



P2P Systems and Overlay Networks



gnutella.com



Vasilios Darlagiannis
CERTH/ITI
Seminars 2010

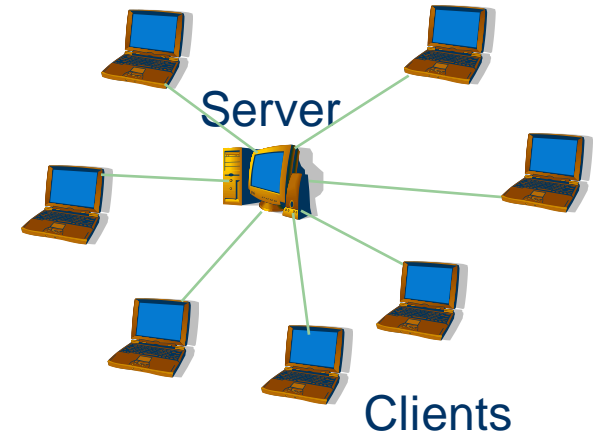


Overview

- Introduction - Motivation
- Design Principles
- Core P2P Operations
- Unstructured Overlays
 - Gnutella
 - ...
- Structured Overlays
 - Chord
 - Omicron
 - ...
- Successful applications
- Discussion on usefulness of the P2P paradigm

Client/Server paradigm

- Limitations
 - Scalability is hard to achieve
 - Presents a single point of failure
 - Requires administration
 - Unused resources at the network edge
- P2P systems try to address these limitations



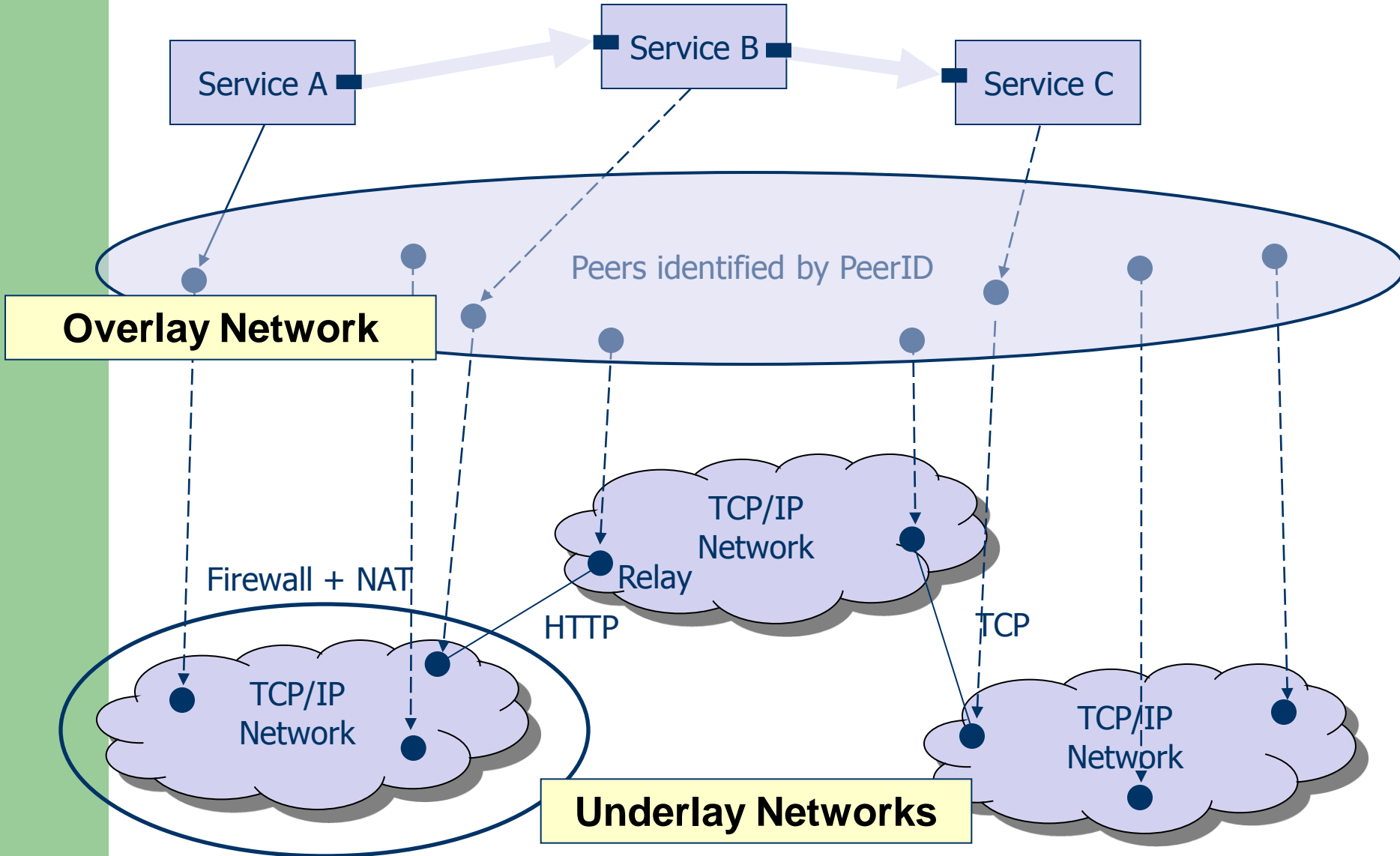
P2P: Definition

- Peer-to-Peer (P2P) is
 - Any distributed network architecture
 - composed of participants that make a portion of their resources directly available to other network participants,
 - without the need for central coordination instances.
- Peers are
 - both suppliers and consumers of resources
 - in contrast to the traditional client–server model where only servers supply, and clients consume.

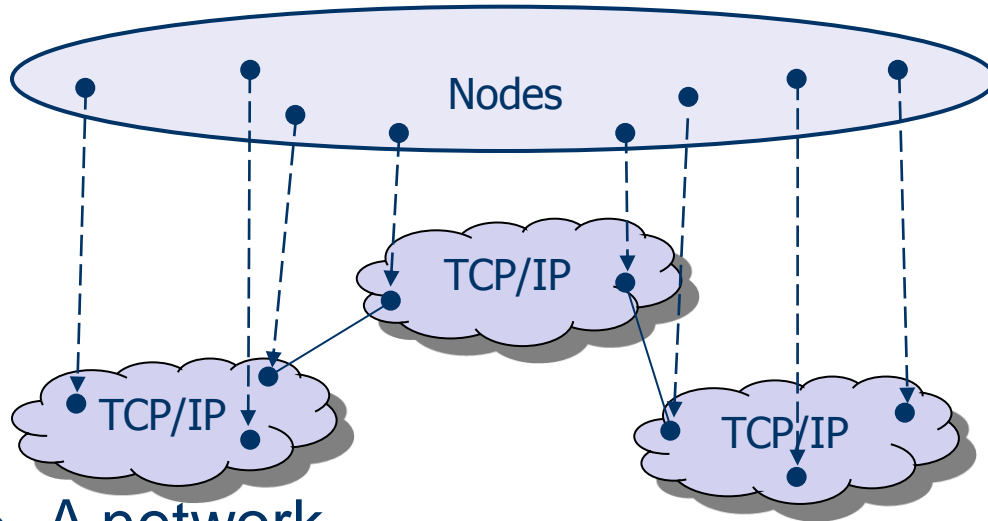
P2P main characteristics

- The concept P2P may refer to:
 - Distributed systems and
 - Communication paradigm
- Main characteristics
 - Systems with loosely-coupled (no fixed relationship), autonomous devices
 - Devices have their own semi-independent agenda
 - Comply to some general rules
 - but local policies define their behavior
 - (At least) limited coordination and cooperation needed
- Key abstraction
 - Application-layer Overlay Networks

P2P Overlay Networks



Overlay Networks



Overlay Network

Underlay Networks

- A network
 - provides services (service model)
 - defines how nodes interact
 - deals with addressing, routing, ...
- Overlay networks
 - built **on top** of one or more existing networks
 - adds an additional layer of
 - abstraction
 - indirection/virtualization

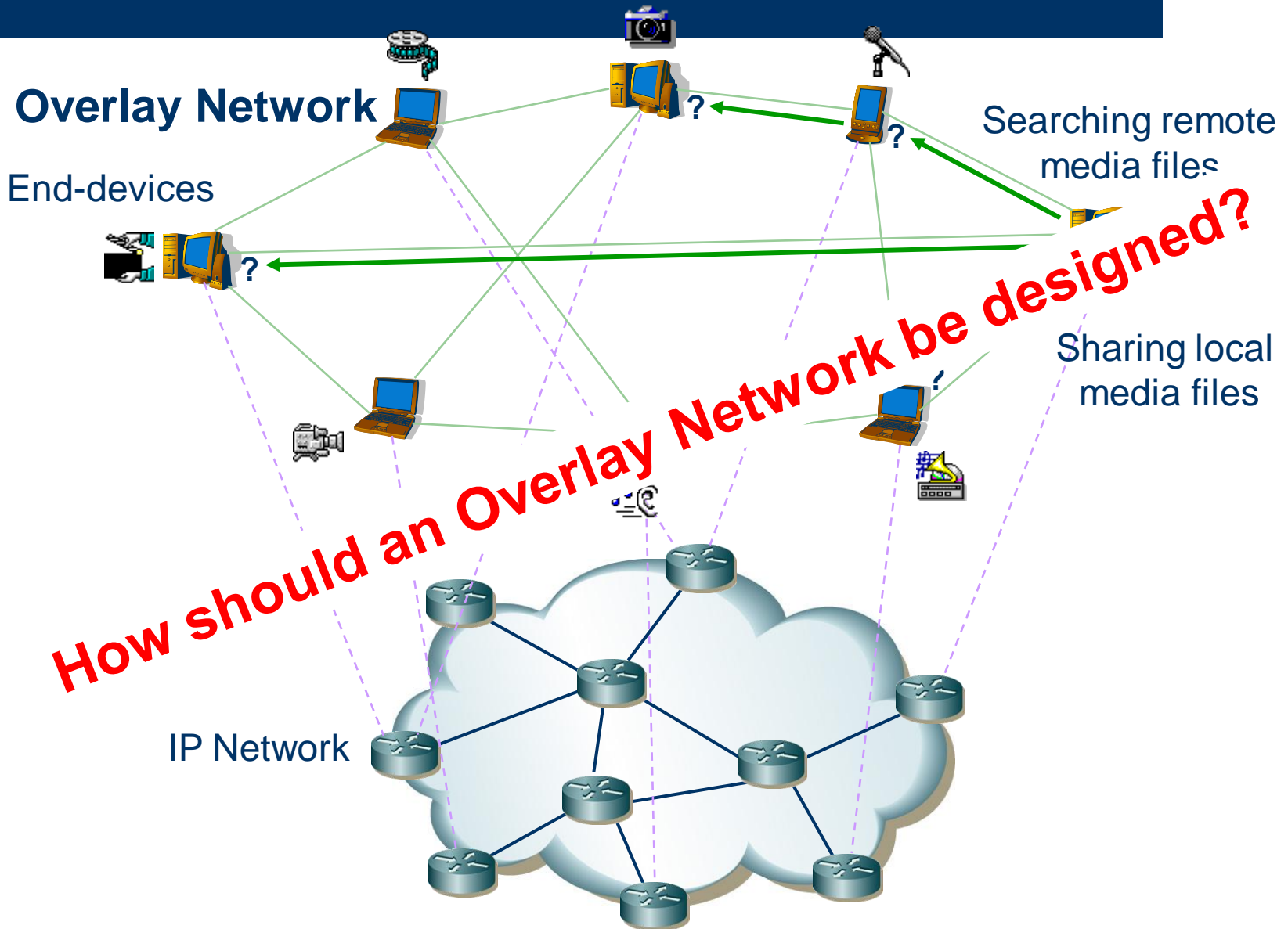
Overlay Networks: Benefits

- Do not have to
 - deploy new equipment
 - modify existing software/protocols
- Allow for easy bootstrapping
 - Make use of existing environment by adding new layer
- Not all nodes must support it
 - Incrementally deployable
- E.g.,
 - adding IP on top of Ethernet does not require modifying Ethernet protocol or driver

