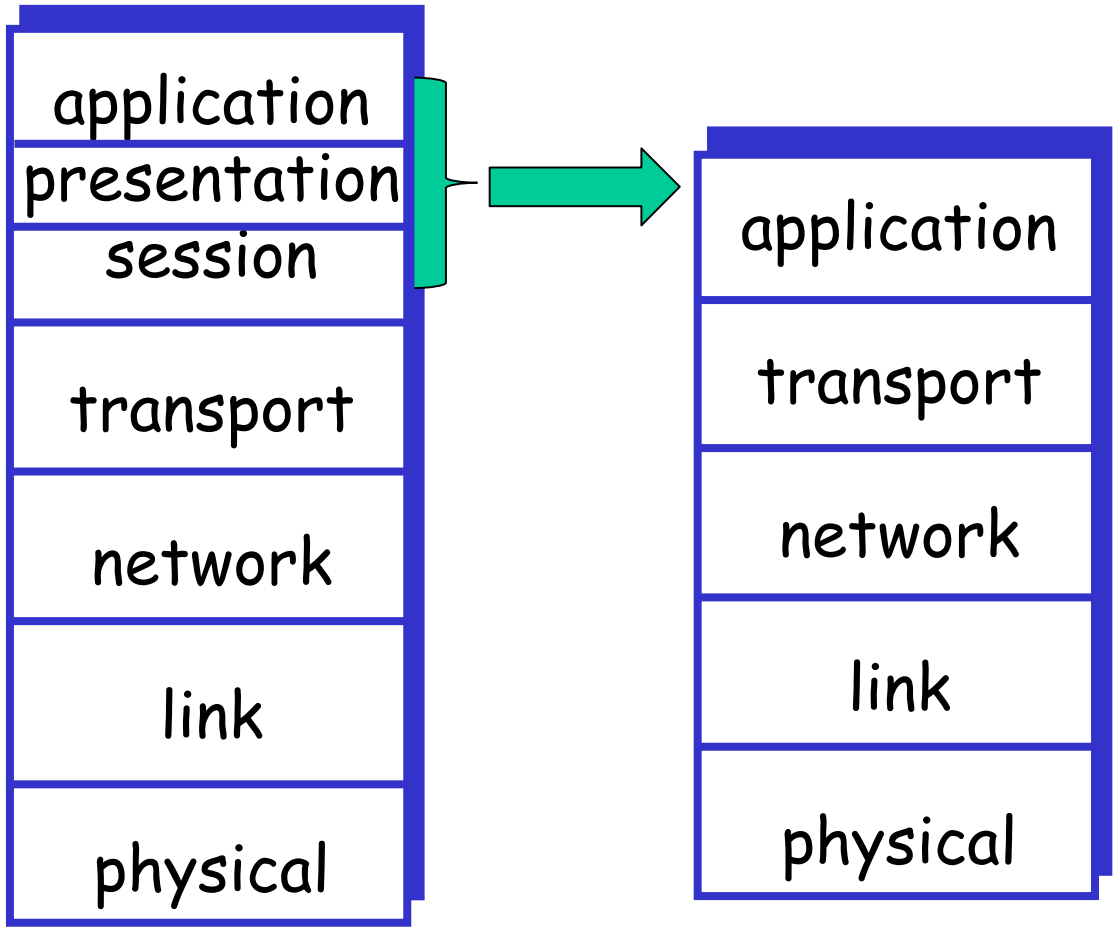
- Internet protocol stack
- Encapsulation
- Connection oriented VS connectionless services
- Circuit Switching
- Packet Switching
- Store-and-forward switches
- Multiplexing: TDM, FDM, Statistical multiplexing, CDMA

