Second part

Example topics to discuss include

- Peer-to-peer systems
- Large-scale distributed systems and services
- Large-scale distributed systems: use cases
- Border Gateway Protocol (BGP) and internet-wide network measurements
- Autoscaling and cloud computing

Notes based on notes by K.W. Ross, J. Kurose, D. Rubenstein, and others

Today's lecture

- General peer-to-peer (P2P) overview
  - Systems and fundamentals
- Peer-to-peer application areas
  - File sharing, communication, computation, overlays, DHTs (including storage and lookup services), and more ...
- Large-scale content delivery
  - BitTorrent, peer-assisted streaming, content delivery networks (CDNs), and file hosting services
- Other large-scale distributed systems and services
  - [if time] E.g., Bootnets, hadoop, planetlab, facebook, google, and youtube

Material today

- These notes, and …
- Reading
  - Cohen, "Incentives build robustness in BitTorrent", Proc. Workshop on Economics of Peer-to-Peer Systems, 2003.
  - Stoica et al., "Chord: A scalable peer-to-peer lookup service for Internet applications", Proc. ACM SIGCOMM, 2001.
  - Castro et al. "SplitStream: High-bandwidth multicast in a cooperative environment", Proc. ACM SOSP, 2003.
  - Mahanti et al., "A Tale of the Tails: Power-laws in Internet Measurements", IEEE Network, 2013.
  - Parvez et al., "Insights on Media Streaming Progress using BitTorrent-like Protocols for On-Demand Streaming", IEEE/ACM Transactions on Networking (ToN), 2012.
  - Dán and Carlsson, "Dynamic Content Allocation for Cloud-assisted Service of Periodic Workloads", Proc. IEEE INFOCOM, 2014.

Defintion of P2P

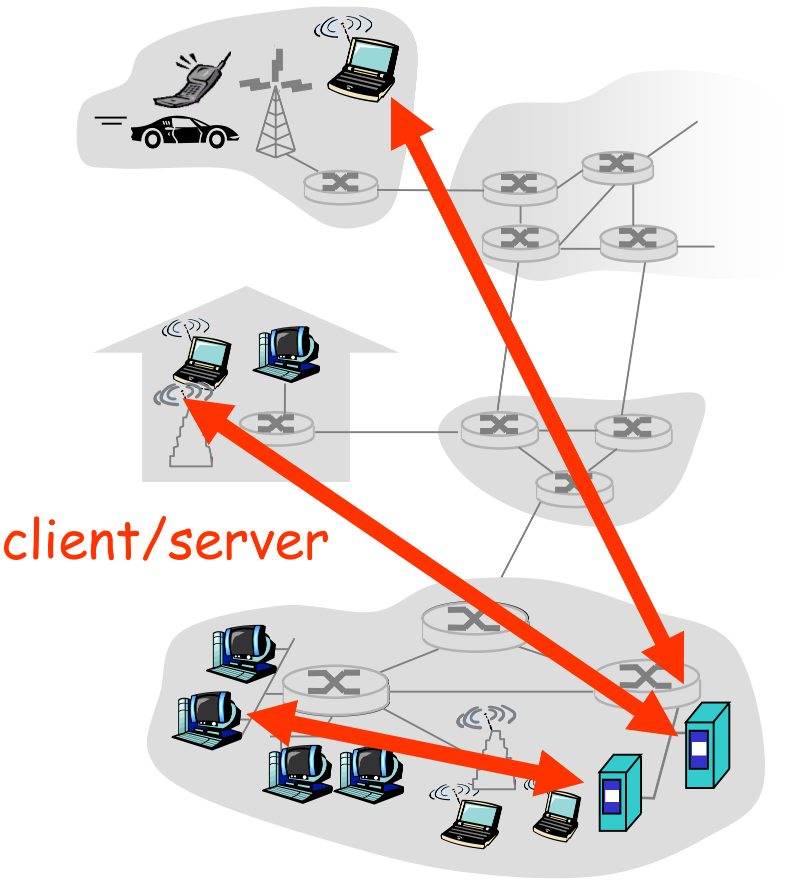
1) Significant autonomy from central servers