Overlay Networks: Drawbacks

- Overhead
 - Adds another layer in networking stack
 - Additional packet headers, processing
- Complexity
 - Layering does not eliminate complexity, it only manages it
 - More layers of functionality
 - more possible unintended interaction between layers
 - Misleading behavior
 - E.g. corruption drops on wireless links interpreted as congestion drops by TCP
- Redundancy
 - Features may be available at various layer
- May provide restricted functionality
 - Some features a “lower layer” does not provide can not be added on top
 - E.g. real-time capabilities (for QoS)

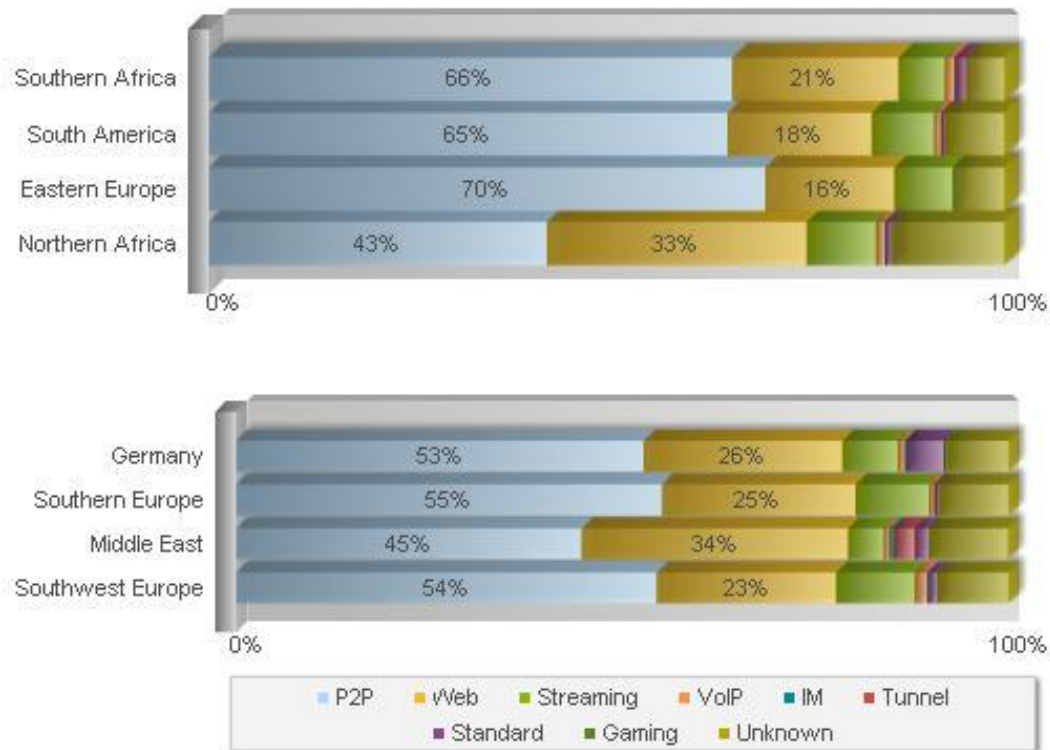
Peer-to-Peer Overlay Networks



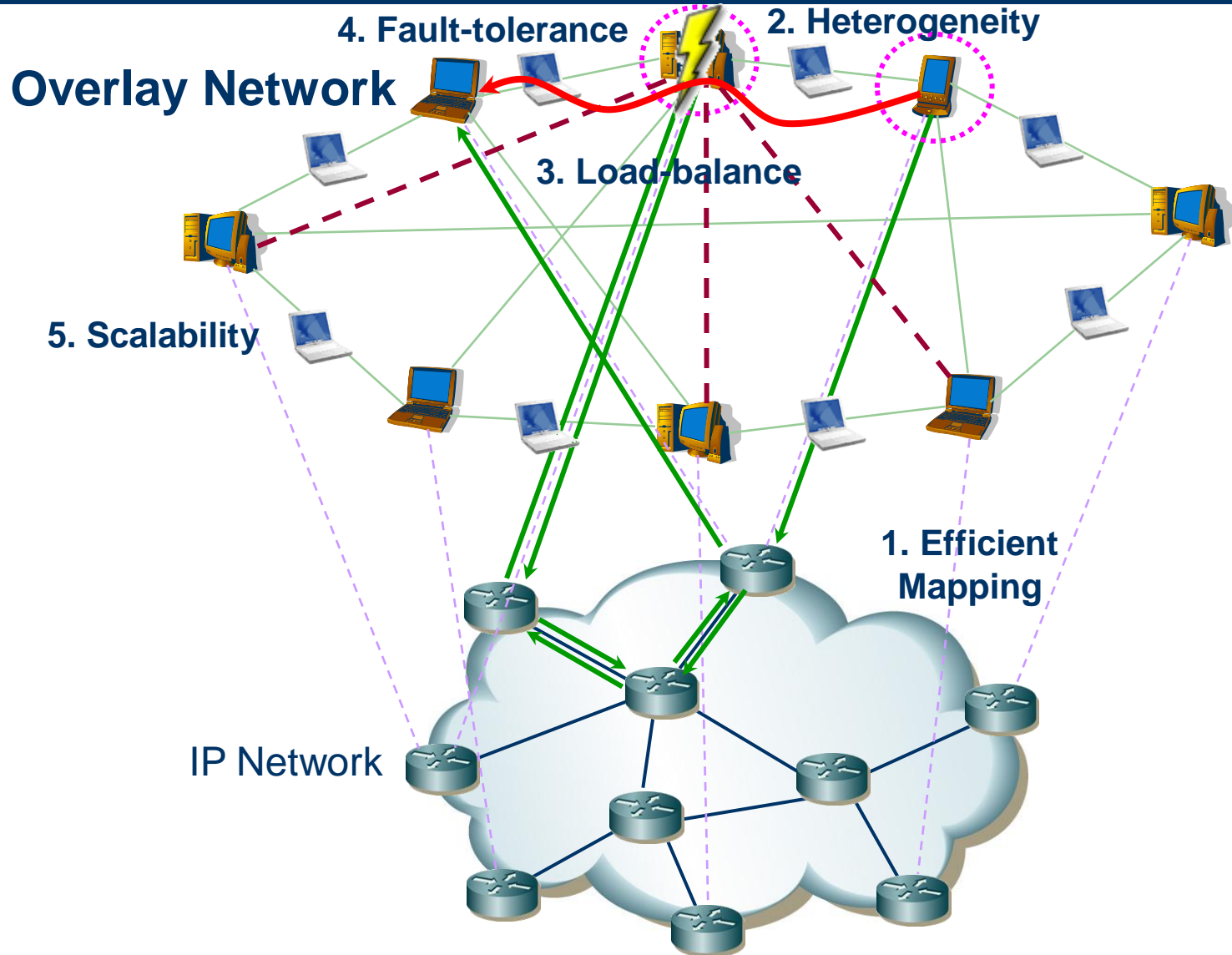
Internet Traffic Study 2008/2009

- P2P generates most traffic in all regions
- Same picture holds the last 9 years
- Changes expected by end of 2010

Distribution of protocol classes



Critical Requirements for Overlays



Properties of P2P Network Graphs

- Ideal network characteristics (general)
 - Small network diameter (worst-case distance)
 - Small average distance / path length
 - Small node degree
 - High connectivity (and high fault tolerance)
 - Support load balancing of traffic
 - Symmetry
- Hard to obtain in reality
 - Trade-offs

Trade-offs in designing Overlays

- Time – Space
 - e.g. local information vs. complete replication of indices
- Security – Privacy
 - e.g. fully logged operations vs. totally untraceable
- Efficiency – Completeness
 - e.g. exact key-based matching vs. range queries
- Scope – Network load
 - e.g. TTL based requests vs. exhaustive search
- Efficiency – Autonomy
 - e.g. hierarchical vs. pure P2P overlays
- Reliability – Low maintenance overhead
 - e.g. deterministic vs. probabilistic operations
- ...

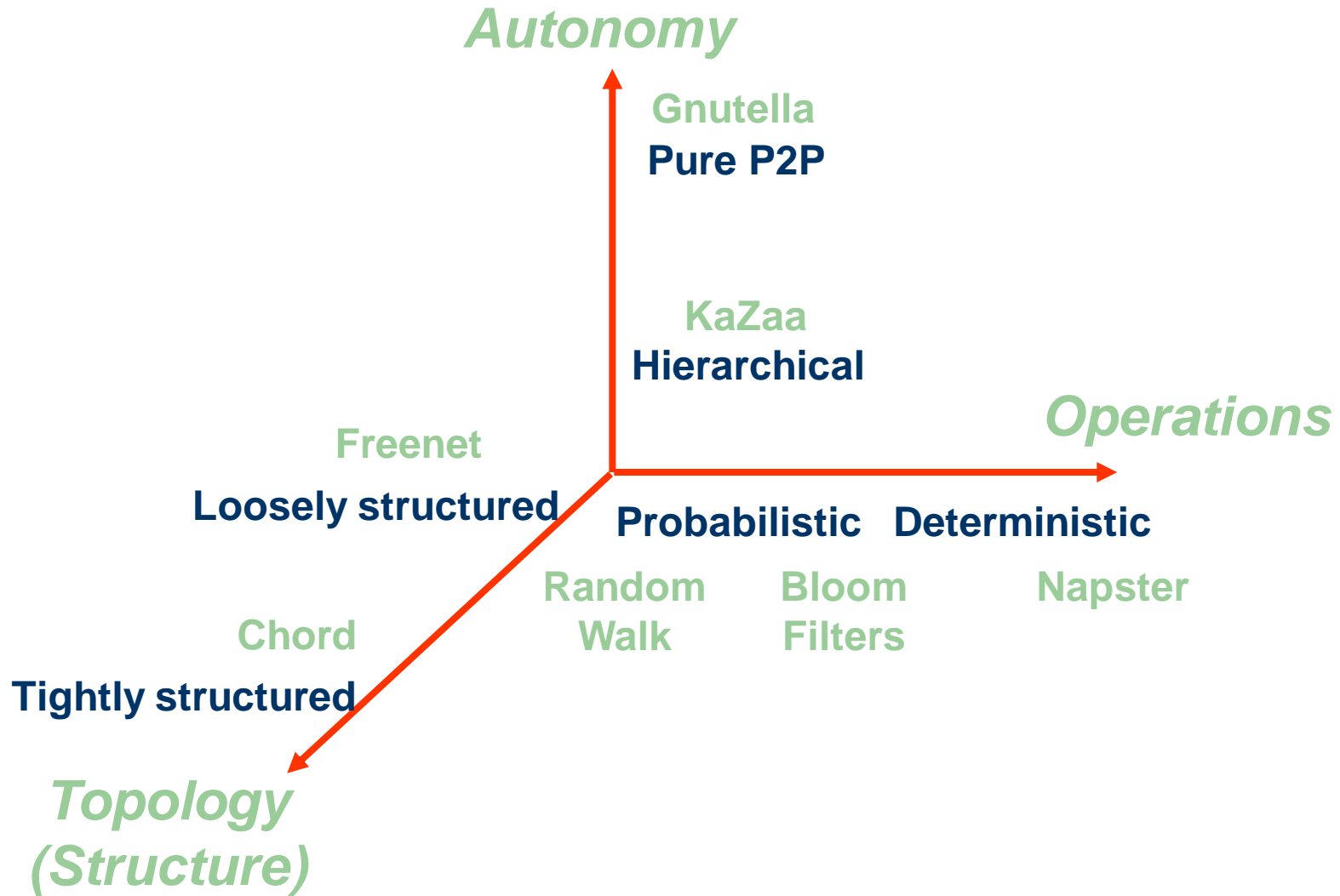
Design mechanisms of P2P Overlays

- Topology structure
 - Loosely structured, tightly structured
- Indexing scheme
 - Distributed Hash Tables (DHTs), Caches, Bloom filters
- Communication paradigms
 - Flooding, random walks, DHT-directed
- Clustering
 - Groups of interest, network proximity, etc.
- Rules/Policies
 - Reputation-, trust-, rate-based
- Roles
 - Service-, operation-based

