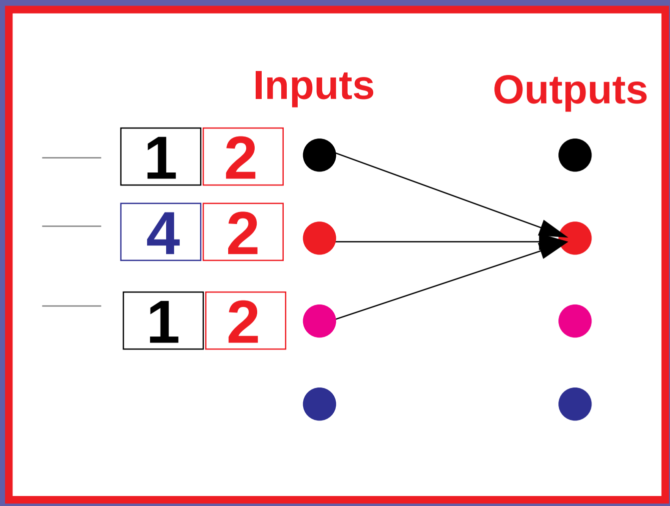
# Should we use shared memory or input-queueing?

Because of a *shortage of memory bandwidth*, most multigigabit and terabit switches and routers use either:

1. Input Queueing, or
2. Combined Input and Output Queueing.

# Head of Line Blocking

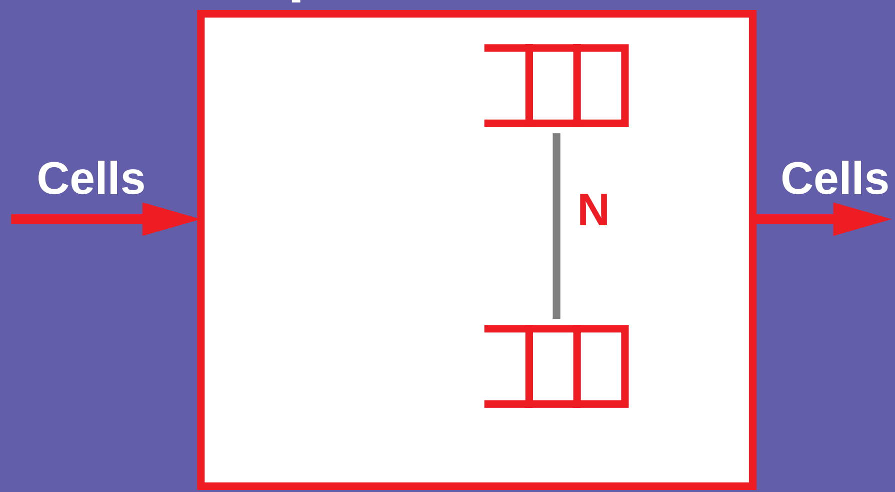
## The Problem




$$\rho_{max} = 2 - \sqrt{2} = 58\%$$

## A Solution....

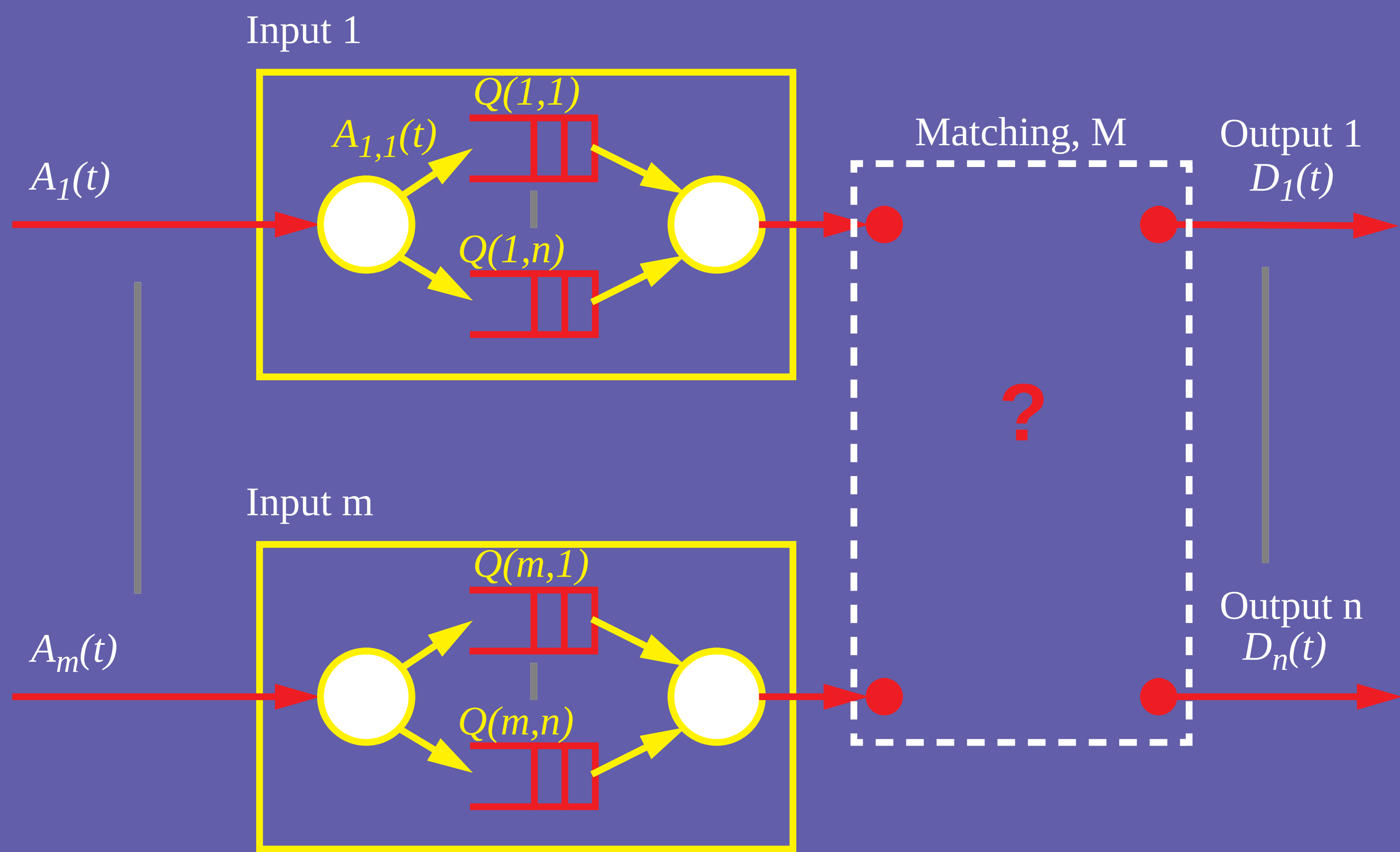
Input Cell Buffer