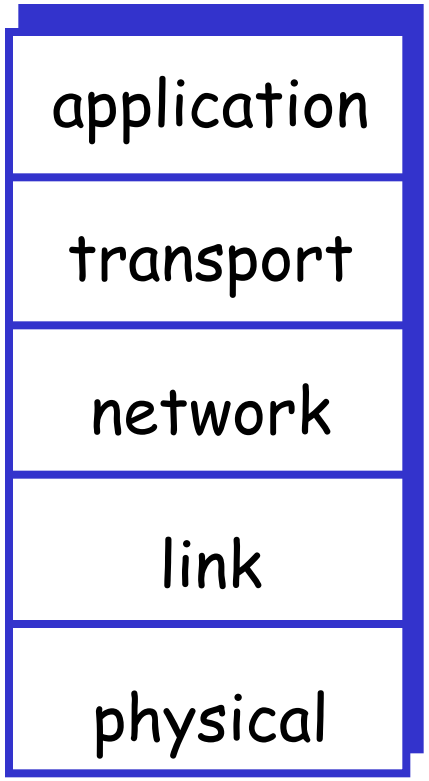
OSI VS Internet protocol stack



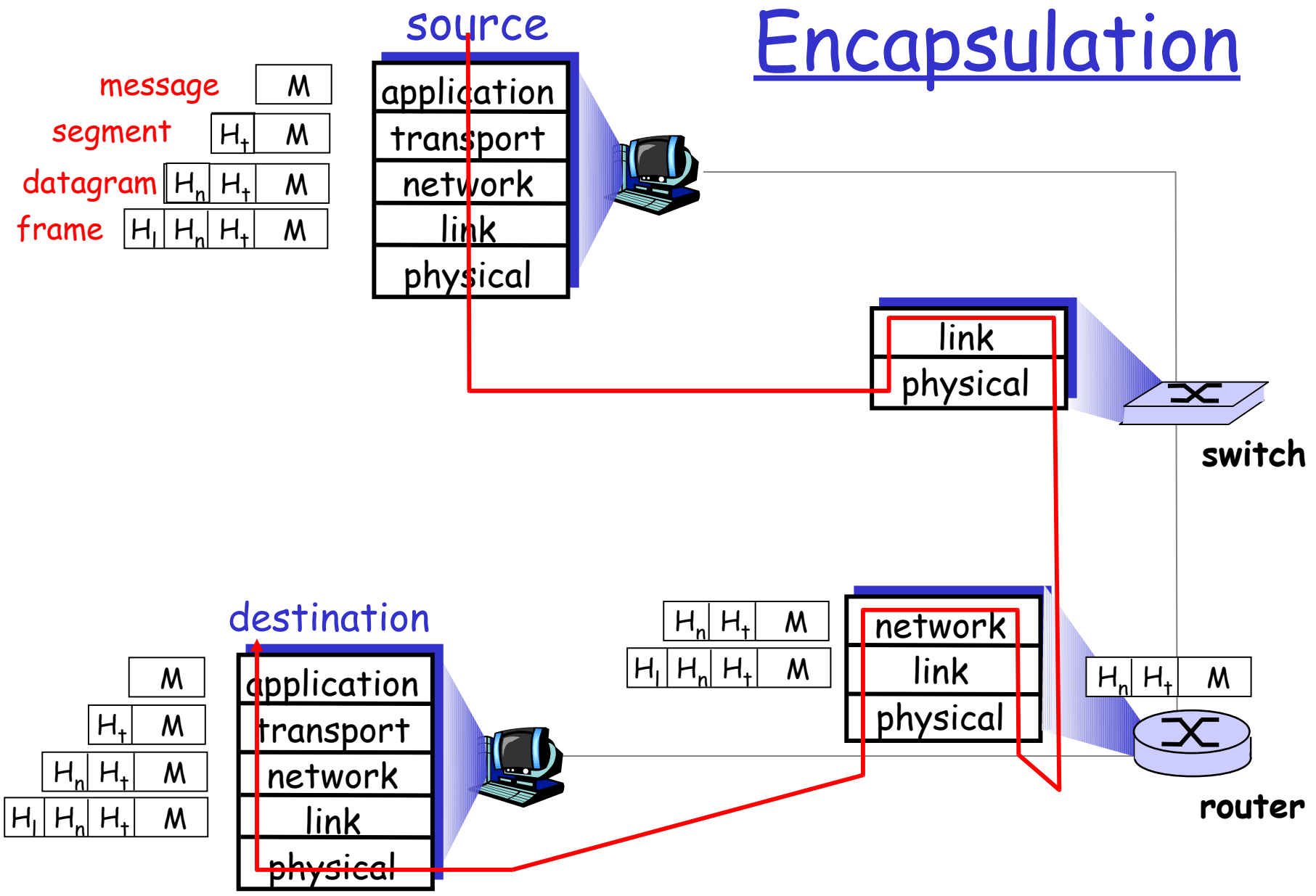
- Internet stack "missing" these layers!
  - these services, *if needed,* must be implemented in application
  - needed?

Internet protocol stack

- application: supporting network applications
  - FTP, SMTP, HTTP,DNS protocols
- transport: process-process data transfer
  - TCP, UDP protocols
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - PPP, Media Access Control (Ethernet, DSL, ISDN, FDDI)
- physical: bits "on the wire"



Encapsulation



Why layering?