Overlay Networks Design Approaches

<i>Client-Server</i>	<i>Peer-to-Peer</i>			
<ol style="list-style-type: none"> 1. Server is the central entity and only provider of service and content. → Network managed by the Server 2. Server as the higher performance system. 3. Clients as the lower performance system <p>Example: WWW</p>	<ol style="list-style-type: none"> 1. Resources are shared between the peers 2. Resources can be accessed directly from other peers 3. Peer is provider and requestor (Servent concept) 			
	<i>Unstructured P2P</i>			<i>Structured P2P</i>
	<i>Centralized P2P</i>	<i>Pure P2P</i>	<i>Hybrid P2P</i>	<i>DHT-Based</i>
	<ol style="list-style-type: none"> 1. All features of Peer-to-Peer included 2. Central entity is necessary to provide the service 3. Central entity is some kind of index/group database <p>Example: Napster</p>	<ol style="list-style-type: none"> 1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → no central entities <p>Example: Gnutella 0.4, Freenet</p>	<ol style="list-style-type: none"> 1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → dynamic central entities <p>Examples: Gnutella 0.6, Fasttrack, edonkey</p>	<ol style="list-style-type: none"> 1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities 4. Connections in the overlay are “fixed” 5. Distributed indexing (content is not relocated) <p>Examples: Chord, CAN</p>

Classification of P2P design mechanisms



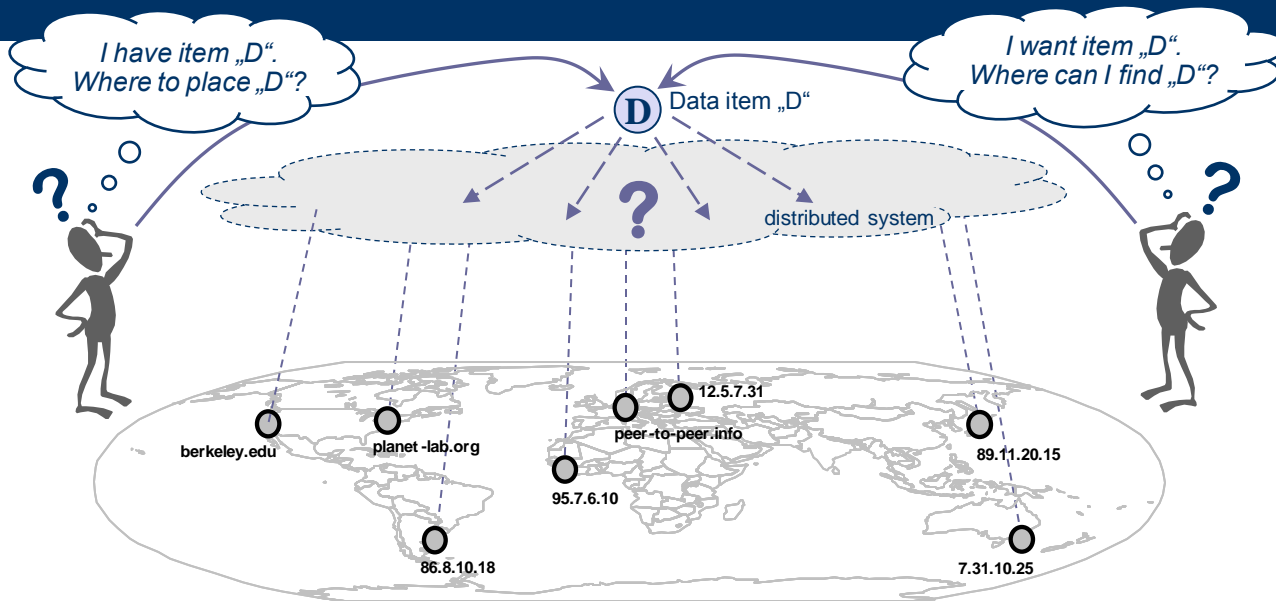
Components of P2P Overlays

- Identification scheme
 - Nodes, resources, services, clusters, etc.
- Routing tables
 - Size, selection of entries, complexity
- Indexing structure
 - Compressed, cached, complete, semantics support
- Communication protocols
 - Recursive, iterative

P2P Core Functionality


- Infrastructure-less connectivity
- Dynamic network management
- Sharing of services and resources
- Management of shared resources
- Load balancing
- Finding shared services and resources

Data Management and Retrieval



Essential challenge in (most) Peer-to-Peer systems?

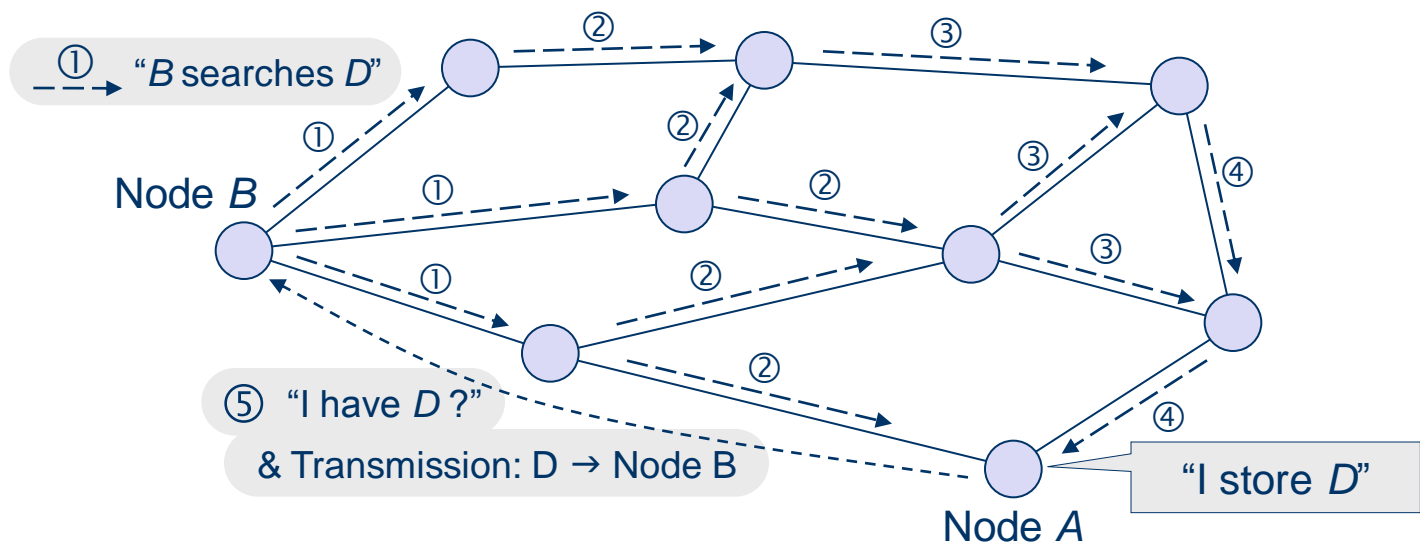
- Location of a data item among systems distributed
 - Where shall the item be stored by the provider?
 - How does a requester find the actual location of an item?
- Scalability:
 - keep the complexity for communication and storage scalable
- Robustness and resilience
 - in case of faults and frequent changes



Finding Information in Unstructured P2P systems

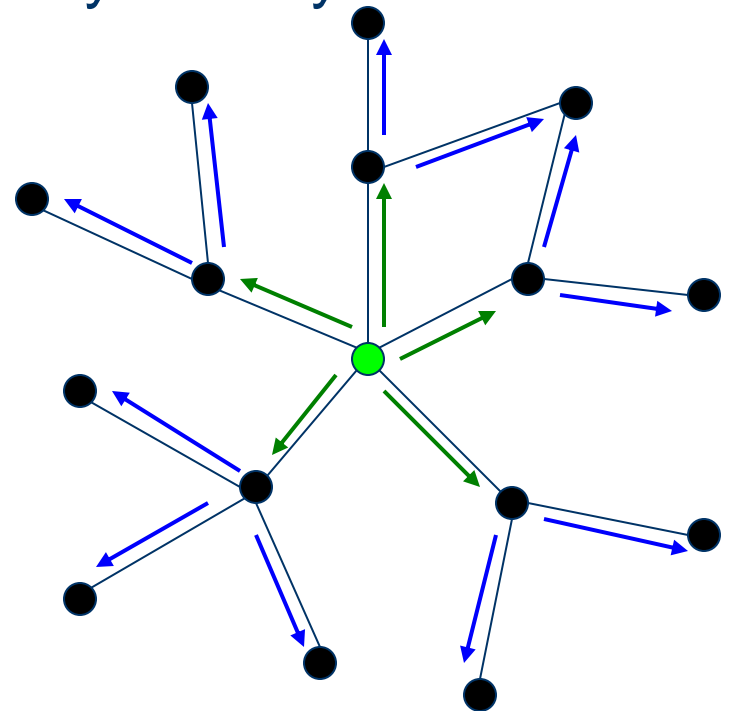
Flooding Search

- Fully Decentralized Approach: Flooding Search
 - No information about location of data in the intermediate nodes
 - Necessity for broad search
 - ① Node B (requester) asks neighboring nodes for item D
 - ②-④ Nodes forward request to further nodes (breadth-first search / flooding)
 - ⑤ Node A (provider of item D) sends D to requesting node B



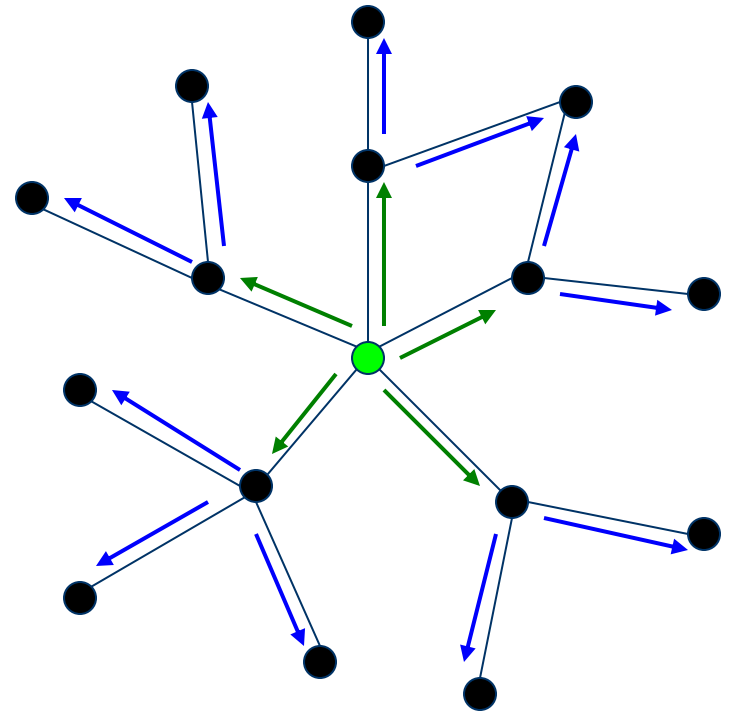
Gnutella: Protocol 0.4 – Characteristics