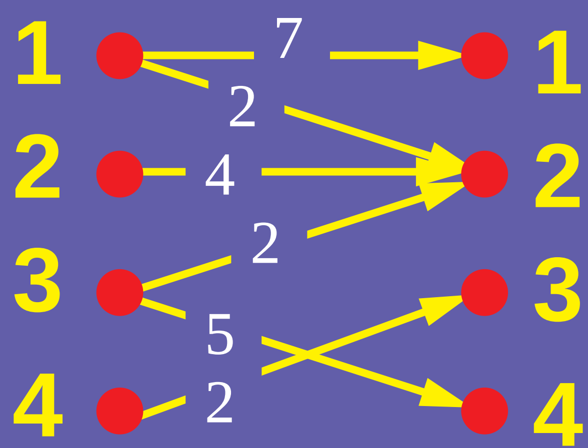

*"Virtual Output Queueing"*

$$\rho_{max} = 100\%$$

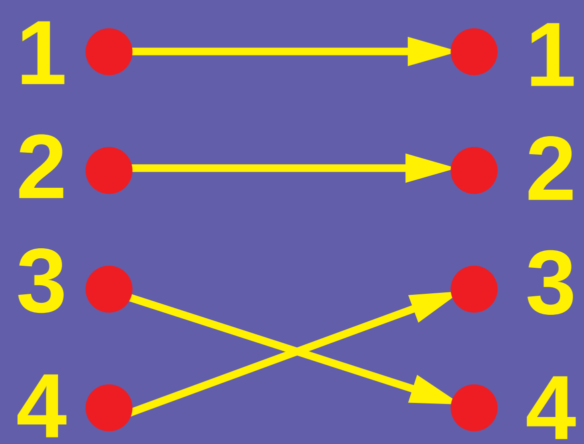
# ....but requires scheduling...



# ....which is equivalent to graph matching




**Request Graph**




**Bipartite Matching**

(Weight = 18)

