# Multiplexing Methods

Daubing the Information

*“I think there is a world market for about five computers.”*

**Thomas Watson,**
**chairman of IBM 1943**

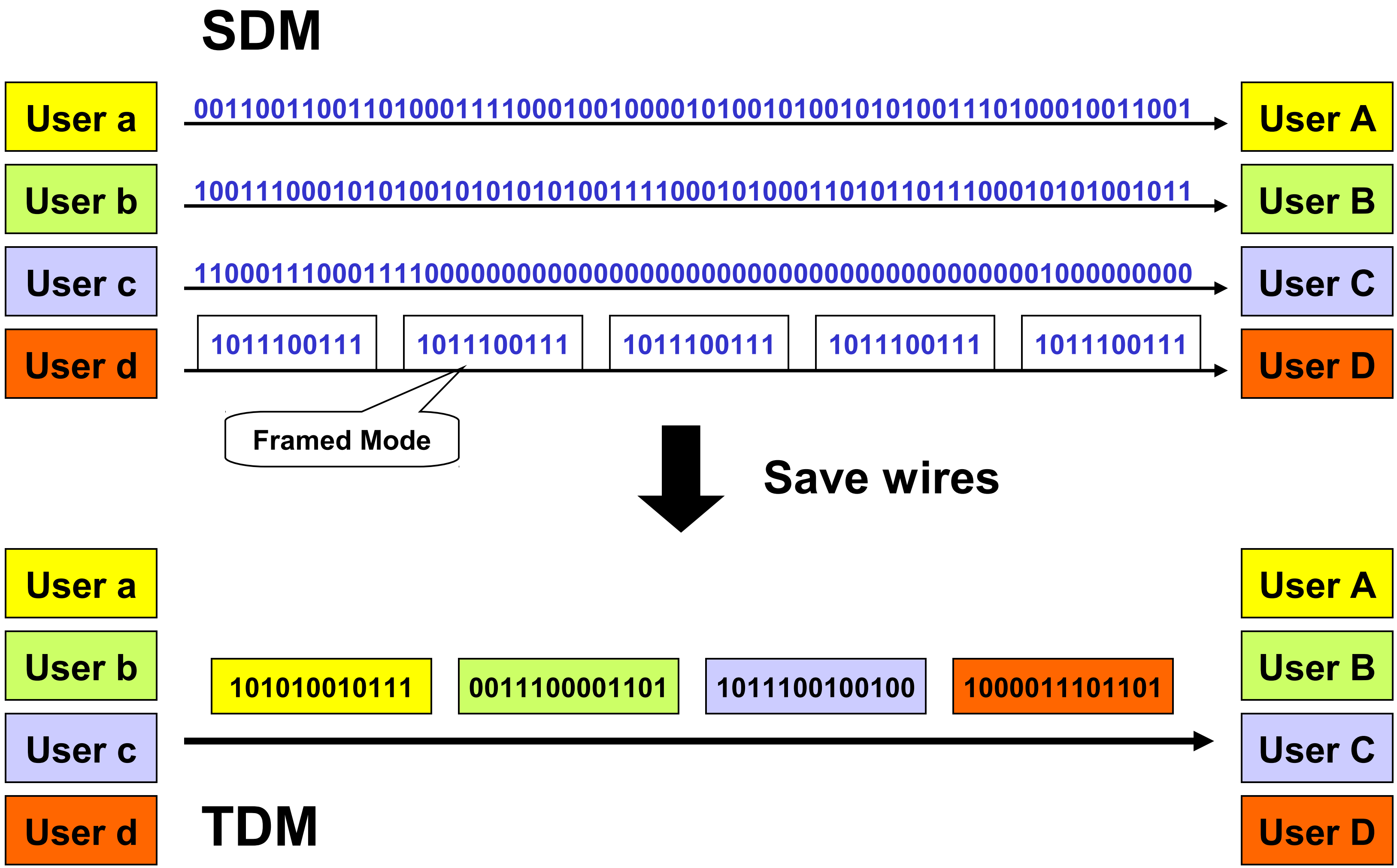
# Multiplexing Types

- TDM
  - Most important
  - Statistical and Deterministic

- SDM
- FDM and (D)WDM
- CDM

Will be covered in other chapters

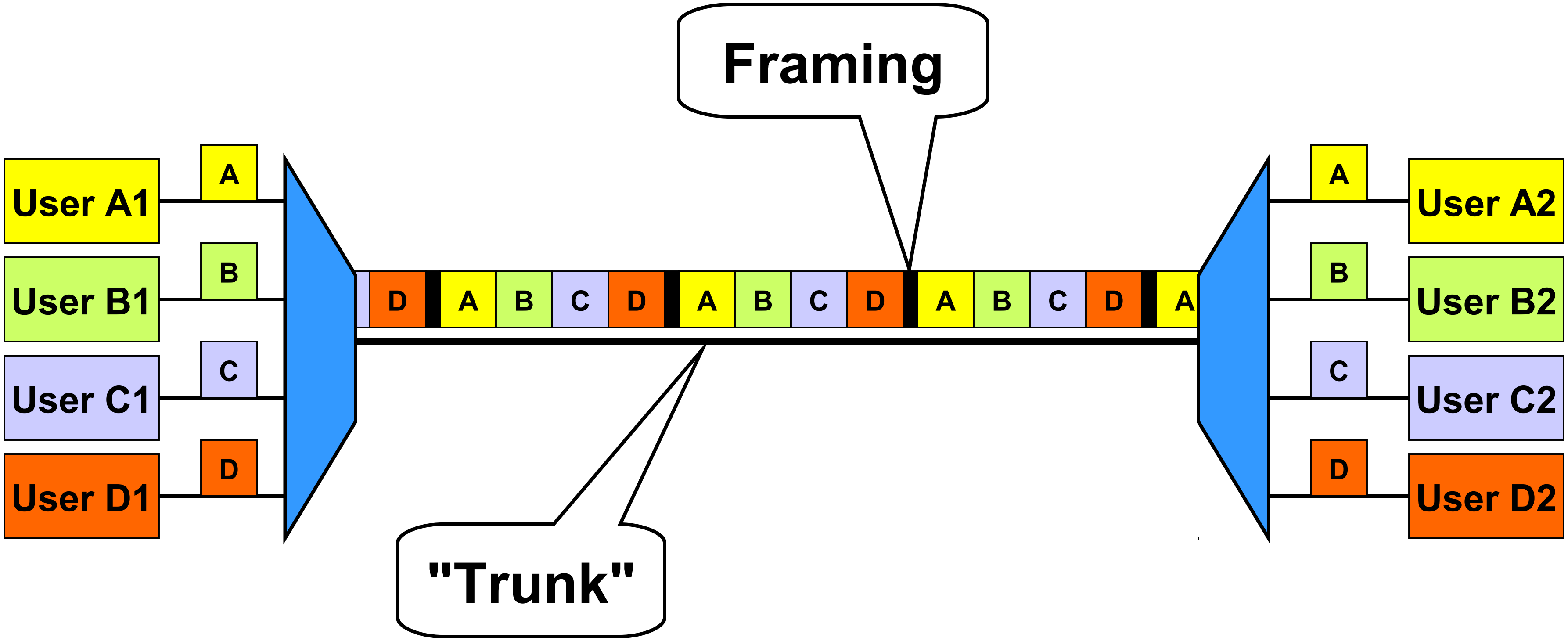
# TDM (1)


SDM
User a
0011001100110100011110001001000010100101001010100111010001001 1001
User A
User b
10011100010101001010101010011110001010001101011011100010101001011
User B
User c
1100011100011110000000000000000000000000000000000000000001000000000
User C
User d
1011100111
1011100111
1011100111
1011100111
1011100111
User D
Framed Mode
Save wires
User a
User b
User c
User d
101010010111
0011100001101
1011100100100
1000011101101
User A
User B
User C
User D
TDM

