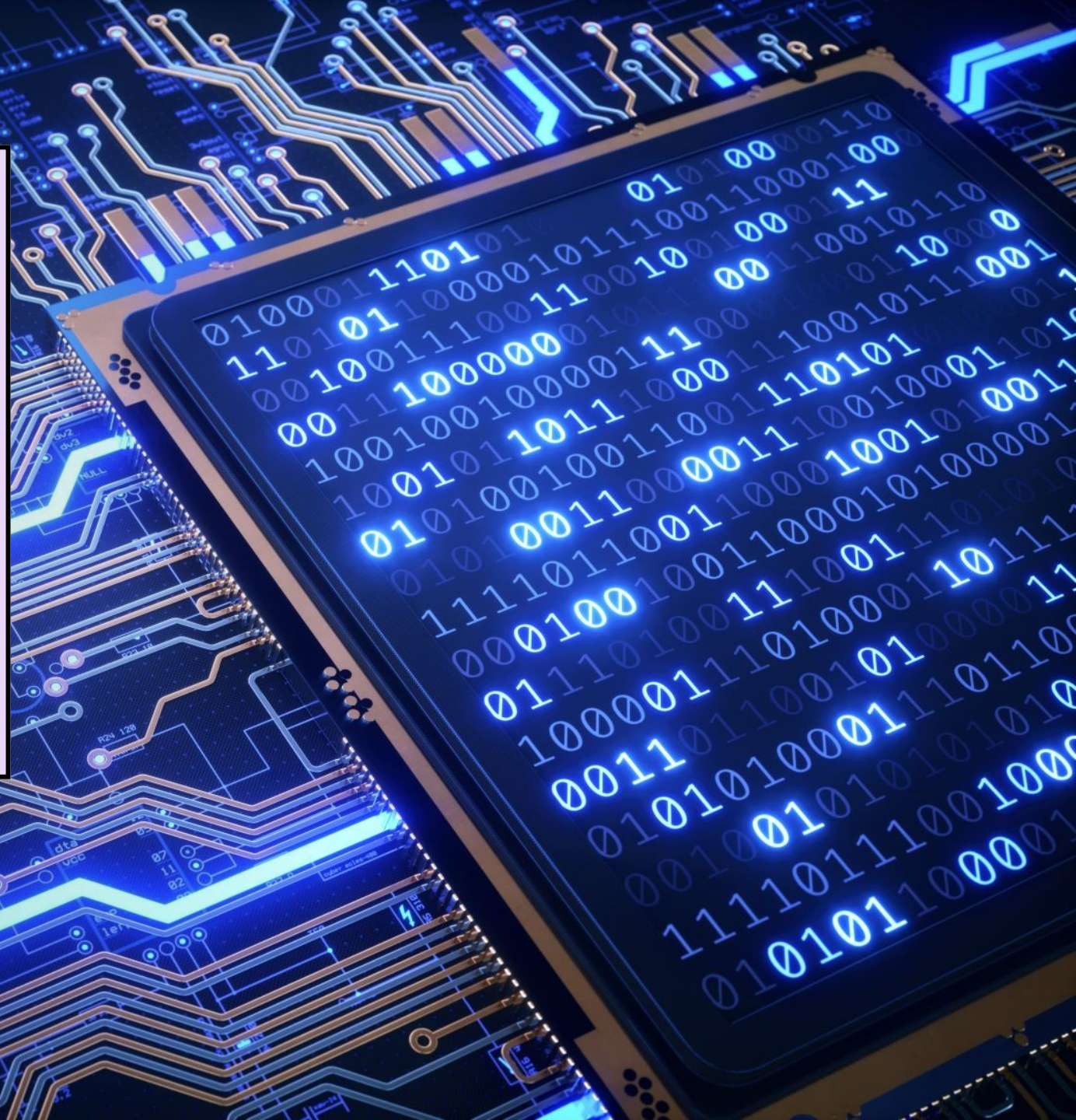


COMPUTER NETWORKS

APPLICATION LAYER



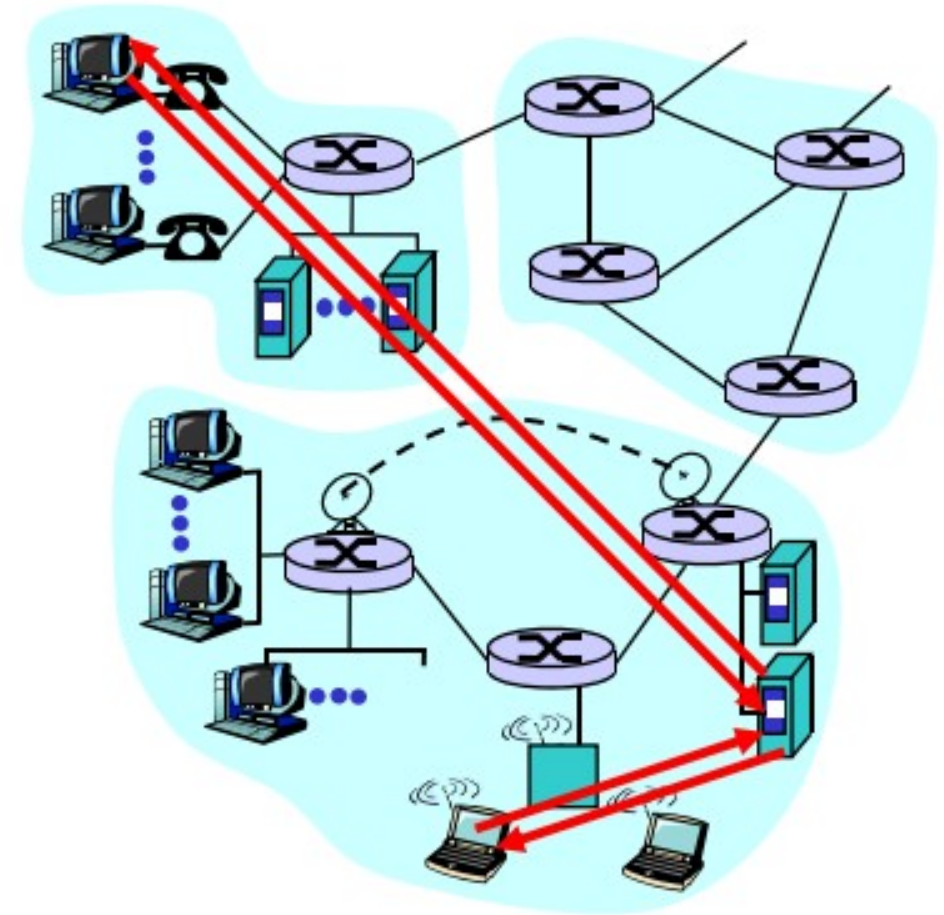
○ Topic of the day

- Peer to Peer Paradigm
 - Centralized Network
 - Decentralized Networks(Unstructured & Structured)
- Distributed Hash Tables
- P2P Network Protocols(Chord, Pastry, Kademlia)
- BitTorrent (Trackerless & with Tracker)
- BlockChain (Cryptocurrencies, Smart Contracts, Supply Chain)