- Message broadcast for node discovery and search requests
 - flooding
 - (to all connected nodes) is used to distribute information
 - nodes recognize message they already have forwarded
 - by their GUID and
 - do not forward them twice
- Hop limit by TTL
 - originally TTL = 7



Expanding Ring

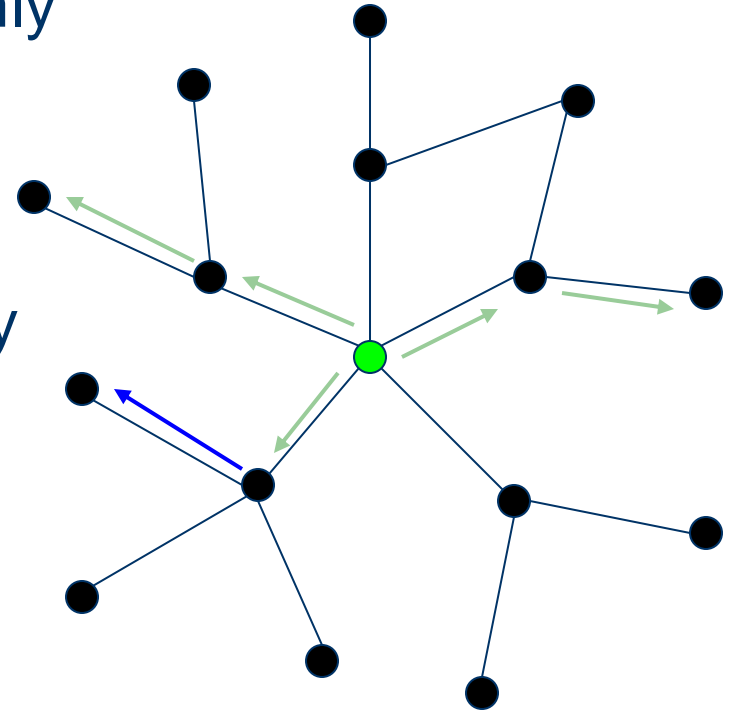
- Mechanism
 - Successive floods with increasing TTL
 - Start with small TTL
 - If no success increase TTL
 - .. etc.
- Properties
 - Improved performance when objects follow Zipf law popularity distribution and located accordingly
 - Message overhead is high



Random Walk

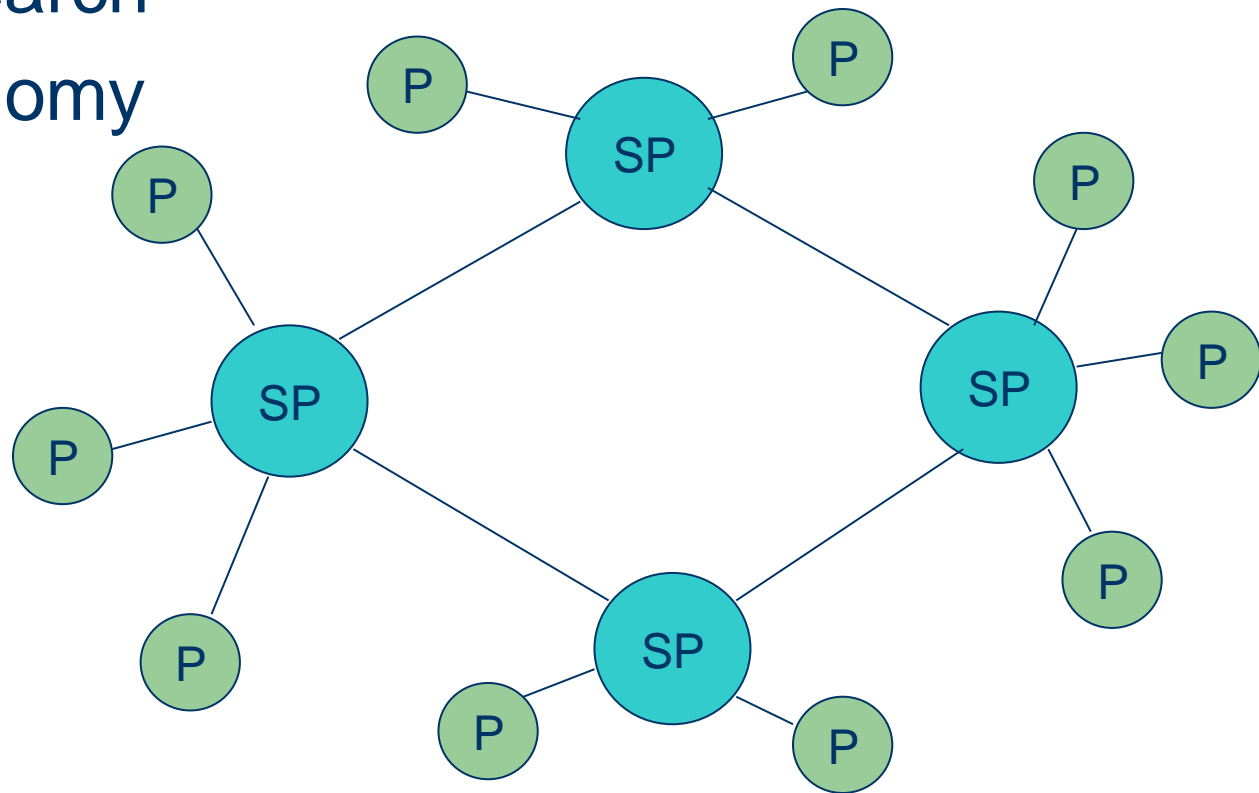
Algorithm and variations

- Forward the query to a randomly selected neighbor
 - Message overhead is reduced significantly
 - Increased latency
- Multiple random walks (k-query messages)
 - reduces latency
 - generates more load
- Termination mechanism
 - TTL-based
 - Periodically checking requester before next submission



Hierarchical (Super-peer) Overlays

- Utilized in Gnutella 0.6, KaZaA, eDonkey, etc.
- Consider non-uniform distributions
- Efficient search
- Less autonomy



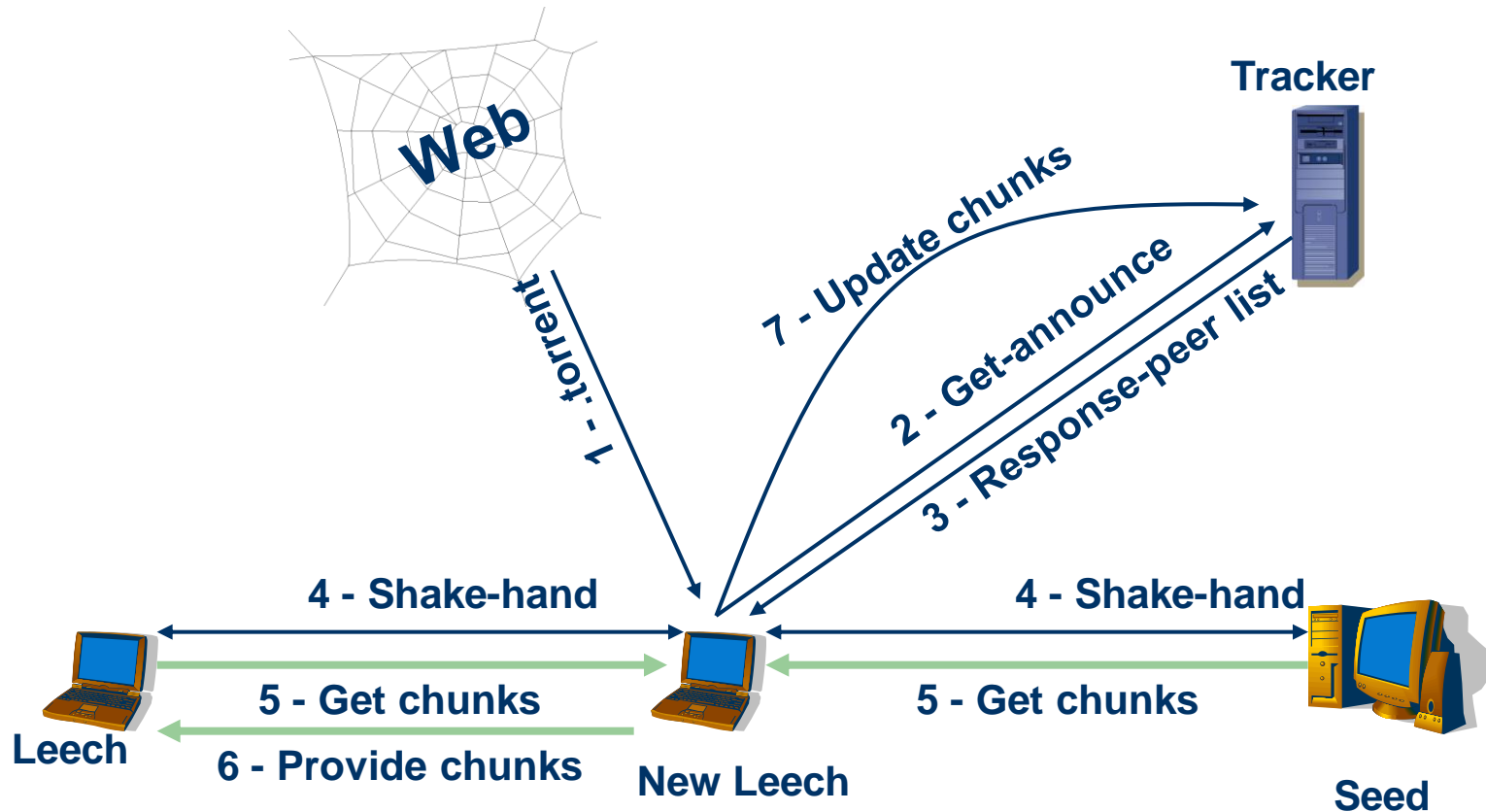
File sharing with BitTorrent

- Cooperative File Sharing
 - Counter free-riders
- Characteristics
 - no virtual currency
 - file is split into chunks
 - tit-for-tat exchange strategy
 - if give you – you give me
 - attempt to reach Pareto efficiency
 - no one can get faster download speeds without hurting someone else's download speed
 - nodes download rarest chunks first
 - new nodes download random chunks first