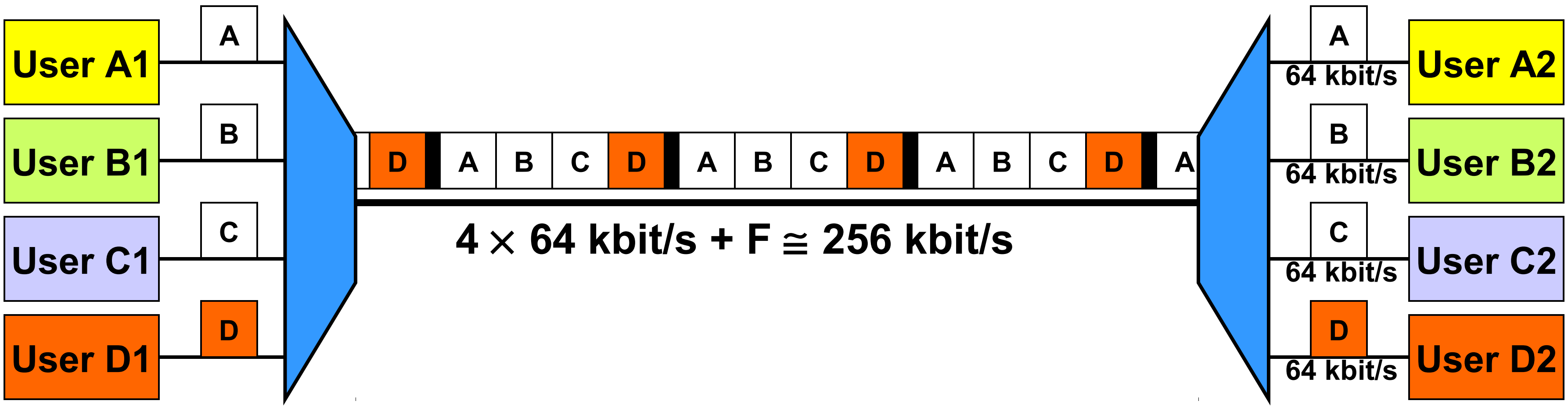
# TDM (2)

- Requires framed link layer
- Saves wires
- Is slower than SDM
- Requires multiplexers and demultiplexers
- Two fundamentally different methods:
  - Deterministic TDM
  - Statistical TDM

# Deterministic TDM (1)



# Deterministic TDM (2)

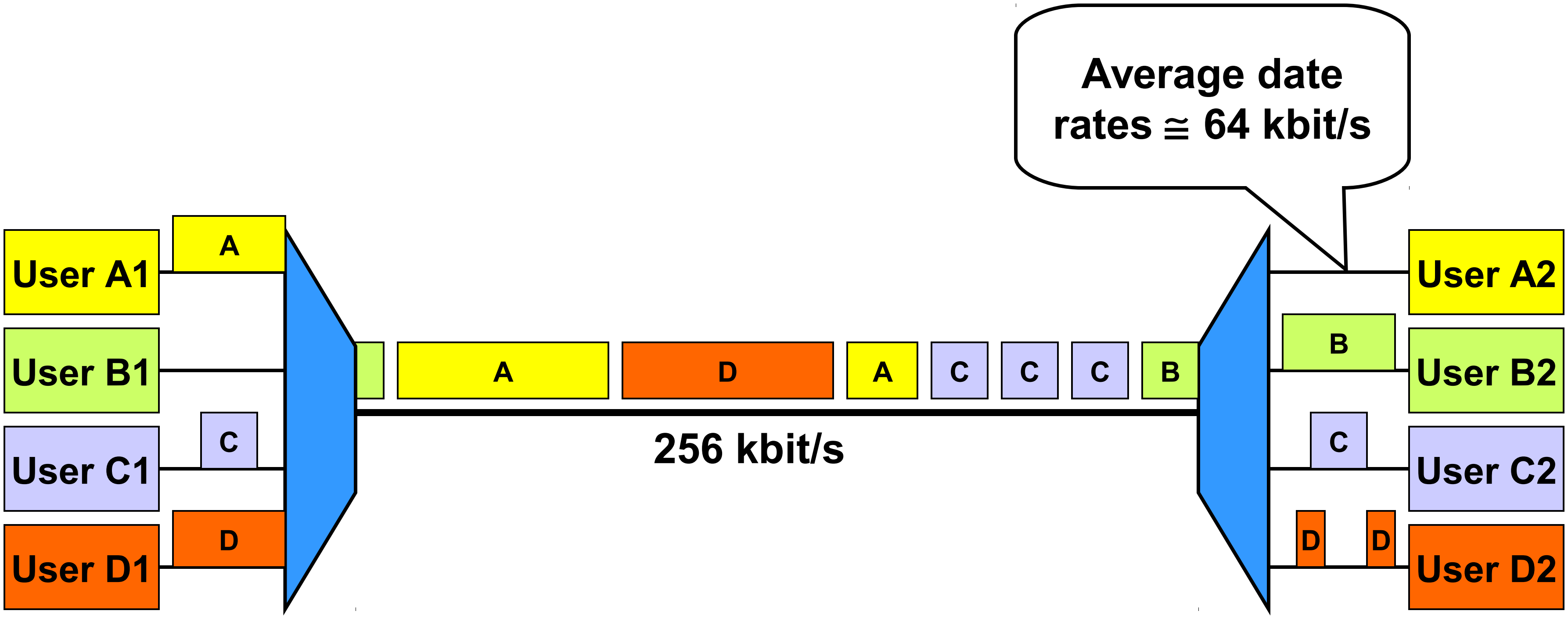


- Trunk speed = Number of slots × User access rate
- Each user gets a constant timeslot of the trunk

# Deterministic TDM – Facts

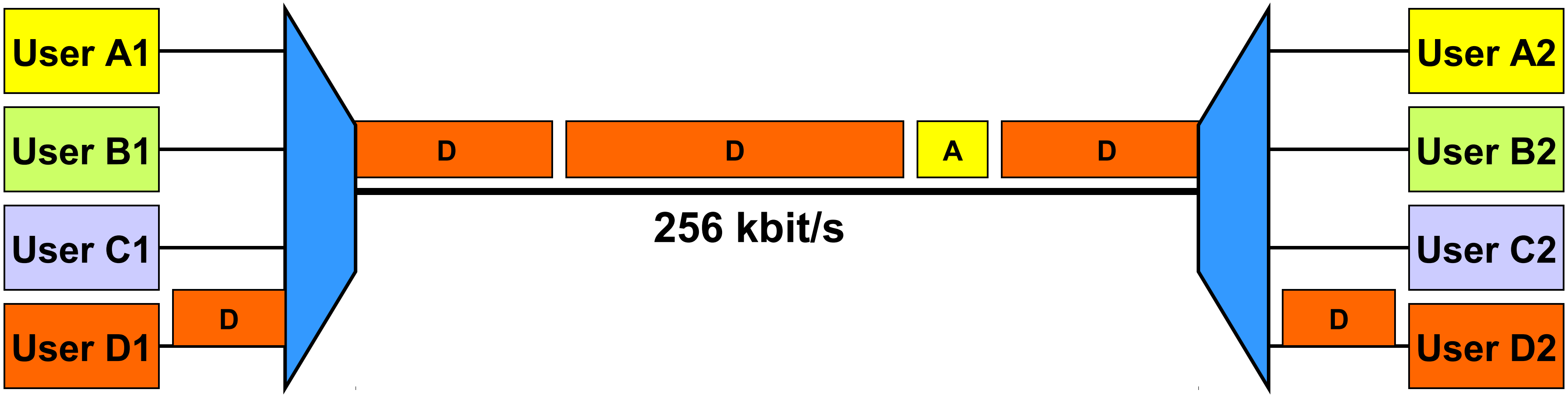
- Order is maintained
- Frames must have same size
- No addressing information required
- Inherently connection-oriented
- No buffers necessary (QoS)
- Protocol transparent
- Bad utilization of trunk

# Statistical TDM (1)



- Trunk speed dimensioned for average usage
- Each user can send packets whenever she wants