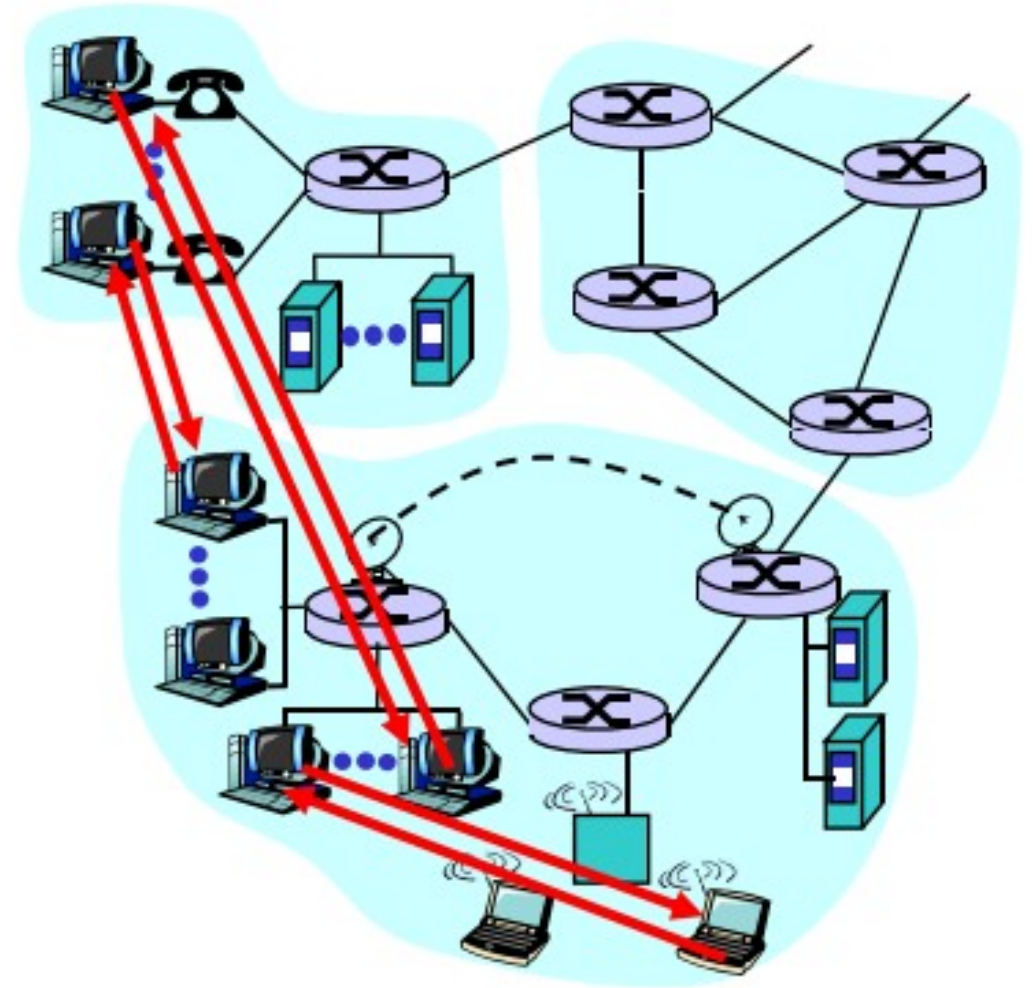
Client-Server Architecture

- 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 directly communicate with eac



○ Peer to Peer Architecture

- No always-on server
- arbitrary end systems directly communicate
- peers are intermittently connected and change IP addresses
- example: Gnutella
- Highly scalable
- Difficult to manage





Examples

- Freenet: (1999) censorship-resistant network providing anonymity
- Napster: (1999-2001) music file sharing service
- Gnutella: (2000) File sharing system
- Fast Track(Kazaa)
- BitTorrent
- WinMax
- GNUnet
- BitCoin
- Ethereum
- InterPlanetary File System



○ Types

- Centralized P2P Network
 - Hybrid system
 - Directory System: Listing of peers centralized(Client-Server)
 - Easy & Simple Maintenance
 - Central point of Failure
 - Traffic issues on the server
 - Vulnerable to attacks
- Decentralized P2P Network
 - No centralized Directory
 - Overlay network of peers(Logical Network)
 - 2 types depending on how the nodes are linked in the overlay network
 - Structured and Unstructured



○ Centralized P2P Network

Napster

- File Transfer P2P
- File search centralized:
 - Peers register content at central server
 - Peers query same central server to locate content

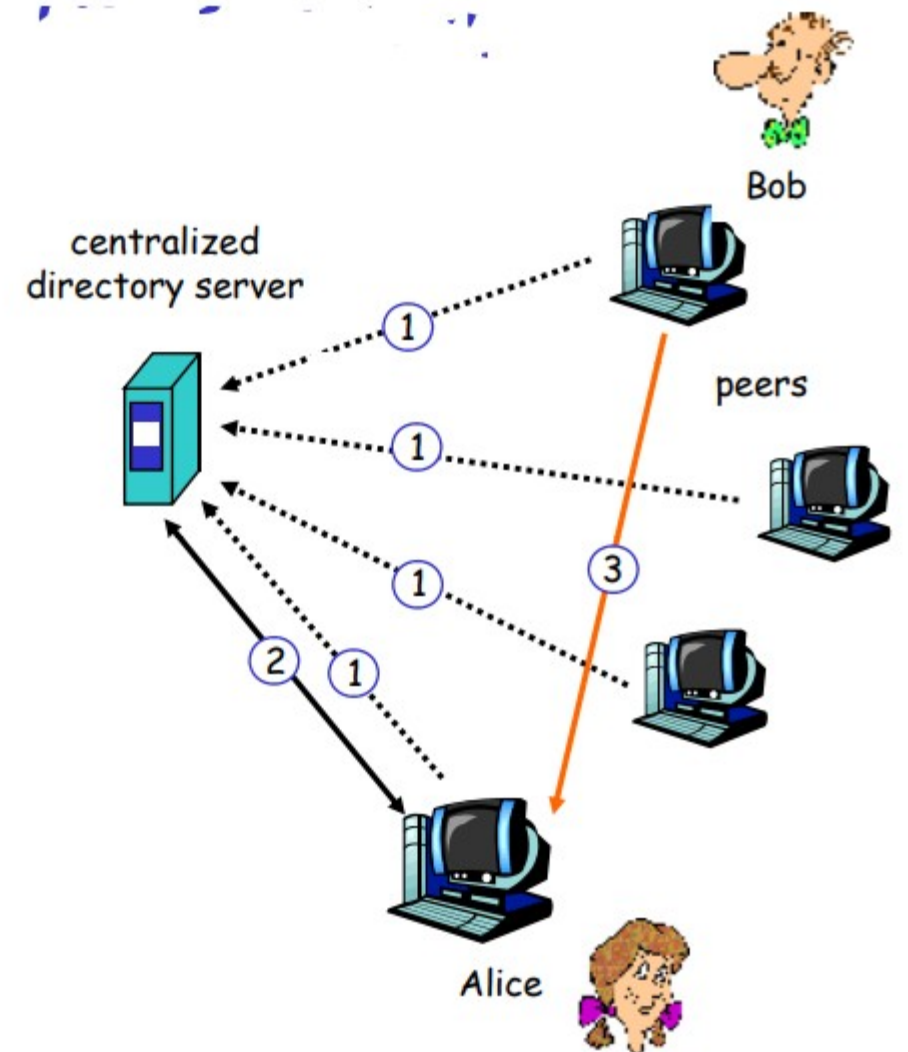
Instant messaging

- Chatting between two users is P2P
- Presence detection/location centralized:
 - User registers its IP address with central server when it comes online
 - User contacts central server to find IP addresses of buddies



Original “Napster” design

- When peer connects, it informs centralized directory server
- informs central server:
 - IP address
 - Content peers
- Alice queries for “Hey Jude”
- 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 decentralized



○ Unstructured Network

- Randomly linked nodes
- Search is not efficient
- Network flooding with query as links are random
- Generate a lot of traffic
- Query resolution takes time or sometimes not resolved
- Examples:
 - Gnutella
 - Freenet





Query flooding: Gnutella

- ❑ fully distributed
 - no central server
- ❑ public domain protocol
- ❑ many Gnutella clients implementing protocol

