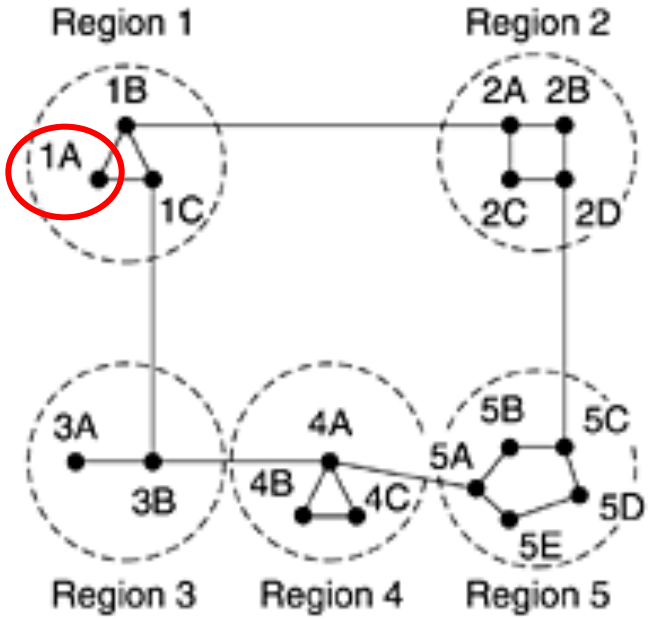
Chapter5 Network Layer(4)

王昊翔 hxwang@scut.edu.cn
School of Computer Science & Engineering ,SCUT
Communication & Computer Network key-Lab of GD

Outline of the lecture

- Hierarchical routing
- Broadcast routing
- Multicast routing
- Mobile routing
- Adhoc routing
- P2P

Hierarchical Routing



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Reduce table!

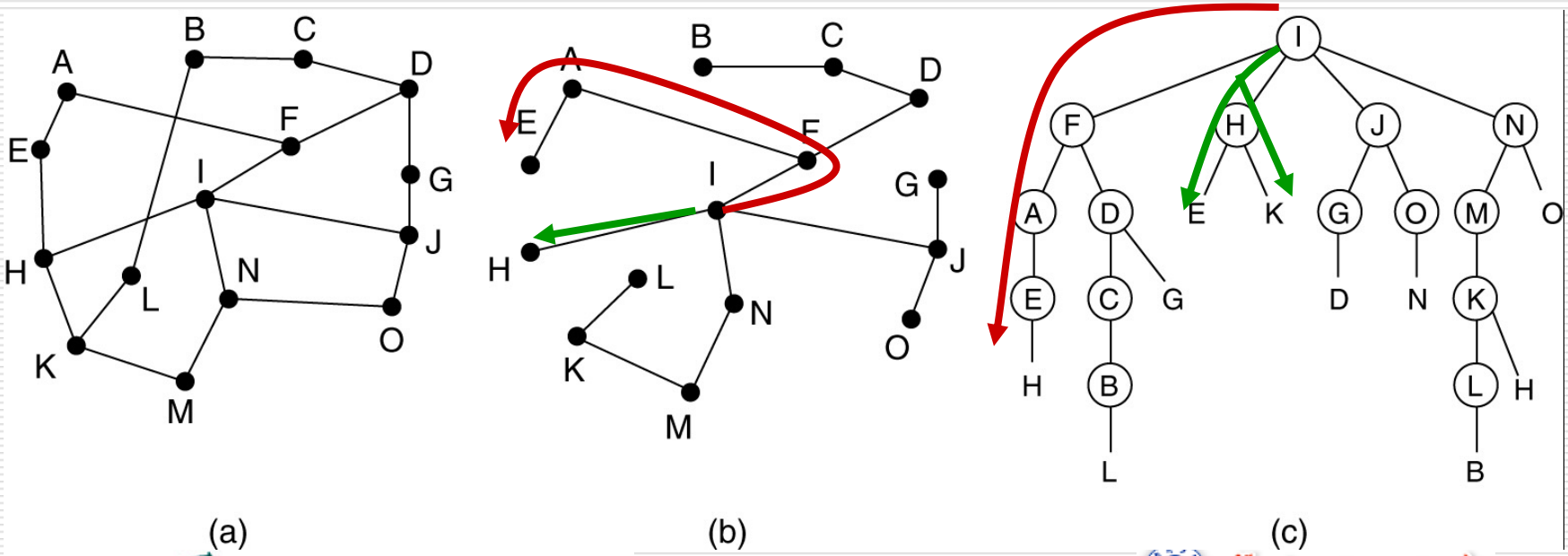
(c)