# Statistical TDM (2)

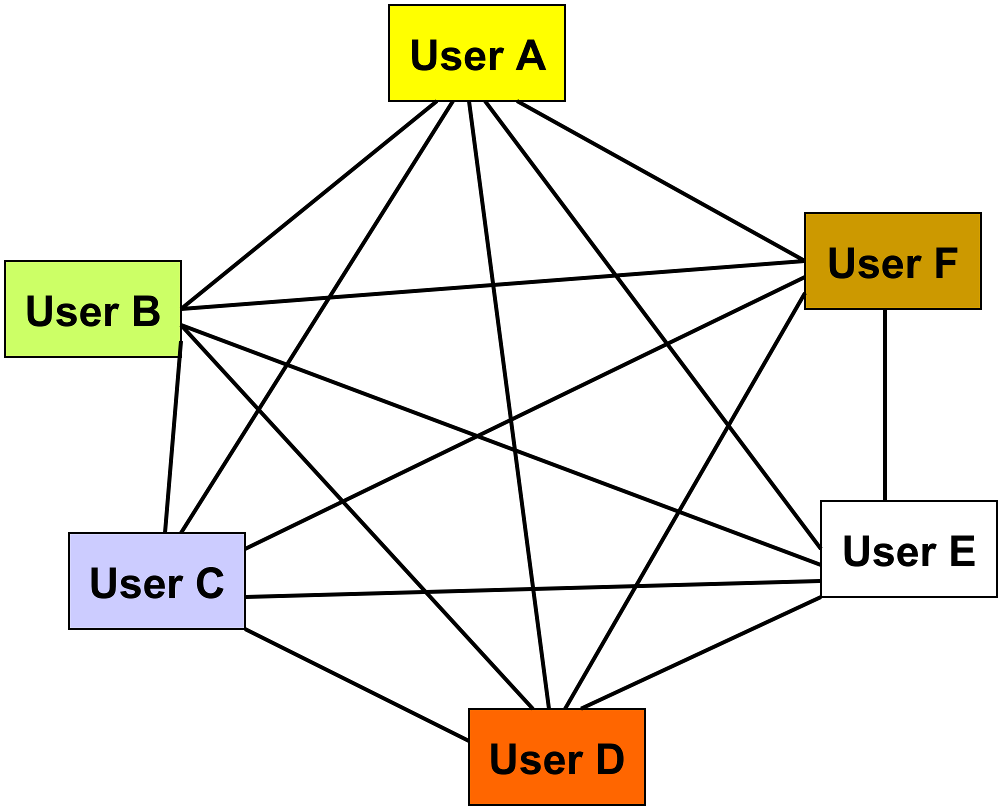


- If other users are silent, one (or a few) users can fully utilize their access rate

# Statistical TDM – Facts

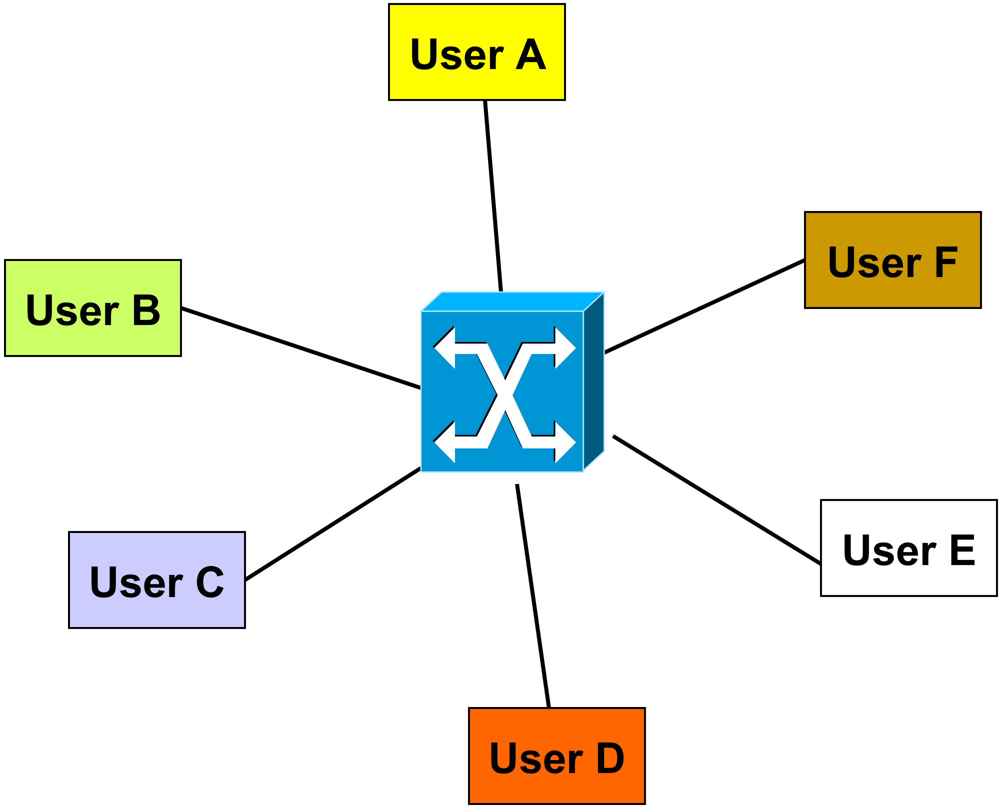
- Good utilization of trunk
  - Statistically dimensioned
- Frames can have different size
- Multiplexers require buffers
- Variable delays
- Address information required
- Not protocol transparent

# Networking: Fully Meshed



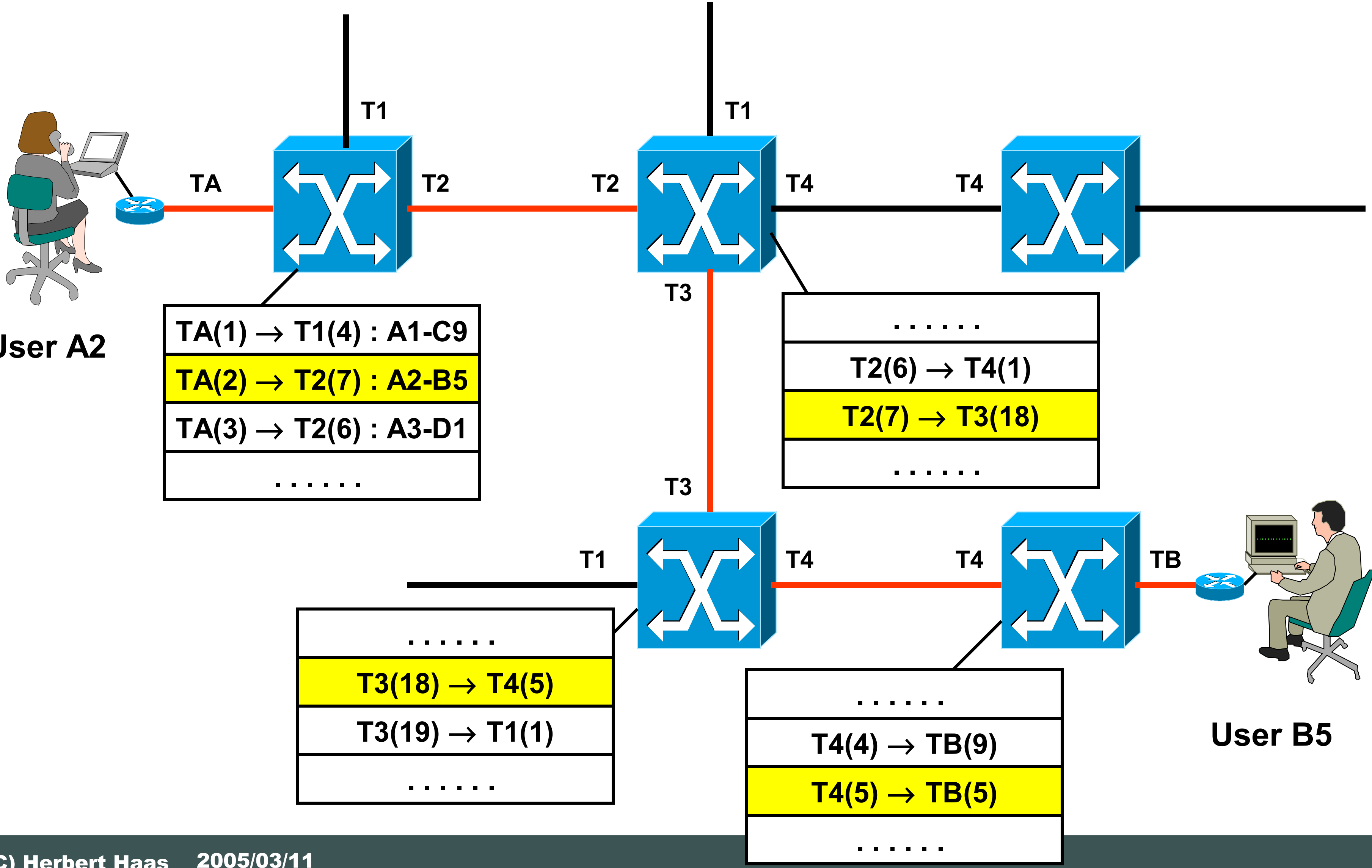
- Metcalfe's Law: n(n-1)/2 links
- Good fault tolerance
- Expensive