BitTorrent Concepts

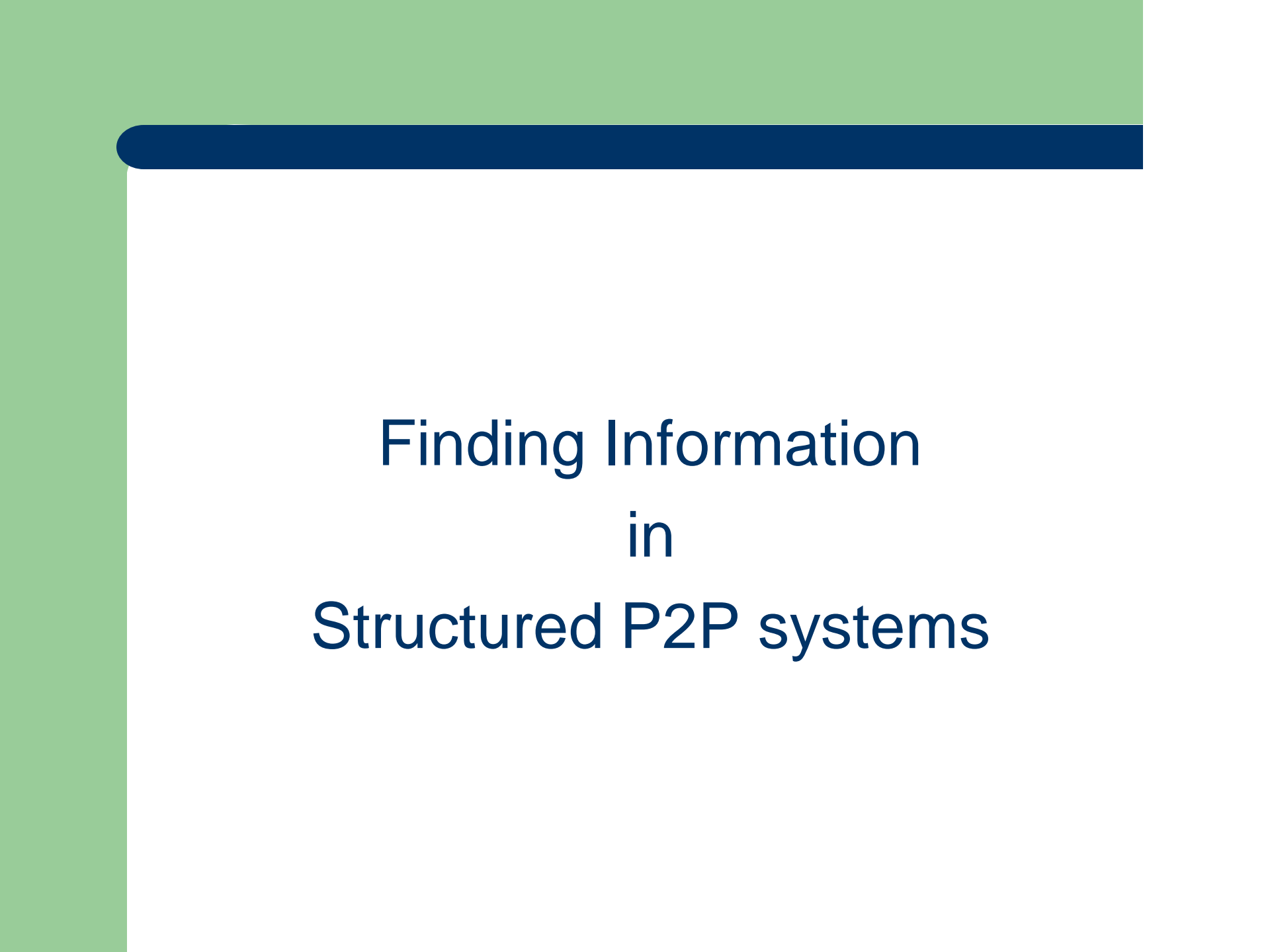
- Torrent:
 - group of peers exchanging chunks of a file
- For each shared file
 - tracker
 - non-content-sharing node
 - actively tracks all seeders and leeches
 - seeders
 - have complete copies of the desired content
 - leeches
 - incomplete copies of the desired content
 - leeches try to download missing chunks

BitTorrent: Operation Scenario



BitTorrent: Evaluation

- Strengths
 - Good bandwidth utilization
 - Limit free riding – tit-for-tat
 - Limit leech attack – coupling upload & download
 - Preferred selection for legal content distribution
- Drawbacks
 - Small files – latency, overhead
 - Central tracker server needed to bootstrap swarm
 - Single point of failure
 - Potentially a scalability issue
 - Robustness
 - System progress dependent on altruistic nature of seeds (and peers)
 - Cannot totally avoid malicious attacks and leeches

The slide features a light green background with a dark blue horizontal bar at the top and a dark blue vertical bar on the left side. The text is centered in a dark blue font.

Finding Information in Structured P2P systems

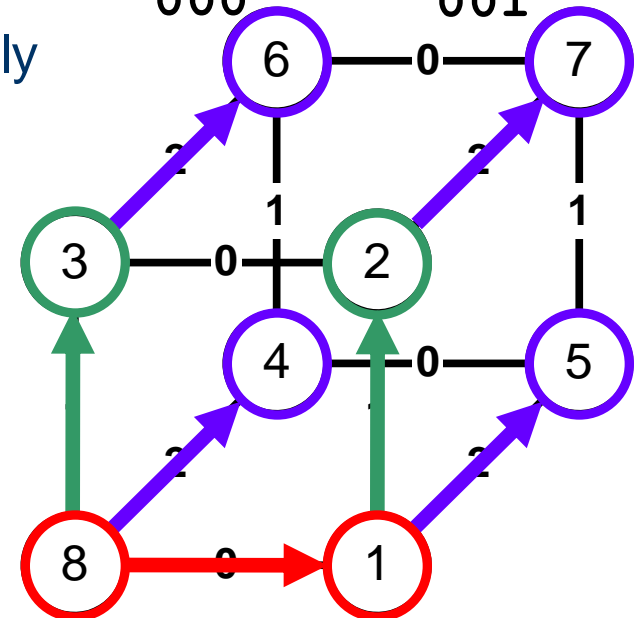
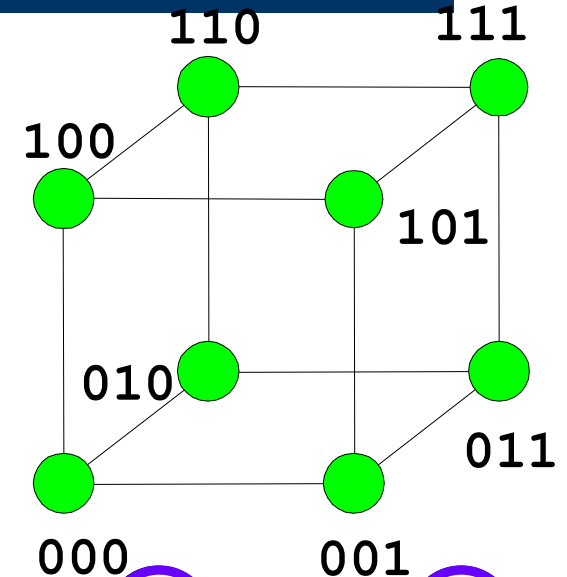
Structured Overlay Networks

- Structured (tightly structured) network topologies
 - Hypercubes
 - De Bruijn
 - Butterflies
 - Meshes,.....

 - DHTs
 -
- Topologies are also met in traditional distributed and parallel systems
 - Different requirements than P2P systems

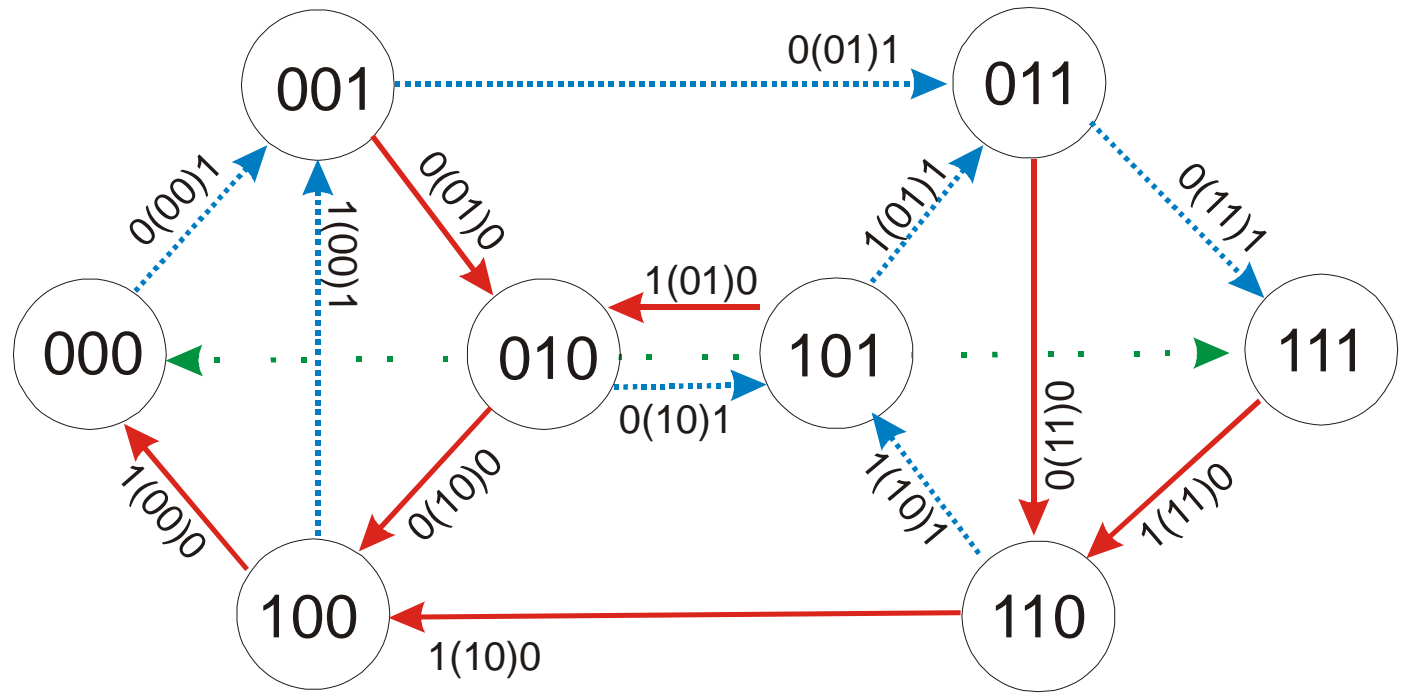
Hypercube Topology

- n-dimensional binary hypercube
 - (or n-cube)
 - 2^n vertices labeled by n-bit binary strings
 - Edges joining two vertices whenever their labels differ in a single bit
- Characteristics
 - Vertex degree grows logarithmically with the number of vertices
 - Logarithmic growth of the diameter
 - Vertex- and edge-symmetric

