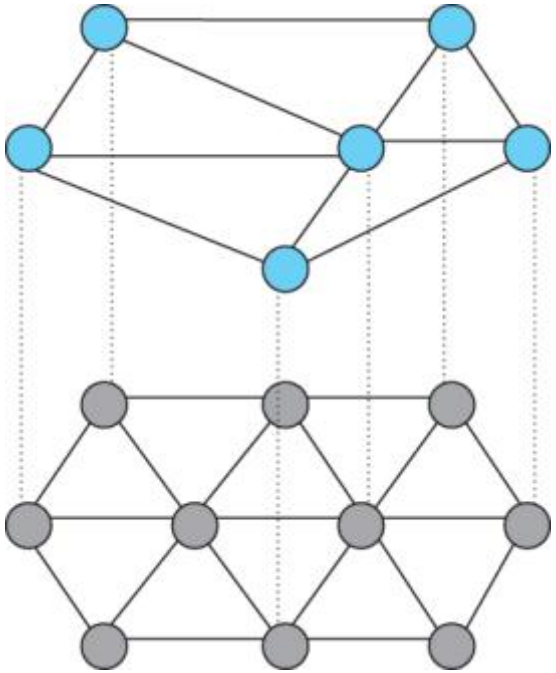
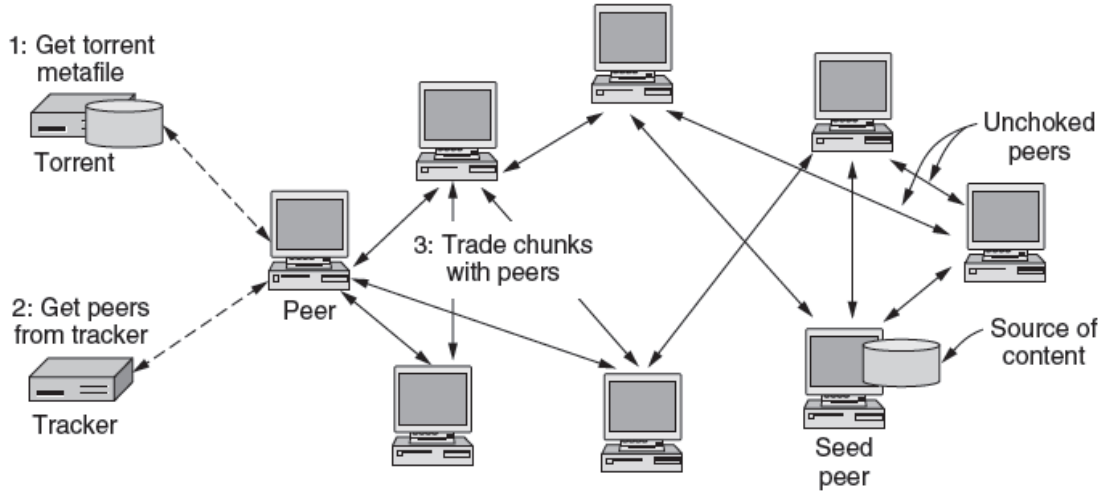
# Overlay networks and P2P





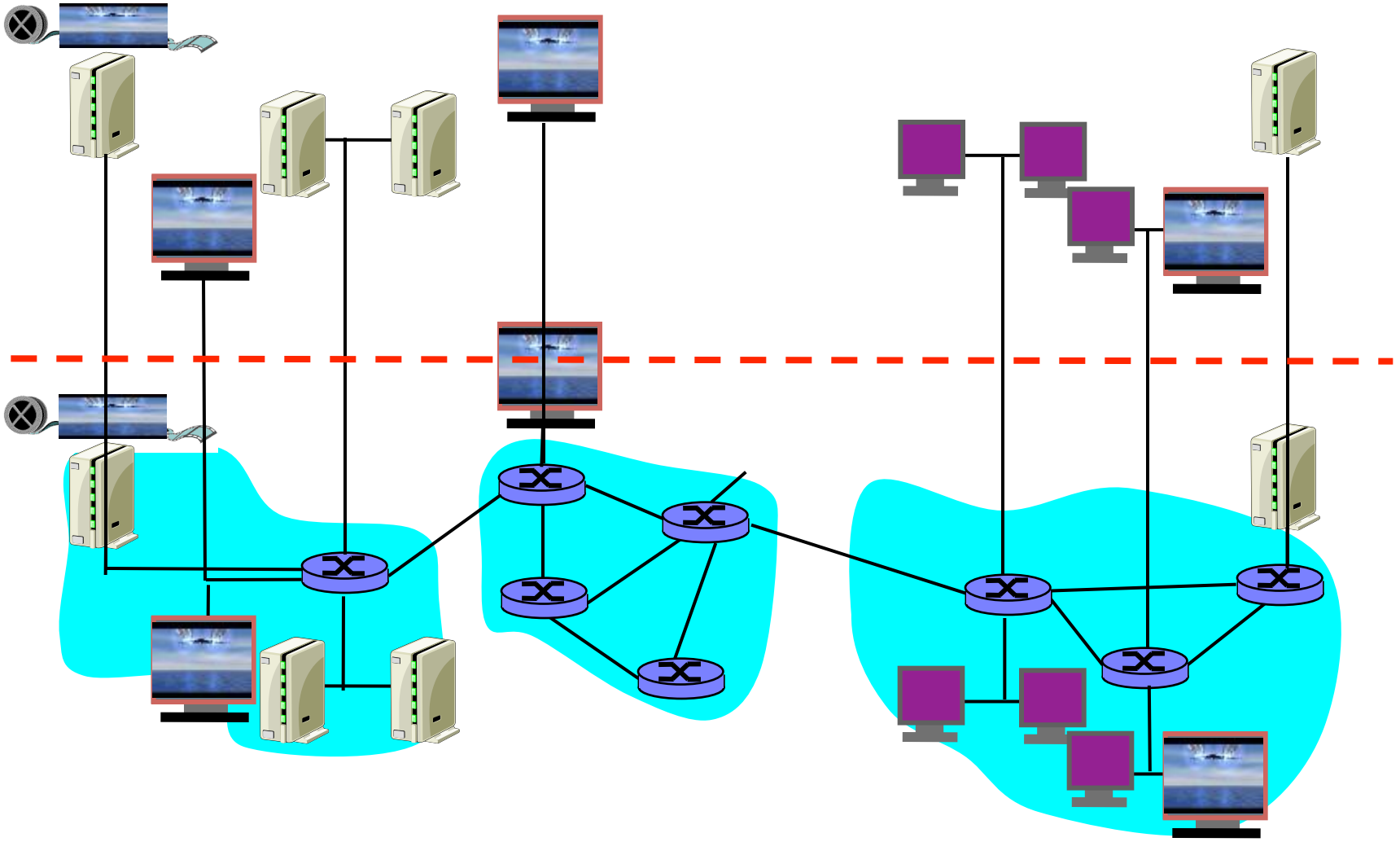
# Overview

- Overlay networks
  - Logical network running on top of physical network
  - Support alternate routing strategies
  - Experimental protocols
- Peer-to-peer (P2P) networks
  - Directory-based
  - Unstructured
  - Structured