Broadcast Routing

- **Possible applications:**
  - **Distributing weather reports, stock market updates, live radio programs.**
- **Broadcast routing can be done in one of five ways:**
  - **Broadcasting each of the packets by sending a unique packet to each destination.**
  - **Flooding the network with the packet so that each machine will receive it.**
  - **Use multi-destination routing to send the packets to particular machines on the network.**
  - **Use a sink tree (or spanning tree) to direct the packets**
  - **Use reverse path forwarding to help control the "flood".**

Reverse path forwarding

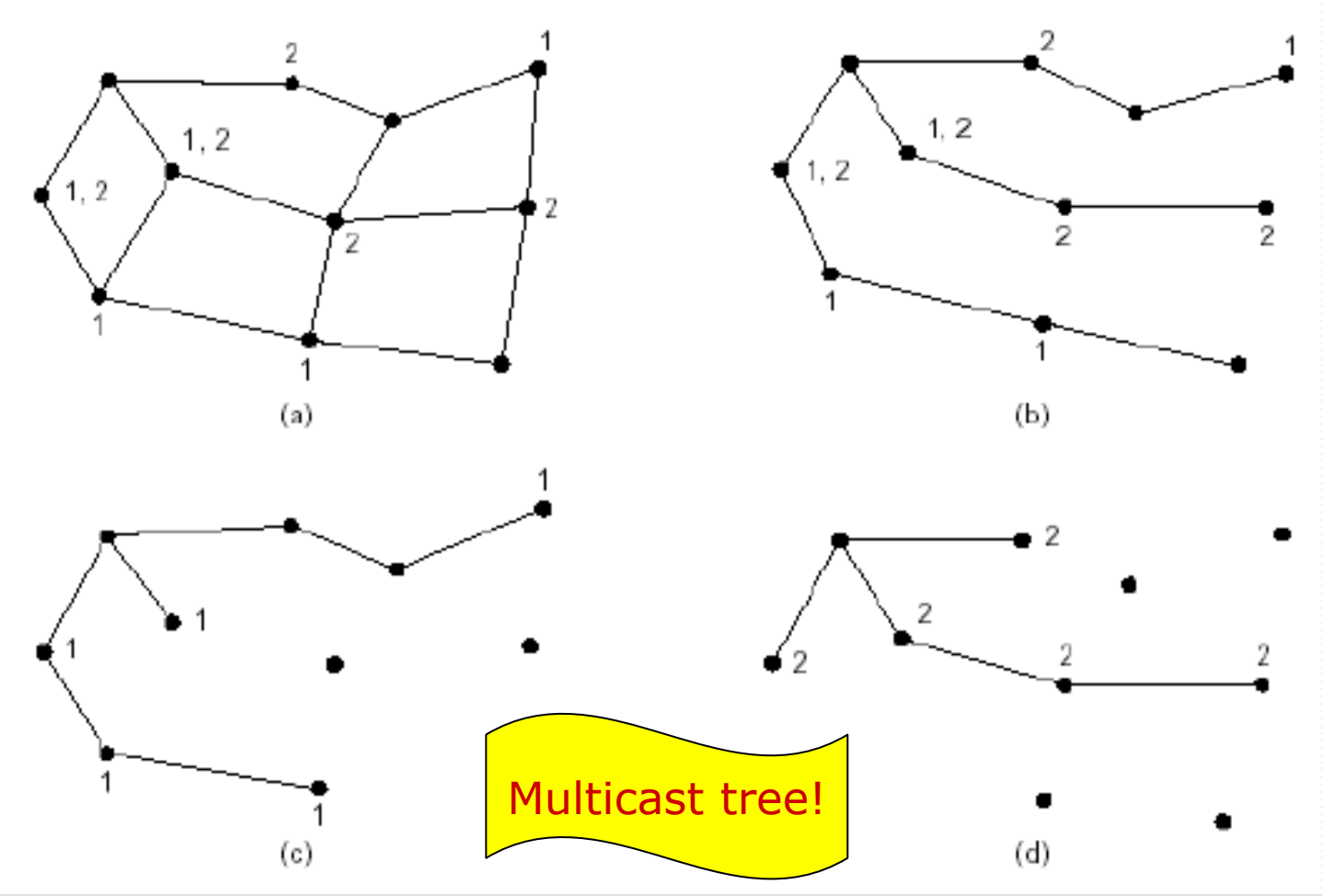
- 逆向路径转发
- **Basic idea: When a broadcast packet arrives at a router from the line that is normally used for sending packet to the source of the broadcast, it is forwarded onto all lines except the one it arrived on. Otherwise, it is discarded.**