Dealing with complex systems:

- Abstraction
- explicit structure allows identification, relationship of complex system's pieces
  - layered reference model for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system

Connection Oriented services

- Establish end to end logical or physical connection before any data are sent
- Involves handshaking
- Reliable data transfer may be involved (e.g. TCP)

Data link layer examples:

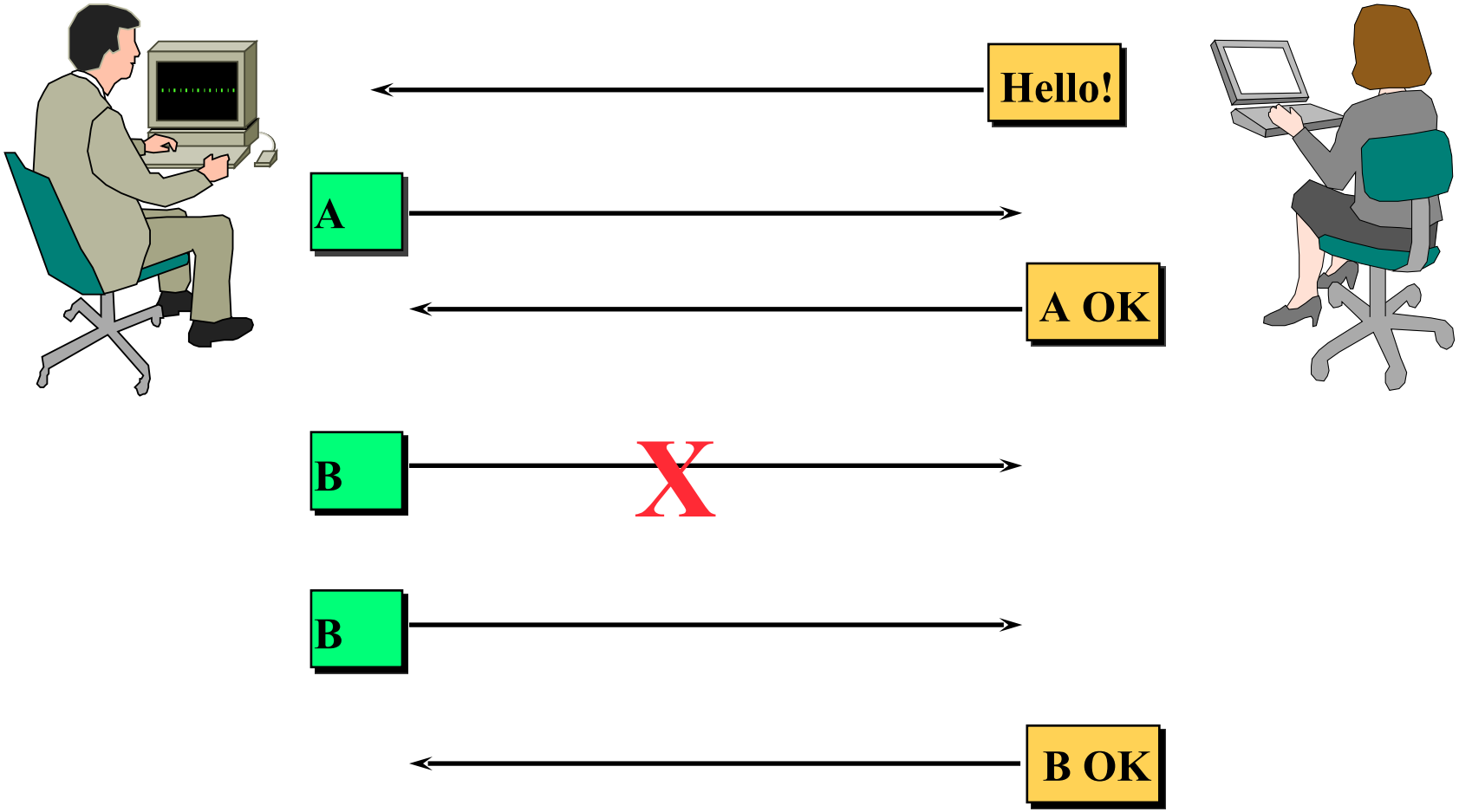
- Circuit mode communication
- Virtual Circuits (packet switching). Same path! We just need a VCI.