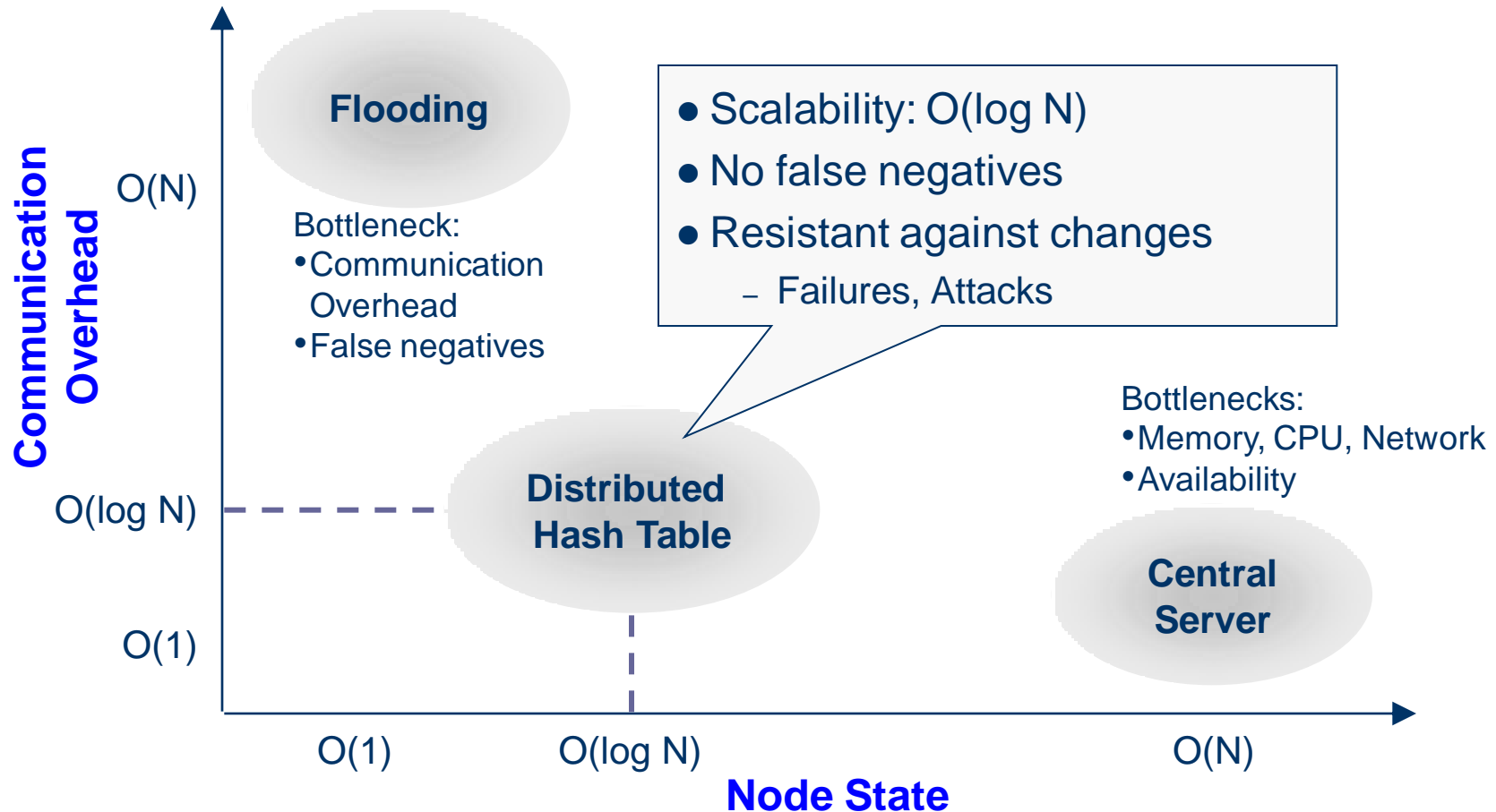


De Bruijn graphs

- Lexicographic graphs
 - Adjacency is based on left shift by 1 position
 - E.g. node 001 points to nodes 01x (010, 011)
- Characteristics
 - Average distance is very close to the diameter
 - Constant vertex degree
 - Logarithmic diameter



Distributed Indexing



DHT: Addressing Space

Mapping of content/nodes into linear space

- Usually: $0, \dots, 2^m-1 \gg$ number of objects to be stored
- Mapping of data and nodes into an address space (with hash function)
 - E.g., $\text{Hash}(\text{String}) \bmod 2^m: \text{H}(\text{„my data“}) \rightarrow 2313$
- Association of parts of address space to DHT nodes

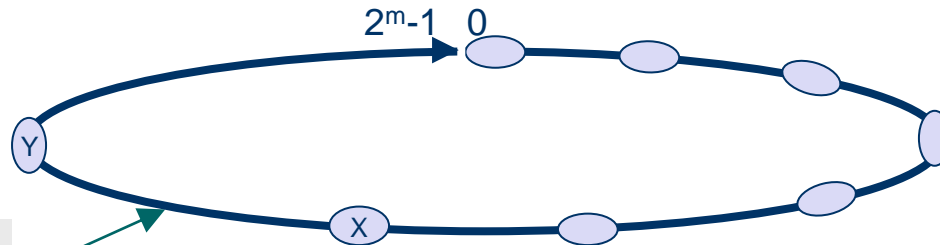


$H(\text{Node Y}) = 3485$

Data item "D":
 $H(\text{"D"}) = 3107$

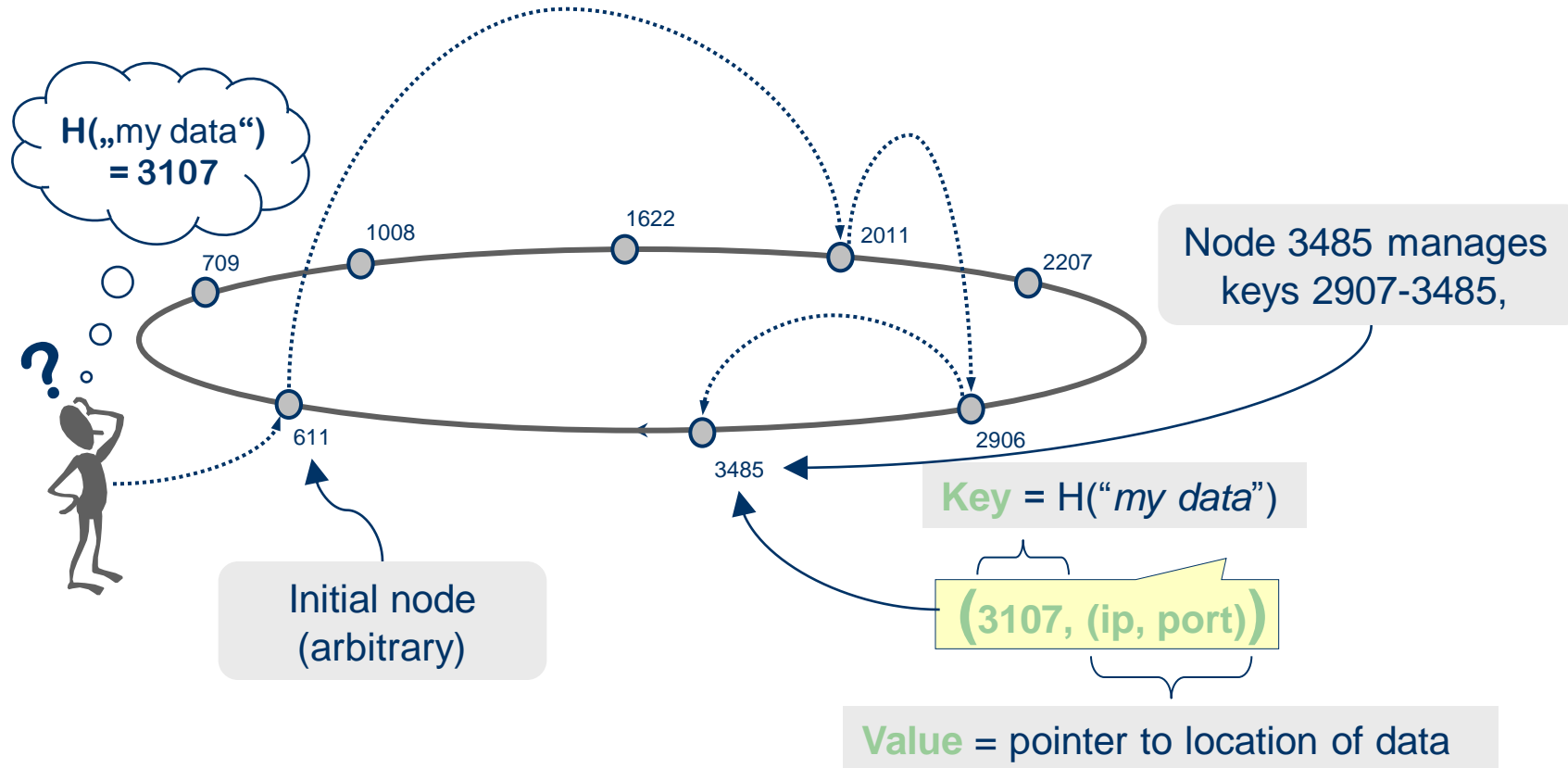
$H(\text{Node X}) = 2906$

Often, the address space is viewed as a circle.



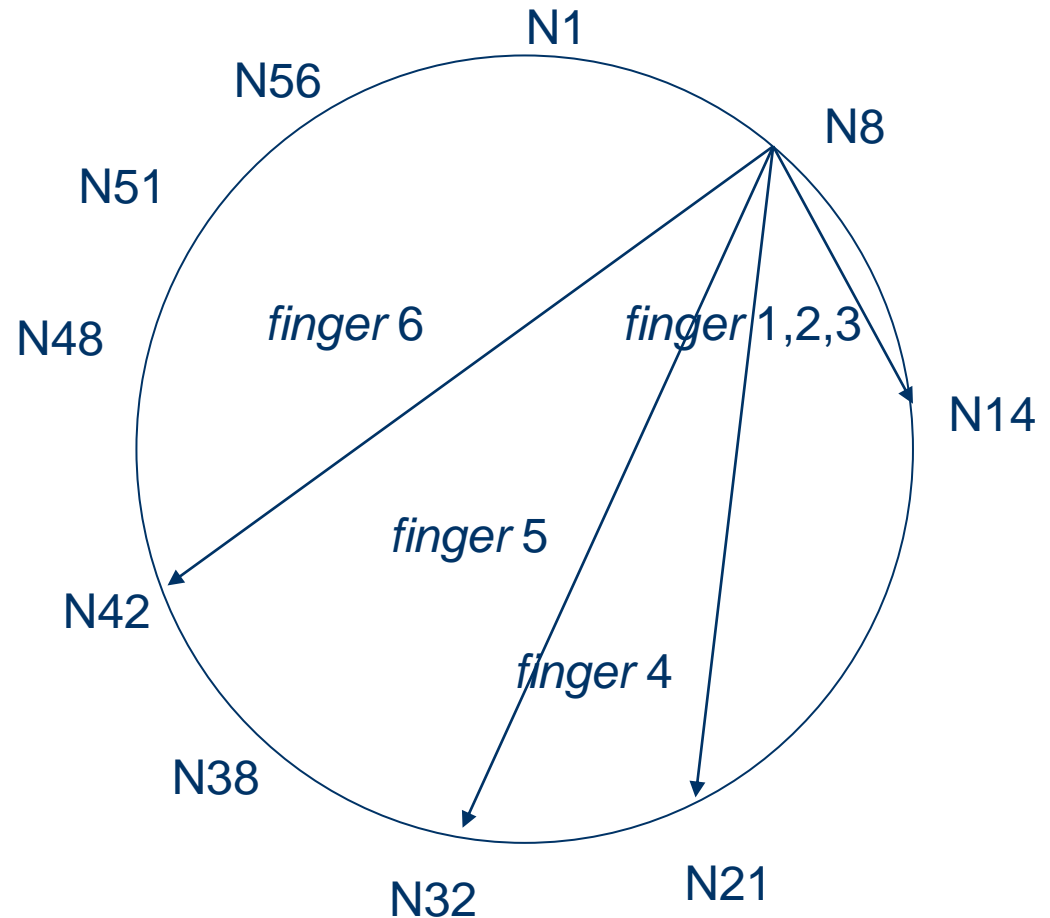
DHT: Routing to destination

- Hash(query)
- Use shortcuts to reach destination in minimum steps (typically $O(\log(n))$)



Chord: Ring-based DHT

- Build $\log(n)$ fingers
- $\text{finger}[k] = \text{first node that succeeds } (n+2^{k-1}) \bmod 2^m$
- Ring invariant must hold



DHT Desirable Properties

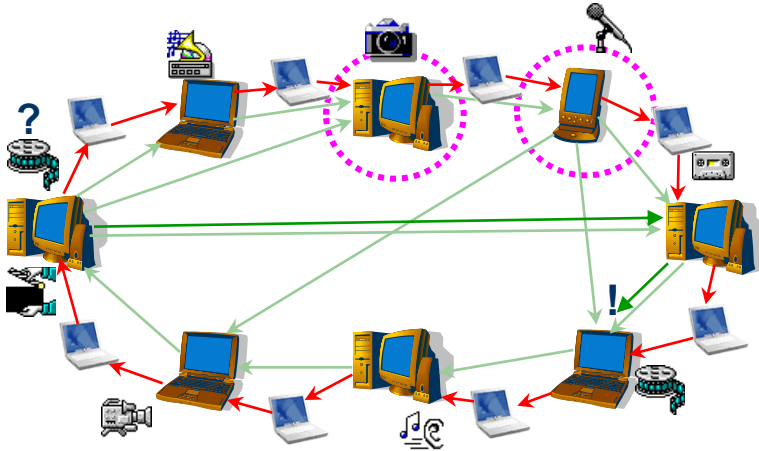
- Keys should be mapped evenly to all nodes in the network (load balance)
- Each node should maintain information about only a few other nodes (scalability, low update cost)
- Messages should be routed to a node efficiently (small number of hops)
- Node arrival/departures should only affect a few nodes