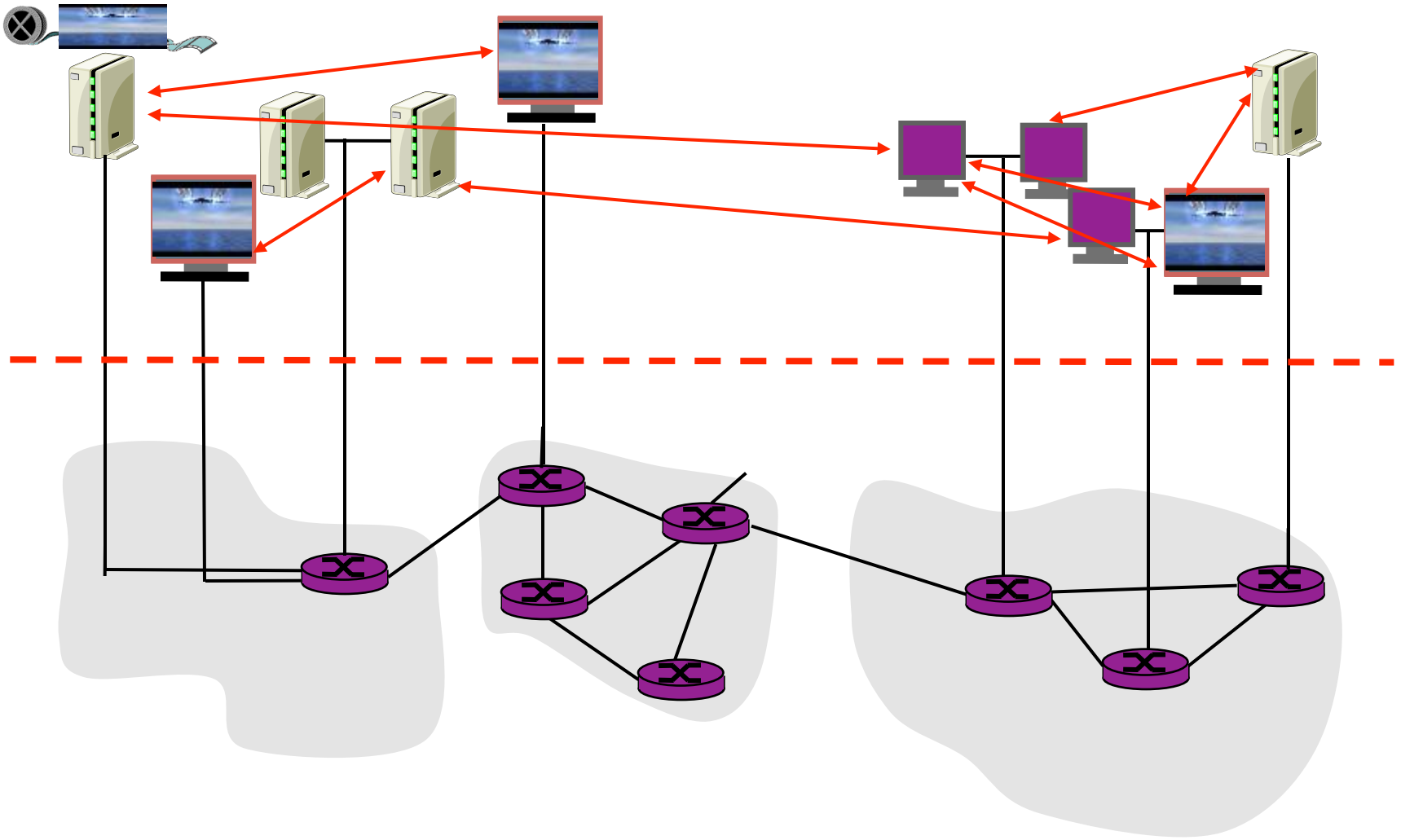
# Internet ossification

- The Internet
  - Originally so researchers to experiment with packet switching
  - Now commercial interests dominate
  - Massive size, can't change software in all the routers
- How to investigate and deploy new features?
  - Migration to IPv6
  - Multicast routing
  - Virtual private networks
  - Optimizing routing between small set of hosts
  - New services such as peer-to-peer

# Overlay network

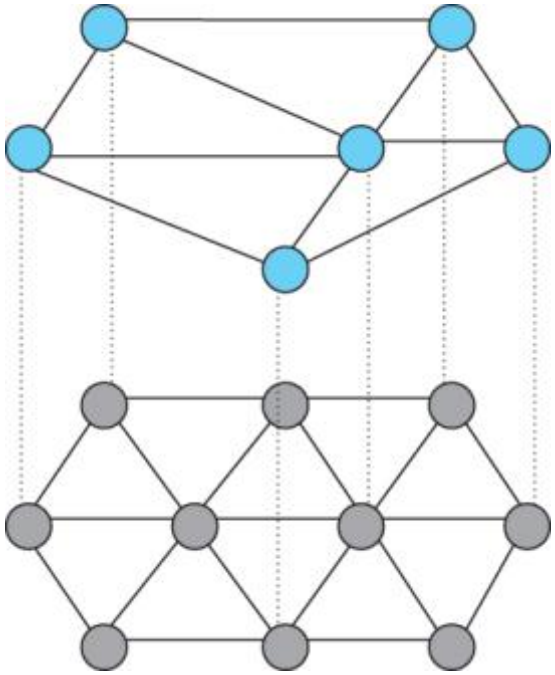


# Application level focus



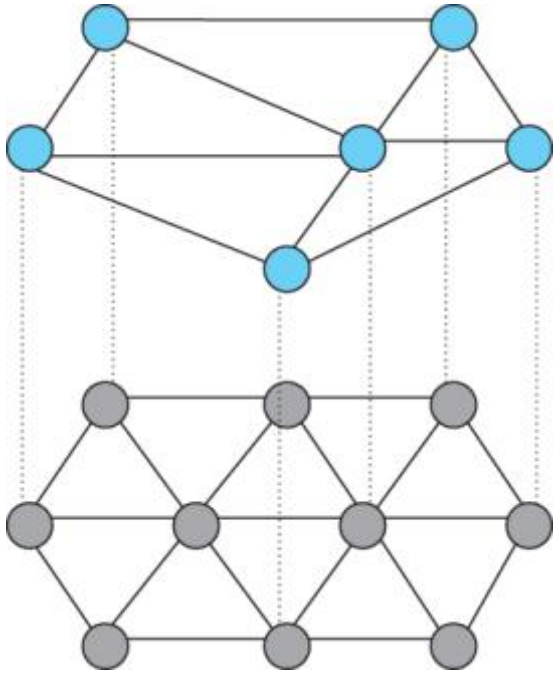
# Overlay networks

- Many logical networks may coexist
  - Over same underlying network
  - Each providing its own particular service
- Nodes in overlay network
  - Often end hosts
  - Act as a traffic forwarding agent
  - Provide a service such as file sharing
- Which nodes are in the network?
  - Party providing the service (e.g. Akamai)
  - Collection of end users (e.g. P2P sharing)



# Overlay networks

- Logical network built on top of a physical network
  - Overlay link is a connection between two nodes
    - Link makes sense from standpoint of the logical network
    - Actual packets on overlay link may transit a series of physical links
  - Internet started as overlay network of the old telephone network