Transport layer examples:

- TCP

Transmission Control Protocol (TCP)

(belongs to transport layer)



Connectionless services

- No handshaking!
- Each data packet carries information about the destination address (datagram)

Network layer example:

- IP protocol

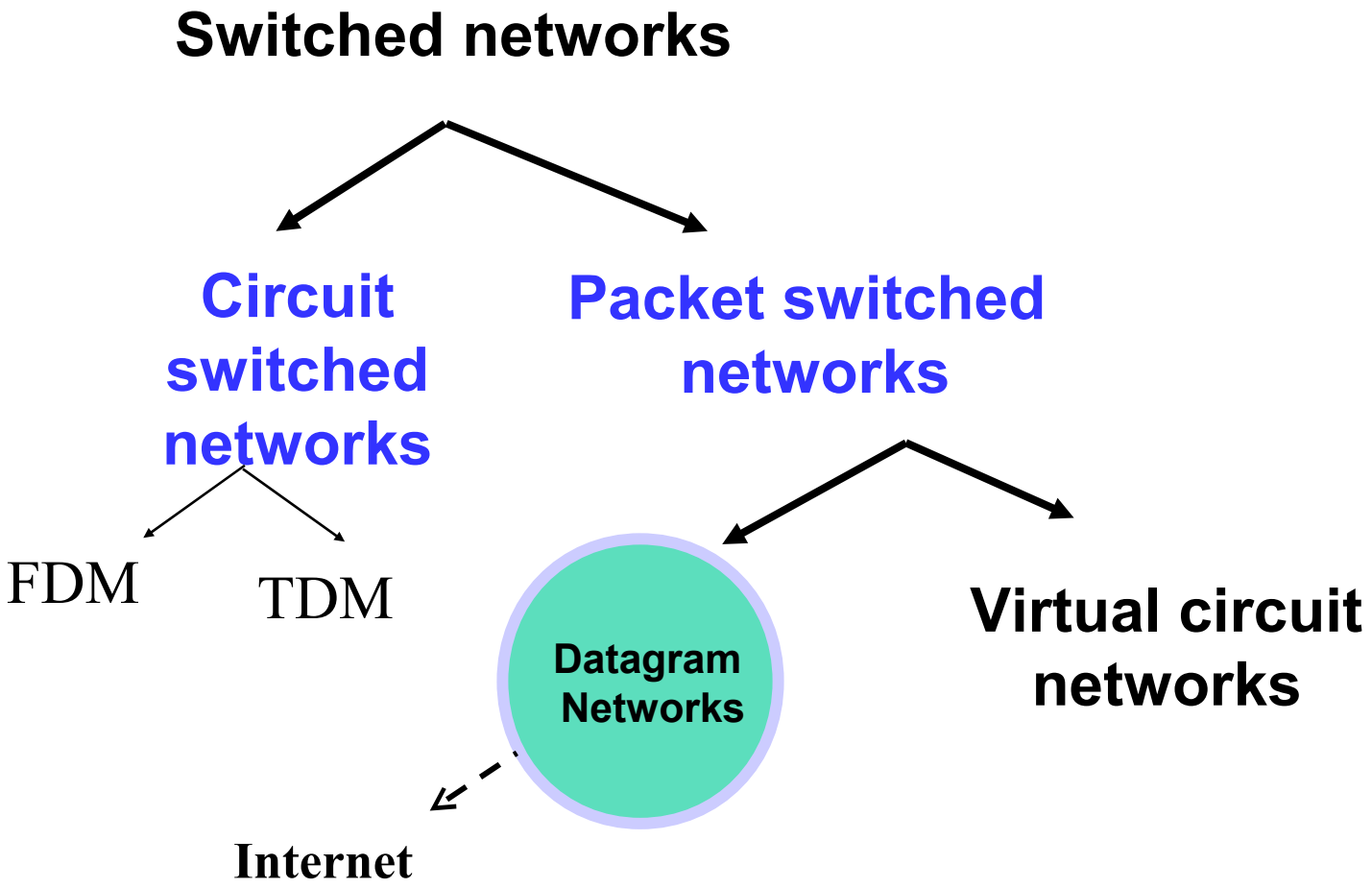
Transport layer examples:

- UDP

Connection oriented VS connectionless services

- The distinction takes place in several layers
- Packet switching examples in both categories
- Connection oriented service on connectionless service??

