# Organisatorisches (1)

- **Lehrstuhl „Echtzeitsysteme und Kommunikation“**
  - http://ivs.cs.uni-magdeburg.de/EuK
  - nett@ivs.cs.uni-magdeburg.de
  - Sekretariat Frau Duckstein, 29/405, Tel. 67-18345
- **Web-Adresse**

  http://ivs.cs.uni-magdeburg.de/EuK/lehre/lehrveranstaltungen/
  - Folien der Vorlesung (in englisch)
  - praktische Übungsaufgaben
  - Mitteilungen (z.B. Bekanntgabe der Übungsgruppeneinteilung)
  - Literaturhinweise
- **Übungen**
  - Übungsleiter: Herr Lindhorst (E-mail: lindhors@ivs.cs.ovgu.de)

    Herr Dornemann, Herr Penzlin
  - Wöchentlich ab 15. KW (13. April)

# Organisatorisches (2)

- **Inhalte der Übungen:**

  – Vermittlung praktischer Kenntnisse in der Anwendung

  Programmieren von TCP-Sockets. Während des Semesters wird in der Übung ein Webserver programmiert, der weiterhin einen Chat-Service anbietet. Dazu werden in regelmäßigen Abständen Teilaufgaben zu bearbeiten sein. Die Programmierung erfolgt unter Linux in C. Grundlegende C-Kenntnisse werden vorausgesetzt, bzw. die Bereitschaft, sich solche anzueignen.

  – Wenn erforderlich, gewünscht ...: Aufarbeitung des Vorlesungstoffs

  Die Übungen finden in unserem Labor statt (Raum 425).

- **Kriterien für erfolgreiche Teilnahme:**

  –schriftliche Prüfung (Scheinerwerb: Note <= 4.0)

  –Zulassungsvoraussetzungen:

  - erfolgreiche Umsetzung der praktischen Aufgaben, sowie Vorstellung und Erklärung der Implementierung

# Introduction (1)

**Computer network:**

An *interconnected* collection of *autonomous* computers

**Interconnected computers:**

Computers are said to be interconnected if they are able to exchange information

**Examples for the physical connection medium:**

copper wire, fiber optics (wired)

radio communication (micro waves), infrared (wireless)

**Autonomous computers:**

No computer can forcibly start, stop, or control computations (actions) on another one.

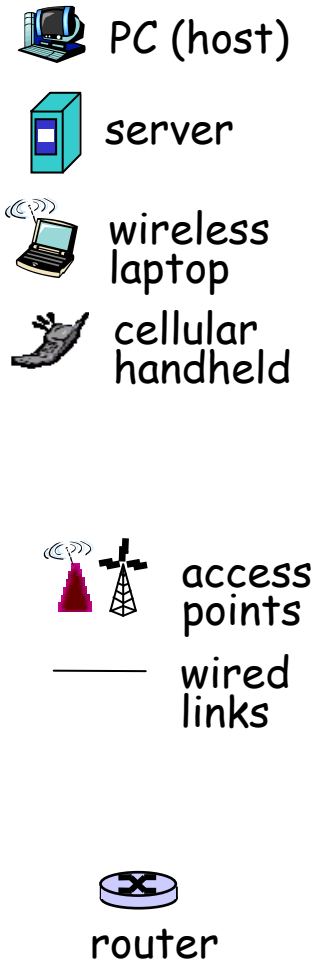
**Distributed system:**

The existence of multiple autonomous computers is transparent (not visible) to the user,i.e. no explicit path addressing (e.g. URL), no explicit moving of objects (files) etc.