# Networking: Switching



- Only 6 links
- Switch supports either deterministic or statistical TDM

# Circuit Switching



| |
|---|
| TA(1) → T1(4) : A1-C9 |
| TA(2) → T2(7) : A2-B5 |
| TA(3) → T2(6) : A3-D1 |
| . . . . . . |

| |
|---|
| . . . . . . |
| T2(6) → T4(1) |
| T2(7) → T3(18) |
| . . . . . . |

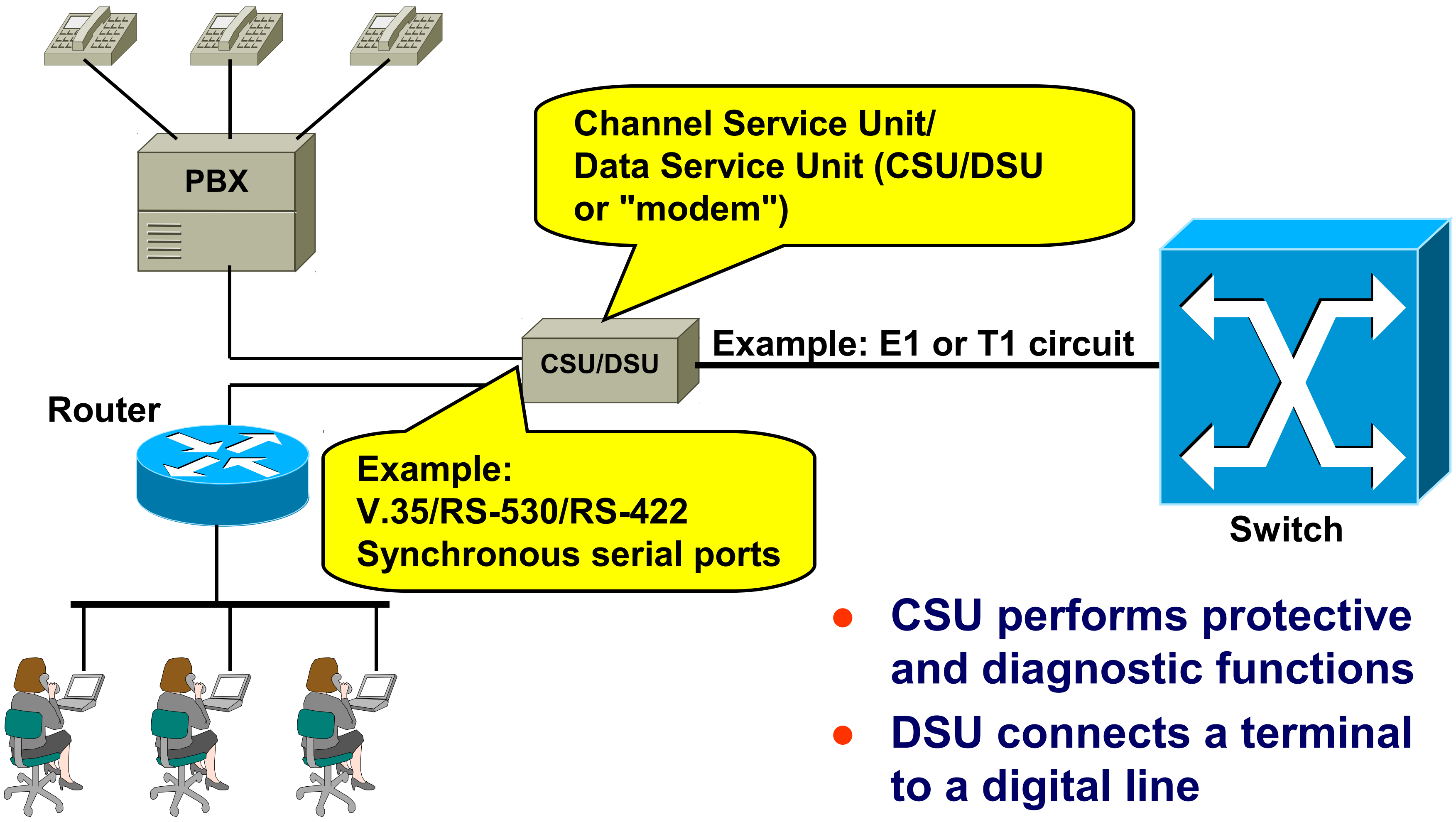
| |
|---|
| . . . . . . |
| T3(18) → T4(5) |
| T3(19) → T1(1) |
| . . . . . . |

| |
|---|
| . . . . . . |
| T4(4) → TB(9) |
| T4(5) → TB(5) |
| . . . . . . |

# Circuit Switching – Facts

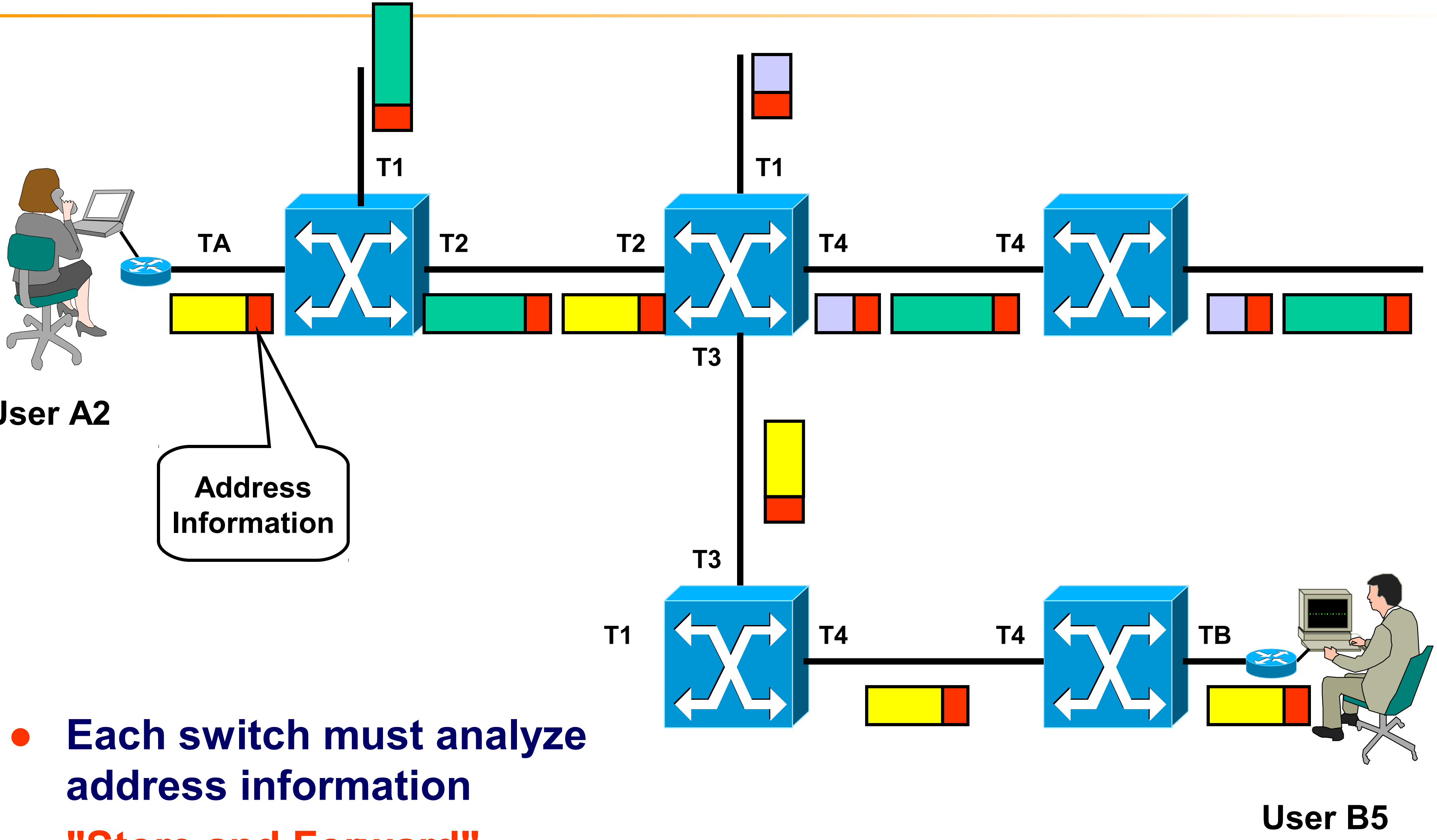
- Based on deterministic TDM
  - Minimal delay
  - Protocol transparent
  - Possibly bad utilization
  - Good for isochronous traffic (voice)
- Switching table entries
  - Static (manually configured)
  - Dynamic (signaling protocol)
  - Scales with number of connections!