Overlay network

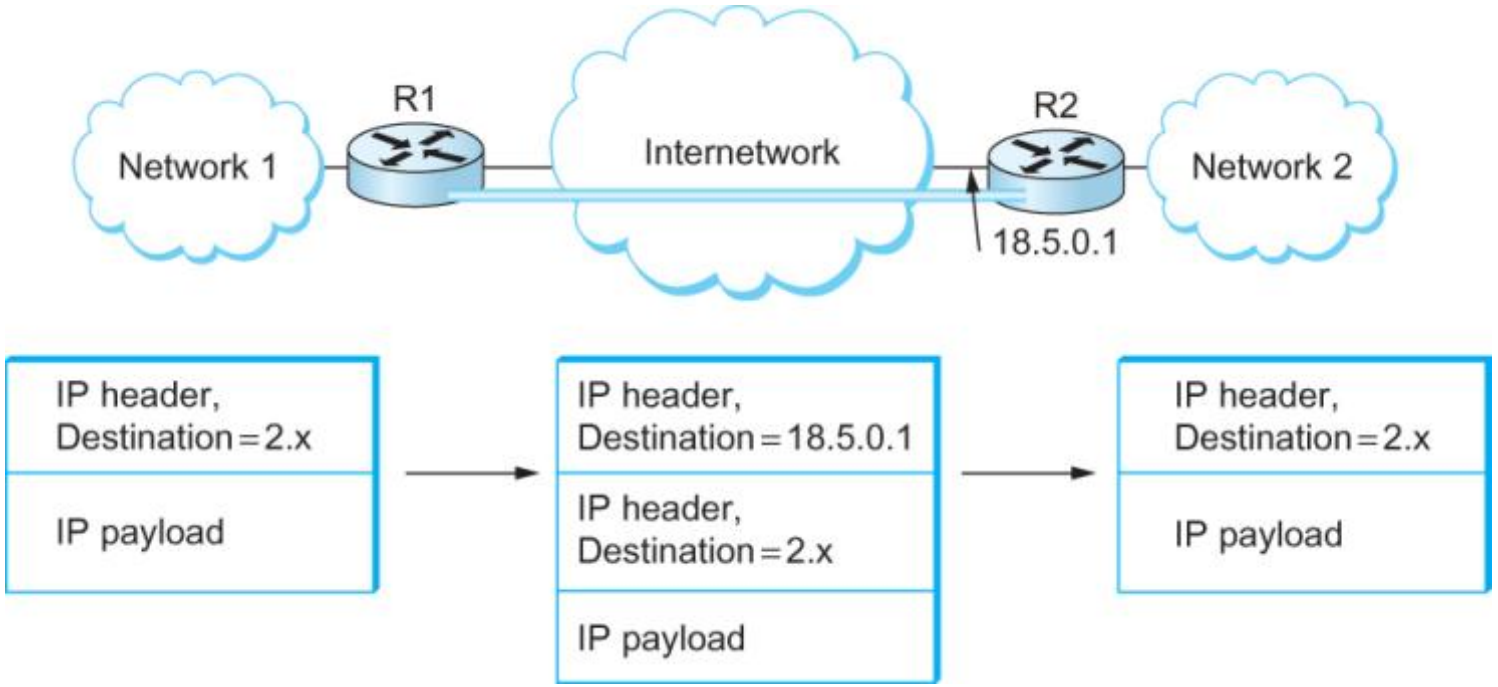
Physical network

# Using overlays for routing

- Routing overlay
  - Purely to support alternative routing strategy
  - No application-level processing at overlay nodes
- Examples:
  - IPv6
  - Virtual Private Network (VPN)
  - Mobile IP
  - Multicast
- Relies on creating "tunnels" through the network

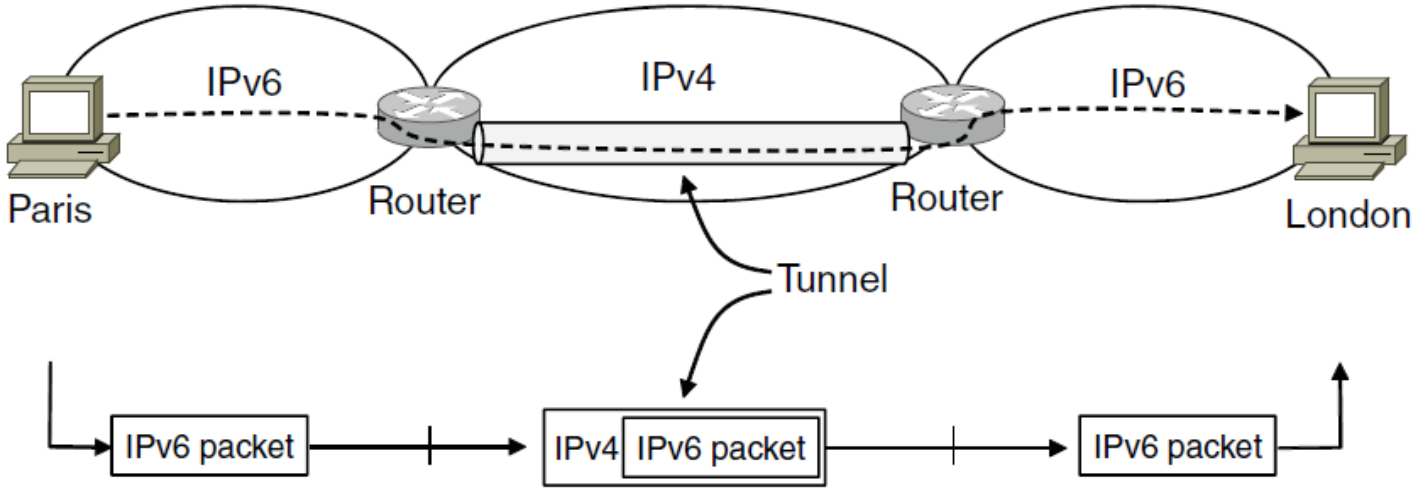
# IP tunneling

- IP tunnel
  - A virtual point-to-point link
  - A packet gets encapsulated inside another packet



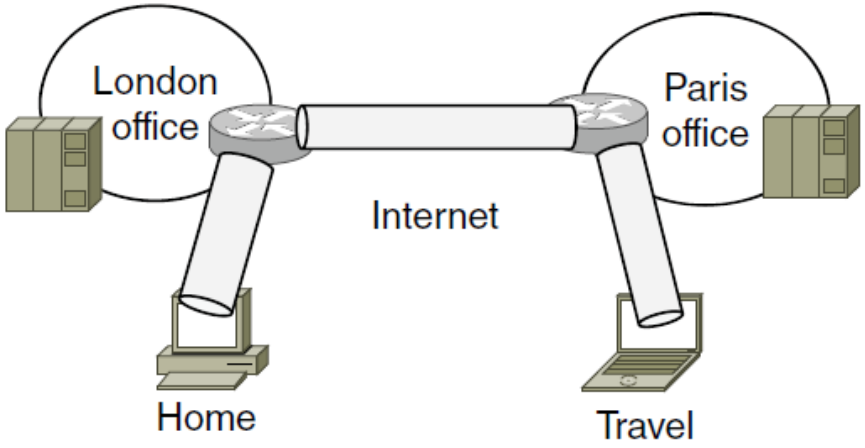
# Deploying IPv6

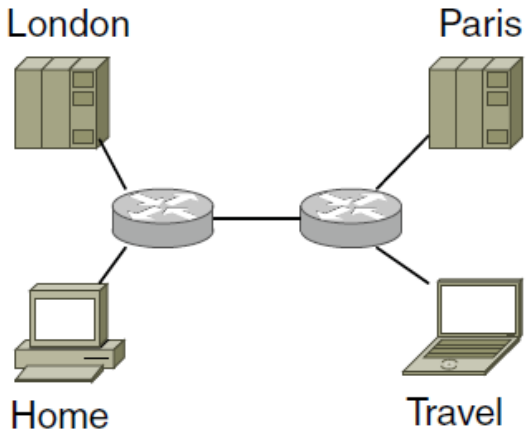
- 6-Bone
  - Overlay network to support IPv6
  - Tunnels IPv6 packets over routers not supporting IPv6



# Creating private networks

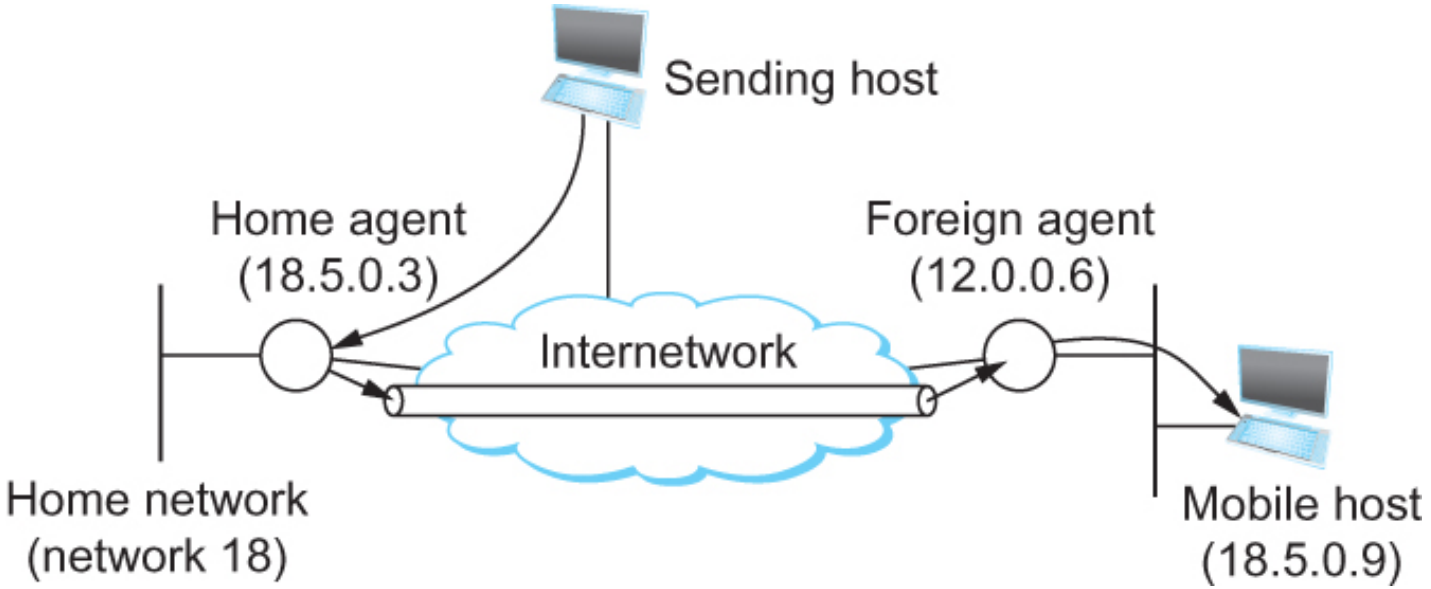
- Virtual Private Networks (VPNs)
  - A secure private network running over the public Internet
  - Equip each office with a firewall that creates tunnels between all pairs of offices
  - Roaming users can connect to firewall using VPN software