Internet Multicasting

- IP supports multicasting, using class D addresses.
- Each class D address identifies a group of hosts.
  - 28 bits are available for identifying groups, so support 250 million groups.
- It's important to IP multicast:
  - Member management (IGMP/MLD)
  - Multicast routing table (PIM-DM/SM)
- Multicasting is implemented by special multicast routers.
  - Application-layer multicast

Multicast routing

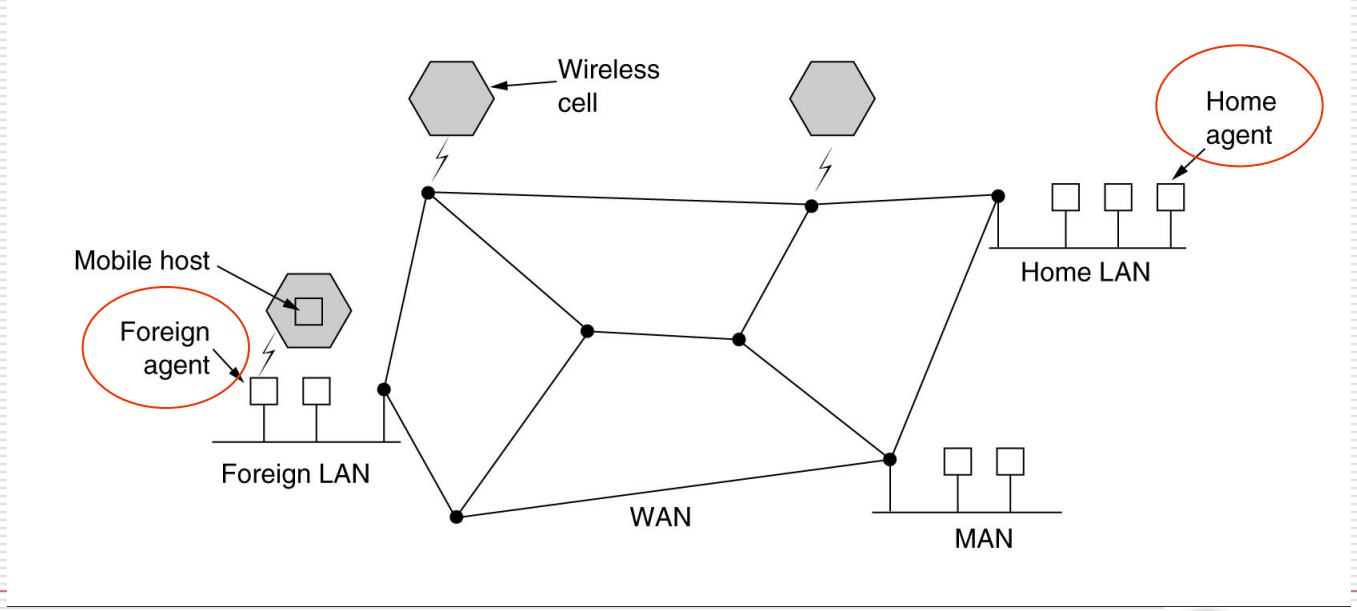


Routing for Mobile Hosts

- What is a mobile host? We include both:
  - A migratory host that physically connects to the network, but moves around (our laptops, stationary).
  - A roaming hosts actually compute on the run and want to maintain its connection as it moves around
- All hosts have a permanent home location and a permanent home address that can be used to determine their home locations.
- The routing goal in systems with mobile users is to make it possible to send packets to mobile users using their home addresses, and have the packets efficiently reach them wherever they may be.

Routing for Mobile Hosts (cont'd)