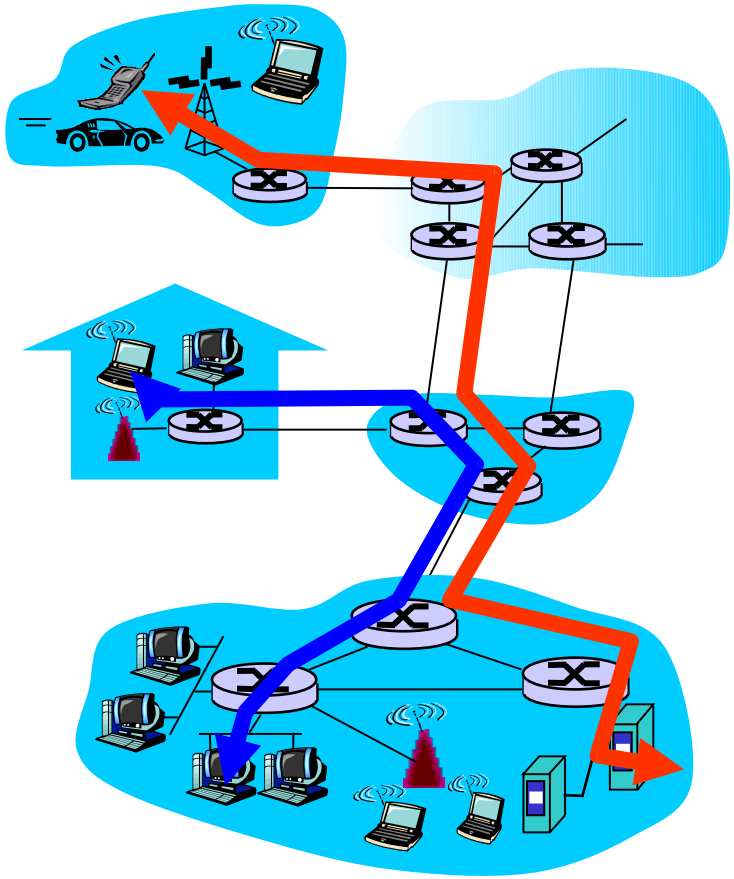
  TCP/IP


Switched networks
Circuit switched networks
Packet switched networks
FDM
TDM
Datagram Networks
Virtual circuit networks
Internet

Circuit Switching

End-end resources reserved for "call"

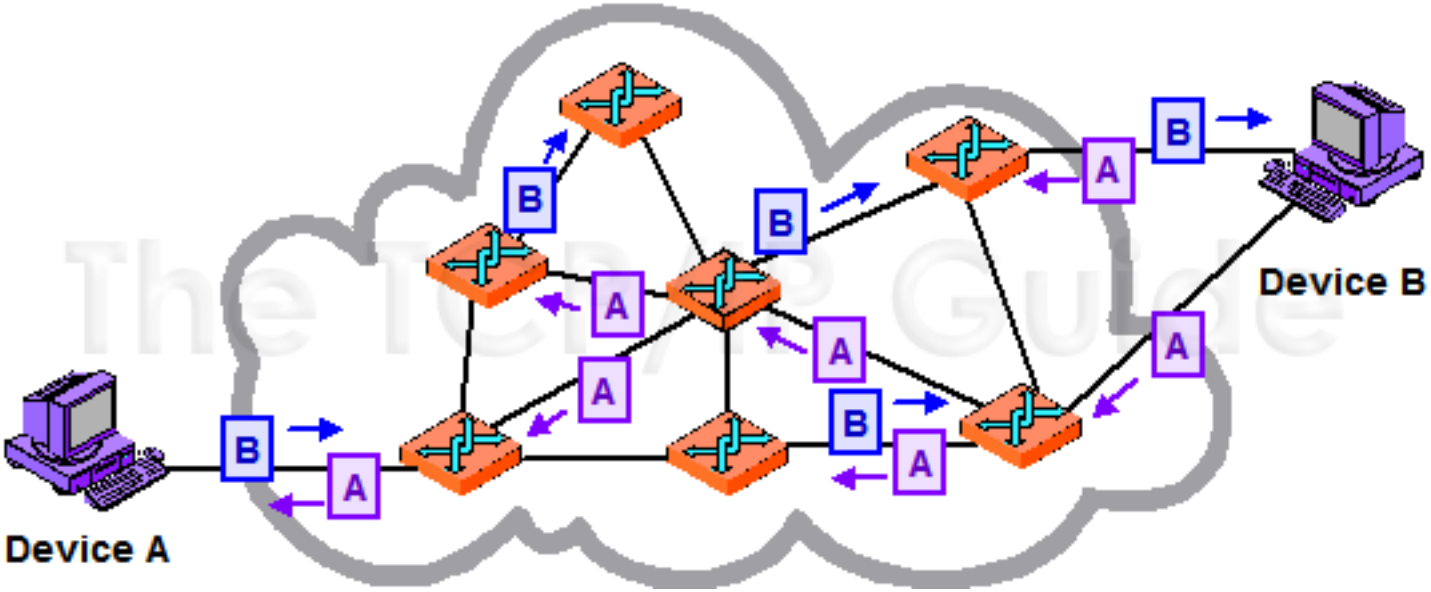
- link bandwidth, switch capacity
- dedicated resources: no sharing (??)
- circuit-like (guaranteed) performance
- call setup required



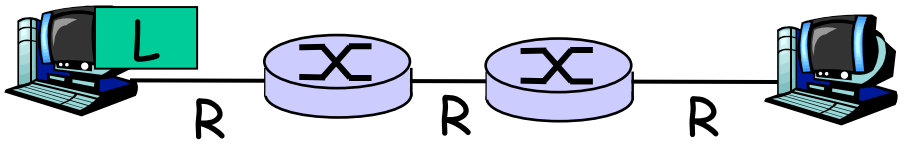
Packet Switching

each end-end data stream divided into *packets*

- user A, B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*
- Same route??