# Typical User-Configuration



- CSU performs protective and diagnostic functions
- DSU connects a terminal to a digital line

# Packet Switching

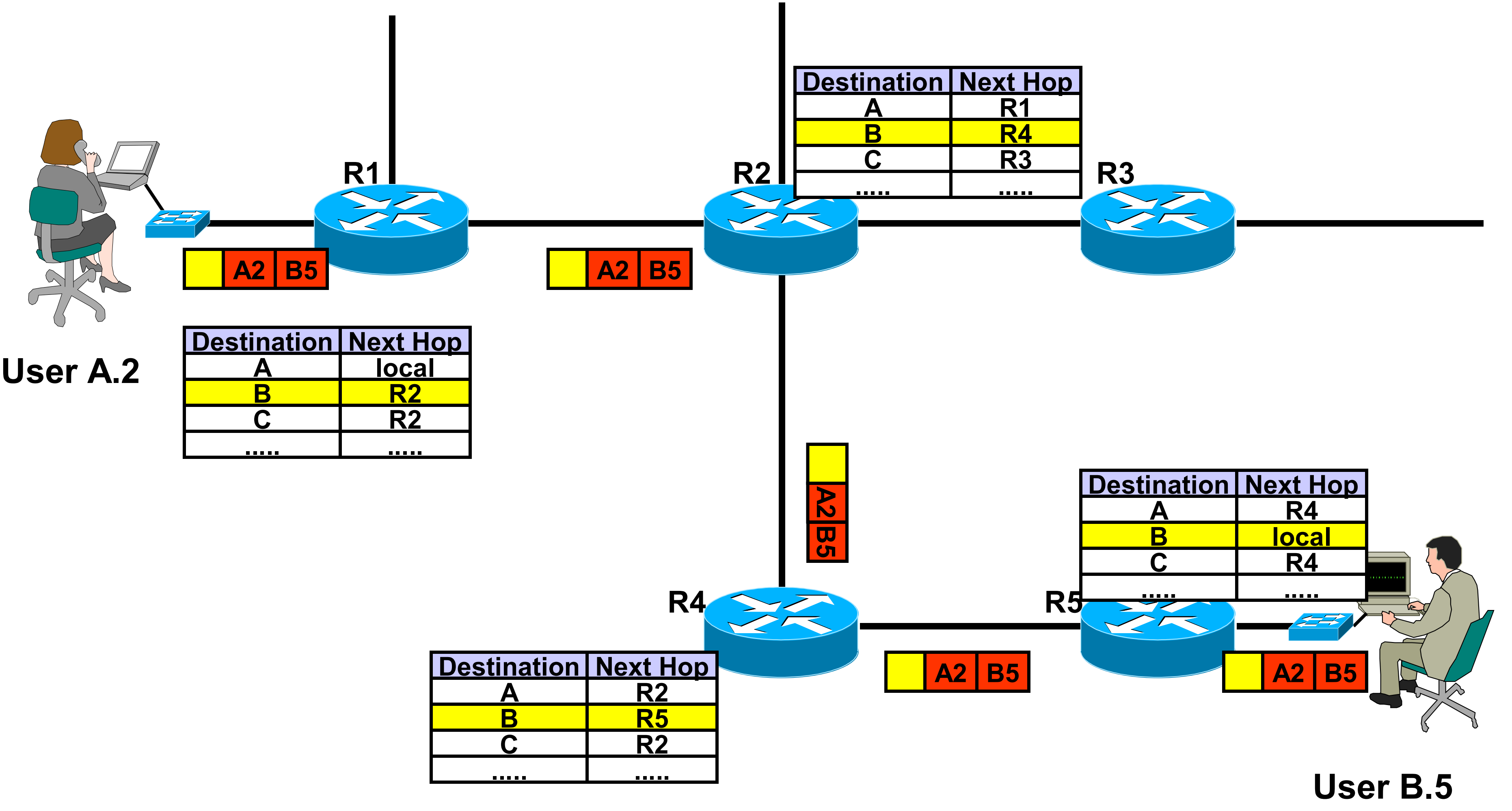


- Each switch must analyze address information
- "Store and Forward"

# Technology Differences

- Datagram Principle
  - Global and routable addresses
  - Connectionless
  - Routing Table
- Virtual Call Principle
  - Local addresses
  - Connectionoriented
  - Switching Table

# Datagram



| Destination | Next Hop |
|---|---|
| A | R1 |
| B | R4 |
| C | R3 |
| ..... | ..... |

| Destination | Next Hop |
|---|---|
| A | local |
| B | R2 |
| C | R2 |
| ..... | ..... |

| Destination | Next Hop |
|---|---|
| A | R4 |
| B | local |
| C | R4 |
| ..... | ..... |

| Destination | Next Hop |
|---|---|
| A | R2 |
| B | R5 |
| C | R2 |
| ..... | ..... |

# Datagram – Facts (1)

- Addresses contain topological information
  - Must be globally unique
- Routing table is configured
  - Static (manually)
  - Dynamic (routing protocols)
- Endless circling in case of routing loops
  - Important issue among routing protocols
- Requires "routable" or "routed" protocols

# Datagram – Facts (2)

- No connection establishment necessary
  - Faster delivery of first data
  - No resource reservation (bad QoS)
- Sequence not guaranteed
  - Rerouting on topology change
  - Load sharing on redundant paths
  - End stations must care

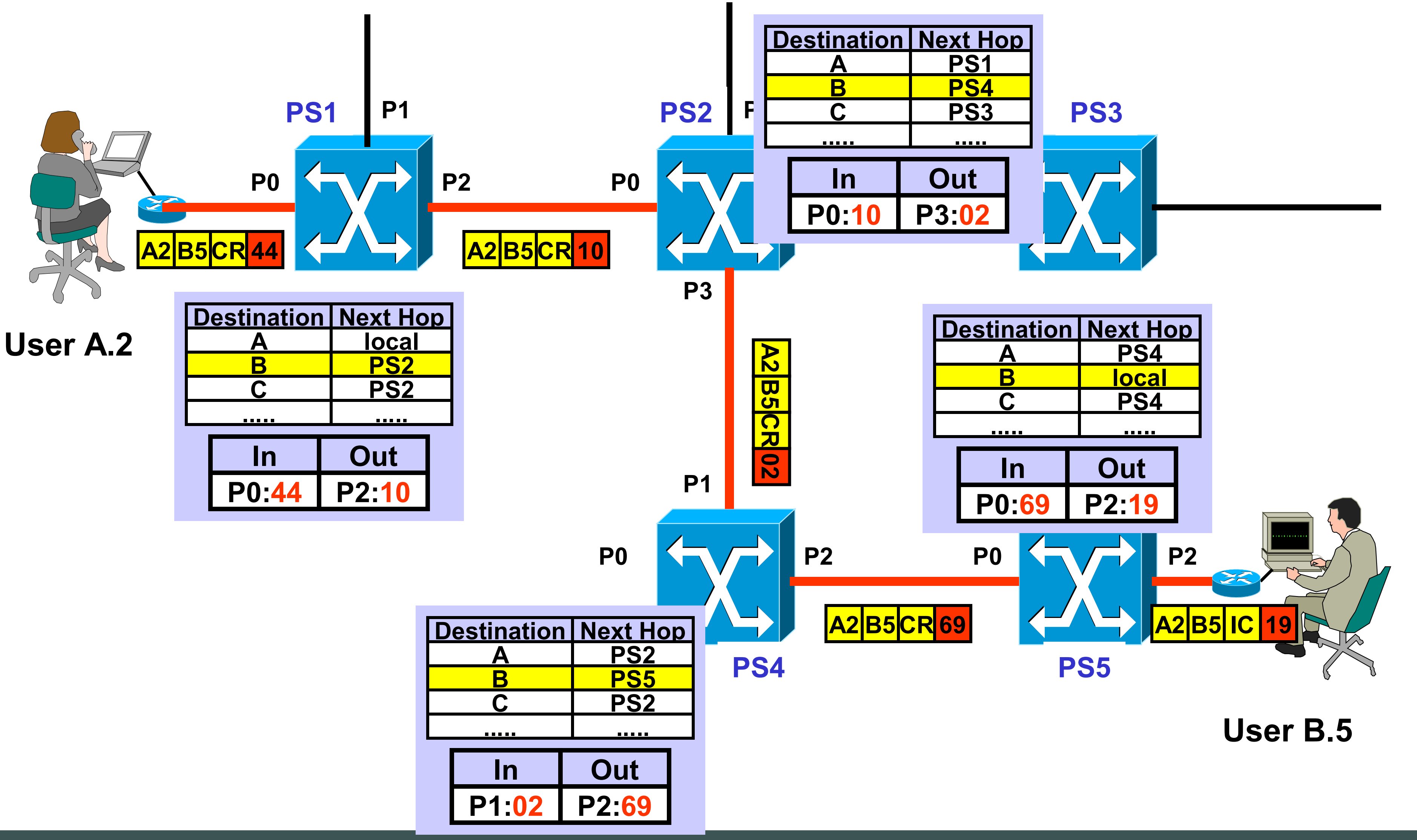
# Datagram – Facts (3)