DHTs: Core Components

- Hash table
 - Uniform distribution
 - Shifted view for each node (adding a node-related offset)
- Mapping function
 - Node Ids and item keys share the same key-space
 - Rules for associating keys to particular nodes
- Routing tables
 - Per-node routing tables that refer to other nodes
 - Rules for updating tables as nodes join and leave/fail
- Routing algorithms (operations on keys):
 - XOR-based (e.g. Kademlia)
 - Shift operations (e.g. D2B)
 - Distance-based (e.g. Chord)
 - Prefix-based (e.g. Pastry)

Motivation for Omicron

Distributed Hash Tables (DHTs)



Issues

- Heterogeneity
- Maintenance cost

● Goal

- Design of an effective P2P Overlay Network
- Merge Super-Peer and DHT properties

● Challenge

- Handle efficiently the large number of conflicting requirements, e.g.
- Heterogeneity versus load-balance

Hierarchical Networks



Issues

- Potential bottlenecks
- Fault-tolerance

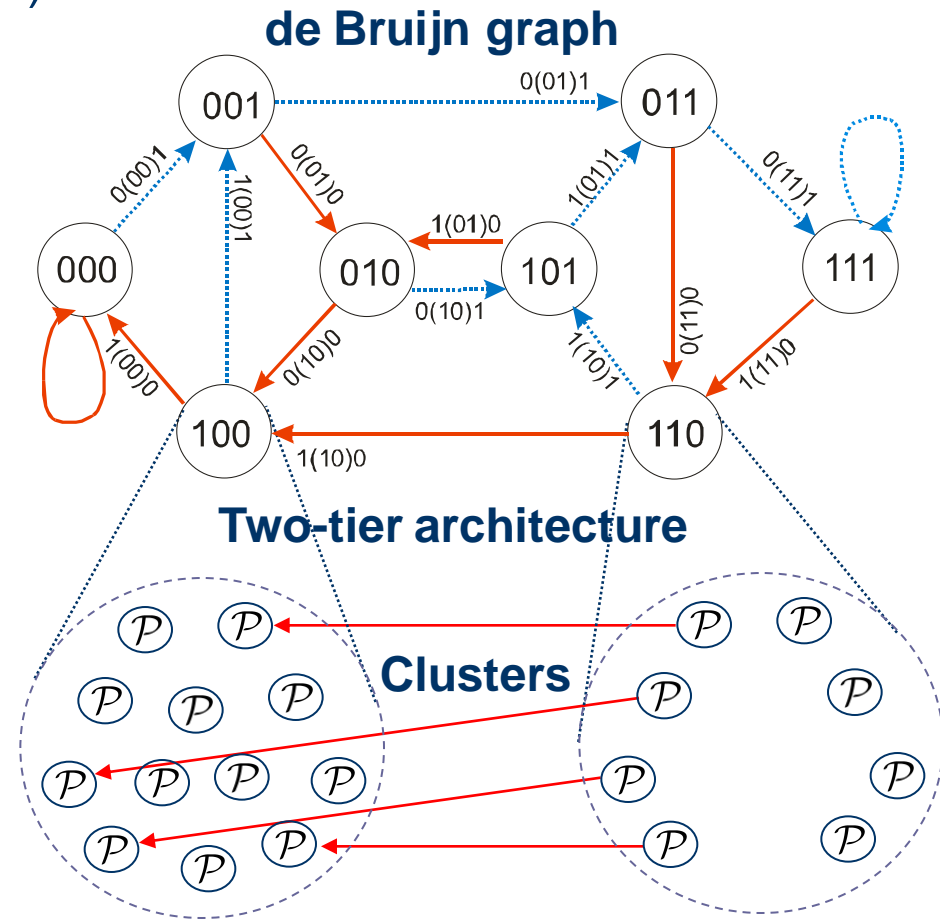
Omicron: Two-tier Overlay

Structured macro level (de Bruijn)

- Scalable
 - Asymptotically optimal
 - Diameter
 - Average node distance
 - Fixed node degree
- Stable nodes are necessary

Clustered micro level

- Redundancy and fault-tolerance
- Locality aware
- Finer load balance
- Handling hot spots



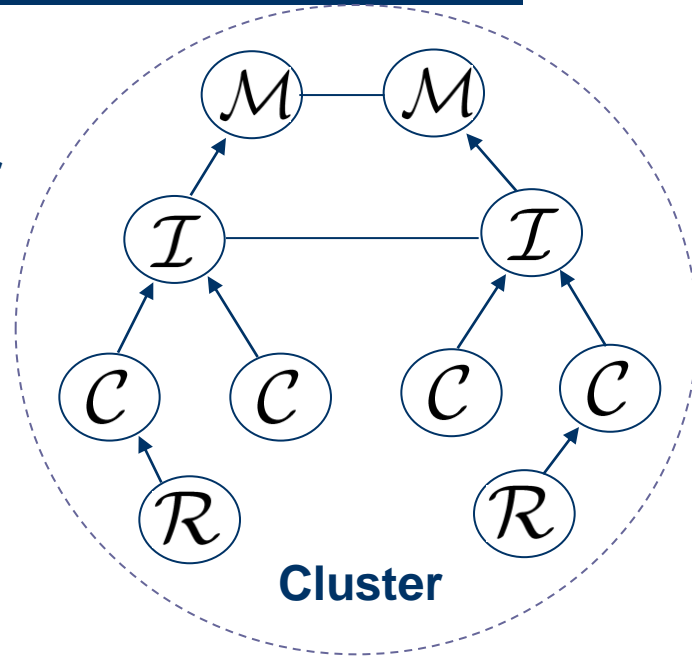


Ⓜ **Maintainer**

Ⓢ **Indexer**

Ⓒ **Cacher**

Ⓡ **Router**



- **Common overlay network operations**

- Maintaining structure (topology)
- Routing queries
- Indexing advertised items
- Caching popular items

Organized
Maintenance,
Indexing,
Caching and
Routing in
Overlay
Networks

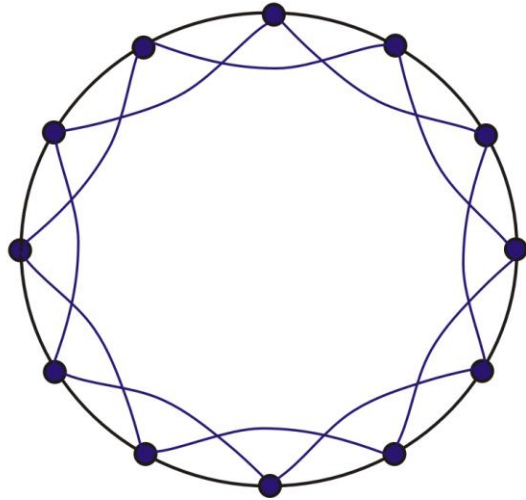
Fuzzynet: Motivation

Motivation

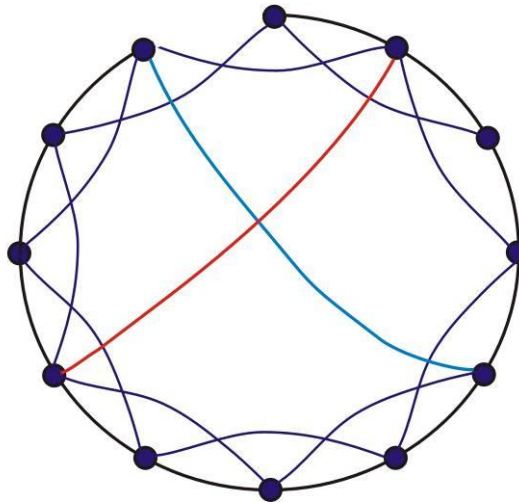
- Advantages of the ring
 - Easy Navigation (greedy routing)
 - Clear responsibility ranges
 - Easy to bootstrap long-range links
- BUT!
 - Keeping the ring invariant is a difficult task:
 - Expensive maintenance (periodic, eager)
 - Non-transitivity effect ($A \rightarrow B$, $B \rightarrow C$, but not $A \rightarrow C$)
 - Firewalled peers, NATs
 - Routing anomalies

Small-World Graphs (Networks)

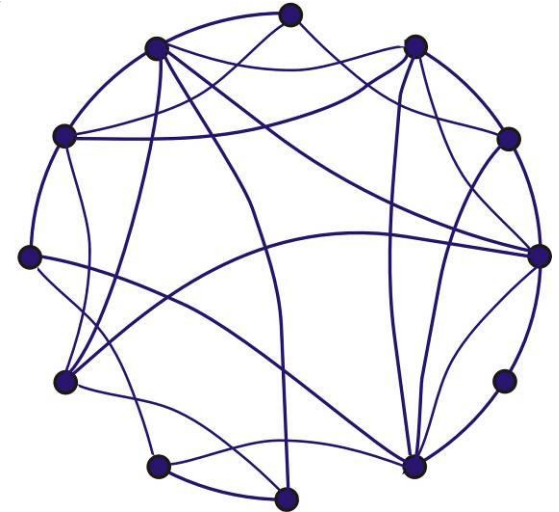
- Regular Graph



- slightly "rewired"



- Random Graph



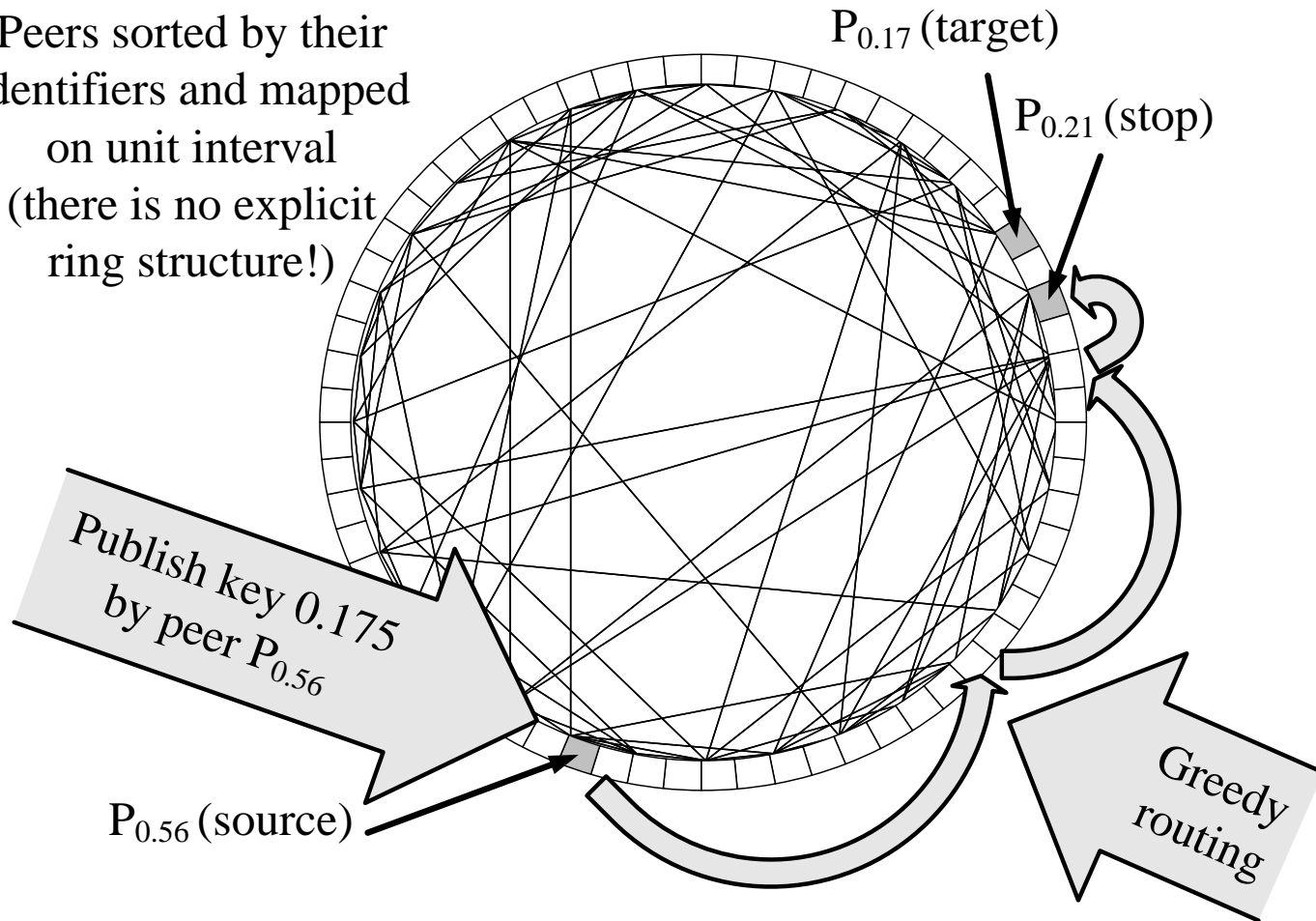
	Regular Graph	Slightly rewired graph	Random graph
Clustering Coefficient	high	high	low
Path Length	high	low	low