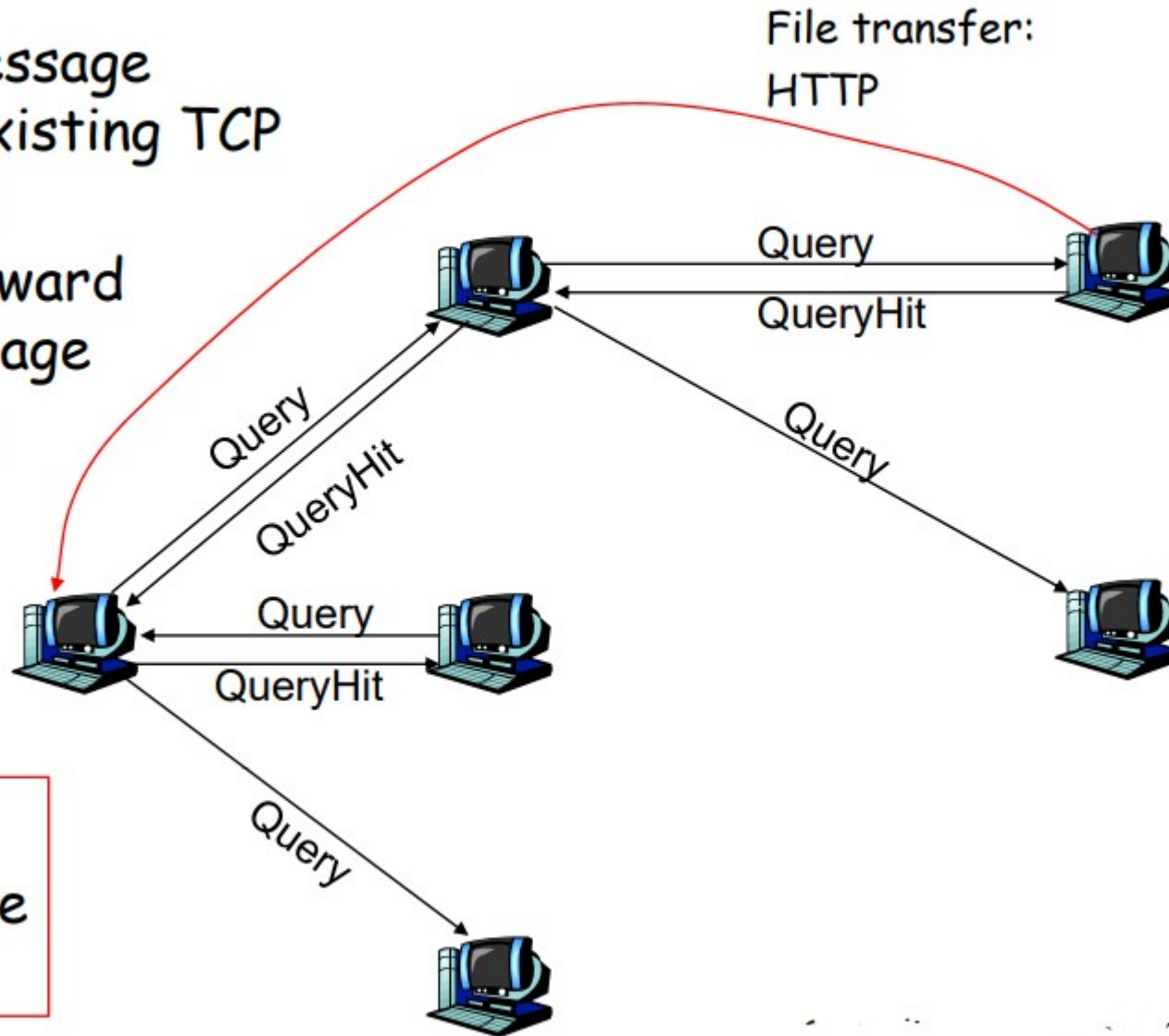
overlay network: graph

- ❑ edge between peer X and Y if there's a TCP connection
- ❑ all active peers and edges is overlay net
- ❑ Edge is not a physical link
- ❑ Given peer will typically be connected with < 10 overlay neighbors



Gnutella: protocol

- ❑ Query message sent over existing TCP connections
- ❑ peers forward Query message
- ❑ QueryHit sent over reverse path



Scalability:
limited scope
flooding



○ Gnutella: Peer joining

1. Joining peer X must find some other peer in Gnutella network: use list of candidate peers
2. X sequentially attempts to make TCP with peers on list until connection setup with Y
3. X sends Ping message to Y; Y forwards Ping message.
4. All peers receiving Ping message respond with Pong message
5. X receives many Pong messages. It can then setup additional TCP connections



○ Structured Networks

- Predefined rules to link nodes
- Effective and efficient resolution of queries
- Most Common technique- DHT
- Many Applications of DHTs
 - Distributed Data Structures DDS
 - Content Distribution Systems CDS
 - DNS
 - P2P File Sharing - BitTorrent



○ What is a DHT?

- Hash Table
 - data structure that maps “keys” to “values”
 - essential building block in software systems
- Distributed Hash Table (DHT)
 - similar, but spread across many hosts
- Interface
 - insert(key, value)
 - lookup(key)



○ How do DHTs work?

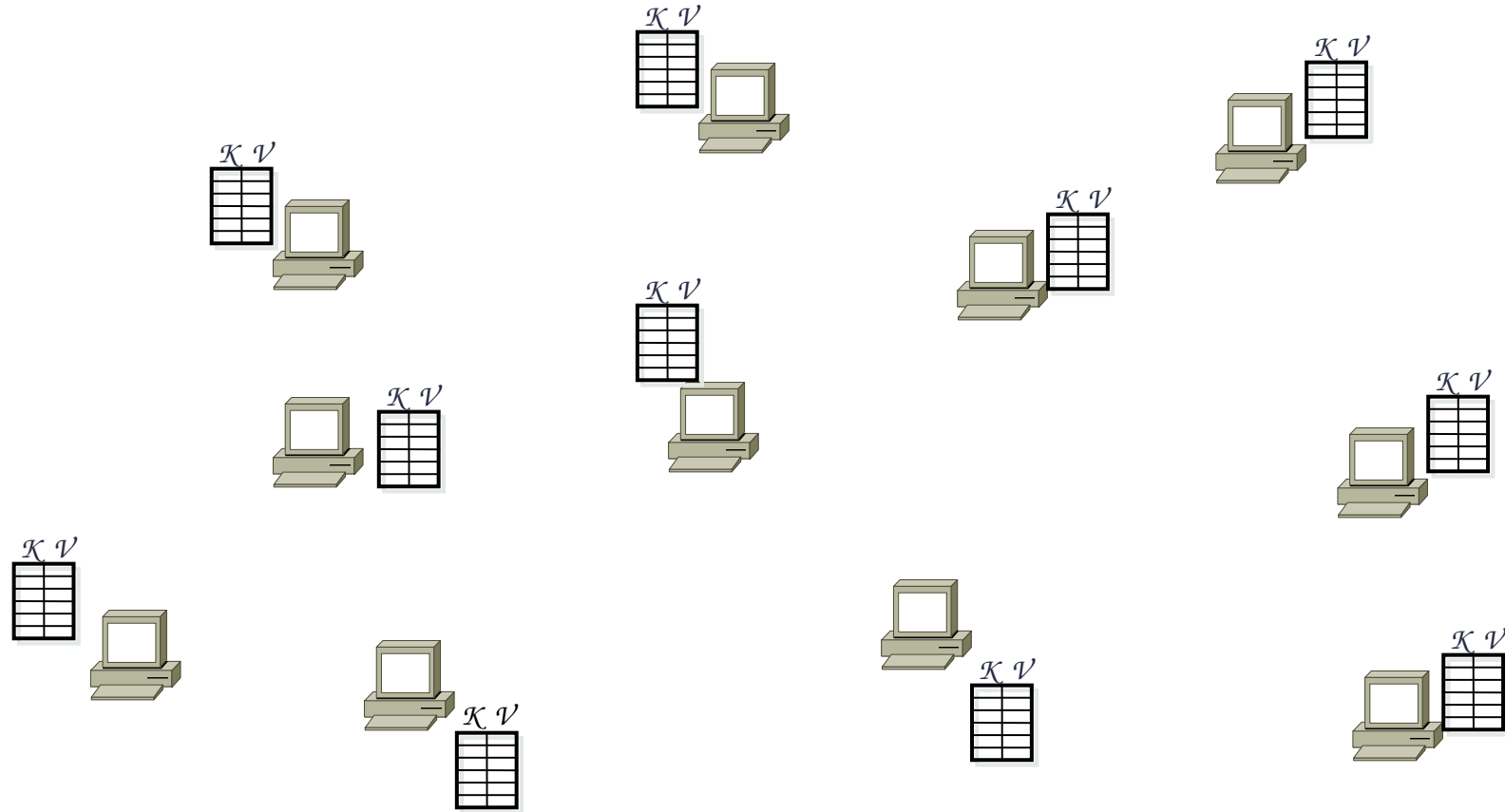
Every DHT node supports a single operation:

- Given *key* as input; route messages to node holding *key*
 - DHTs are *content-addressable*



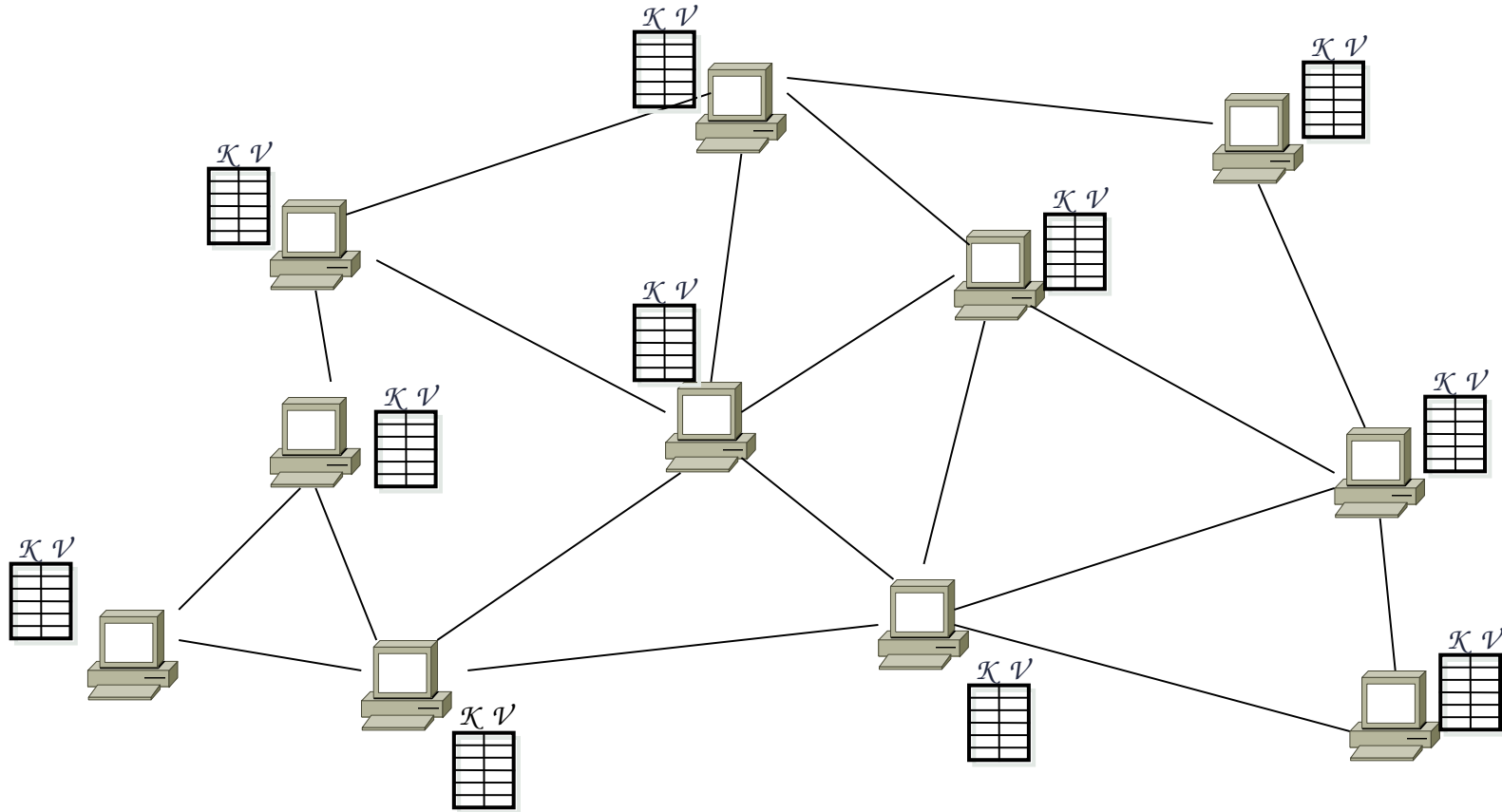


DHT: basic idea





DHT: basic idea

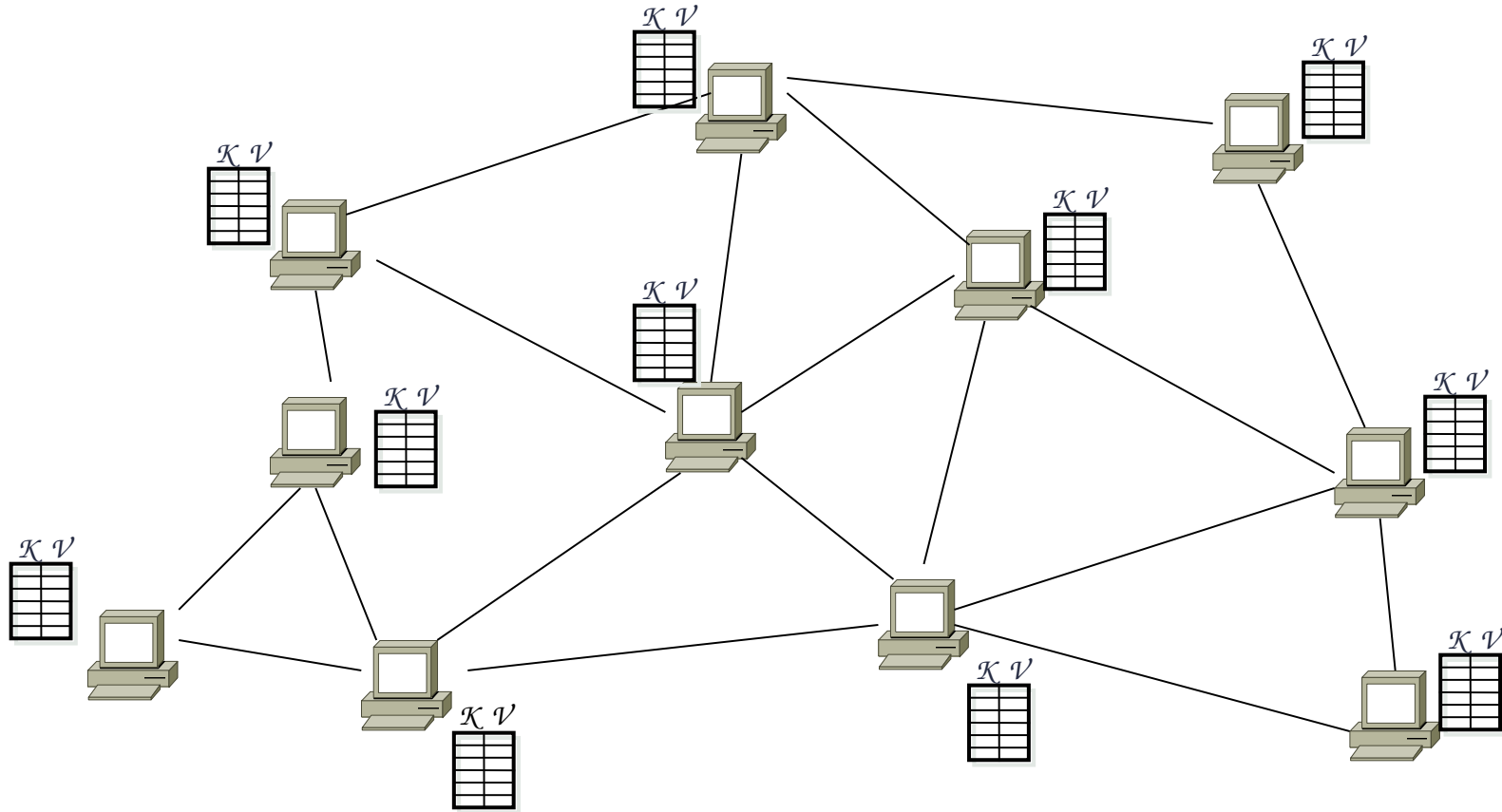


Neighboring nodes are “connected” at the application-level





DHT: basic idea

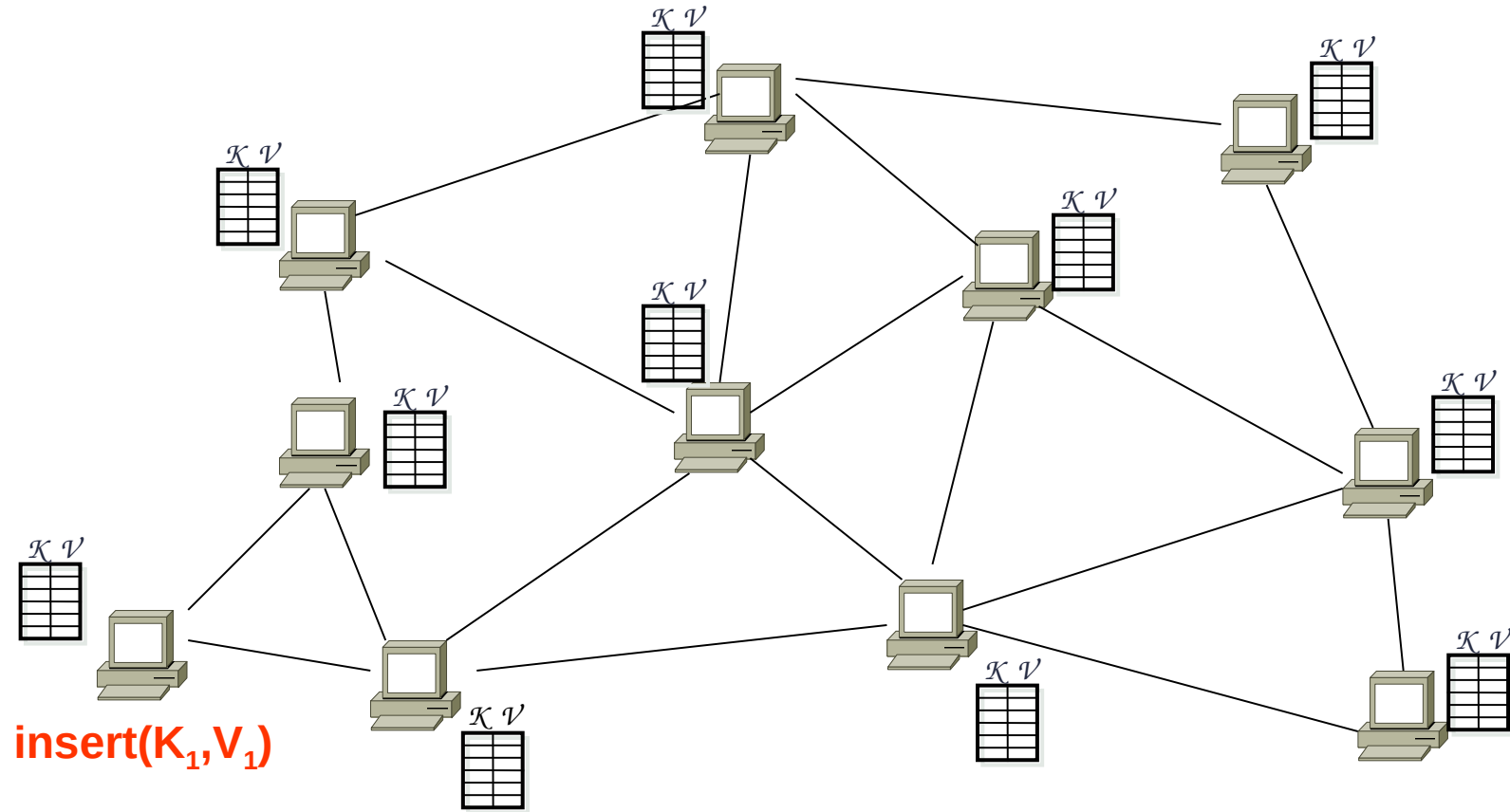


Operation: take *key* as input; route messages to node holding *key*





DHT: basic idea