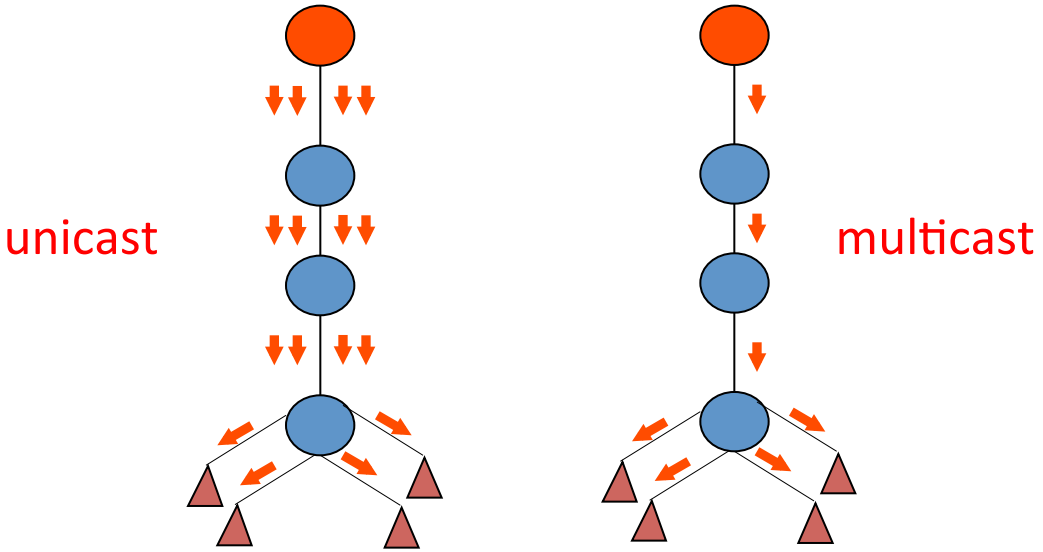
# Routing to mobile hosts

- Mobile IP
  - Route packets to a roaming mobile host
  - Home agent tunnels traffic to foreign agent



# Delivering to multiple hosts

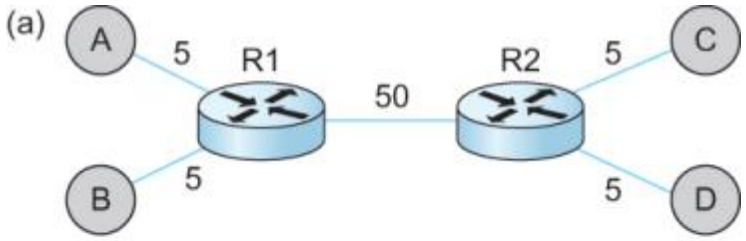
- Multicast
  - Deliver same content to many hosts avoiding redundancy



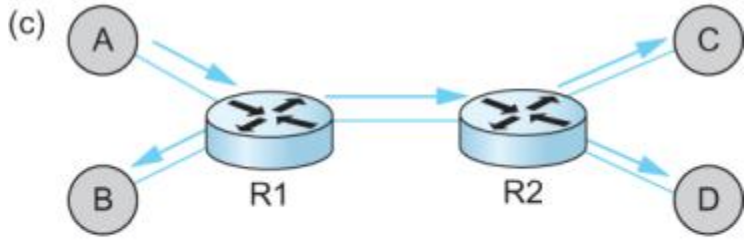

- IP multicast
  - Special addressing, forwarding, and routing
  - Not widely deployed, MBone tunneled between nodes

# End-system multicast

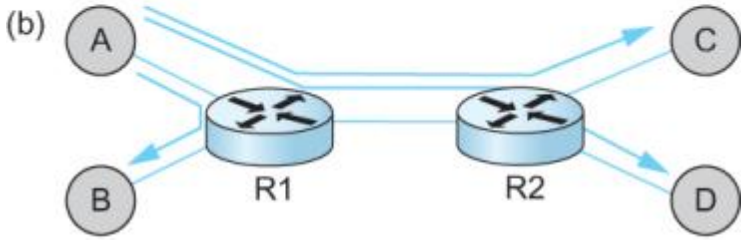
- End-system multicast
  - IP multicast not widely deployed, have hosts do it instead
  - End hosts form their own multicast tree
  - Hosts help forward data onto to others in multicast group



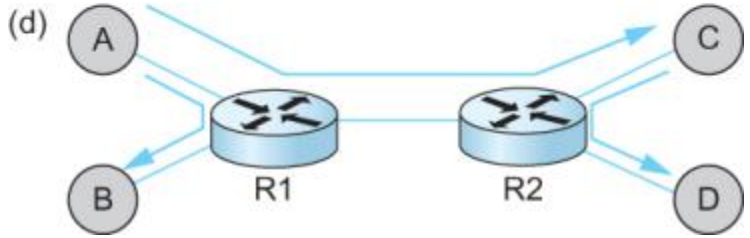

*Network with link costs.*




*Multicast with router support.*



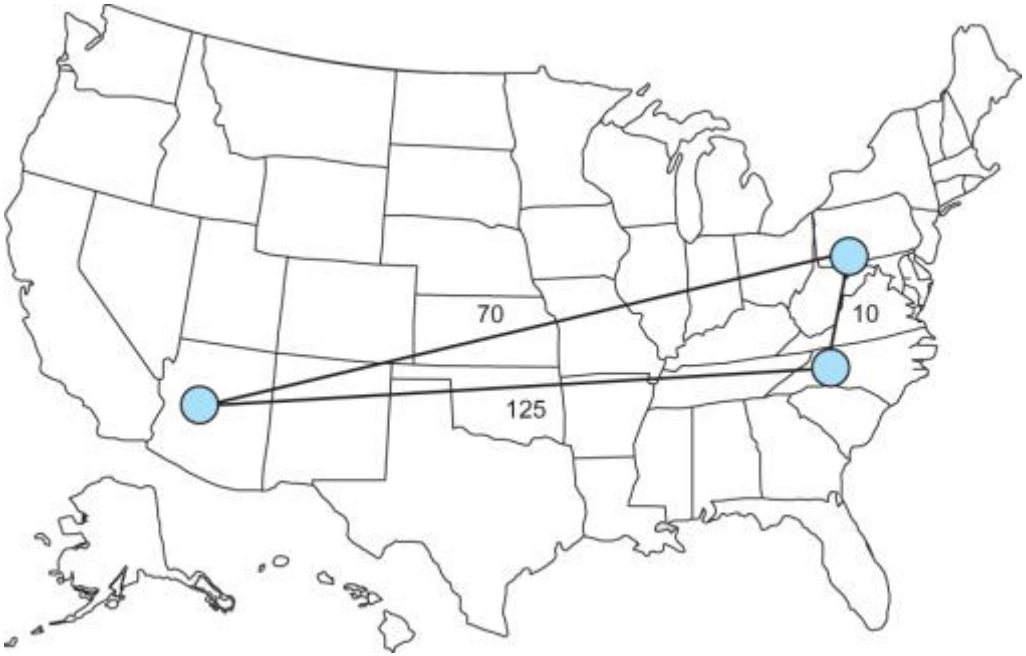

*Naïve delivery using unicast.*




*End-system multicast,*
*C forwards traffic on to D.*

# Improving performance

- Resilient Overlay Networks (RON)
  - Build overlay networks between small set of nodes (~10s)
  - Monitor latency, bandwidth and drop probability between every pair of nodes
  - Use data to select optimal route between nodes in set