# Practical Algorithms

1. ***i*SLIP** — Weight = 1
   — Iterative round-robin
   — Simple to implement

   ***Simple, fast, efficient***

2. ***i*LQF** — Weight = Occupancy
3. ***i*OCF** — Weight = Cell Age

   ***Good for non-uniform traffic. Complex!***

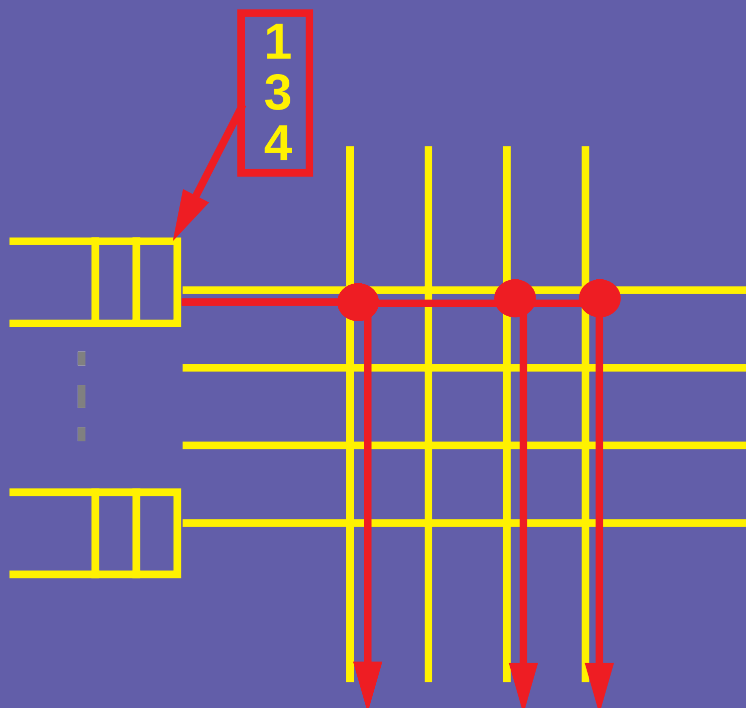
4. ***MCFF*** — Weight = Backlog

   ***Good for non-uniform traffic. Simple!***

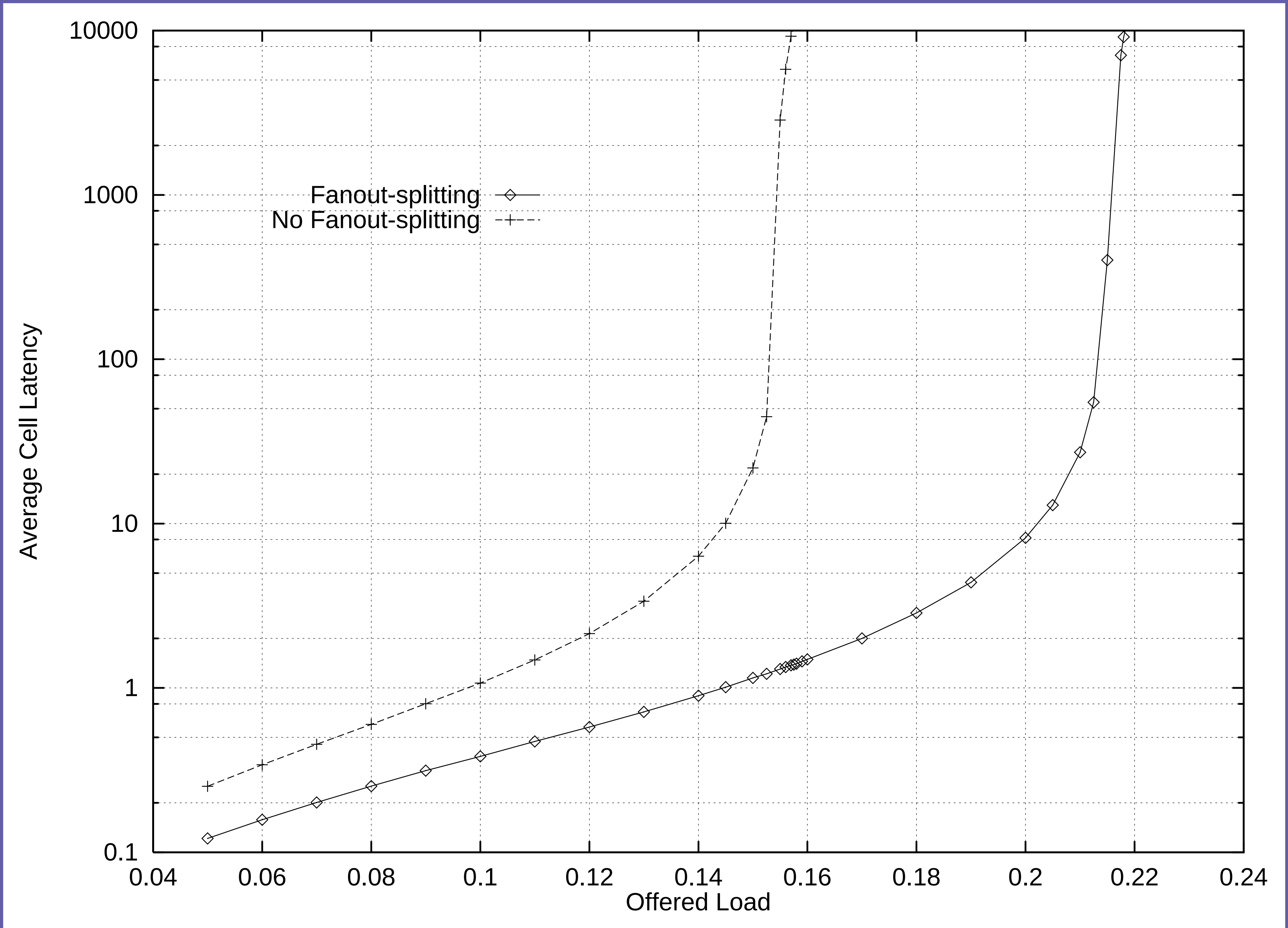
# Multicast Traffic

## Queue Architecture

1. Making use of the crossbar
2. Why treat multicast differently?
3. Why maintain a single FIFO queue?
4. Fanout-splitting