2) Exploits resources at the edges of the Internet

- storage and content
- CPU cycles
- human presence

3) Resources at edge have intermittent connectivity, being added & removed

Client-server architecture



server:

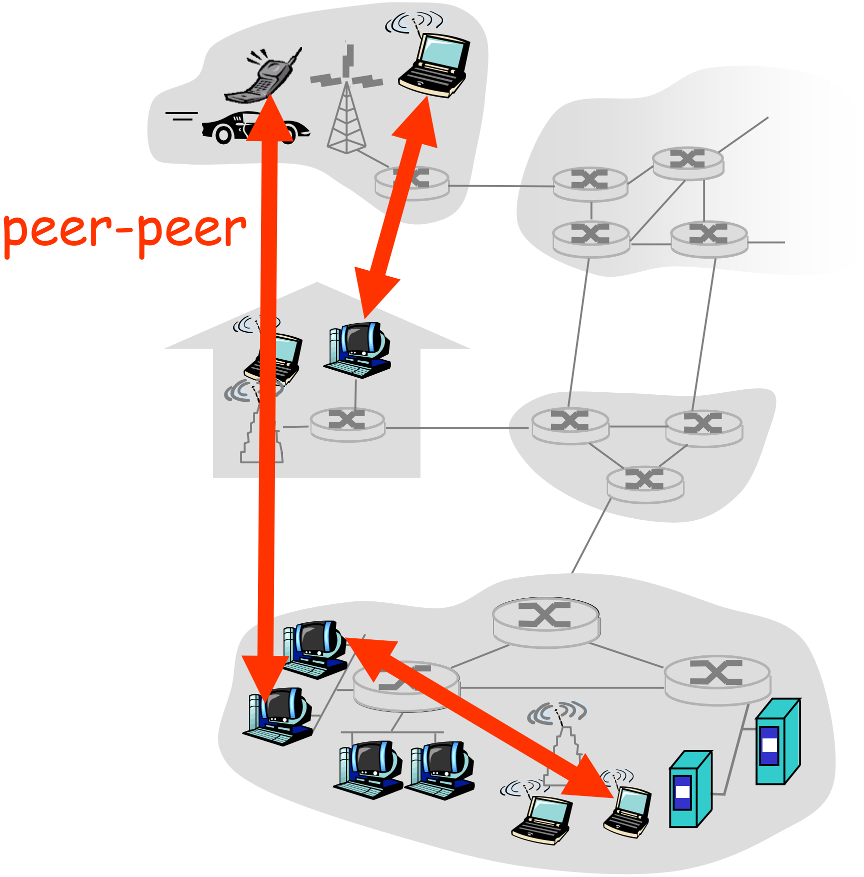
- always-on host
- permanent IP address
- server farms for scaling

clients:

- communicate with server
- may be intermittently connected
- may have dynamic IP addresses
- do not communicate directly with each other

Pure P2P architecture

- *no* always-on server
- arbitrary end systems directly communicate
- peers are intermittently connected and change IP addresses



Hybrid of client-server and P2P

Skype

- voice-over-IP P2P application
- centralized server: finding address of remote party:
- client-client connection: direct (not through server)

Instant messaging

- chatting between two users is P2P
- centralized service: client presence detection/location
  - user registers its IP address with central server when it comes online
  - user contacts central server to find IP addresses of buddies

Some example areas/systems

- P2P file sharing
  - Napster, Gnutella, KaZaA, BitTorrent, etc
- P2P communication
  - Instant messaging, Skype
- P2P computation
  - seti@home

- DHTs & their apps
  - Chord, CAN, Pastry, Tapestry
- P2P apps built over emerging overlays
  - PlanetLab