- Best effort service
  - Router may drop packets
  - Reliable data transport requires good transport layer ("Dumb network, smart hosts")
- Simple protocols
  - Easy to implement (Internet's success)
- Proactive flow control difficult
  - Since routes might change

# Examples

- IP
- IPX
- Appletalk
- OSI CLNP

# Virtual Call – CR



PS1:

| Destination | Next Hop |
|---|---|
| A | local |
| B | PS2 |
| C | PS2 |
| ..... | ..... |

| In | Out |
|---|---|
| P0:44 | P2:10 |

PS2:

| Destination | Next Hop |
|---|---|
| A | PS1 |
| B | PS4 |
| C | PS3 |
| ..... | ..... |

| In | Out |
|---|---|
| P0:10 | P3:02 |

PS4:

| Destination | Next Hop |
|---|---|
| A | PS2 |
| B | PS5 |
| C | PS2 |
| ..... | ..... |

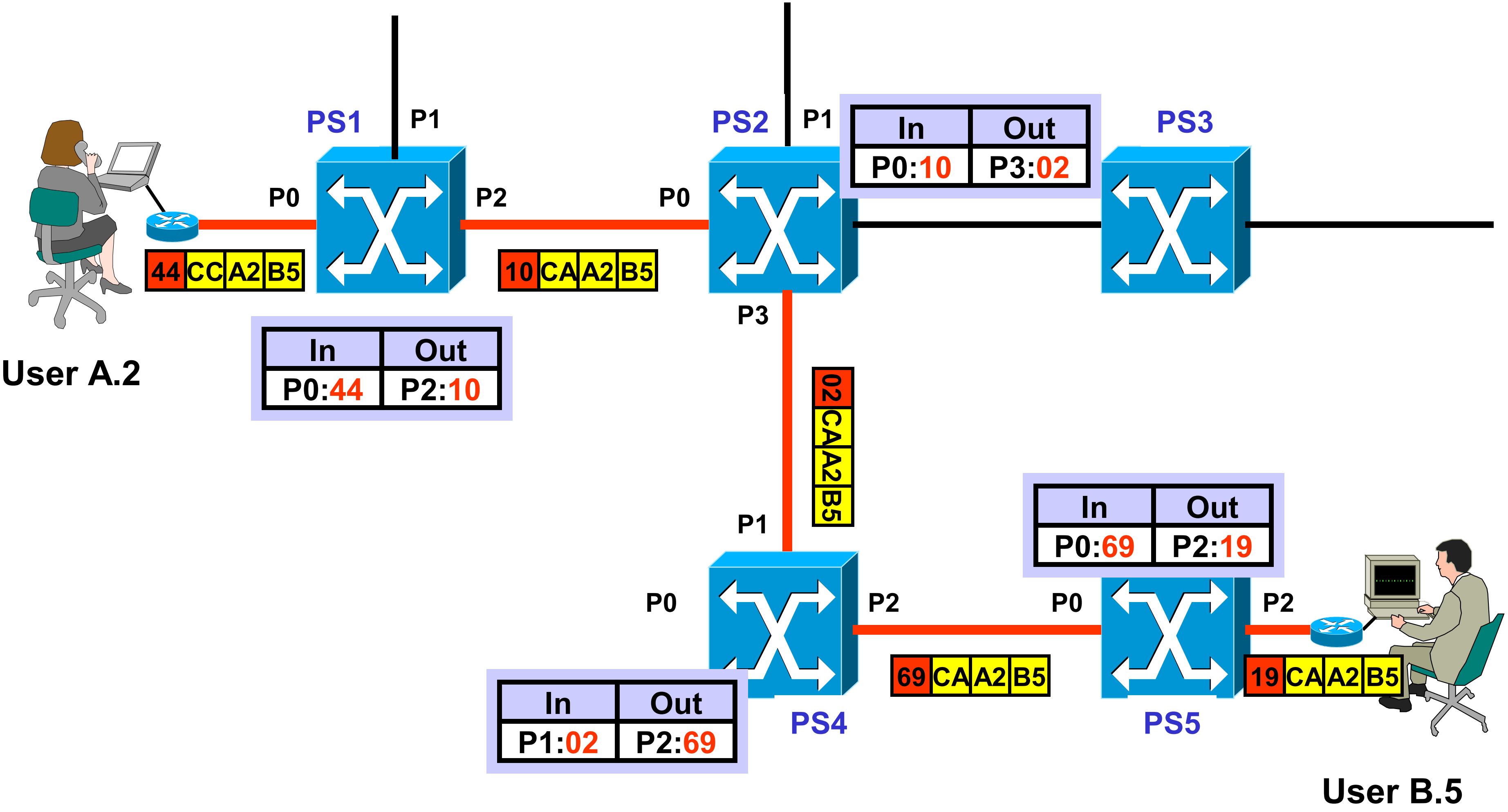
| In | Out |
|---|---|
| P1:02 | P2:69 |

PS5:

| Destination | Next Hop |
|---|---|
| A | PS4 |
| B | local |
| C | PS4 |
| ..... | ..... |