# Fanout-Splitting



# Multicast Traffic

1. ***Residue Concentration***

2. ***Tetris-based schedulers***

# Gigabit and Terabit Routing

## 1. Accelerating Lookups:

- Label-Swapping
- Longest-matching prefixes

## 2. Switched Backplanes

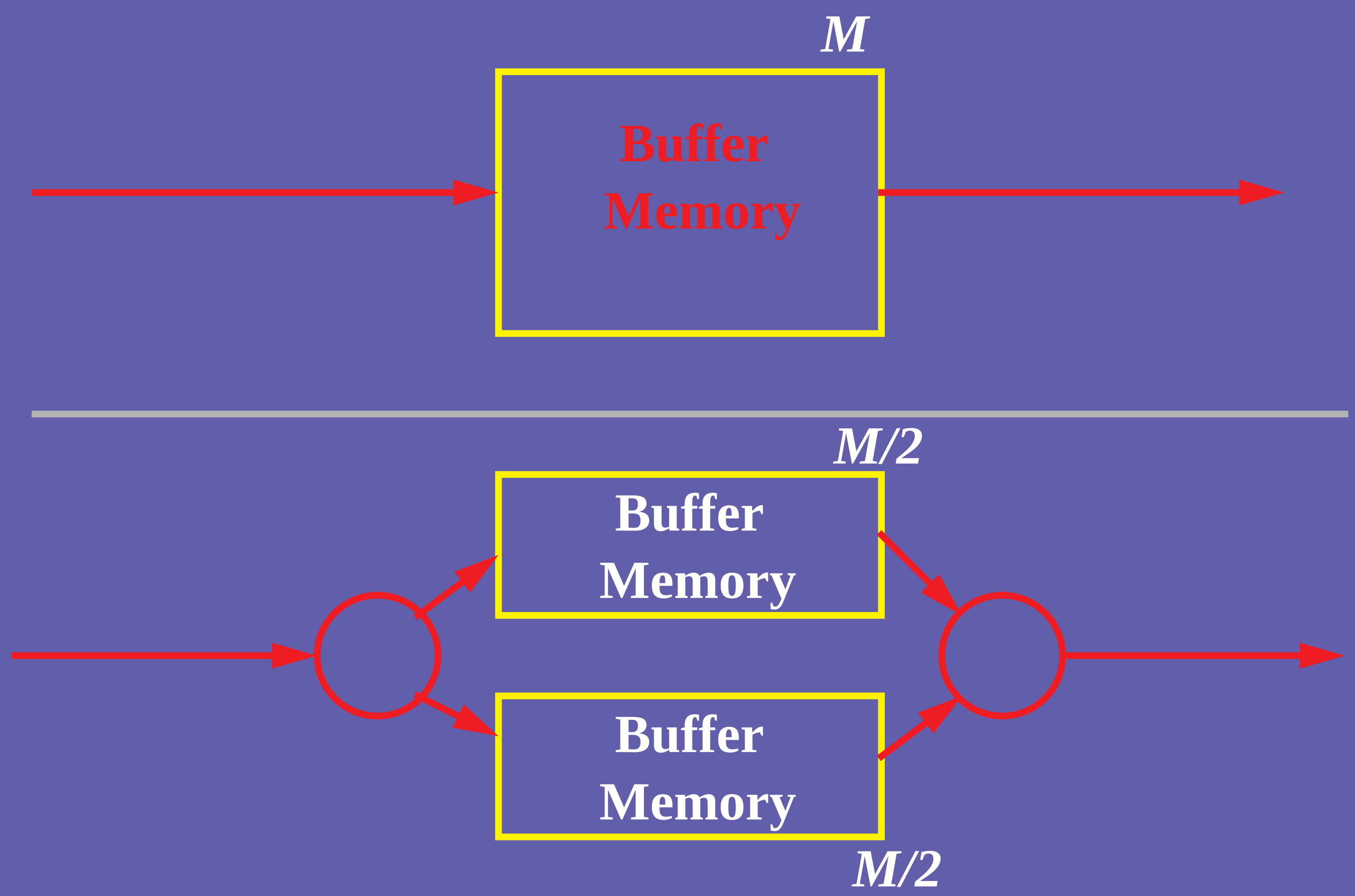
- Input Queueing
  - — Theory
  - — Unicast
  - — Multicast
- Fast Buffering
- Speedup