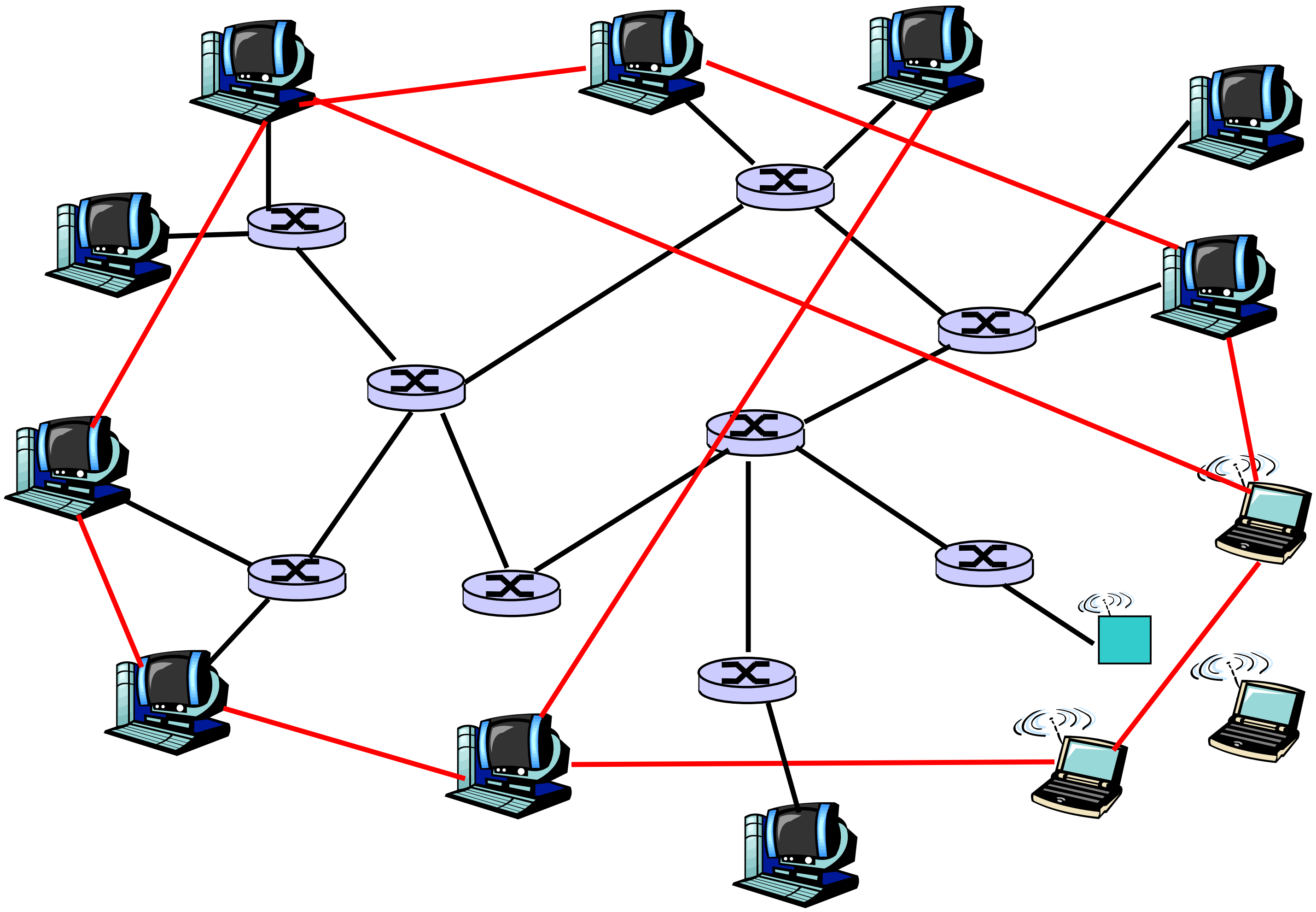
Wireless ad-hoc networking not covered here

Overview of P2P

- overlay networks
- P2P applications

Overlay networks

— overlay edge



Overlay graph

Virtual edge

- TCP connection
- or simply a pointer to an IP address

Overlay maintenance

- Periodically ping to make sure neighbor is still alive
- Or verify liveness while messaging
- If neighbor goes down, may want to establish new edge
- New node needs to bootstrap