| In | Out |
|---|---|
| P0:69 | P2:19 |

# Virtual Call – CA



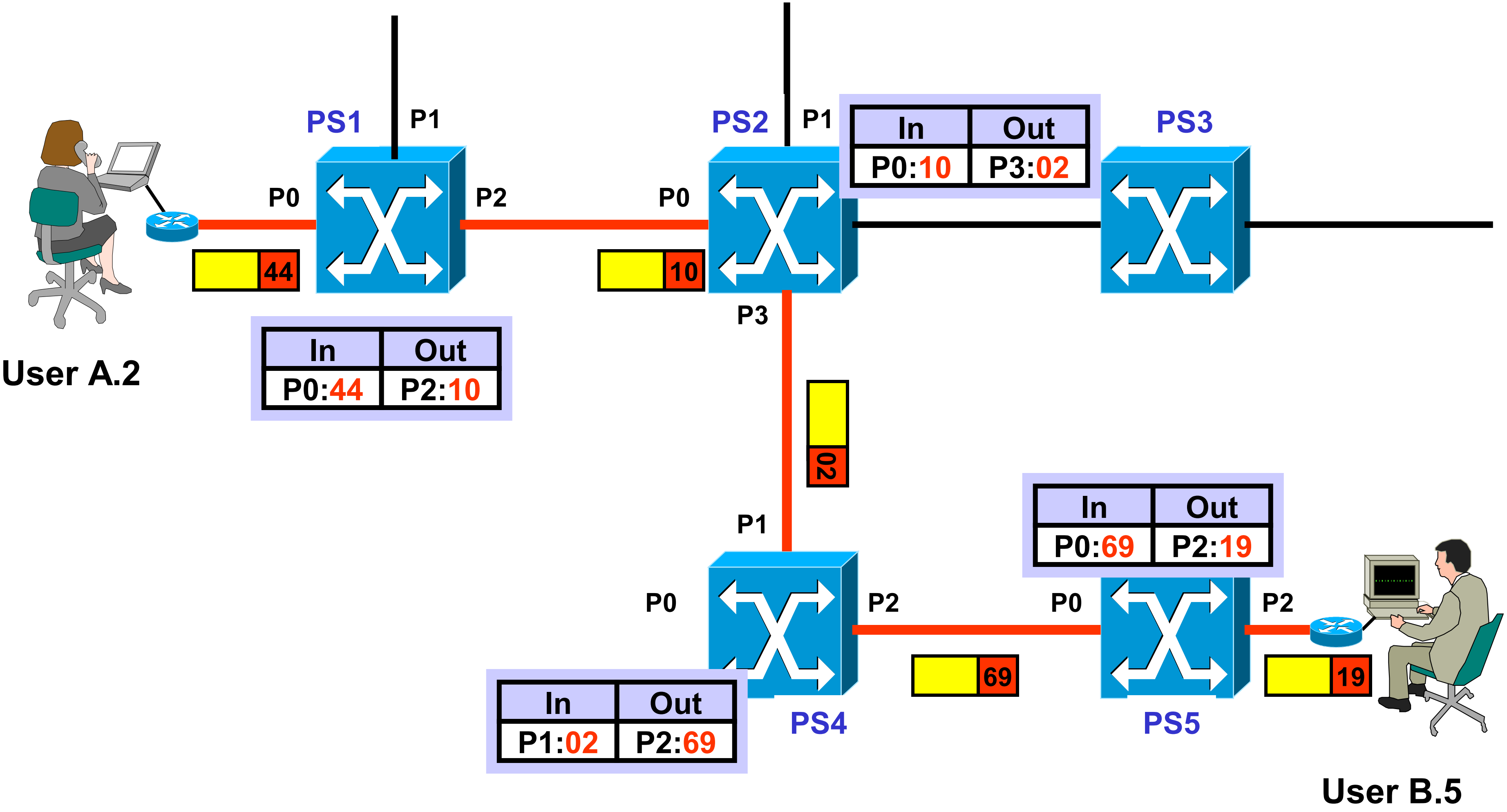
| In | Out |
|---|---|
| P0:44 | P2:10 |

| In | Out |
|---|---|
| P0:10 | P3:02 |

| In | Out |
|---|---|
| P1:02 | P2:69 |

| In | Out |
|---|---|
| P0:69 | P2:19 |

# Virtual Call – Data



| In | Out |
|---|---|
| P0:44 | P2:10 |

| In | Out |
|---|---|
| P0:10 | P3:02 |

| In | Out |
|---|---|
| P1:02 | P2:69 |

| In | Out |
|---|---|
| P0:69 | P2:19 |

# Virtual Call – Facts (1)

- Connection establishment
  - Through routing process (!)
  - Globally unique topology-related addresses necessary
  - Creates entries in switching tables
  - Can reservate switching resources (QoS)
- Packet switching relies on local identifiers
  - Not topology related
  - Only unique per port

# Virtual Call – Facts (2)

- Packet switching is much faster than packet forwarding of routers
  - Routing process is complex, typically implemented in software
  - Switching is simple, typically implemented in hardware

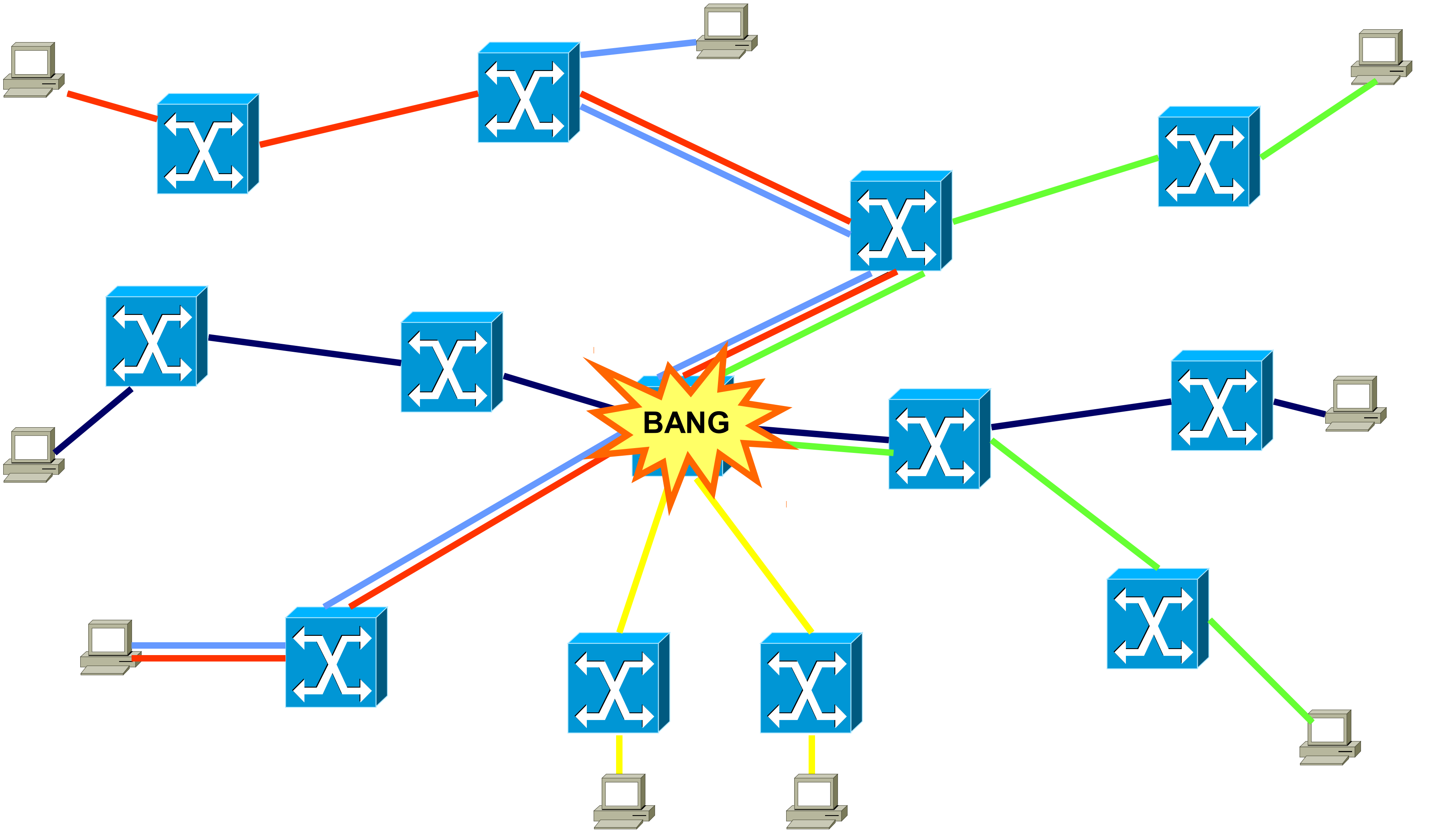
# Virtual Call – Facts (3)

- Connection can be regarded as virtual pipe
  - Sequence is guaranteed
  - Resources can be guaranteed
- Network failures disrupt pipe
  - Connection re-establishment necessary
  - Datagram networks are more robust

# Virtual Call – Facts (4)

- Virtual call multiplex
  - Multiple virtual pipes per switch and interface possible
  - Pipes are locally distinguished through connection identifier
- Other names for connection identifier
  - LCN (X.25)
  - DLCI (Frame Relay)
  - VPI/VCI (ATM)