---

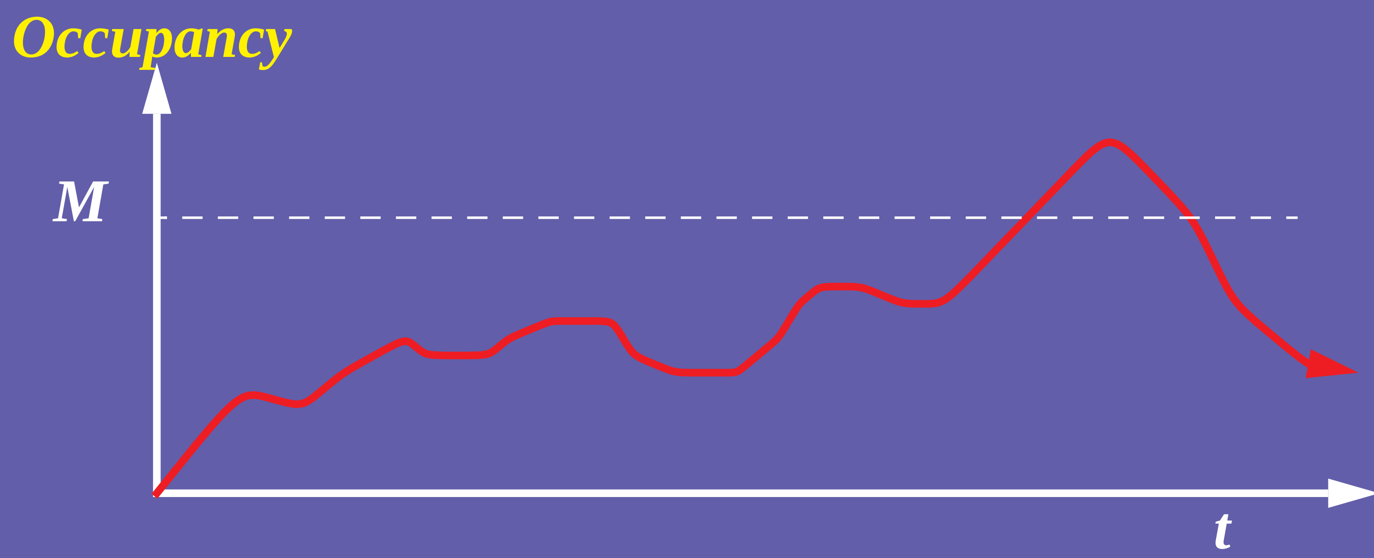
## 3.Our main project: *The Tiny Tera*

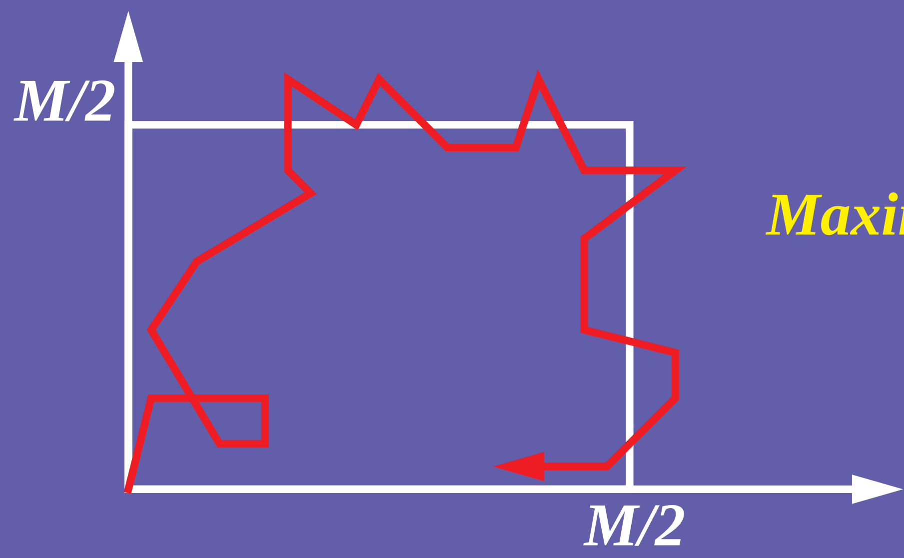
# Fast Buffering

## *Ping-pong Memory*



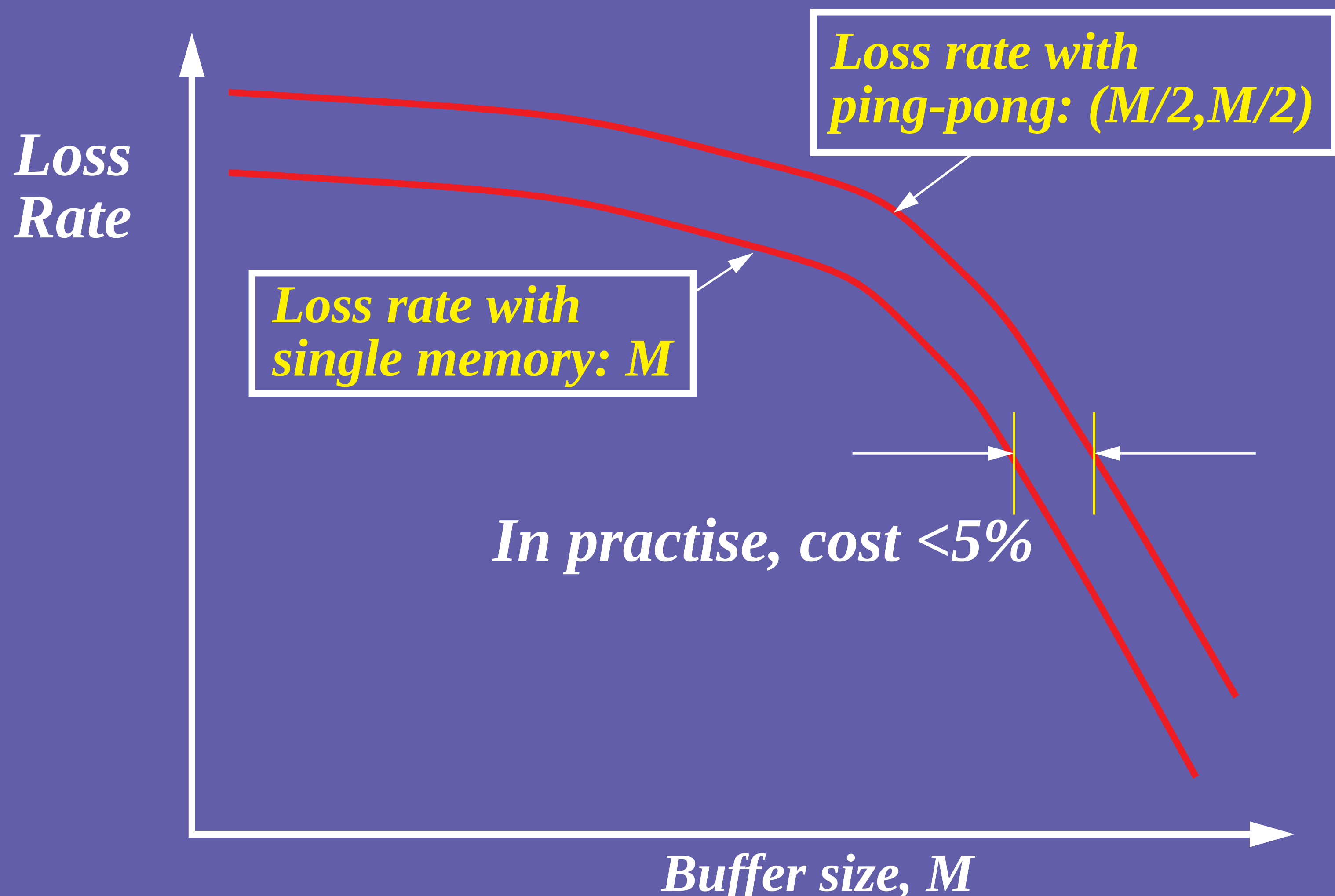
# Fast Buffering
## *Ping-pong Memory*




***Maximum "cost" = M/2***

# Fast Buffering
## *Ping-pong Memory*



# Gigabit and Terabit Routing

## 1. Accelerating Lookups:

- Label-Swapping
- Longest-matching prefixes

## 2. Switched Backplanes

- Input Queueing
  - Theory
  - Unicast
  - Multicast