*Sometimes it may be better to take several hops compared to going direct.*

# RON performance

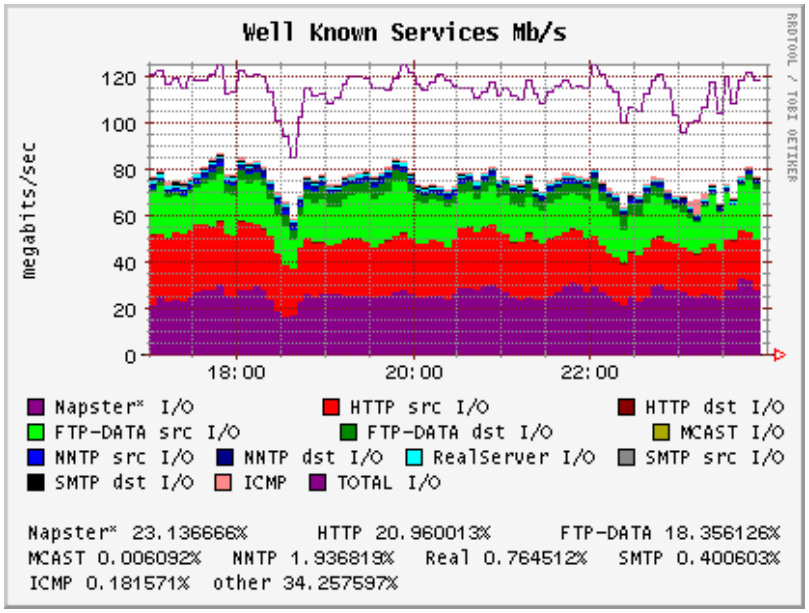
- IP routing does not adapt to congestion
  - RON can reroute when path is congested
- IP routing is sometimes slow to converge
  - RON can quickly direct traffic around problem
- IP routing depends on AS policies
  - RON can pick best performing path ignoring policies
- But RON has some drawbacks:
  - Packets may go through more hops, loading hosts, increasing costs
  - Probing causes network overhead

# Overlay services: P2P

- Peer-to-peer (P2P) networks
  - Community of users pooling resources (storage space, bandwidth, CPU) to provide a service
  - e.g. Sharing MP3 files, Skype
  - Nodes are hosts willing to share, links are tunnels used to transport objects of interest
- Types:
  - Centralized P2P – central server for indexing
  - Pure P2P – all peers are equals
  - Hybrid P2P – some peers are supernodes

# P2P: Napster

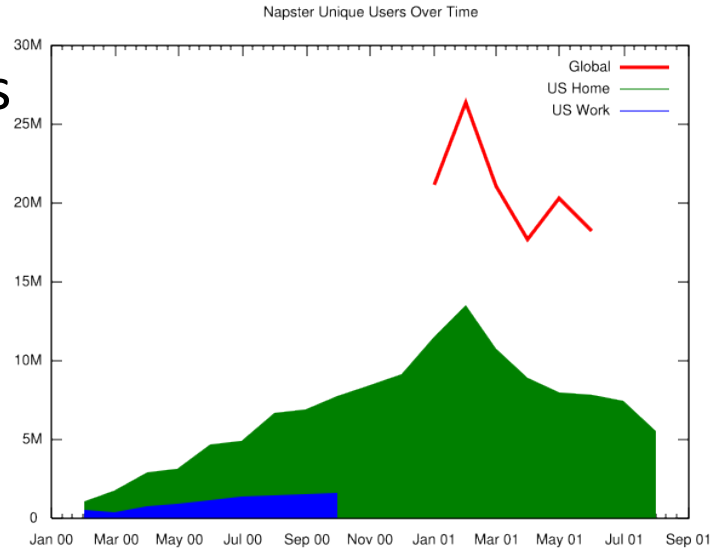

- Napster: the rise
  - Created by Shawn Fanning
    - Christmas break his freshmen year at college
  - Allows people to search and share MP3 files
  - January 1999, Napster version 1.0
  - May 1999
    - Company founded
    - Shawn drops out of school
  - September 1999, first lawsuits
    - No such thing as bad publicity
  - By 2000, 80 million users




*UW-Madison, March 9th, 2000*

# P2P: Napster

- Napster: the fall
  - December 1999
    - RIAA files lawsuit
  - Metallica's "I Disappear" circulates
    - Before official release, starts getting radio play
    - 2000 band files a lawsuit
  - July 2001, shutdown due to lawsuits
  - 2002, relaunched as a paid service
    - Major record labels not keen to license
    - Files bankruptcy
  - Gave rise to many P2P alternatives
  - Forced industry out of stone age
    - iTunes




*Napster users peak, Feb 2001.*

# Napster technology

- User installs software
  - Registers name, password, local directory with music
- Client contacts central Napster server
  - Connects via TCP
  - Provides list of music in user's directory
  - Napster updates its database
- Client searches for music
  - Napster identifies currently online client with file
  - Provides IP addresses so client can download directly

# Napster technology

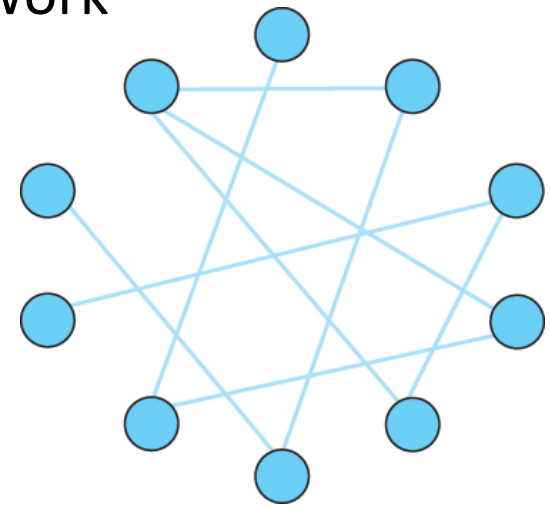
- Central server continually updated
  - Easy to track music currently available and from what peer
  - Good source to prove copyright infringement
  - Single point of failure, performance bottleneck
- Peer-to-peer transfer
  - Key idea of P2P: heavy lifting done between peers
  - No need for Napster to provision lots of capacity
    - Just enough to support indexing/search needs of clients
- Proprietary protocol

# P2P: Gnutella

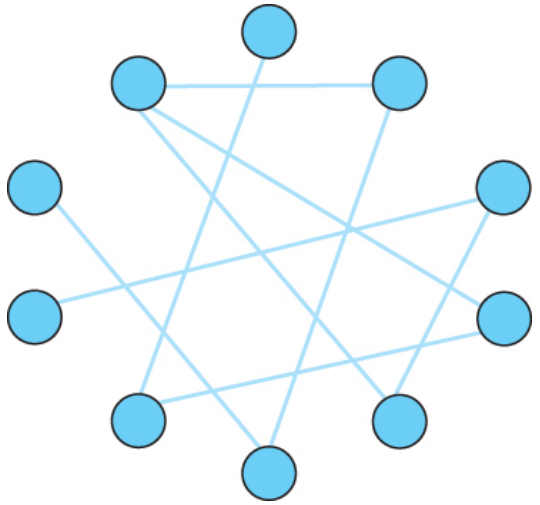
- Gnutella
  - Early 2000, created by Justin Frankel & Tom Pepper at Nullsoft (recently acquired by AOL)
  - Prematurely announced on Slashdot
    - Thousands download client
    - Next day AOL shuts it down over legal concerns
    - Too late, protocol reverse engineered and released as open source
  - Protocol became basis for many clients
    - e.g. Limewire, Bearshare


# P2P: Gnutella

- No centralized object registry
  - Hosts arrange themselves in overlay network
  - Each host runs Gnutella software
- Joining the network
  - Contact a few nodes
    - Pre-existing list shipped with software
    - Web cache of known nodes
  - Send ping to neighbors who ask their neighbors
  - New node gets pong messages back from other nodes
  - Repeat until new node reaches quota of desired neighbors