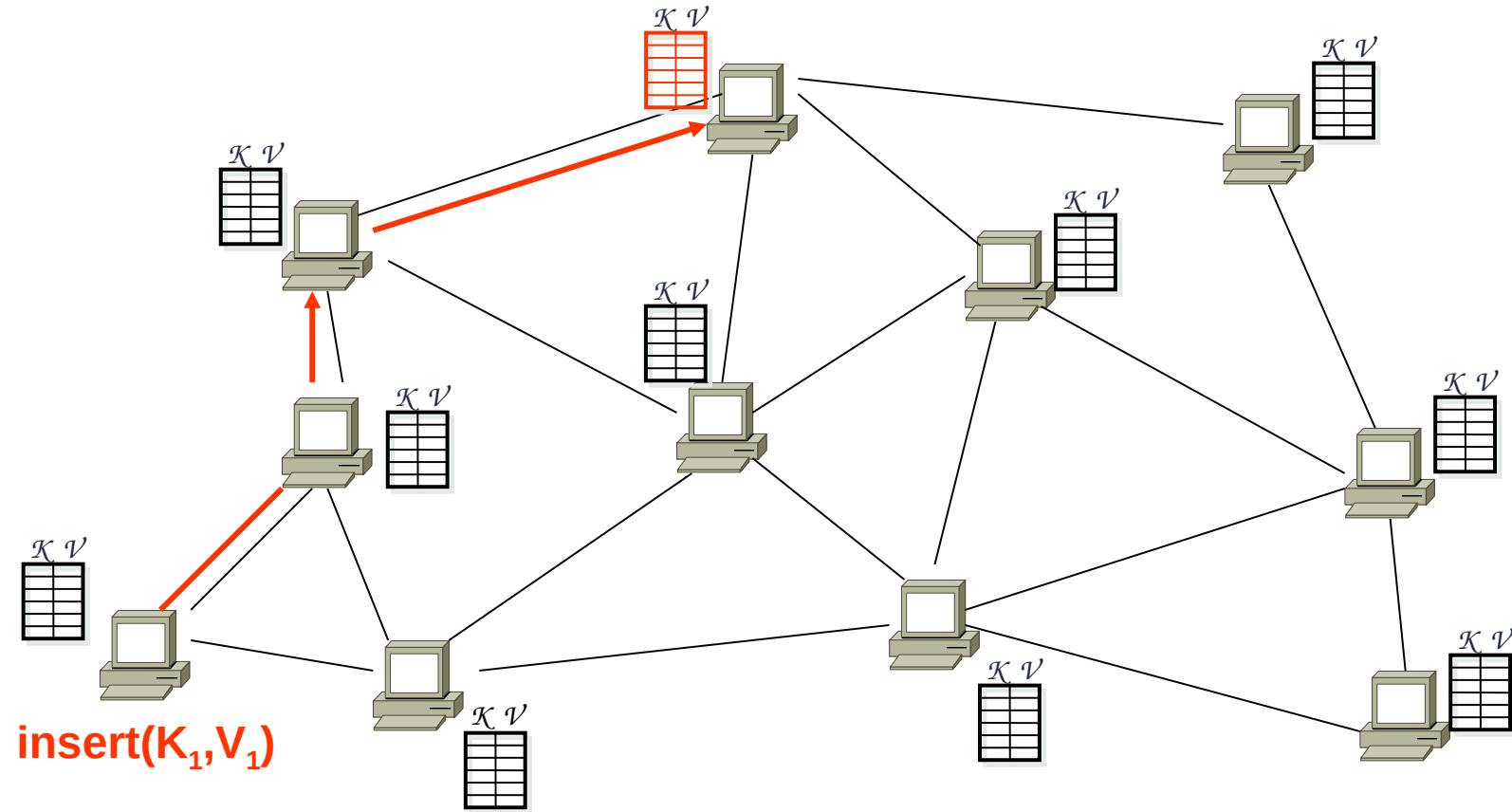


Operation: take *key* as input; route messages to node holding *key*





DHT: basic idea

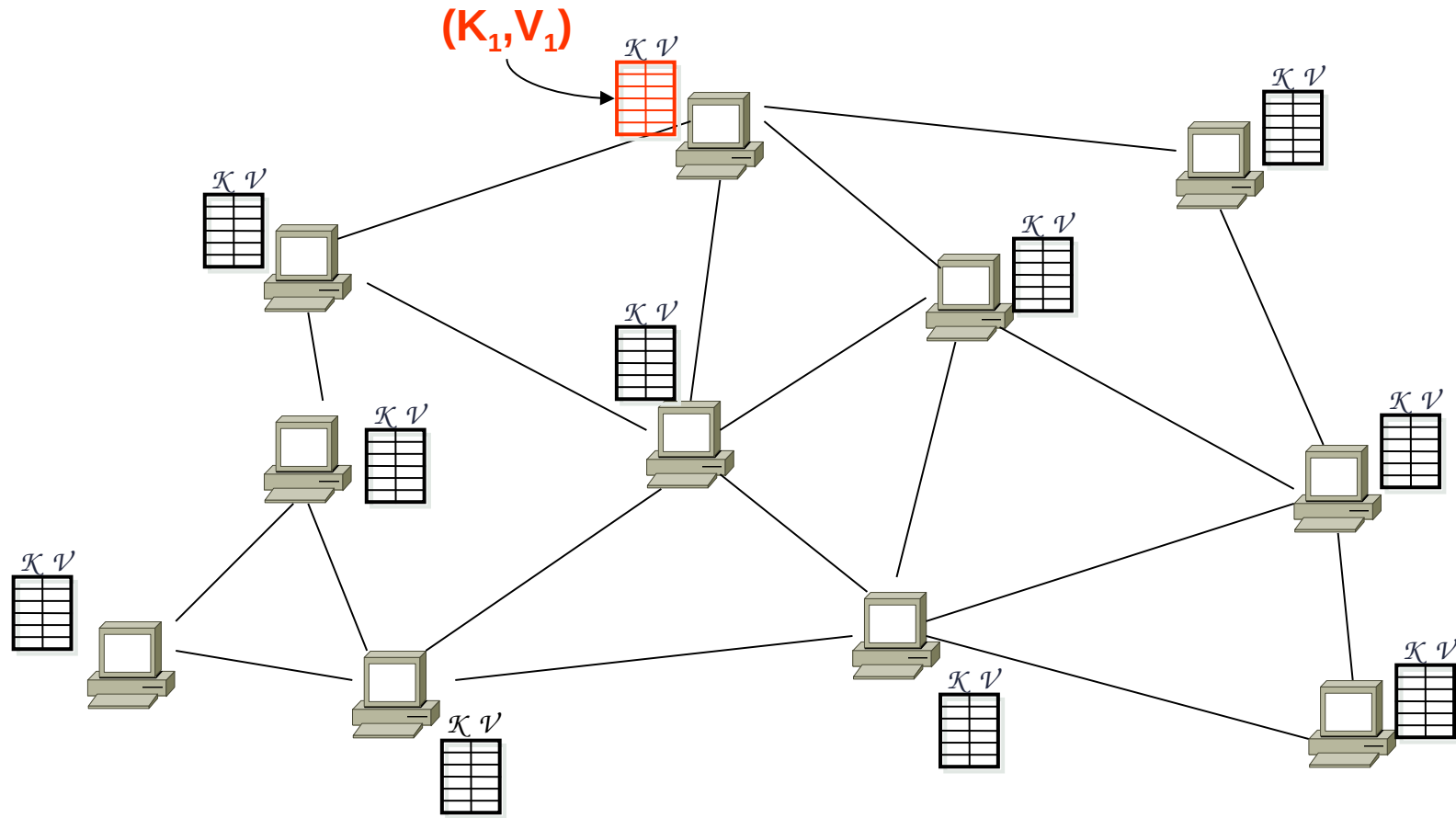


Operation: take *key* as input; route messages to node holding *key*





DHT: basic idea

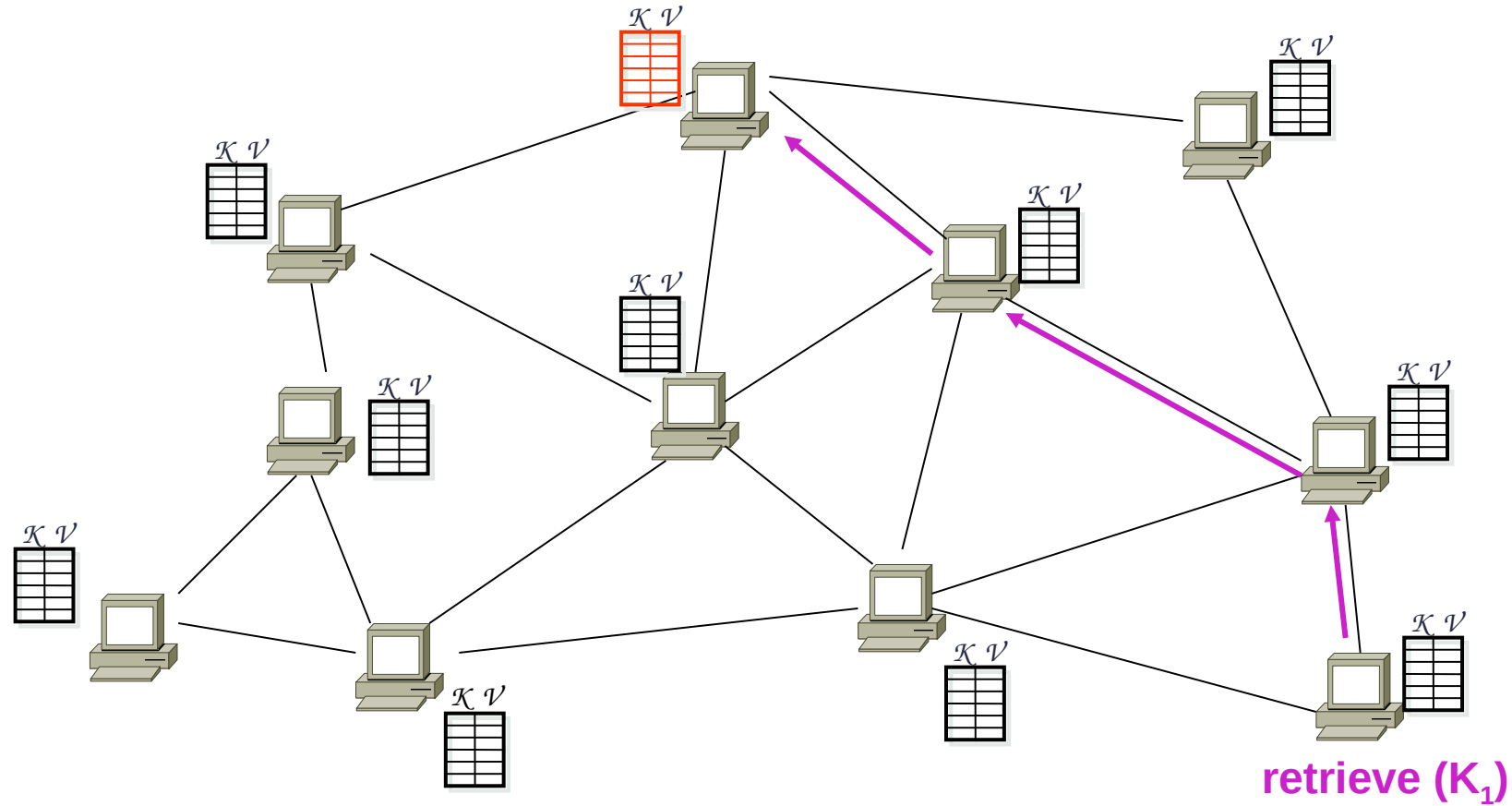


Operation: take *key* as input; route messages to node holding *key*





DHT: basic idea



Operation: take *key* as input; route messages to node holding *key*



○ How to design a DHT?

- State Assignment:
 - what “(*key*, *value*) tables” does a node store?
- Network Topology:
 - how does a node select its neighbors?
- Routing Algorithm:
 - which neighbor to pick while routing to a destination?
- Various DHT algorithms make different choices
 - CAN, Chord, Pastry, Tapestry, Plaxton, Viceroy, Kademlia, Skipnet, Symphony, Koorde, Apocrypha, Land, ORDI ...