**Principle vehicle for discussing computer networks and their protocols:**

The Internet

# What's the Internet: "component's" view (1)

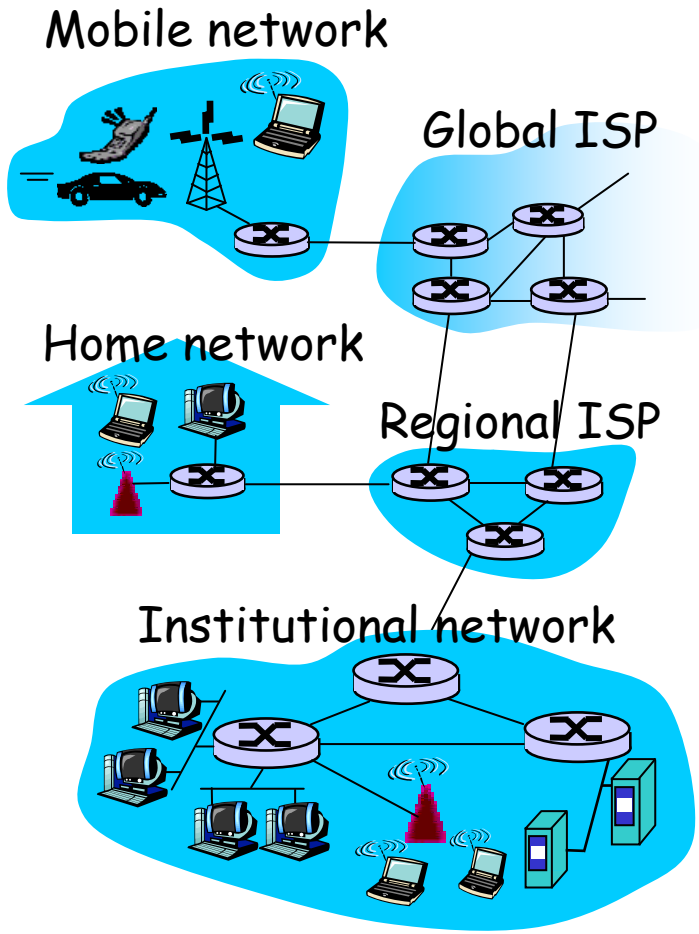



- millions of connected computing devices:

  *hosts = end systems* running *network applications*

- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate *bandwidth*

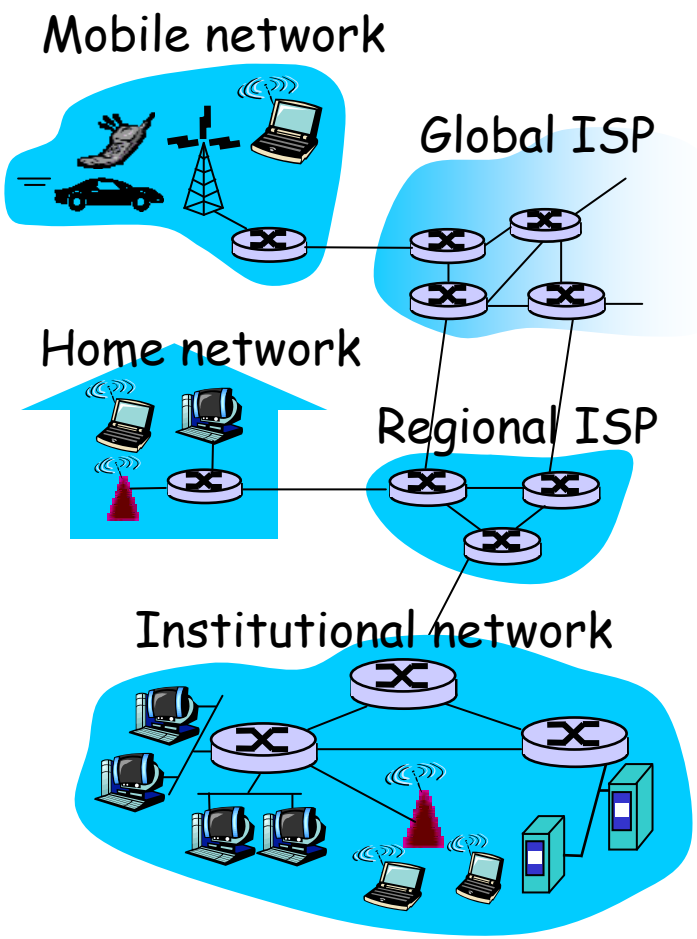
- *routers:* forward packets (chunks of data)