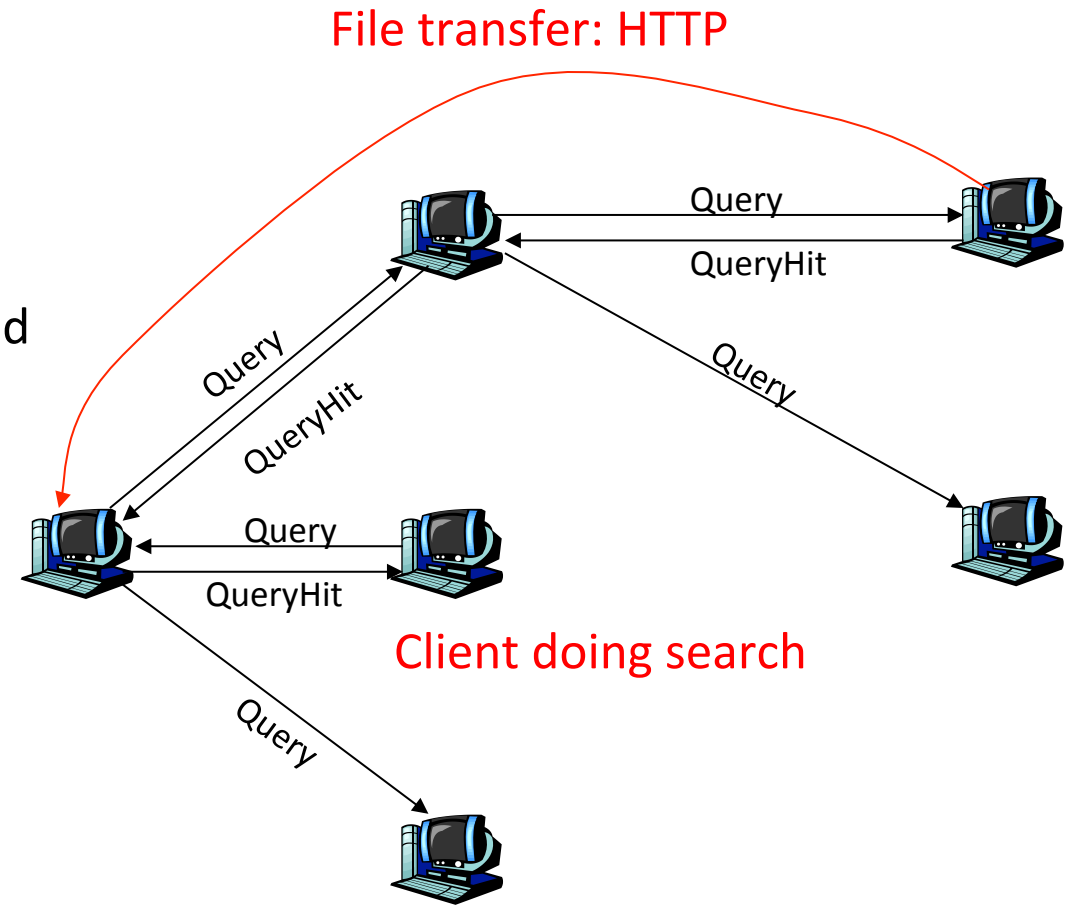
# P2P: Gnutella



- Query flooding
  - Client wants some file
  - Sends query to connected neighbors
  - Neighbors send to their neighbors
  - …
  - Node with file sends response
    - Classic gnutella, response sent along route of the query
    - Client connects to host with file and downloads
    - Or if host is behind a firewall, clients requests host push the file

# Gnutella query flooding

- Query over existing TCP connections
- Flooding scalability:
  - Queries have TTL
    - Remove if been around too long
  - Unique query ID
    - Don't forward recent queries



# Gnutella

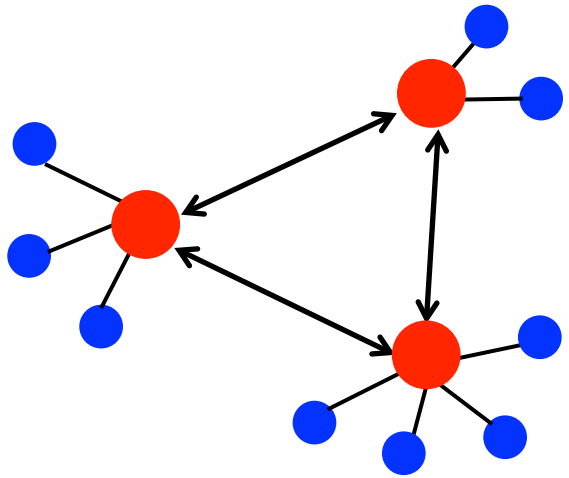
- Advantages
  - Fully decentralized, all nodes are equals
  - Hard to shutdown since no central server
  - Search cost distributed among nodes
- Disadvantages
  - Each search may cause a lot of traffic
  - Search time may be long
  - Paths are unreliable with hosts going up and down

# P2P: FastTrack


- FastTrack
  - 2001, created by Dutch company (Kazaa BV)
  - Same team that later built Skype
  - Hybrid P2P, uses super-nodes to improve scalability
  - e.g. KaZaA, Morpheus
- Smarter query flooding
  - Join by contacting super-node
  - Client sends list of files to its super-node
  - Search: send query to super-node who floods to other super-nodes
  - Fetch file directly from peer(s)

# P2P: FastTrack

- Supernodes
  - Indexes users' shares and performs searches on them
  - Normal nodes can get promoted to super-node
    - Selected based on uptime history
  - Consolidate queries since many nodes only have a few files
    - May be faster for super-node to handle than delegating to normal-node



# P2P: BitTorrent

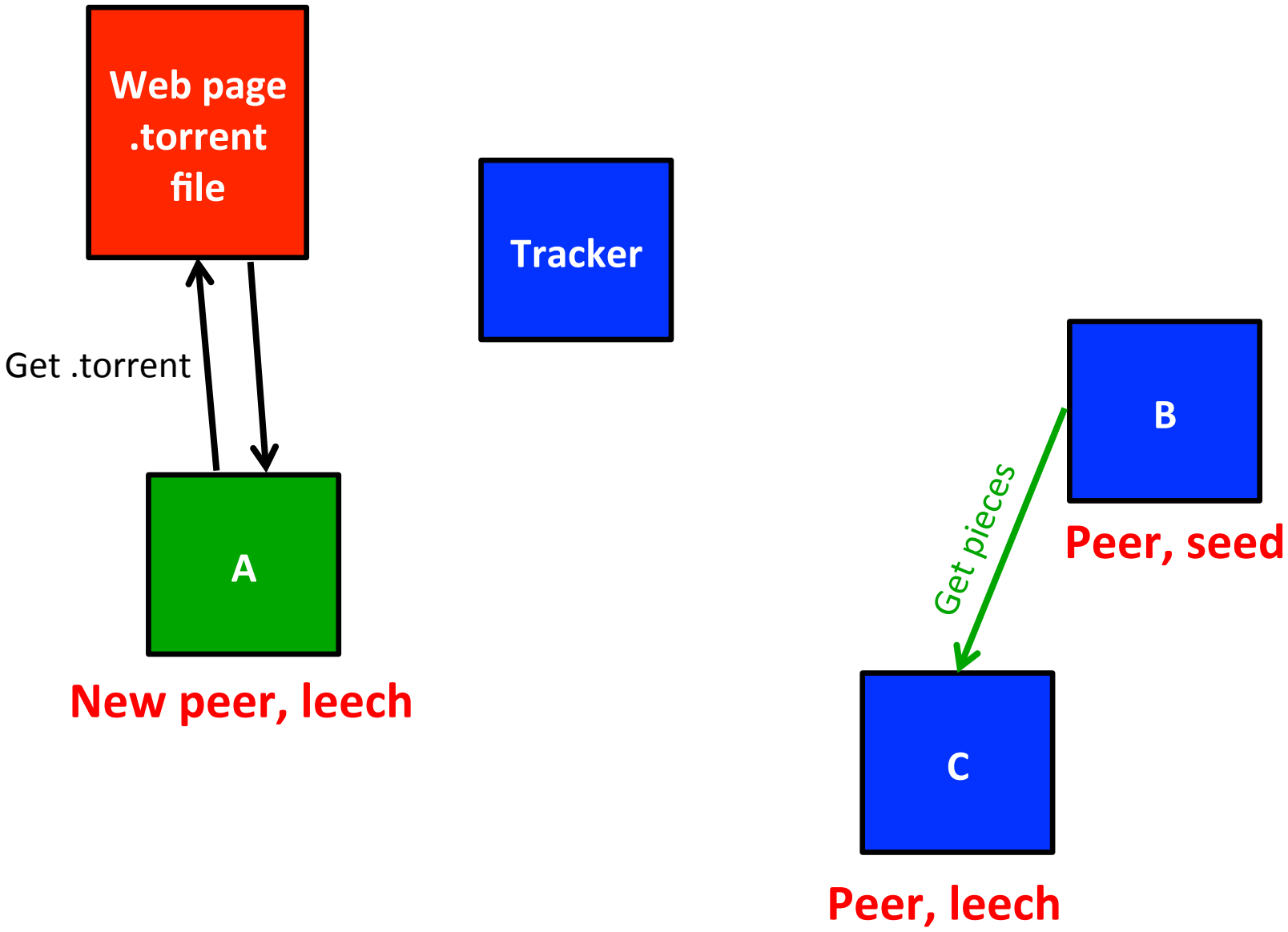
- BitTorrent protocol
  - 2001, Bram Cohen releases first protocol implementation
  - Now supported by many different clients
  - 2011, ~100 million users
- Motivations:
  - Serve up popular content fast
    - Popularity exhibits temporal locality (flash crowd)
  - Efficient fetching, not searching
    - Distribute same file to many peers
    - Single publisher, many downloaders
  - Measures to prevent free-loading