- `nodeID= hash(peer IP address)`
- `key= hash(Object Name)`
- (Key, value) pair is stored.
- Value is the actual Content

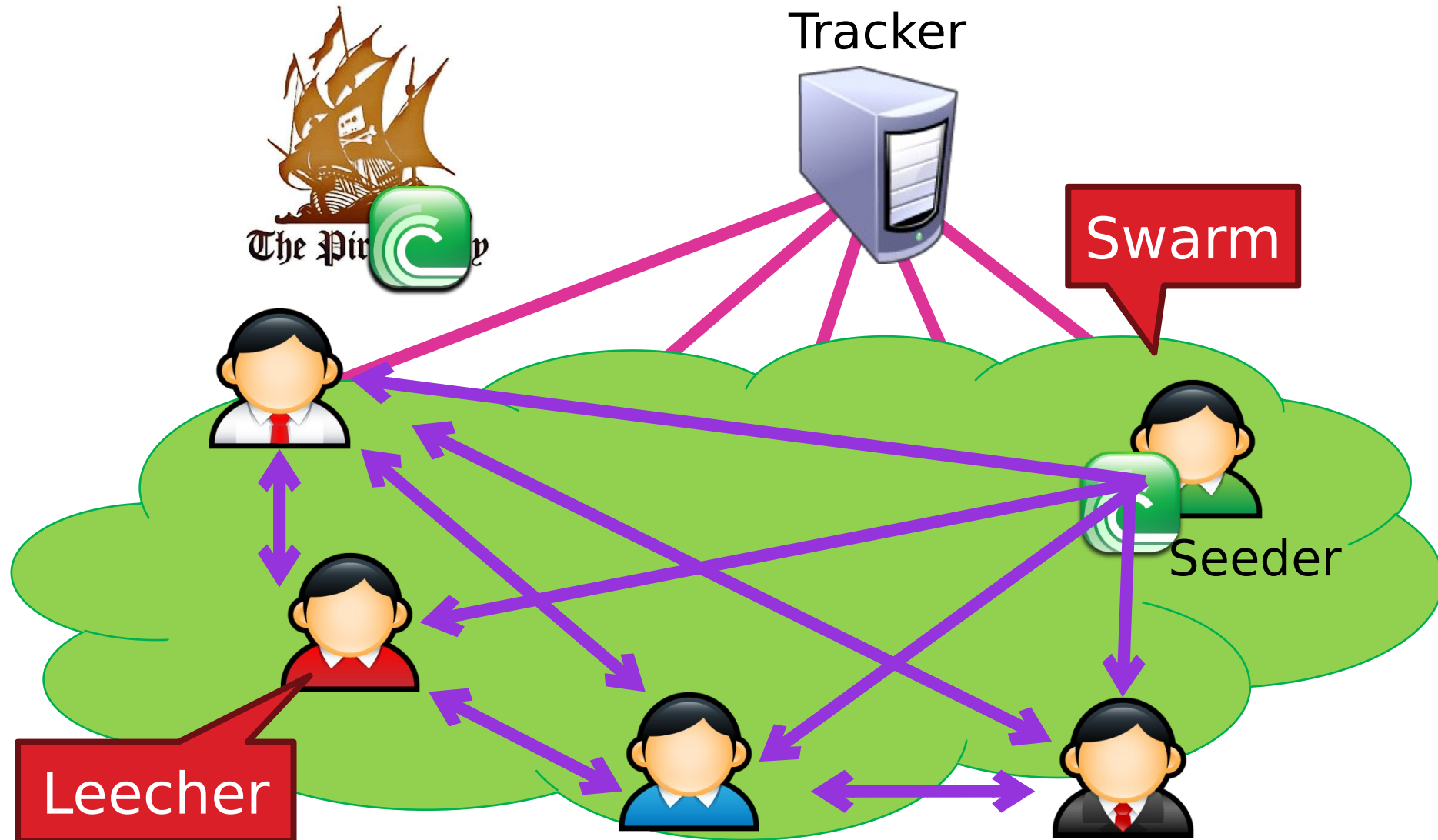


○ What is BitTorrent



- Designed for fast, efficient content distribution
 - Ideal for large files, e.g. movies, DVDs, ISOs, etc.
 - Uses P2P file swarming
- Not a full fledged P2P system
 - Does not support searching for files
 - File swarms must be located out-of-band
 - Trackers acts a centralized swarm coordinators
 - Fully P2P, trackerless torrents are now possible
- Insanely popular
 - 35-70% of all Internet traffic

BitTorrent Overview





○ .torrent File

- Contains all meta-data related to a torrent
 - File name(s), sizes
 - Torrent hash: hash of the whole file
 - URL of tracker(s)
- BitTorrent breaks files into pieces
 - 64 KB – 1 MB per piece
 - .torrent contains the size and SHA-1 hash of each piece
- Basically, a .torrent tells you
 - Everything about a given file
 - Where to go to start downloading

○ Torrent Sites



- Just standard web servers
 - Allow users to upload .torrent files
 - Search, ratings, comments, etc.
- Some also host trackers
- Many famous ones
 - Mostly because they host illegal content
- Legitimate .torrents
 - Linux distros
 - World of Warcraft patches

○ Torrent Trackers

Tracker

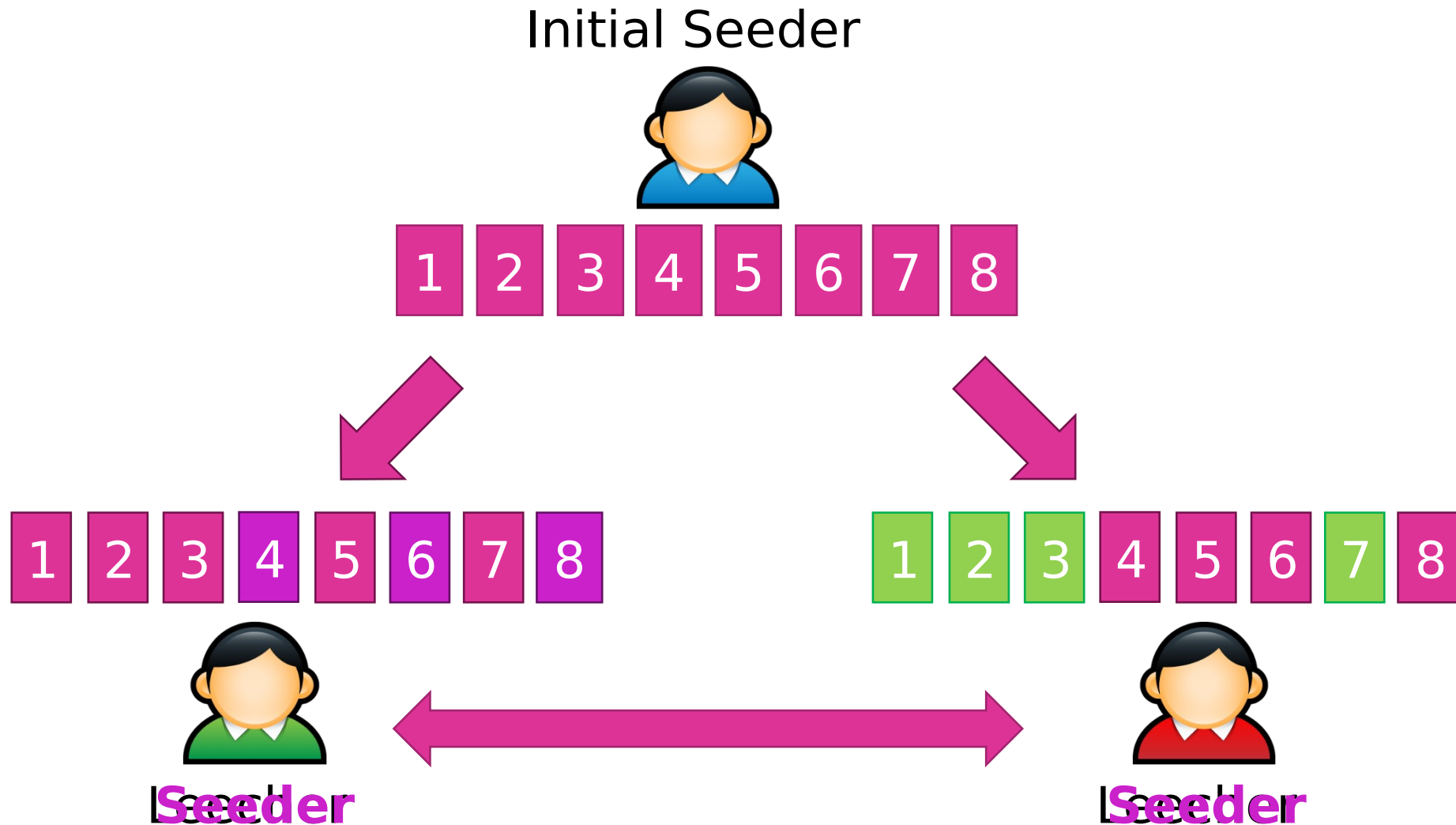


- Really, just a highly specialized webserver
 - BitTorrent protocol is built on top of HTTP
- Keeps a database of swarms
 - Swarms identified by torrent hash
 - State of each peer in each swarm
 - IP address, port, peer ID, TTL
 - Status: leeching or seeding
 - Optional: upload/download stats (to track fairness)
- Returns a random list of peers to new leechers