More about overlays

Unstructured overlays

- e.g., new node randomly chooses three existing nodes as neighbors

Structured overlays

- e.g., edges arranged in restrictive structure

Proximity

- Not necessarily taken into account

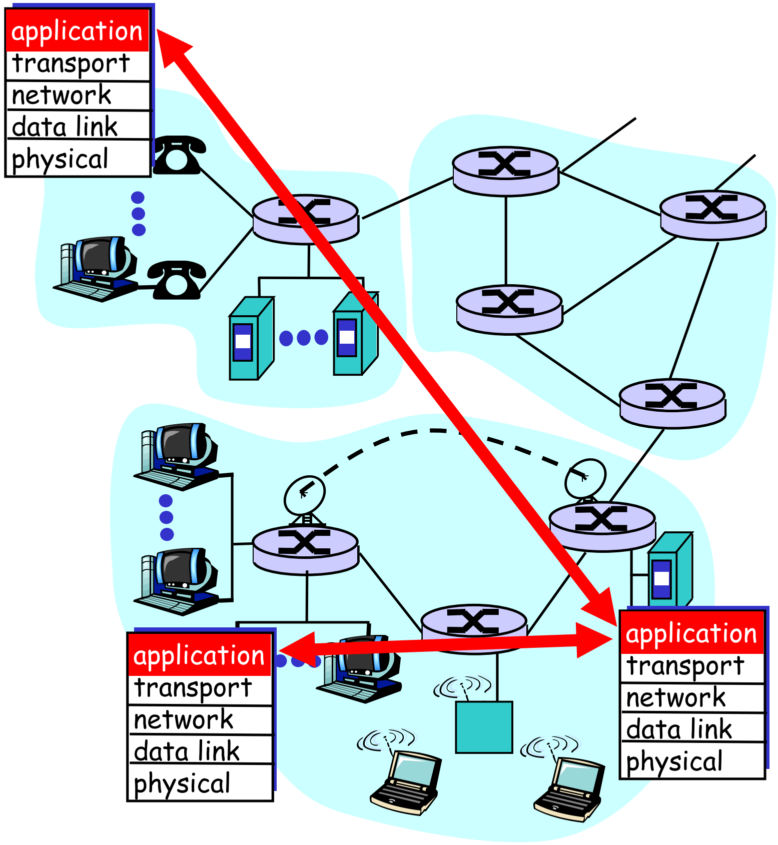
Overlays: all in the application layer

Tremendous design flexibility

- Topology, maintenance
- Message types
- Protocol
- Messaging over TCP or UDP

Underlying physical net is transparent to developer

- But some overlays exploit proximity



Examples of overlays

- DNS
- BGP routers and their peering relationships
- Content distribution networks (CDNs)
- Application-level multicast
  - economical way around barriers to IP multicast

- And P2P apps !