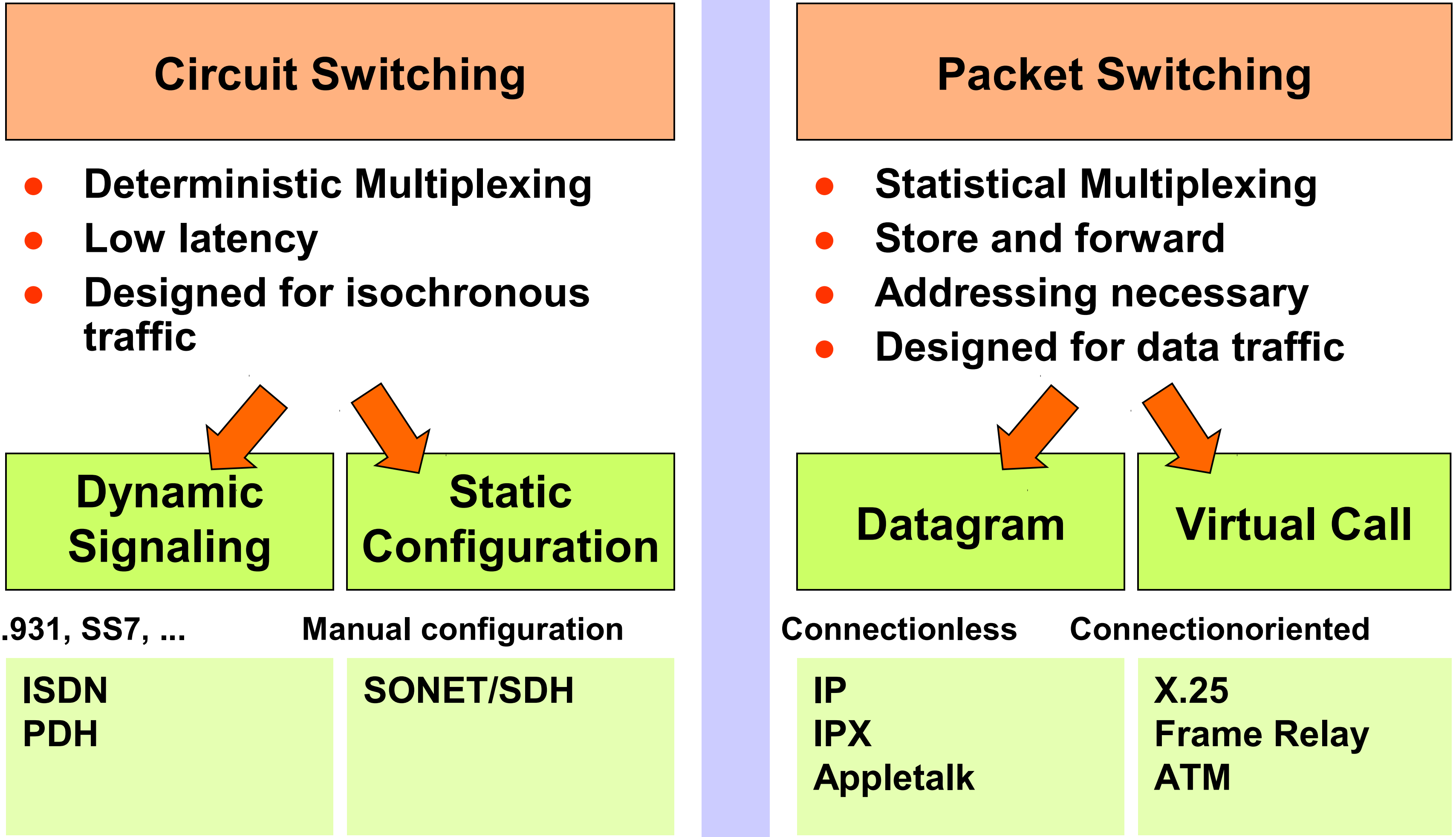
# Example



# Two Service Types

- Switched Virtual Circuit (SVC)
  - Dynamic establishment as shown
  - At the end a proper disconnection procedure necessary
- Permanent Virtual Circuit (PVC)
  - No establishment and disconnection procedures necessary
  - Switching tables preconfigured by administrator

# Taxonomy



## Circuit Switching

- Deterministic Multiplexing
- Low latency
- Designed for isochronous traffic

**Dynamic Signaling**

Q.931, SS7, ...

ISDN
PDH

**Static Configuration**

Manual configuration

SONET/SDH

## Packet Switching

- Statistical Multiplexing
- Store and forward
- Addressing necessary
- Designed for data traffic

**Datagram**

Connectionless

IP
IPX
Appletalk

**Virtual Call**

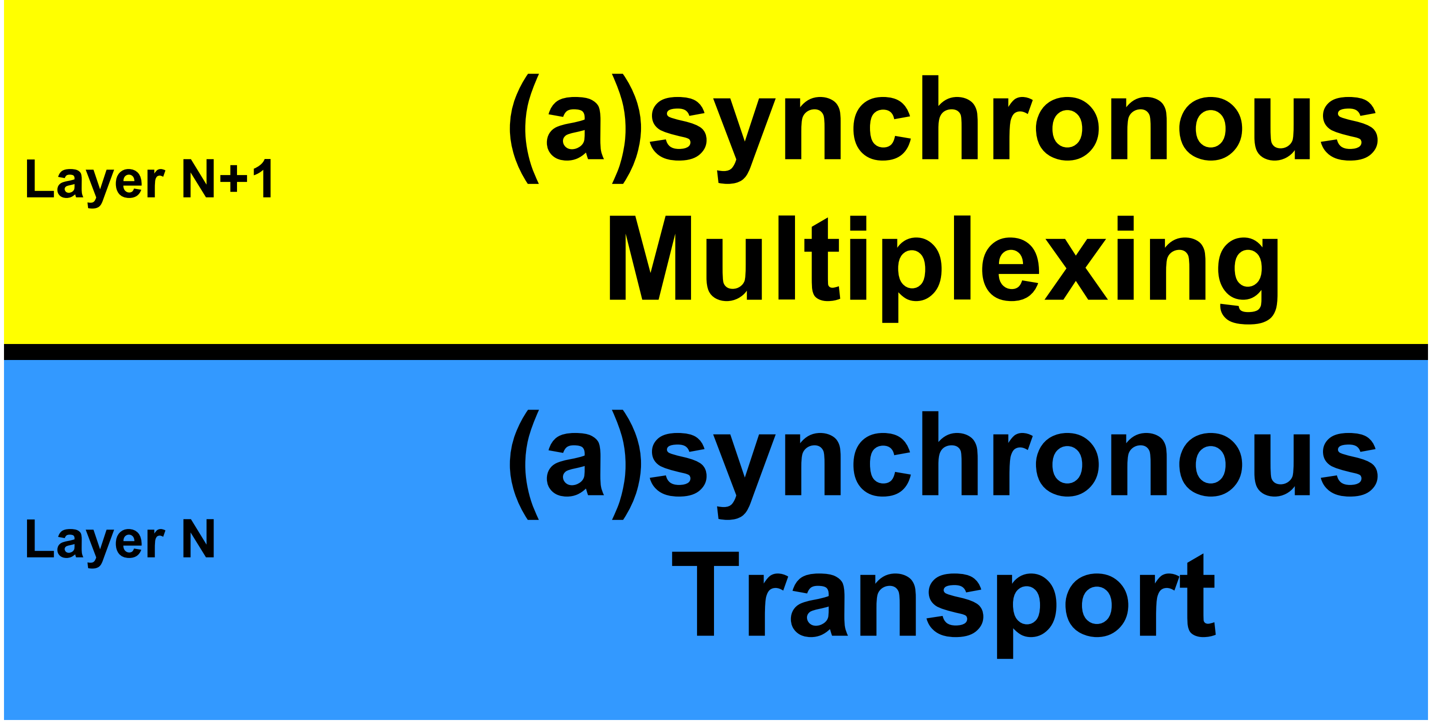
Connectionoriented

X.25
Frame Relay
ATM

# Summary

- Only two worlds: circuit switching or packet switching
  - The first is good for voice the latter is good for data
  - Everybody wants to have the best of both worlds
- Datagram (CL) versus Virtual Call (CO)
  - Different address types (!)

# Synchronization Revisited



Layer N+1

(a)synchronous Multiplexing

Layer N

(a)synchronous Transport

# Quiz

- Derive Metcalfe's law. Which well-known formula looks very similar?
- Let's improve the VC principle! What's the advantage of using more than one label per packet?
- How do hash tables work?
- How can we get the best of both worlds (circuit/packet) ?