# What's the Internet: "component's" view (2)

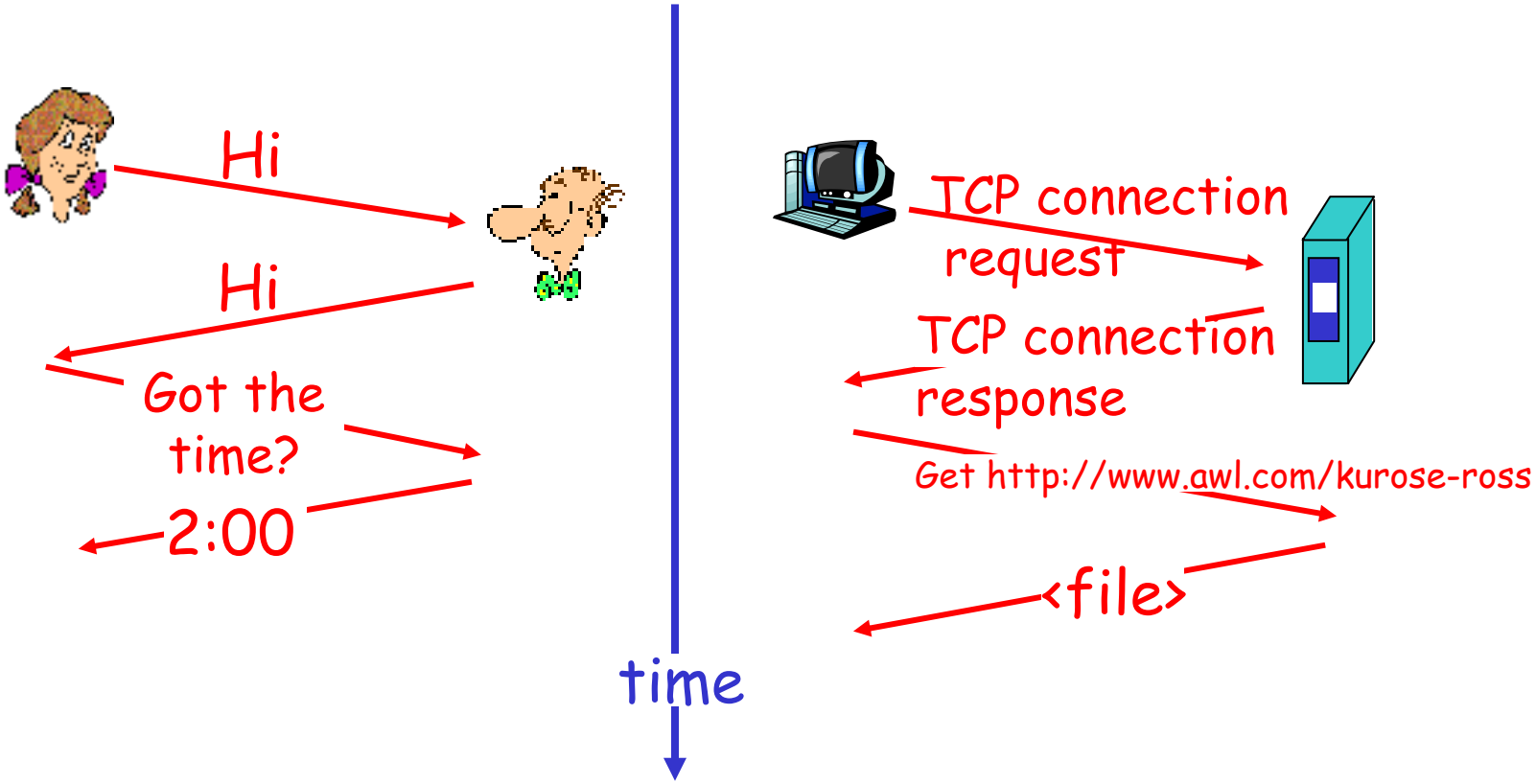
- *protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Ethernet

- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



# What's a protocol?

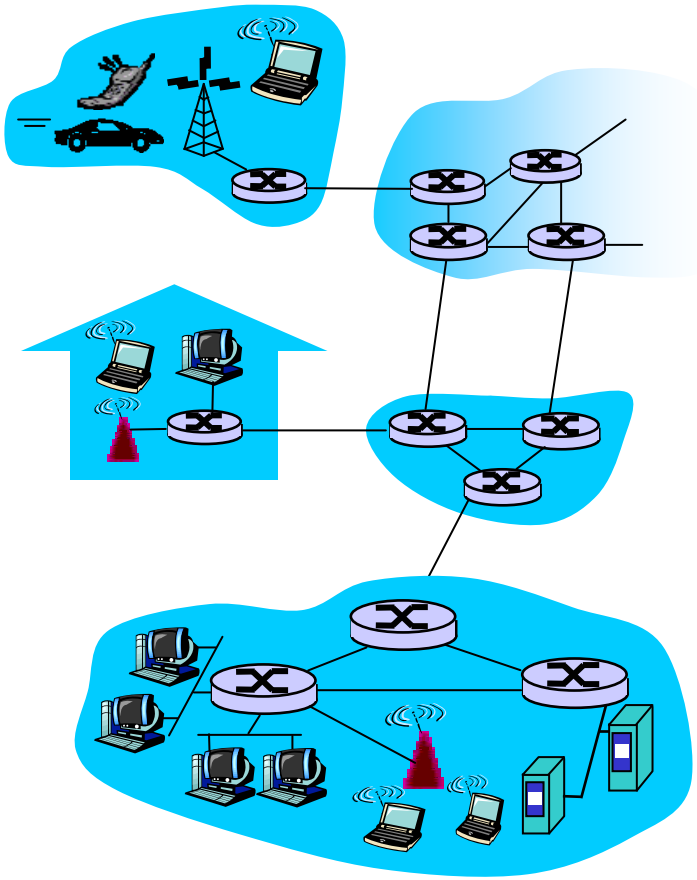
a human protocol and a computer network protocol:




*protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt*

# What's the Internet: a service view

- communication *infrastructure* enables distributed applications:
  - Web, VoIP, email, games, e-commerce, file sharing, Skype
- communication services provided to apps:
  - reliable data delivery from source to destination
  - "best effort" (unreliable) data delivery



# A closer look at network structure

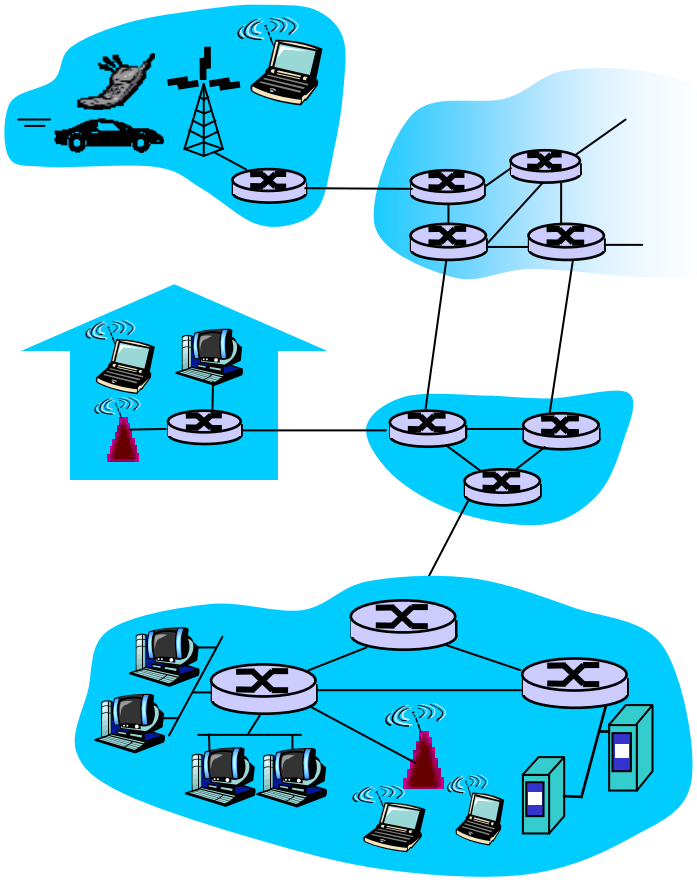
- network edge:

  applications running on hosts

- access networks:

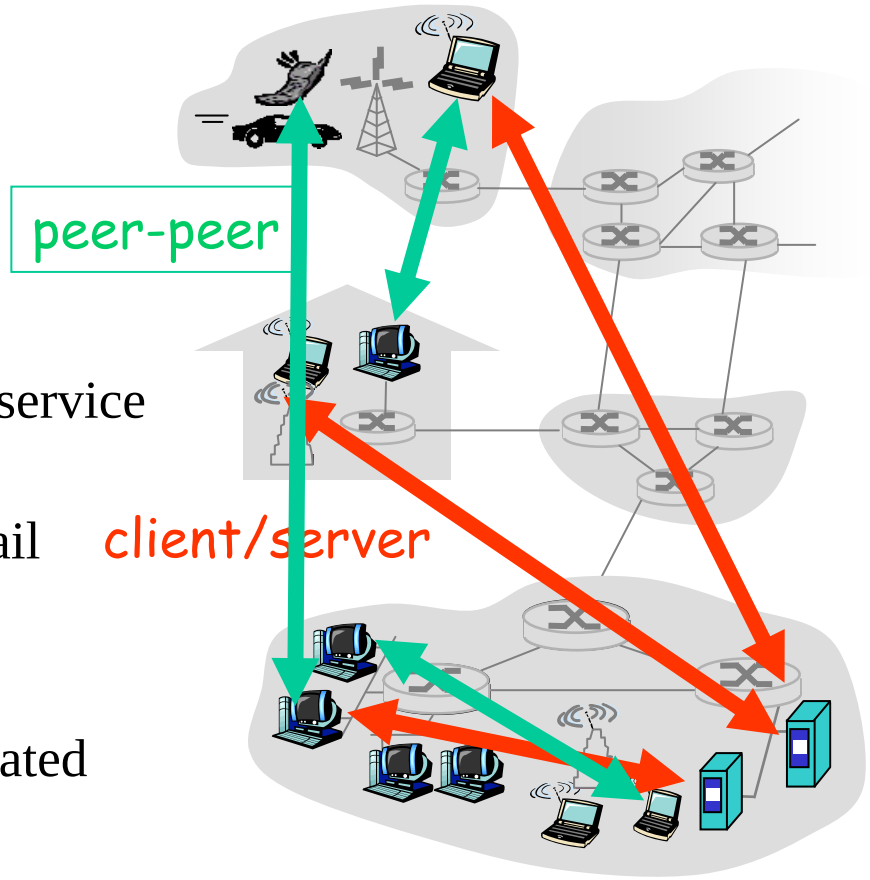
  wired or wireless communication links that connect hosts to their edge router

- network core:
  - interconnected routers
  - network of networks



# The network edge

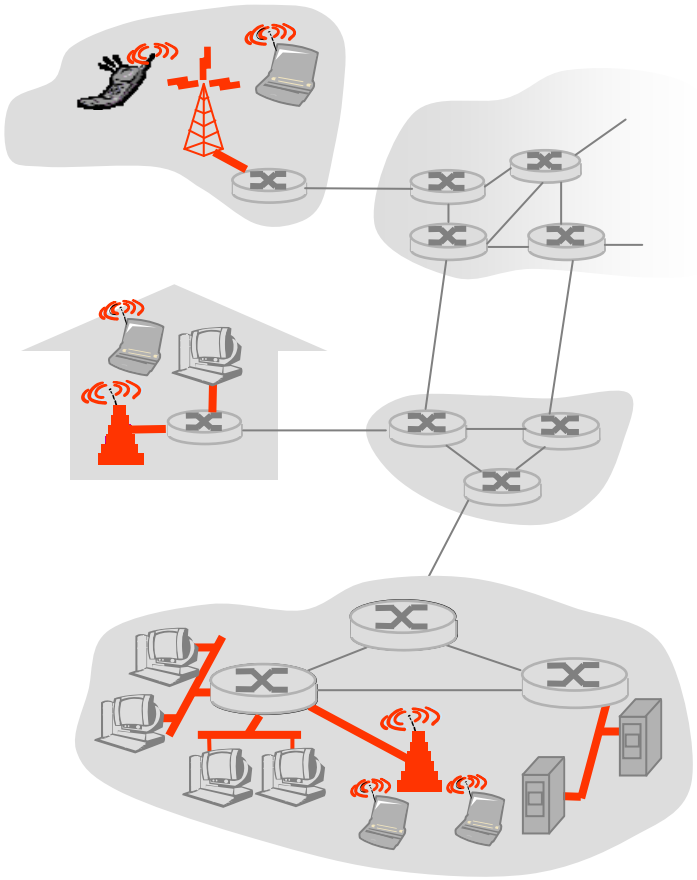
- end systems (hosts):
  - run application programs
  - e.g. Web, email
  - at “edge of network”
- client/server model
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- peer-to-peer model:
  - minimal (or no) use of dedicated servers
  - e.g. VOIP, Skype



# Access networks

*How to connect end systems to edge router:*

- residential access nets
- institutional access networks (school, company)
- mobile access networks

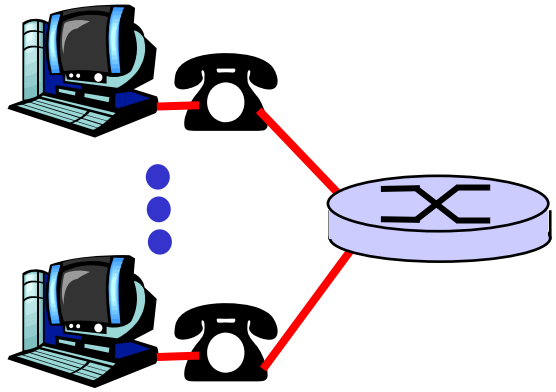


*Important parameters:*

- bandwidth (bits per second) of access network?
- shared or dedicated?