Packet-switching: store-and-forward



- takes L/R seconds to transmit (push out) packet of L bits on to link at R bps
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link
- delay = 3L/R (assuming zero propagation delay)

Example:

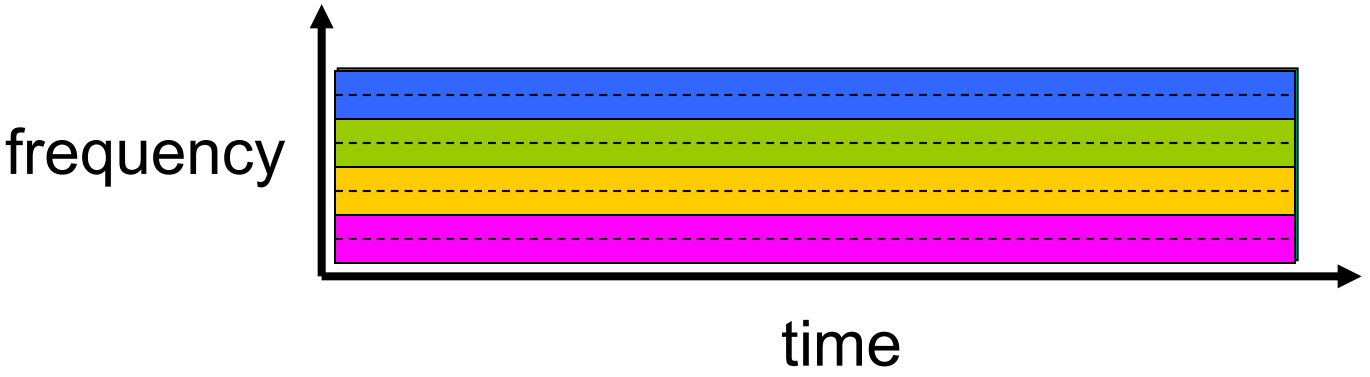
- L = 7.5 Mbits
- R = 1.5 Mbps
- transmission delay = 15 sec

Circuit Switching: FDM and TDM

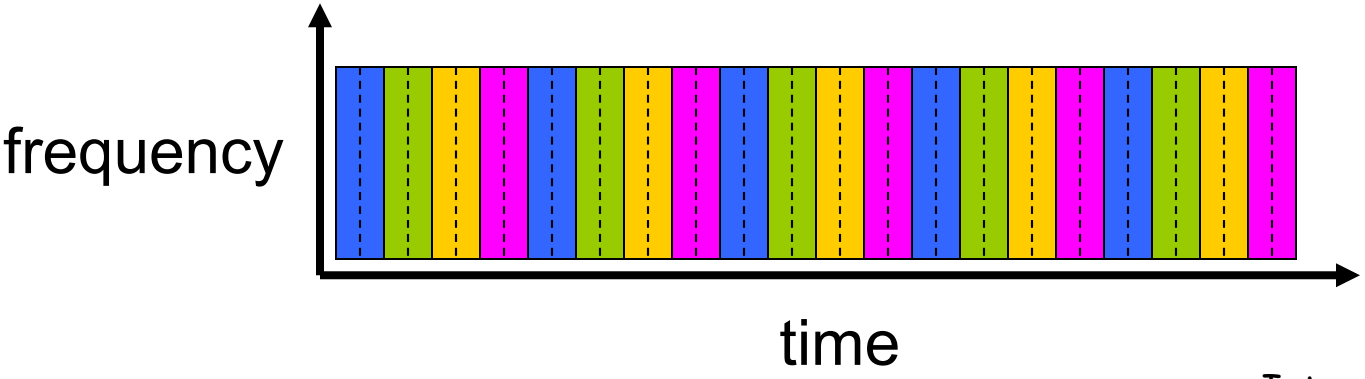
Example:

4 users

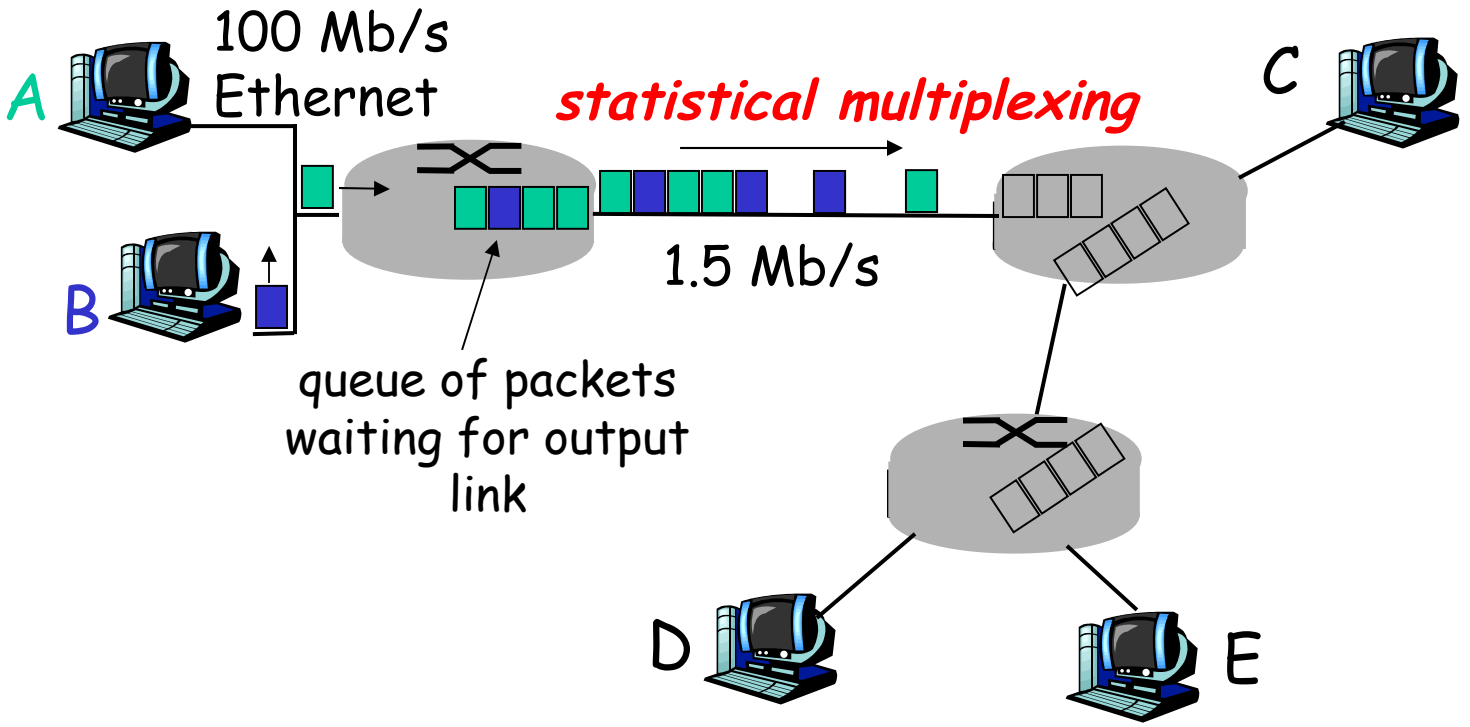
FDM



TDM



Packet Switching: Statistical Multiplexing



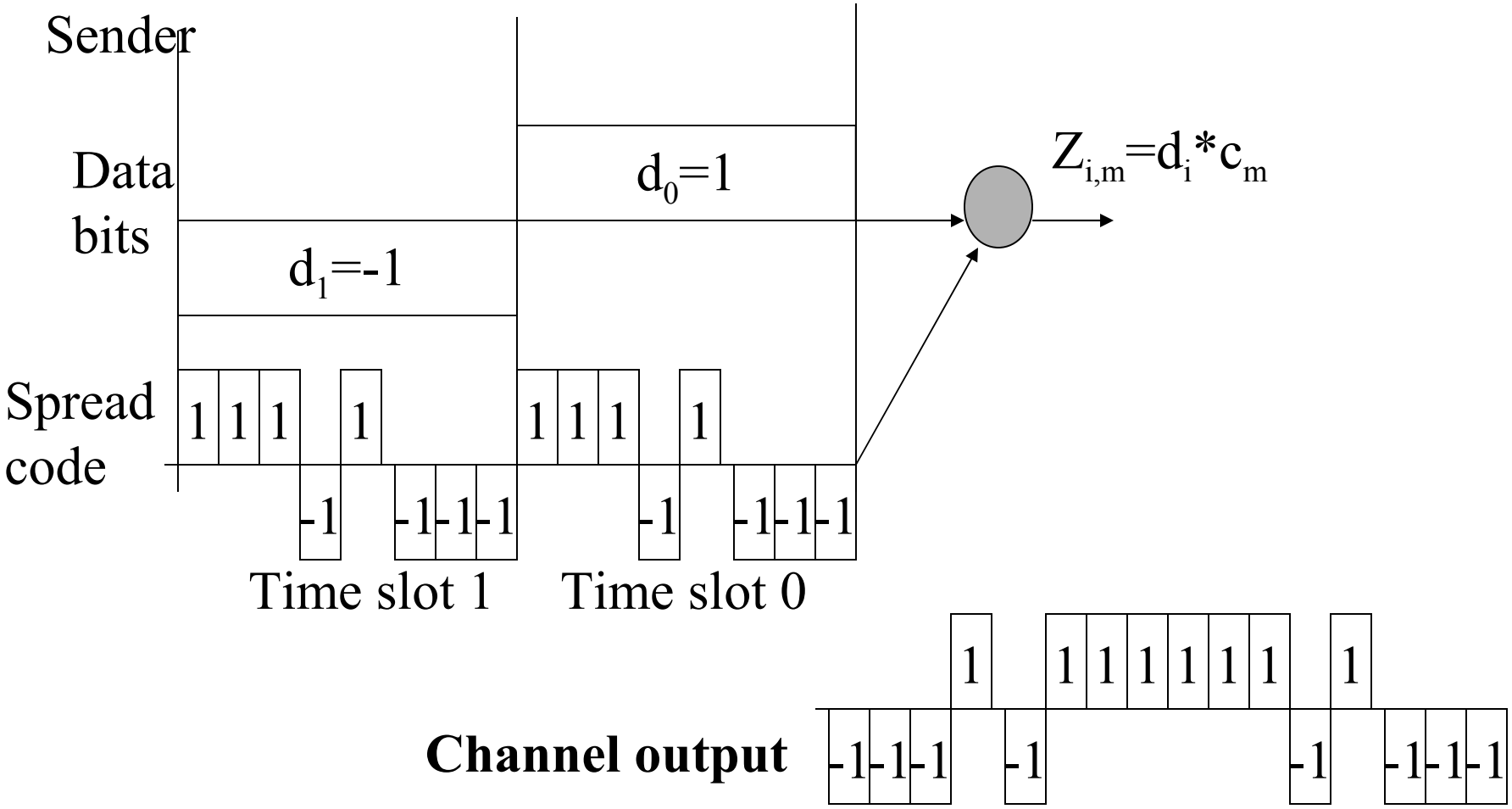
Sequence of A & B packets does not have fixed pattern, bandwidth shared on demand ***statistical multiplexing***.

TDM: each host gets same slot in revolving TDM frame.

Πολλαπλή Πρόσβαση Διαίρεσης Κώδικα (CDMA)

- Στο CDMA ορίζεται σε κάθε κόμβο ένας διαφορετικός κώδικας
- Οι κώδικες είναι **ορθογώνιοι μεταξύ τους** (δηλ. το εσωτερικό γινόμενο μεταξύ οποιωνδήποτε δύο κωδίκων είναι 0)
- Κάθε κόμβος χρησιμοποιεί το δικό του μοναδικό κώδικα για να κωδικοποιήσει τα bits των δεδομένων που στέλνει
- Οι κόμβοι **μπορούν να εκπέμπουν ταυτόχρονα**
- **Πολλαπλοί κόμβοι σε κάθε κανάλι**
- Οι αντίστοιχοι προς αυτούς δέκτες
  - Λαμβάνουν σωστά τα κωδικοποιημένα bits δεδομένων ενός πομπού
    - **Θεωρώντας ότι ο δέκτης γνωρίζει τον κώδικα του πομπού**, παρά τι παρεμβαλλόμενες μεταδόσεις άλλων κόμβων

Παράδειγμα CDMA



Παράδειγμα CDMA (συνέχεια)

- Όταν δεν υπάρχουν παρεμβάλλοντες πομποί
  - Ο δέκτης
    - Λαμβάνει τα κωδικοποιημένα bits
    - Ανακτά τα αρχικά bit δεδομένων, d_i, υπολογίζοντας το

$$d_i = \frac{1}{M} \sum_{m=1}^{M} Z_{i,m} * c_m$$

- **Τα παρεμβάλλοντα εκπεμπόμενα δυαδικά σήματα είναι προσθετικά**