- Fast Buffering
- Speedup

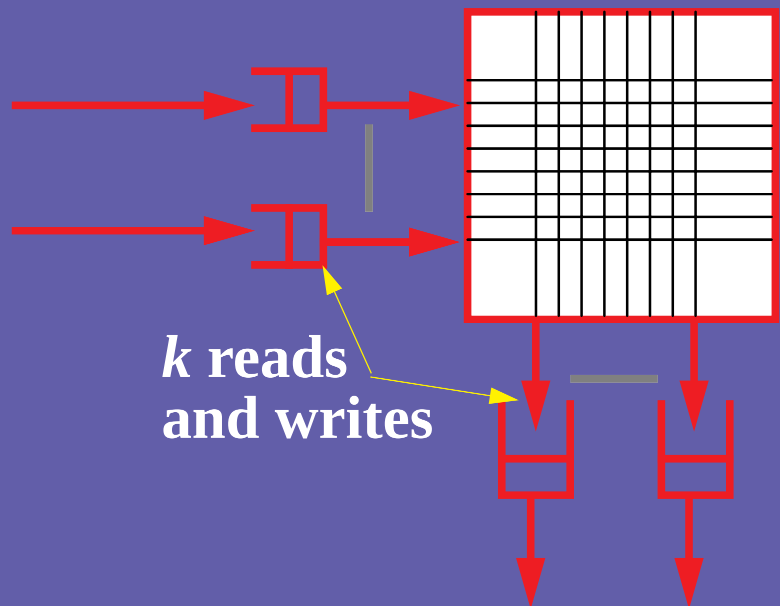
---

## 3. Our main project: *The Tiny Tera*

# Matching Output Queueing with Input- and Output- Queueing

*How much speedup is enough?*

## Combined Input- and Output-Queueing:



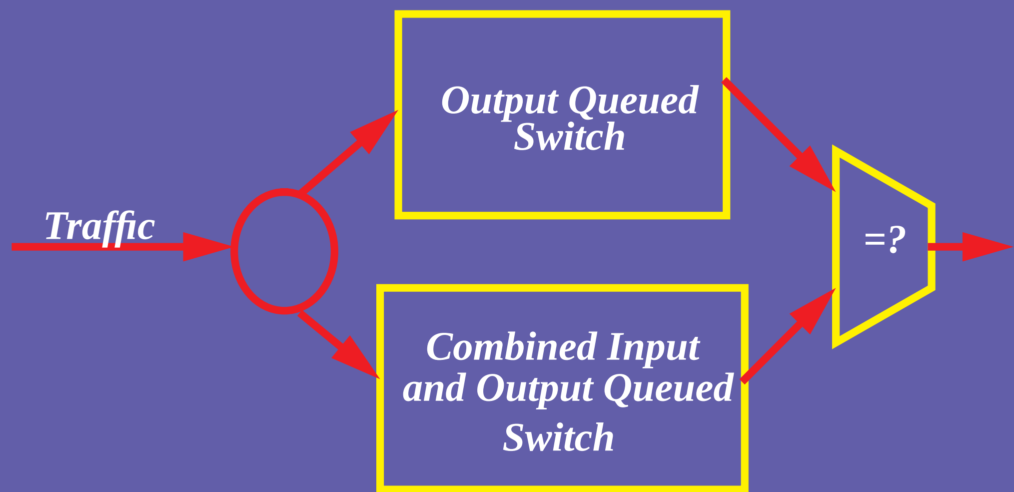
# Matching Output Queueing with Input- and Output- Queueing

***How much speedup is enough?***

**Conventional wisdom suggests:**

***A speedup*** $k = 2 - 4$ ***leads to high throughput***

# Matching Output Queueing with Input- and Output- Queueing




**Fact** *To match output queueing, with FIFO input queues:*
*k = N*

**Fact** *To match output queueing, with virtual output queues:*
*k = 4 is sufficient*

**Conjecture:** *To match output queueing, with VOQs:*
*k = 2 is sufficient*

1. **Accelerating Lookups:**
   - Label-Swapping
   - Longest-matching prefixes
2. **Switched Backplanes**
   - Input Queueing
     - — Theory
     - — Unicast
     - — Multicast
   - Fast Buffering
   - Speedup

---

**3.Our main project: *The Tiny Tera***