# Residential access: point-to-point access

- Dialup via modem
  - up to 56Kbps direct access to router (often less)
  - Can't surf and phone at same time: can't be "always on"

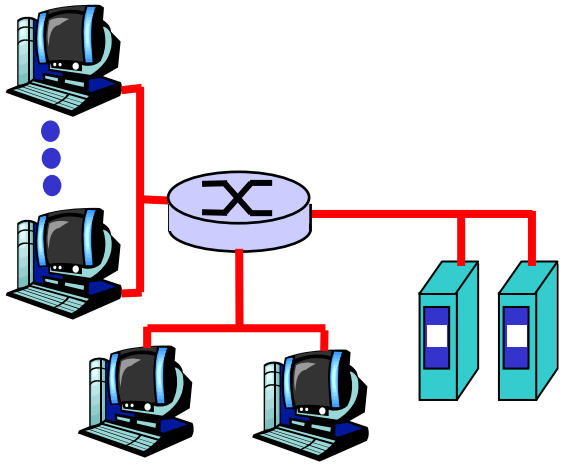


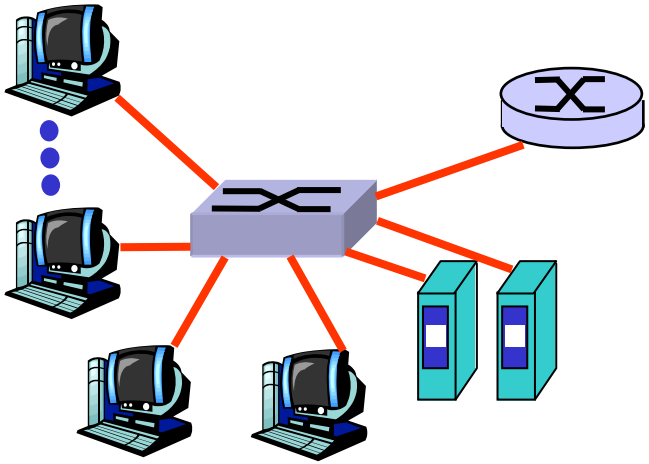
- DSL: digital subscriber line
  - deployment: telephone company or Internet provider
  - DSL divides communication link into 3 separate frequency bands (FDM):
    - up to 1 Mbps upstream (4 kHz bis 50 kHz)
    - up to 8 Mbps downstream (50 kHz bis 1 MHz)
    - normal telephone line (0 to 4 kHz)

# Company access: local area networks

- company/univ local area network (LAN) connects end system to edge router

- Ethernet:
  - 10 Mbs, 100Mbps, 1Gbps, 10Gbps Ethernet
  - modern configuration: end systems connect into *Ethernet switch*