Fuzzynet: Zero maintenance ringless overlay (2)

- Fuzzynet
 - No ring structure (only Small-World “long-range” links)
 - No predefined responsibility ranges
 - Data is probabilistically stored in the data-key vicinity
 - Compatible with any Small-World network
 - Typical DHT replication rate
 - Network construction without the help of the ring (peer order is considered)
 - Lookup (Read) – simple greedy routing
 - Publish (Write) – greedy routing + write burst

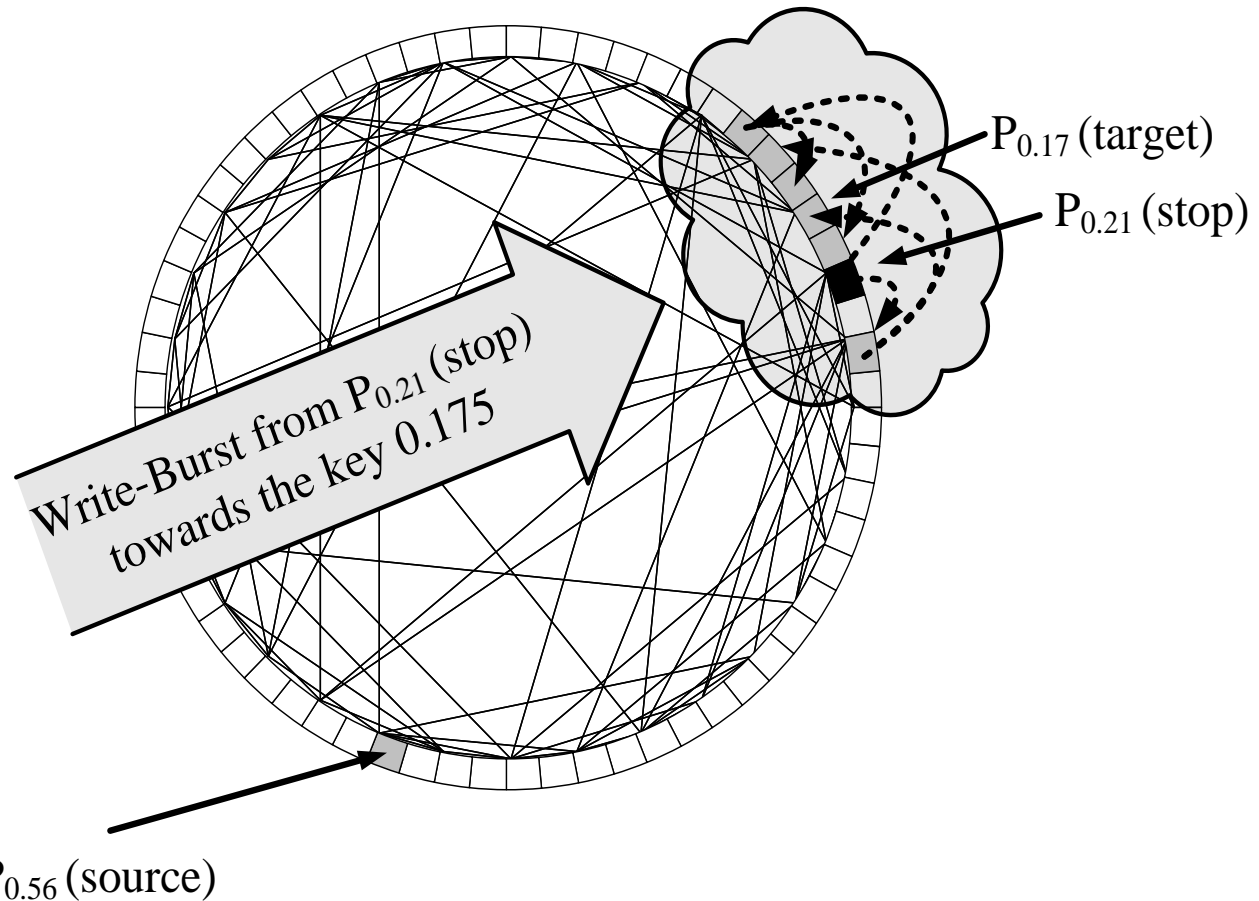
Write phase 1: Greedy-Approach

Peers sorted by their identifiers and mapped on unit interval (there is no explicit ring structure!)



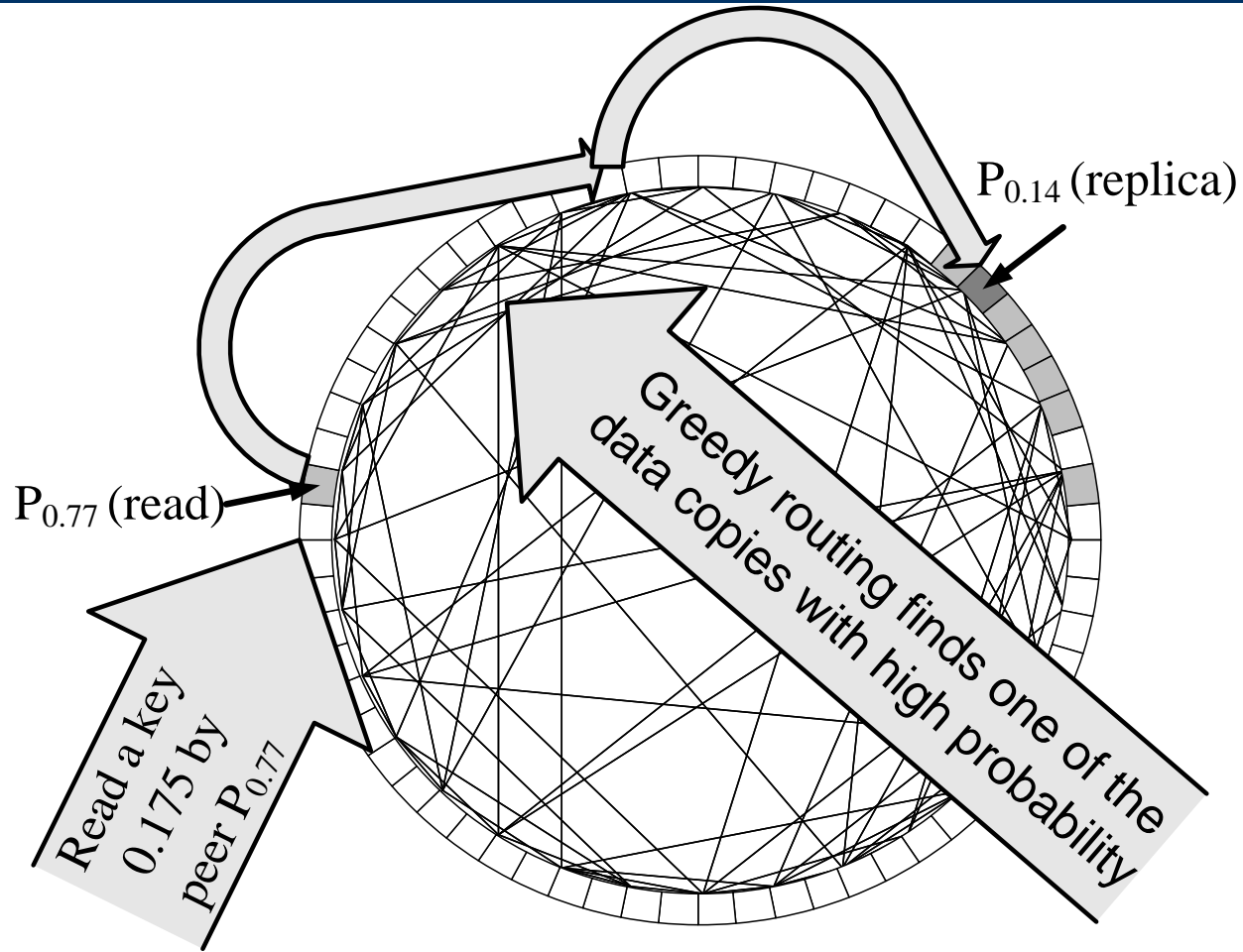
- Routing from the originator peer ($P_{0.56}$) to the *greedy-closest* peer ($P_{0.21}$) where the greedy approach towards the target key 0:175 (*actual-closest* peer $P_{0.17}$) is no further possible.

Write phase 2: Write-Burst



- The *greedy-closest* peer ($P_{0.21}$) seeds the replicas in the cluster vicinity of the key 0.175 using the Write-Burst.

Lookup (read)



- After writing the data in the vicinity of the key 0.175, the lookup (read) from any node will have very high chance finding at least one of the data replicas.

P2P Application & Service Domains

File Sharing: music, video and other data

- Napster, Gnutella, FastTrack (KaZaA, ...), eDonkey, eMule, BitTorrent, eXeem, etc.

Distributed Storage/Distributed File sharing

- (Anonymous) Publication: Freenet
- PAST, OceanStore, etc.

Collaboration

- P2P groupware
- Groove
- P2P content generation
- Online Games
- P2P instant messaging

Distributed Computing - GRID

- P2P CPU cycle sharing
- GRID Computing, ..., distributed simulation
 - SETI@home: search for extraterrestrial intelligence
 - Popular Power: former battle to the influenza virus

Security and Reliability

- Resilient Overlay Network (RON)
- Secure Overlay Services (SOS)

Application Layer Multicast

- Narada

VoIP

- Skype

P2P: When is it useful?

- P2P paradigm provides
 - Scalable means for communication
 - Infrastructure-less deployment of distributed systems
 - Incrementally deployable services
 - Fault-tolerance
 - Anti-censorship means for sharing ideas
 - Utilize spare end-node resources

P2P: When it is more a curse than a blessing?

- P2P may introduce hassles when
 - Business plans require absolute control of provided services
 - Small scale network use cases
 - Unreliable end-nodes
 - Complete authentication is crucial
 - Trust is hard to establish
 - Copyright management issues