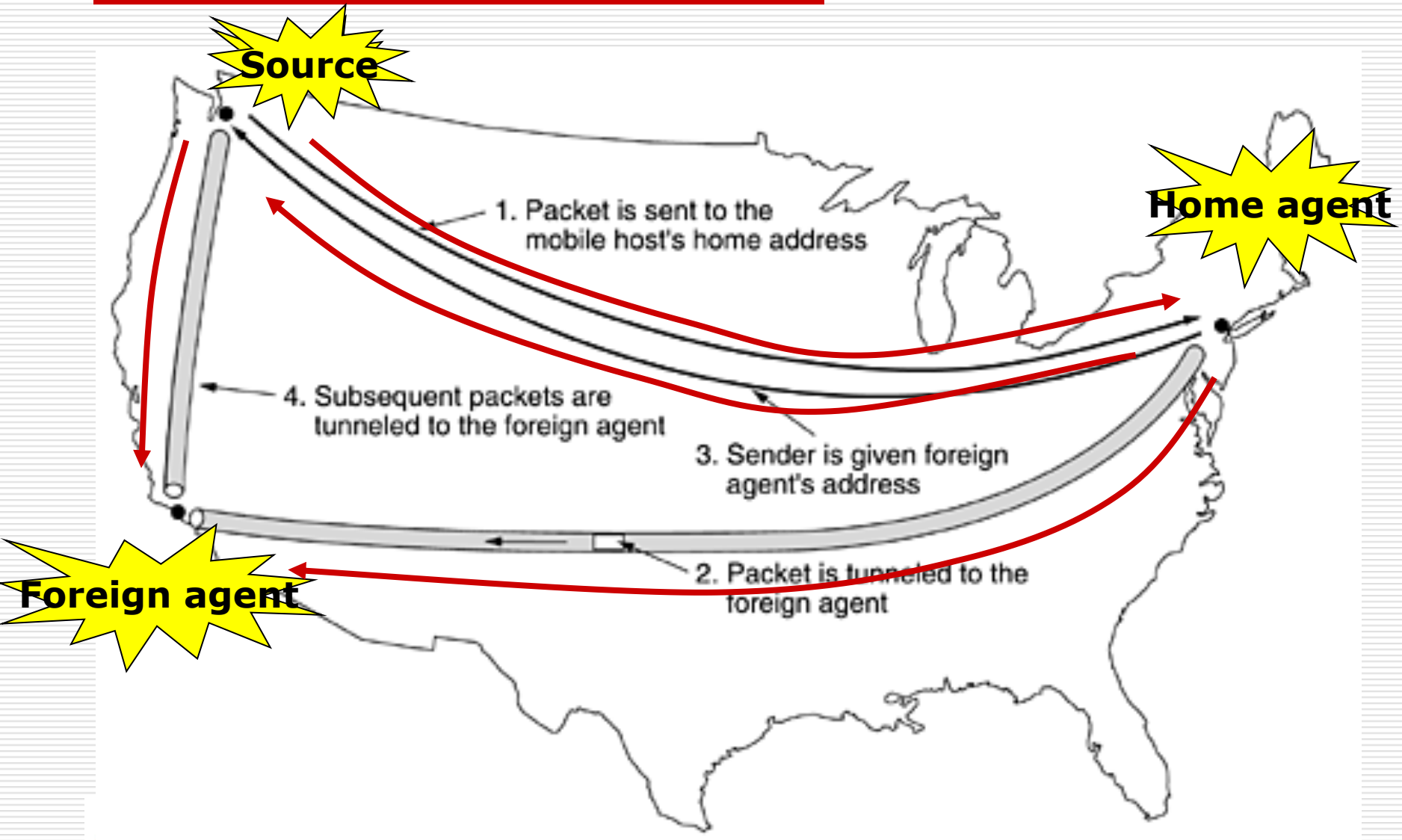
- **The world is divided into small areas**
  - **Each area has one or more foreign agents(外地代理), which keeps track of all mobile users visiting the area.**
  - **Each area also has one home agent(家乡代理), which keeps tracks of users whose home is in the area, but who are currently visiting another area.**



Mobile Host Registration Procedure

- Each foreign agent periodically broadcasts a packet announcing its existence and address.
- The mobile host registers with the foreign agent, giving its home address, current data link layer address, and some security information.
- The foreign agent contacts the mobile host's home agent to inform that the user is currently in the foreign area.
- The home agent examines the security information provided by the foreign agent, if the information is correct, then the foreign agent is told to proceed.
- When the foreign agent gets the acknowledgement from the home agent, it makes an entry in its tables and informs the mobile host that it is now registered.
- When a user leaves an area, it should de-registrate itself.

Packet Routing For Mobile Hosts (tunnel)



Routing in Ad Hoc Networks

- Ad hoc networks (or MANETs, Mobile Ad hoc NETworks) involve both the hosts and the routers are mobile:
  - Military vehicles on battlefield - No infrastructure
  - A fleet of ships at sea - All moving all the time
  - Emergency works at earthquake - The infrastructure destroyed.
  - A gathering of people with notebook computers - In an area lacking 802.11.
- With an ad hoc network, the topology may be changing all the time, routing in ad hoc networks quite different from routing in their fixed counterparts.

Peer-to-Peer Networks

- A peer-to-peer (or P2P) computer network is a network that relies primarily on the computing power and bandwidth of the participants in the network rather than concentrating it in a relatively low number of servers.
- All nodes are symmetric and there is no central control or hierarchy.
- Nobody is willing to host and maintain a centralized database or even a centralized index.