# BitTorrent process

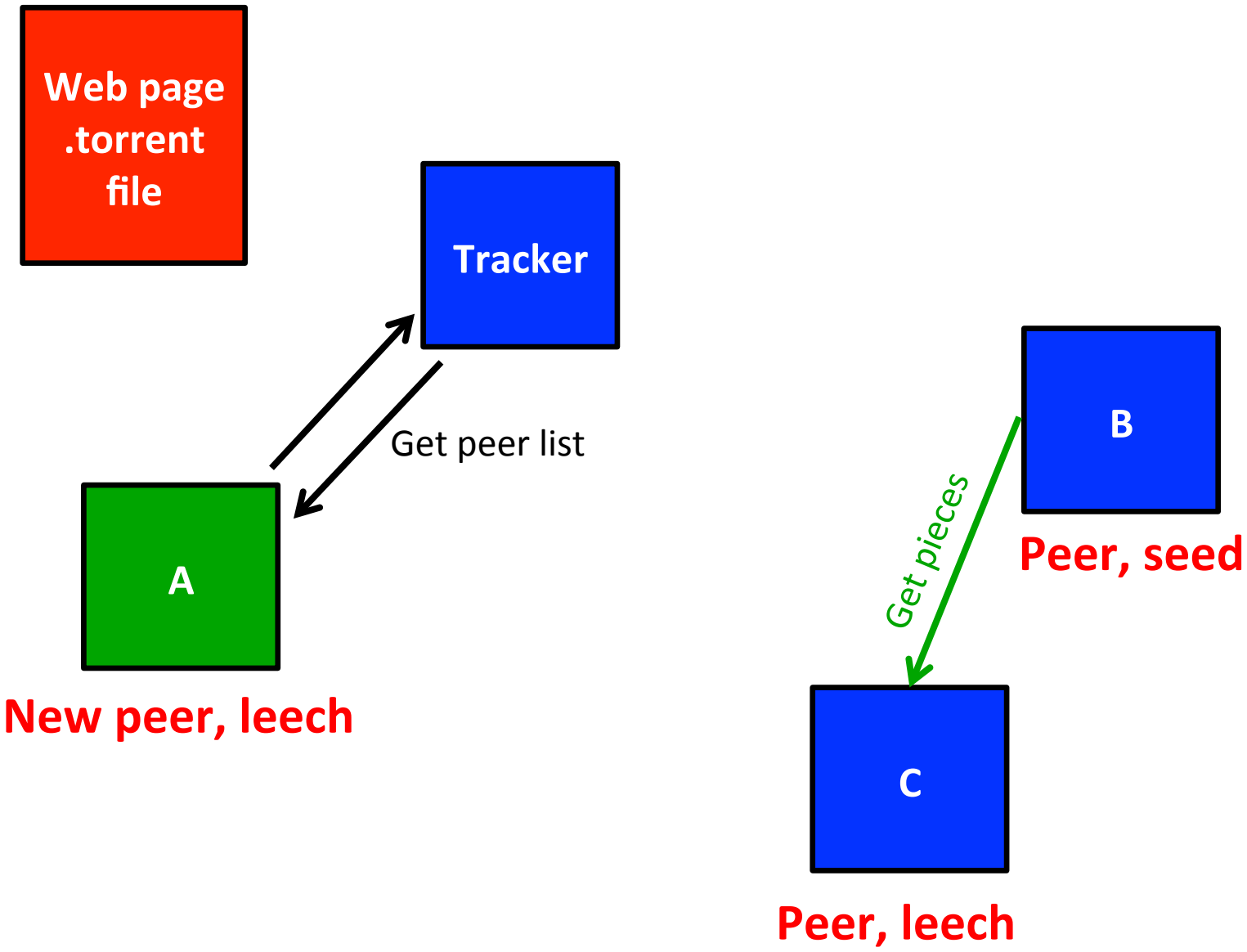
- File divided into many pieces
  - Peers exchange the pieces by uploading and downloading to each other
  - Seed, peer with entire file

- Process:
  - Users find a torrent of interest, open in BitTorrent client
  - Client contacts the tracker listed in torrent file
  - Gets list of peers currently transferring the file
    - Swarm – the peers currently with some/all of the file