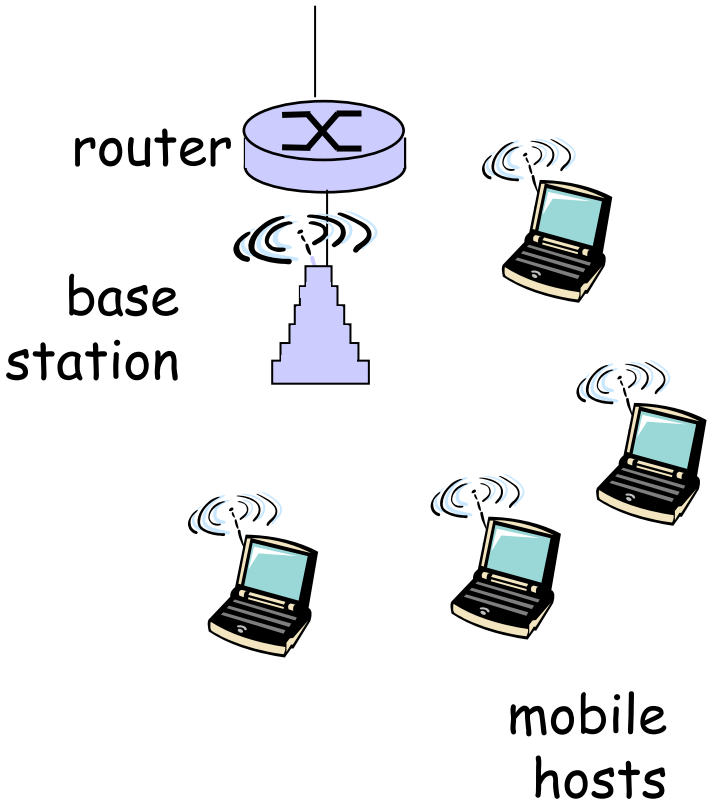




# Wireless access networks

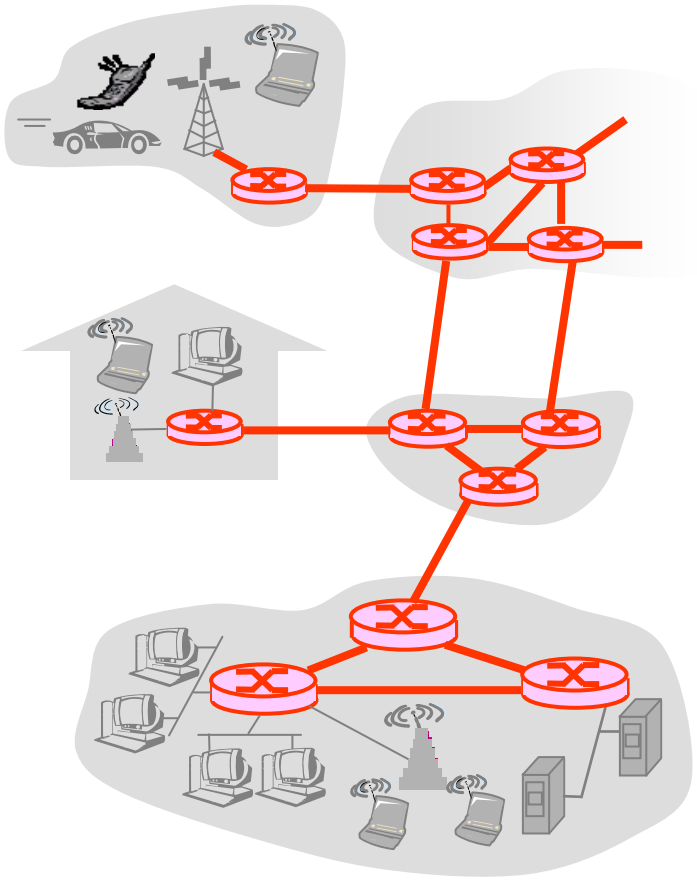
- shared *wireless* access network connects end system to router
  - via base station (access point)

- wireless LANs:
  - 802.11b/g : 11 or 54 Mbps



# Network Core

- mesh of interconnected routers
- *the* fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete “chunks”



# Network Core: Circuit switching

End-to-end resources reserved for "call"

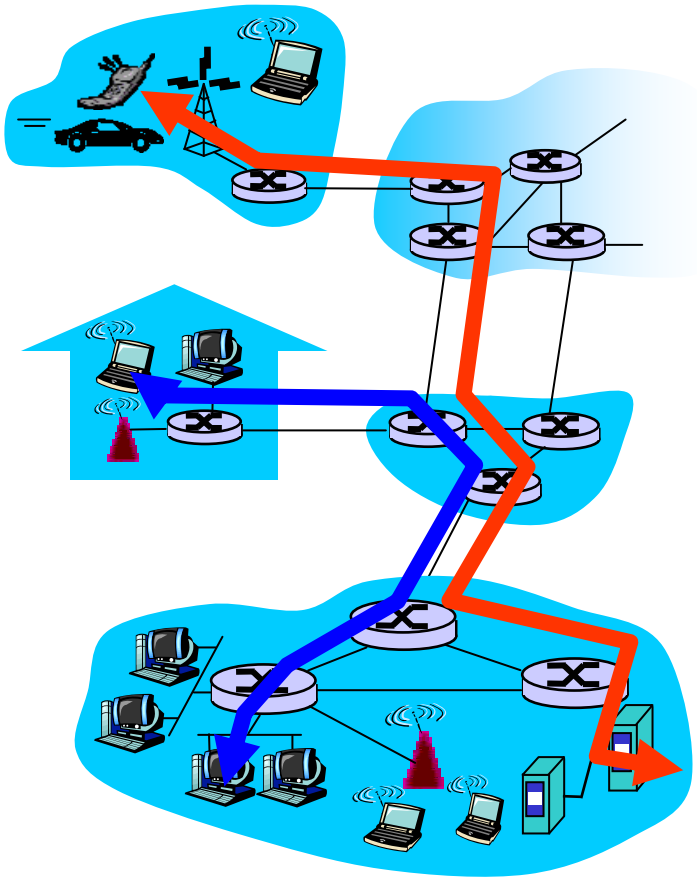
- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required

network resources (e.g., bandwidth) divided into "pieces"