Peer-to-Peer Networks (cont'd)

- Advantages of peer-to-peer networks
  - All clients provide resources, including bandwidth, storage space, and computing power.
  - The distributed nature of peer-to-peer networks also increases robustness in case of failures in pure P2P systems
- Unstructured and structured P2P networks
  - Unstructured P2P network
    - To find a desired piece of data in the network, the query has to be flooded through the network in order to find as many peers as possible that share the data.
    - Most of the popular P2P networks such as Napster, Gnutella and KaZaA are unstructured.
  - Structured P2P networks
    - maintain a Distributed Hash Table (DHT) and by allowing each peer to be responsible for a specific part of the content in the network.
    - Some well known structured P2P networks are Chord, Pastry, Tapestry, CAN and Tulip

http://www.intsci.ac.cn/users/luojw/P2P/

P2P 文件共享

Example

- Alice runs P2P client application on her notebook computer
- Intermittently connects to Internet; gets new IP address for each connection
- 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: HTTP
- While Alice downloads, other users uploading from Alice.
- Alice’s peer is both a Web client and a transient Web server.

All peers are servers = highly scalable!

P2P: centralized directory

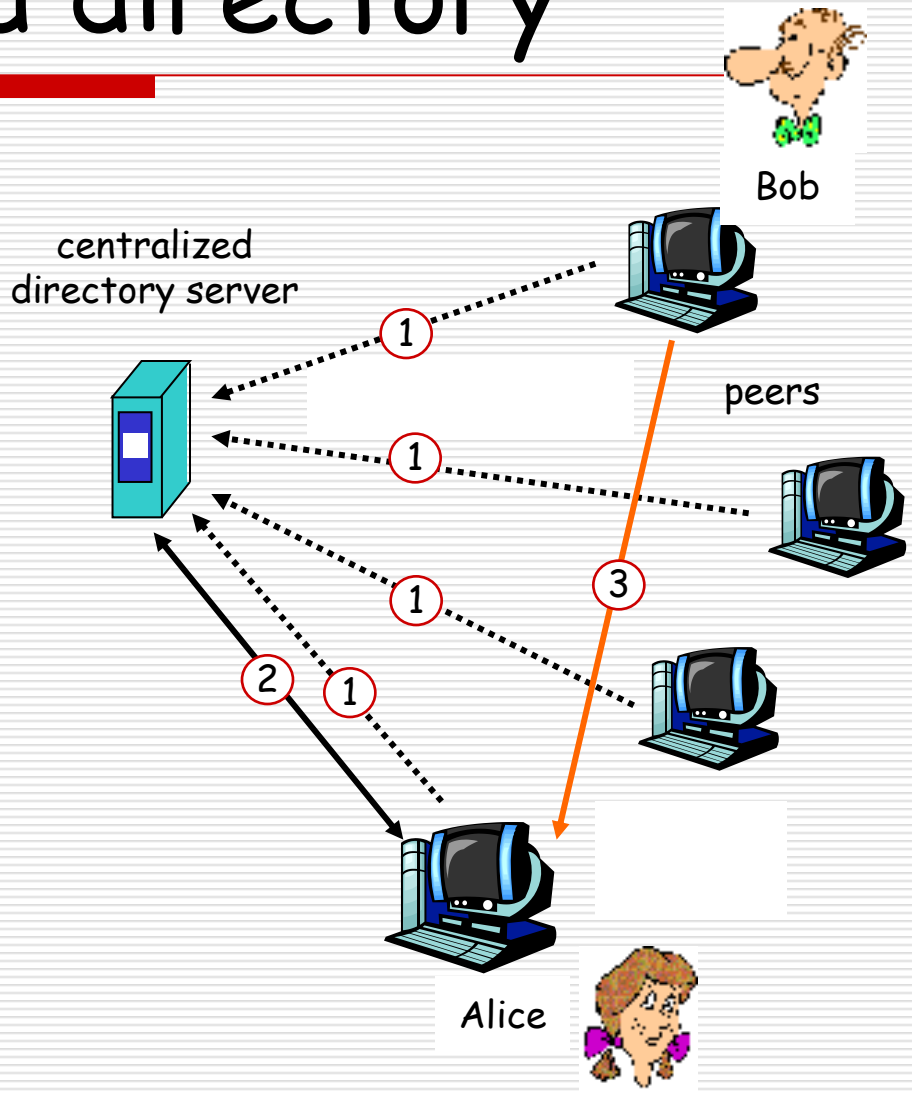
Original "Napster" design

1) when peer connects, it informs central server:

- **IP address**
- **content**

2) Alice queries for "Hey Jude"