○ Peer Selection

- Tracker provides each client with a list of peers
 - Which peers are best?
 - Truthful (not cheating)
 - Fastest bandwidth
- Option 1: learn dynamically
 - Try downloading from many peers
 - Keep only the best peers
 - Strategy used by BitTorrent
- Option 2: use external information
 - E.g. Some torrent clients prefer peers in the same ISP

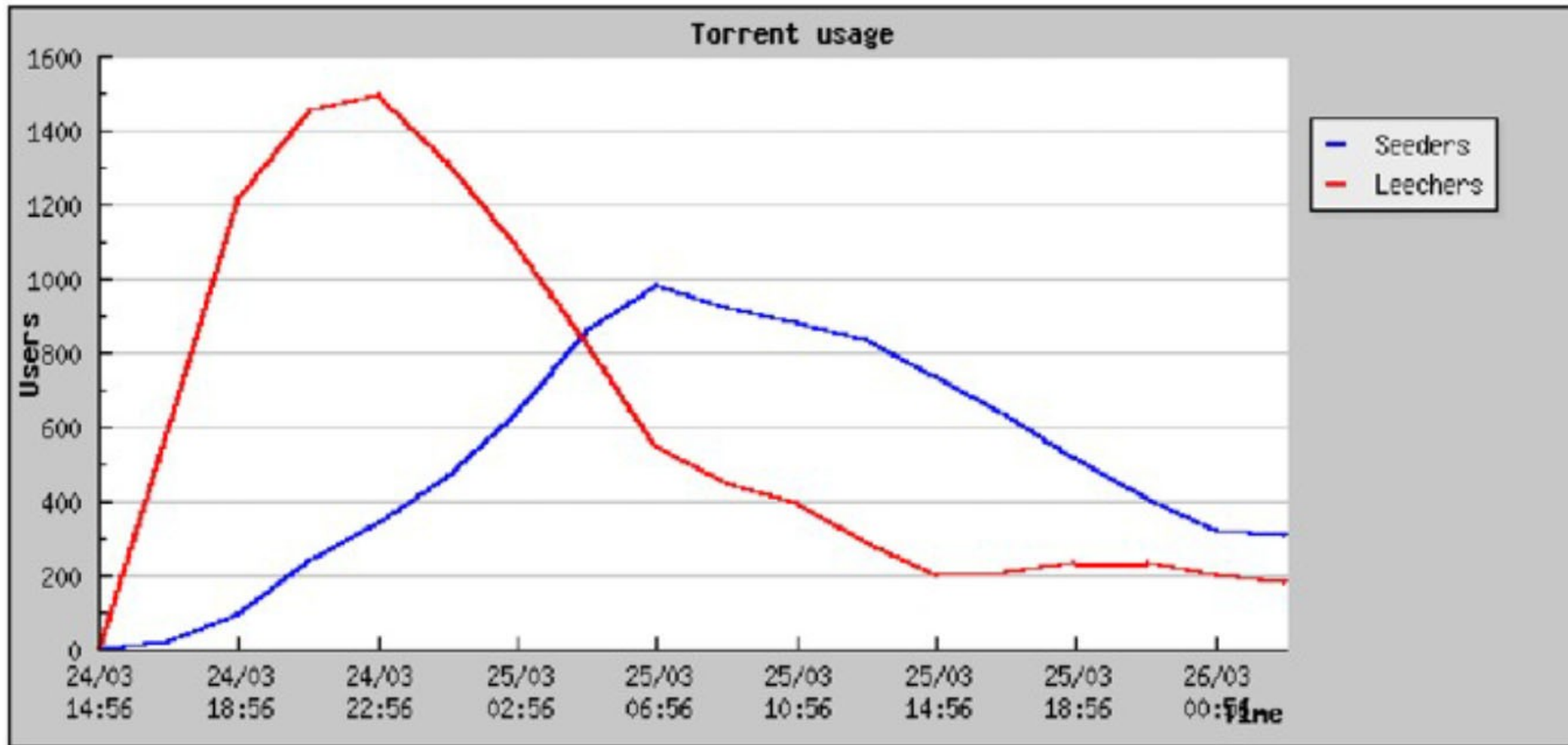
Sharing Pieces



○ The Beauty of BitTorrent

- More leechers = more replicas of pieces
- More replicas = faster downloads
 - Multiple, redundant sources for each piece
- Even while downloading, leechers take load off the seed(s)
 - Great for content distribution
 - Cost is shared among the swarm

Typical Swarm Behavior



○ “Trackerless” BitTorrent

- To be more precise, “BitTorrent without a centralized-tracker”
- E.g.: Azureus
- Uses a Distributed Hash Table (Kademlia DHT)
- Tracker run by a normal end-host (not a web-server anymore)
 - The original seeder could itself be the tracker
 - Or have a node in the DHT randomly picked to act as the tracker



○ BlockChain Technology

- Distributed Ledger
- Cryptocurrencies
- Smart Contracts
- Supply Chain
- E-Voting
- Health Care





JOERI CANT

OCT 05, 2019

Walmart Uses Blockchain Tech to Track Shrimp Supply Chains

United States retail giant Walmart will use blockchain technology to track its Indian-sourced shrimp.

8855 Total views

254 Total shares

Listen to article



2:05





NEWS ANALYSIS

Walmart launches 'world's largest' blockchain-based freight-and-payment network

Walmart Canada's blockchain hits the sweet spot for the supply chain industry as it solves issues such as invoice dispute resolution and real-time data distribution.



By **Lucas Mearian**

Senior Reporter, Computerworld | NOV 19, 2019 3:00 AM PST





IBM Blockchain to Offer Decentralized Smart Contract Option

May 12, 2020 at 08:00 UTC ■ Updated May 12, 2020 at 14:24 UTC



IBM, Walmart Collaborate for FDA Blockchain Pilot Program

June 13, 2019 by Jessica Kent

IBM, Walmart, KPMG, and Merck will collaborate as part of FDA's pilot project to identify, track, and trace prescription medicines using blockchain technology. The program is intended to assist drug supply chain stakeholders in...

94% of Healthcare Execs Say AI, Blockchain Have Advanced Innovation

June 06, 2019 by Jessica Kent

Ninety-four percent of healthcare executives report that artificial intelligence, blockchain, and other emerging technologies have accelerated the pace of innovation over the past three years, according to an Accenture report. However,...

Pistoia Alliance to Use Blockchain for Life Sciences Data Sharing

February 07, 2019 by Jessica Kent

The Pistoia Alliance, a global, non-profit alliance of life sciences stakeholders, has launched the next phase of its blockchain project, in which members will examine how blockchain can improve data identity, data integrity, and data...



References

- <https://slideplayer.com/slide/3299585/>
- BitTorrent
 - “Incentives build robustness in BitTorrent”, Bram Cohen
 - BitTorrent Protocol Specification: <http://www.bittorrent.org/protocol.html>
- Poisoning/Pollution in DHT's:
 - “Index Poisoning Attack in P2P file sharing systems”
 - “Pollution in P2P File Sharing Systems”