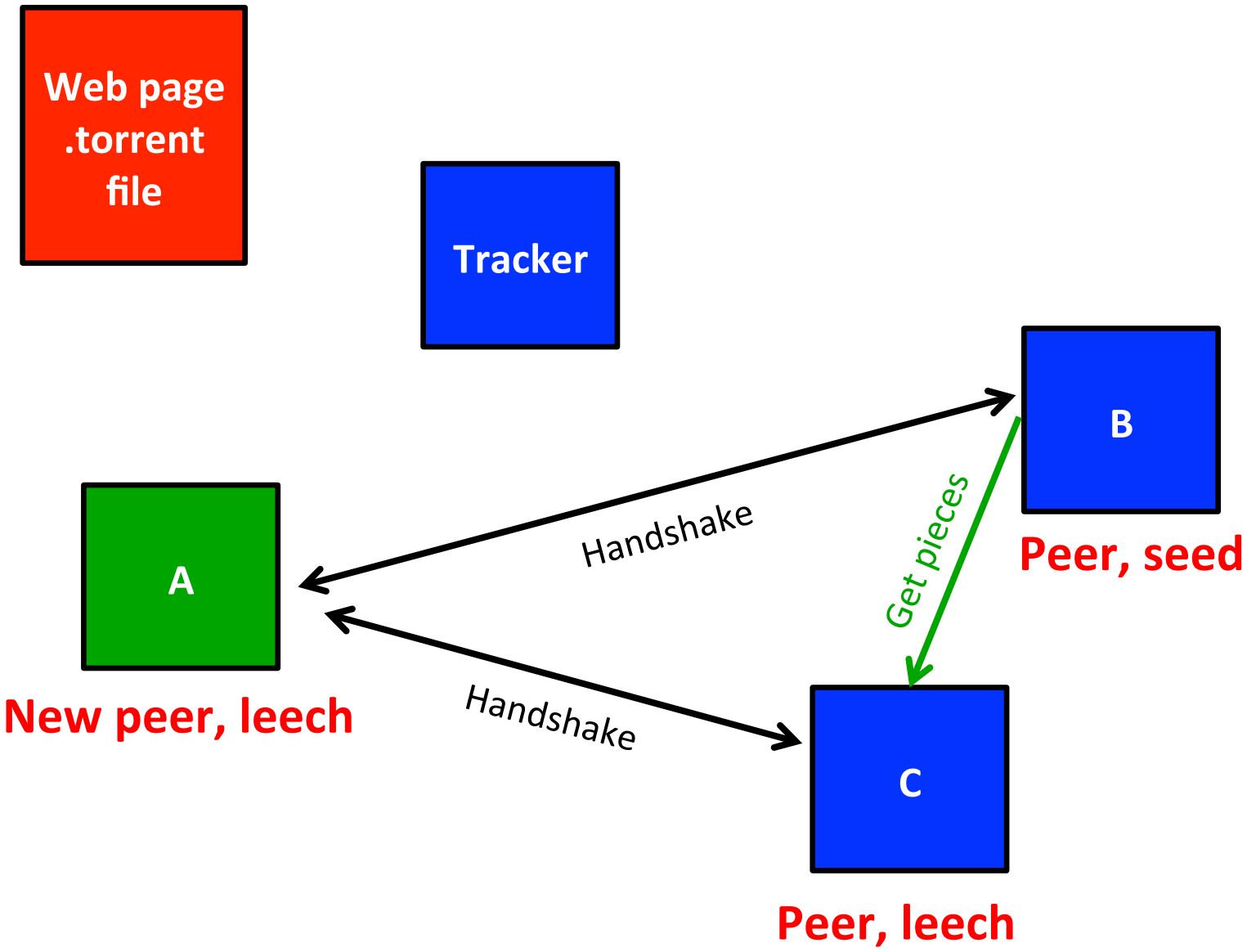
# BitTorrent process



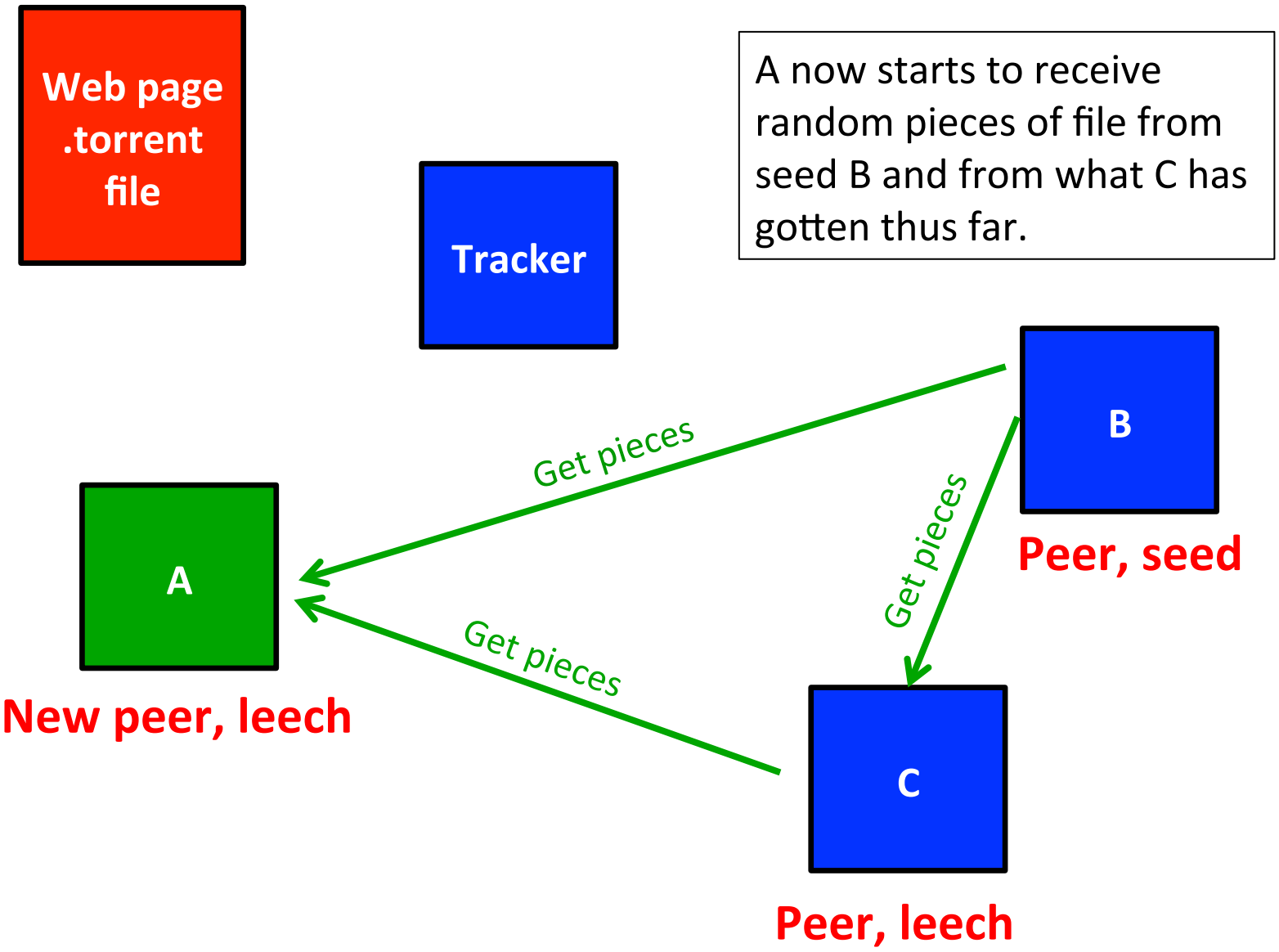
# BitTorrent process



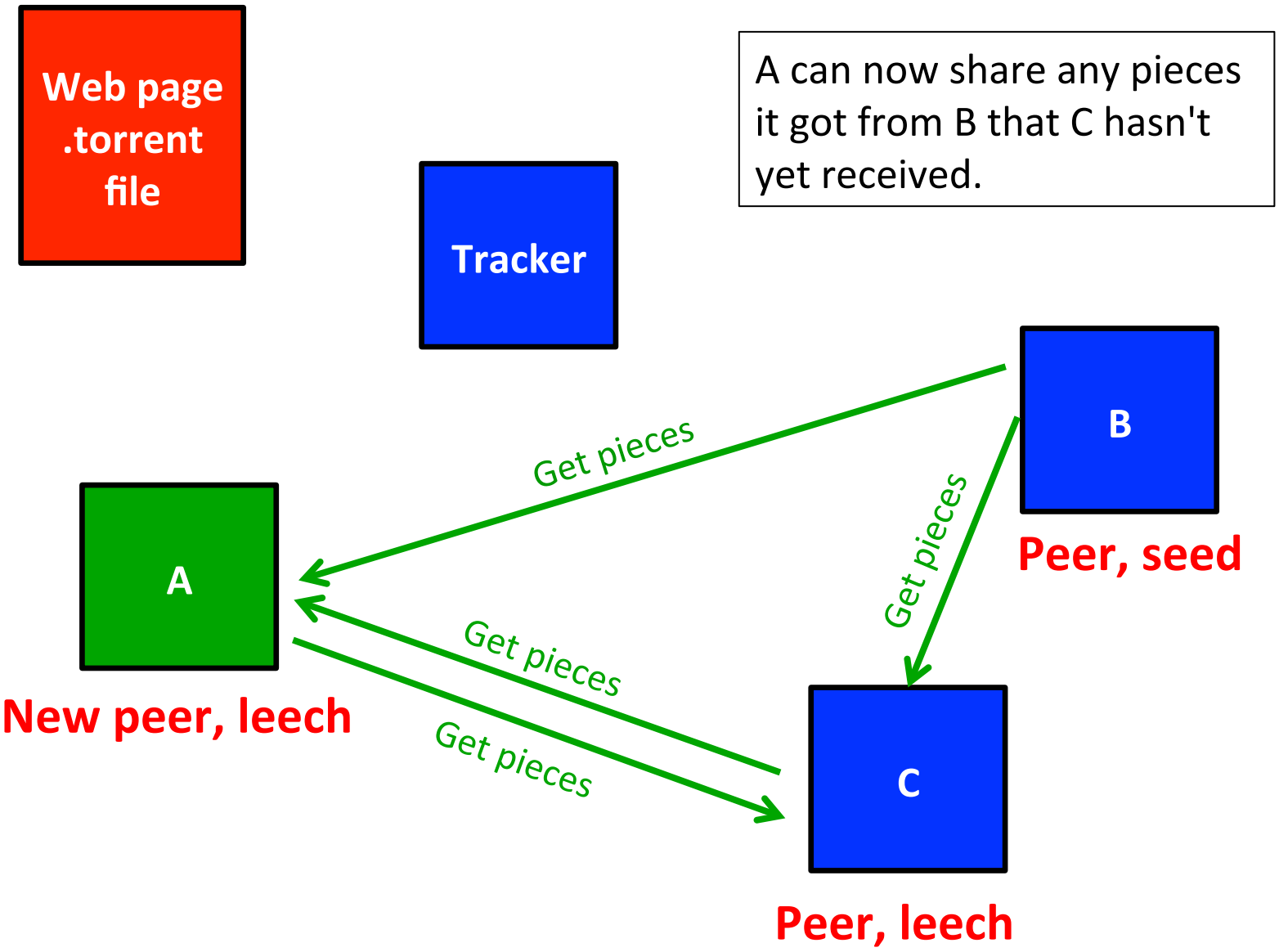
# BitTorrent process



# BitTorrent process



# BitTorrent process



# Summary

- Overlay networks
  - Allow deployment of new routing protocols and services
  - On top of the existing physical internet
  - Much nimbler that relying on upgrades in all routers
- Applications of overlays:
  - IPv6 deployment, Mobile IP, Multicast routing
  - Virtual Private Networks (VPNs), Resilient Overlay Networks (RON)
- P2P services:
  - Varying degrees of centralization
  - Different target applications