3) Alice requests file from Bob



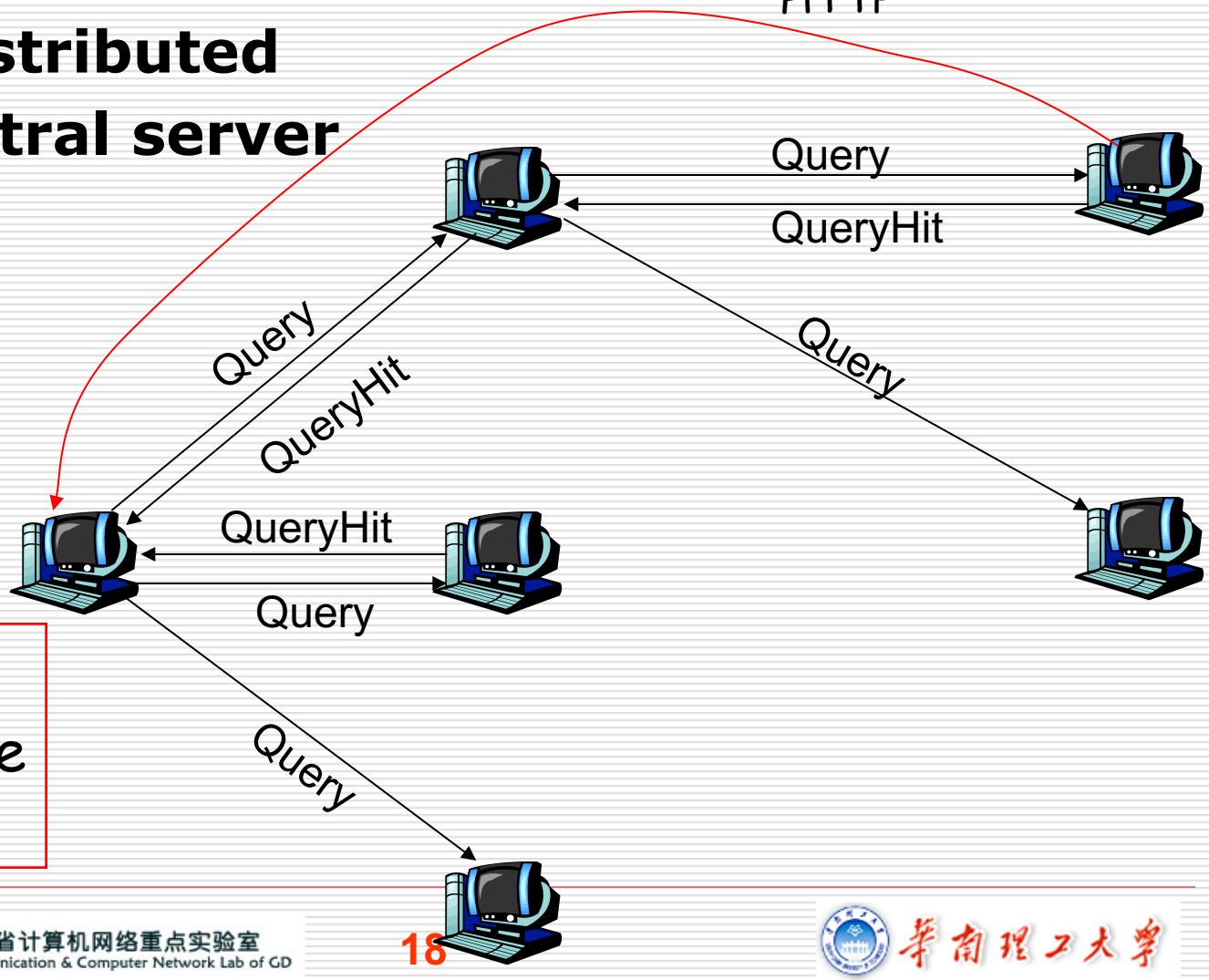
P2P: problems with centralized directory

- Single point of failure
- Performance bottleneck
- Copyright infringement

file transfer is decentralized, but locating content is highly centralized.