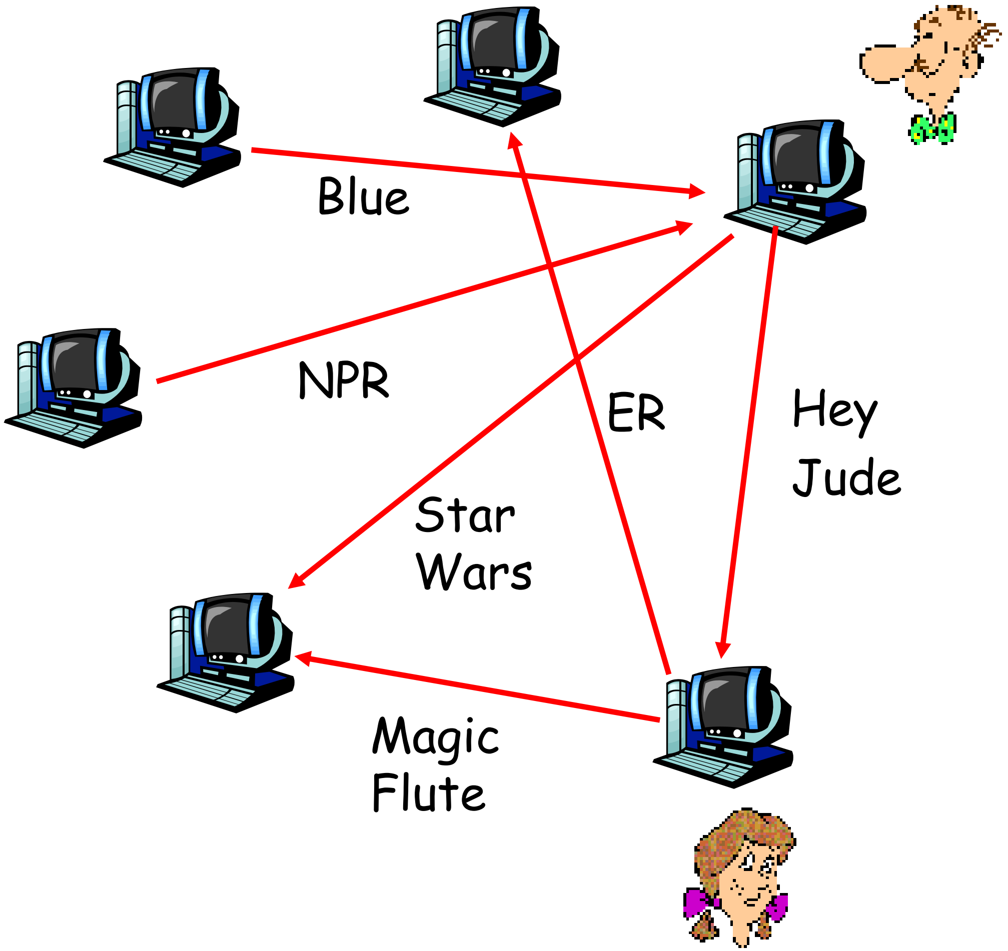
1. Overview of P2P

- overlay networks
- current P2P applications
  - P2P file sharing & copyright issues
  - Instant messaging
  - P2P distributed computing

P2P file sharing

- Alice runs P2P client application on her notebook computer
- Intermittently connects to Internet; gets new IP address for each connection
- Registers her content in P2P system
- Asks for “Hey Jude”
- Application displays other peers that have copy of Hey Jude.
- Alice chooses one of the peers, Bob.
- File is copied from Bob’s PC to Alice’s notebook: P2P
- While Alice downloads, other users uploading from Alice.

Millions of content servers



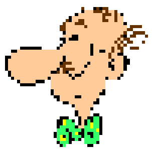
Killer deployments

- Napster
  - disruptive; proof of concept
- Gnutella
  - open source
- KaZaA/FastTrack
- BitTorrent

P2P file sharing software

- Allows Alice to open up a directory in her file system
  - Anyone can retrieve a file from directory
  - Like a Web server
- Allows Alice to copy files from other users' open directories:
  - Like a Web client

- Allows users to search the peers for content based on keyword matches:
  - Like Google



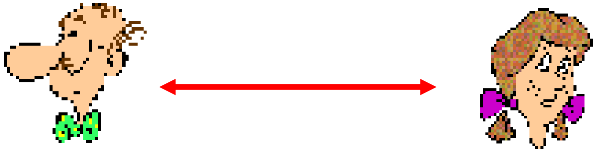
Seems harmless to me …

But often copyright issues !!

Instant Messaging

- Alice runs IM client on her PC
- Intermittently connects to Internet; gets new IP address for each connection
- Registers herself with "system"
- Learns from "system" that Bob in her buddy list is active

- Alice initiates direct TCP connection with Bob: P2P
- Alice and Bob chat.



- Can also be voice, video and text.

P2P Distributed Computing

seti@home

- Search for ET intelligence
- Central site collects radio telescope data
- Data is divided into work chunks of 300 Kbytes
- User obtains client, which runs in backgrd

- Peer sets up TCP connection to central computer, downloads chunk
- Peer does FFT on chunk, uploads results, gets new chunk

Not peer to peer, but exploits resources at network edge