- pieces allocated to calls
- resource piece *idle* if not used by owning call *(no sharing)*

dividing link bandwidth into "pieces"

- frequency division
- time division

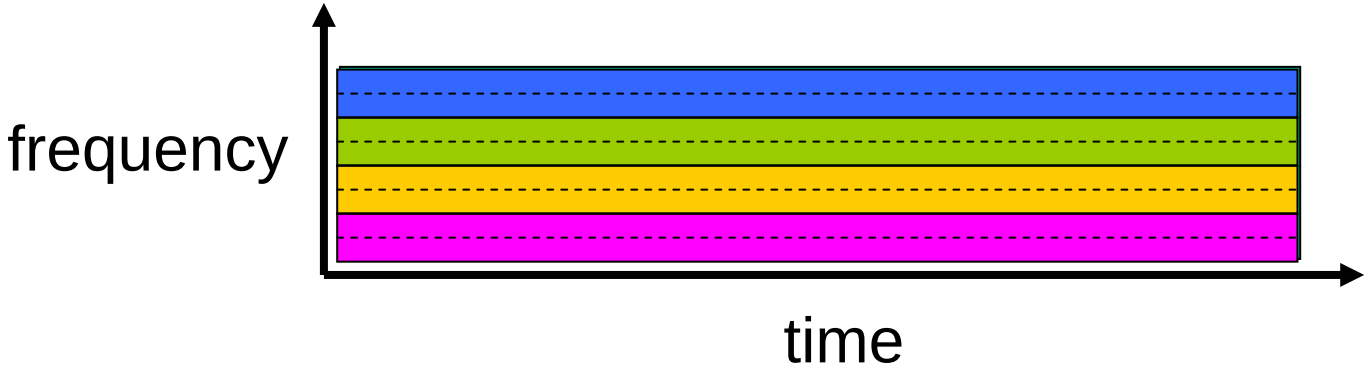


# FDM and TDM

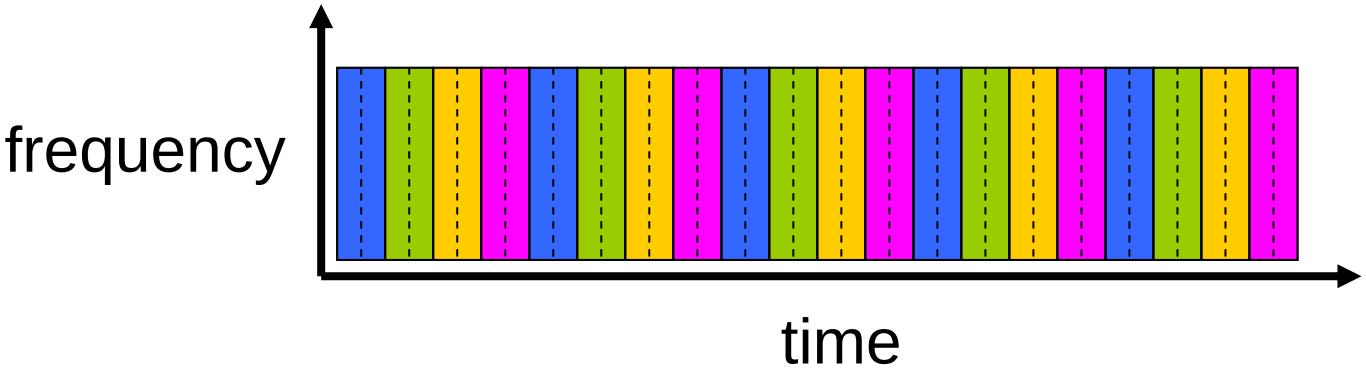
Example:

4 users

FDM




TDM



# Network Core: Packet switching

each end-to-end data stream divided into *packets*

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

resource contention:

- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
  - switch receives complete packet before forwarding

Bandwidth division into "pieces"

Dedicated allocation

Resource reservation