Gnutella: protocol

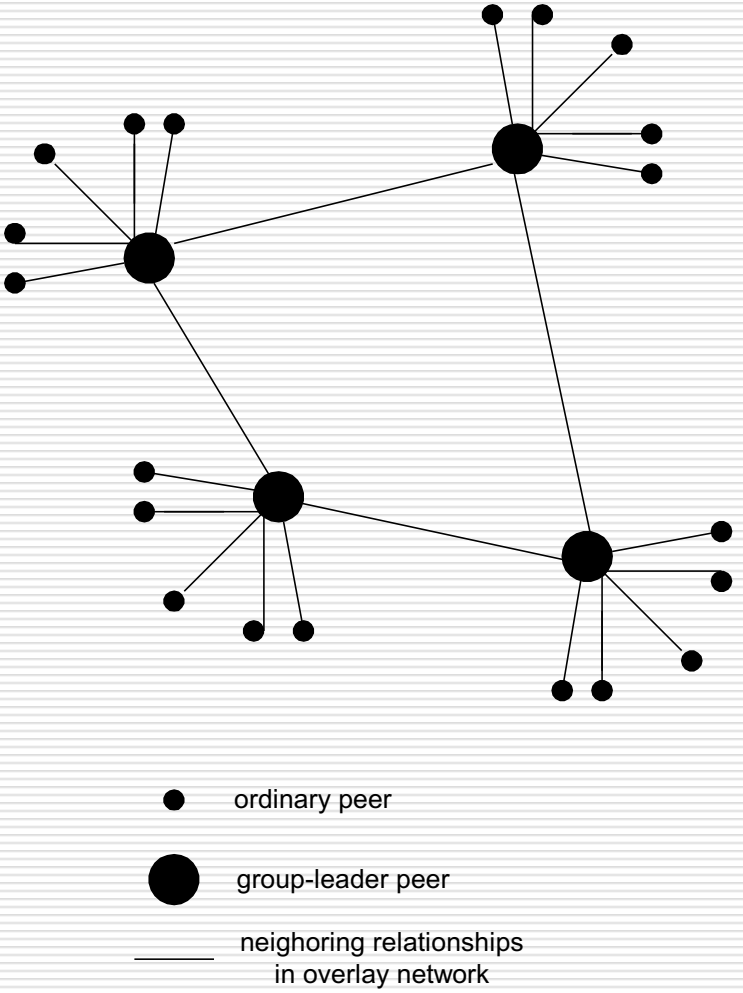
- **fully distributed**
  - **no central server**



Scalability:
limited scope
flooding

Exploiting heterogeneity: KaZaA

- Each peer is either a group leader or assigned to a group leader.
  - TCP connection between peer and its group leader.
  - TCP connections between some pairs of group leaders.
- Group leader tracks the content in all its children.



KaZaA: Querying

- Each file has a hash and a descriptor
- Client sends keyword query to its group leader
- Group leader responds with matches:
  - For each match: metadata, hash, IP address
- If group leader forwards query to other group leaders, they respond with matches
- Client then selects files for downloading
  - HTTP requests using hash as identifier sent to peers holding desired file

Node Lookup in Chord

- The Chord system consists of n participating users.
- Each user node has an IP address that can be hashed to an m-bit number using a hash function.
  - Chord uses SHA-1 for hash.
  - Any IP address is converted to a 160-bit number called the node identifier.
- Conceptually, all the 2^{160} node identifiers are arranged in ascending order in a big circle.
- Some of them correspond to participating nodes, but most of them do not.

Node Lookup in Chord (cont'd)

- Define the function successor(k) as the node identifier of the first actual node following k around the circle clockwise.
- The names of the records are also hashed with hash (i.e., SHA-1) to generate a 160-bit number, called the key.
  - key = hash(name)
- To make a record available to others, the node builds a tuple consisting of (name, my-IP-address) and then asks successor(hash(name)) to store the tuple.
- In this way, the index is distributed over the nodes at random.
- For fault tolerance, p different hash functions could be used to store each tuple at p nodes.

Node Lookup in Chord (cont'd)

- If some user later wants to look up name, he hashes it to get key and then uses successor (key) to find the IP address of the node storing its index tuples.
- Lookup procedure:
  - The requesting node sends a packet to its successor containing its IP address and the key it is looking for.
  - The packet is propagated around the ring until it locates the successor to the node identifier being sought.
  - That node checks to see if it has any information matching the key, and if so, returns it directly to the requesting node.

Node Lookup in Chord (cont'd)

- As a first optimization, each node could hold the IP addresses of both its successor and its predecessor, so that queries could be sent either clockwise or counterclockwise.
- Linearly searching all the nodes is very inefficient in a large peer-to-peer system since the mean number of nodes required per search is n/2.
- To greatly speed up the search, each node also maintains what Chord calls a finger table(指取表).
  - The finger table has m entries, indexed by 0 through m - 1, each one pointing to a different actual node.
  - Each of the entries has two fields: start and the IP address of successor(start)
  - The values of the fields for entry i at node k are:

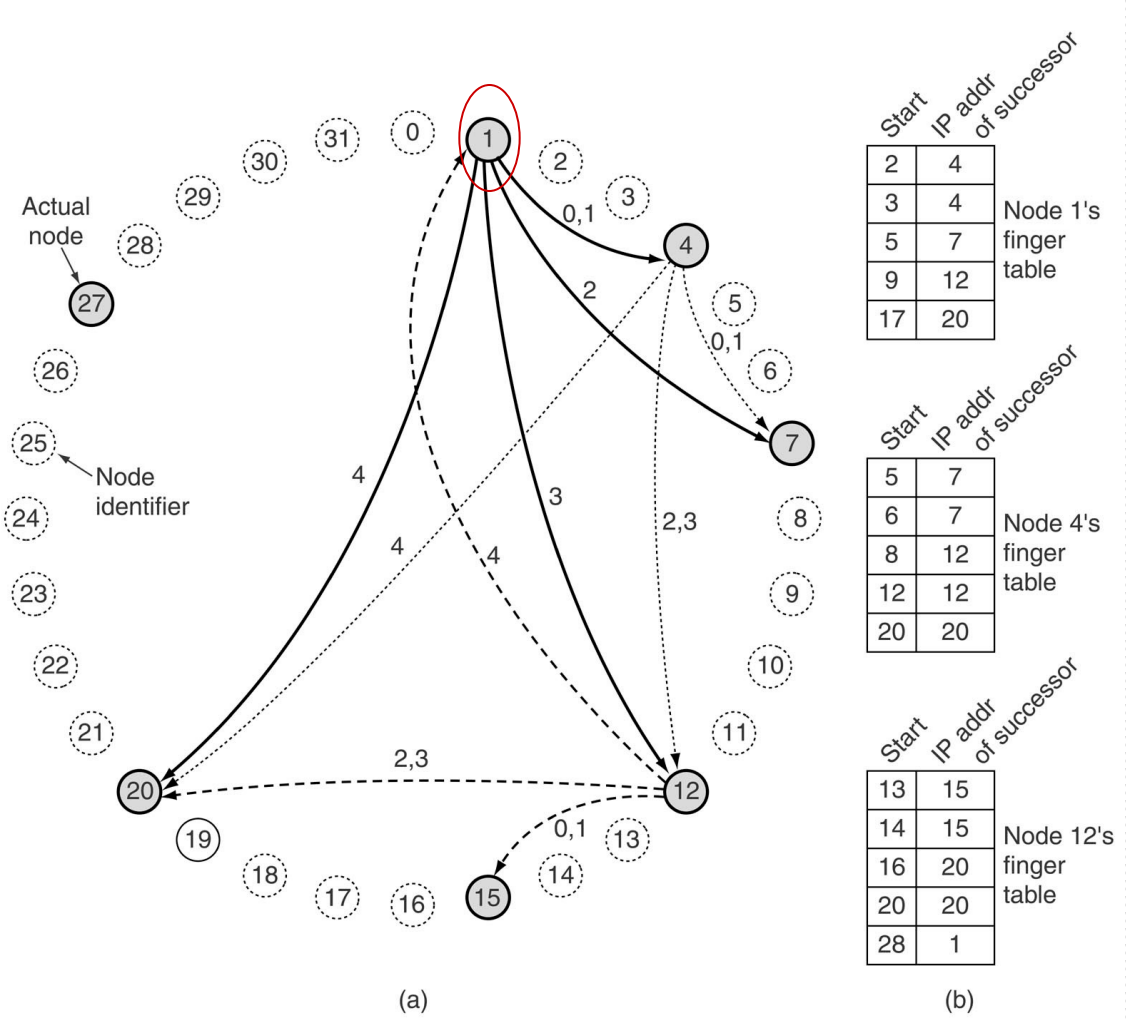
$$start = k + 2^i \quad (modulo\, 2^m)$$

IP address of $successor(start[i])$

Node Lookup in Chord (cont'd)

- Using the finger table, the lookup of key at node k proceeds as follows.
  - If key falls between k and successor (k), then the node holding information about key is successor (k) and the search terminates.
  - Otherwise, the finger table is searched to find the entry whose start field is the closest predecessor of key. A request is then sent directly to the IP address in that finger table entry to ask it to continue the search.
  - The average number of lookups is $\log_2 n$.

Node Lookup in Chord (cont'd)



Lookup key= 3 on node 1

Lookup key=14 on node 1

Lookup key=16 on node 1

New Node Join And Leave

- When a new node, r, wants to join, it must contact some existing node and ask it to look up the IP address of successor (r) for it.
- The new node then asks successor (r) for its predecessor.
- The new node then asks both of these to insert r in between them in the circle.
- When a node leaves gracefully, it hands its keys over to its successor and informs its predecessor of its departure so the predecessor can link to the departing node's successor.
- To alleviate the problem caused by node crashes, each node keeps track not only of its direct successor but also its s direct successors

Summary of the lecture

- Multi-level routing
- broadcast routing
- mobile routing
- adhoc routing
- p2p node look-up

Summary of the first part

- How a packet arrive the destination from source?
- Routing algorithm
  - DV: RIP
  - LS: OSPF
  - BGP
  - Others