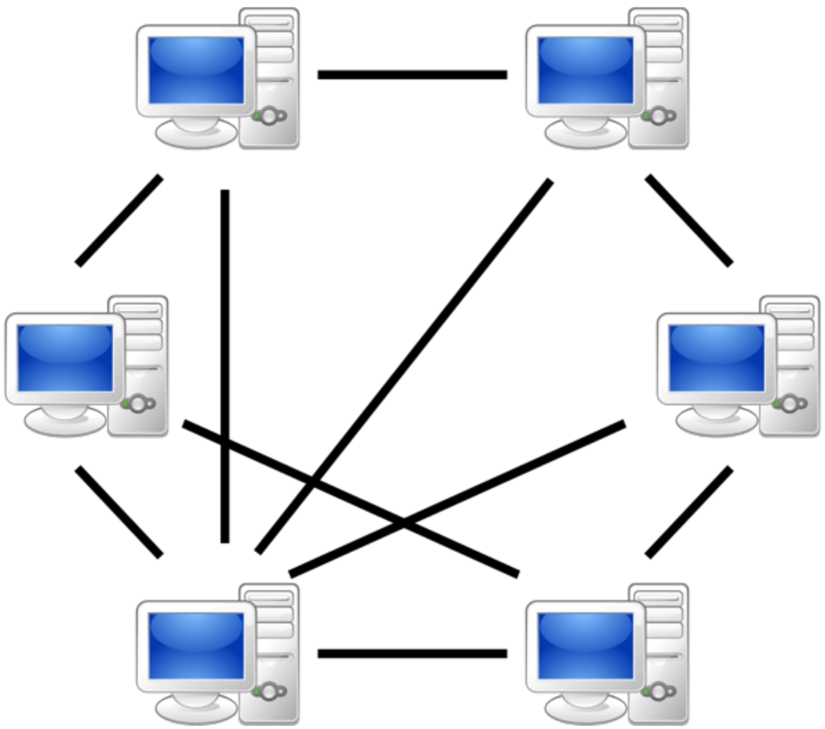
Peer-to-Peer Systems

Tecnologie Internet
a.a. 2022/2023

Peer-to-peer systems

A **P2P system** is a particular type of **distributed system**, i.e. an hardware and software system containing more than one processing element, storage element, concurrent processes, or multiple programs, running under a loosely or tightly controlled regime.



Peer-to-peer systems

The **P2P paradigm** emerged as a highly appealing solution for scalable and high-throughput resource sharing among decentralized computational entities, by using appropriate information and communication systems without the necessity for central coordination.

Leveraging idle resources to do something useful, like cycle sharing or content sharing, is the key concept of the peer-to-peer paradigm.

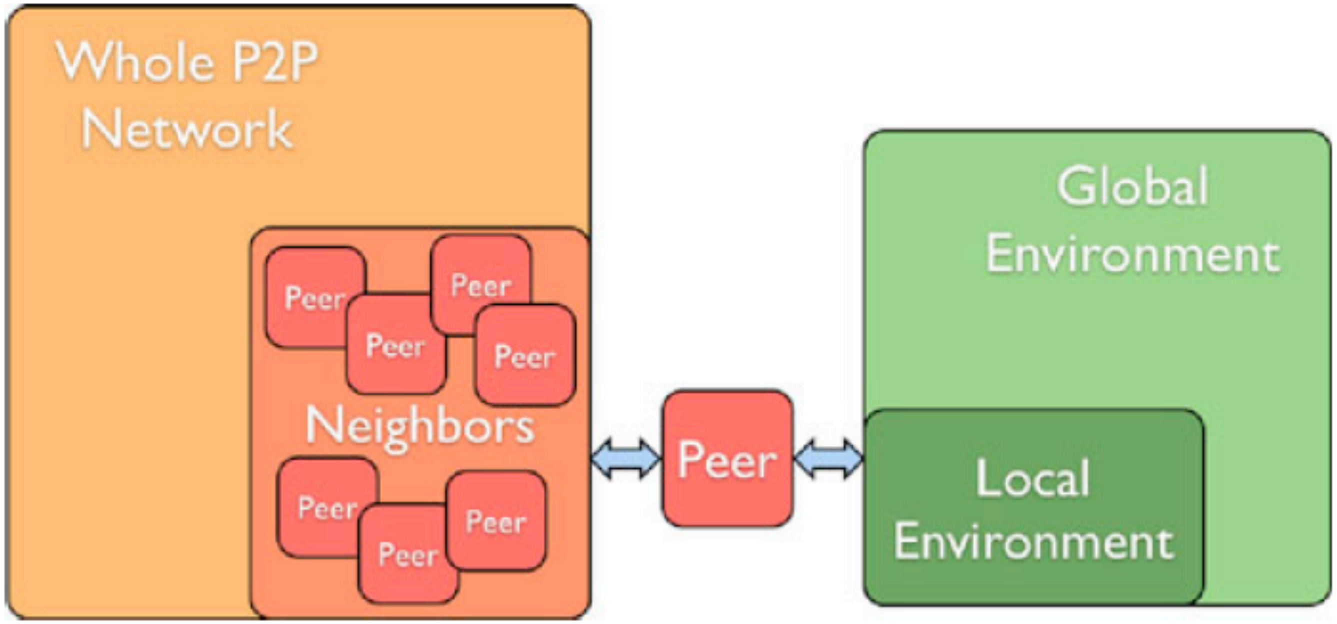
Peer-to-peer file sharing systems like **Napster**, **eMule**, and **BitTorrent** have enjoyed explosive popularity. But the peer-to-peer paradigm can be applied to many other kinds of distributed applications.

For example, it is a viable solution for high performance computing, autonomic network computing, multi-agent systems, Internet streaming applications, etc.

Peer-to-peer systems

A P2P system is a complex system whose elements (peer nodes, or simply nodes or peers) are collectors, providers and consumers of resources.

Resource sharing is based on the collaboration among peers, each one having a limited view of the system but contributing to the correct functioning of the whole system.



Peer-to-peer systems

We distinguish between **three types of peer-to-peer systems**:

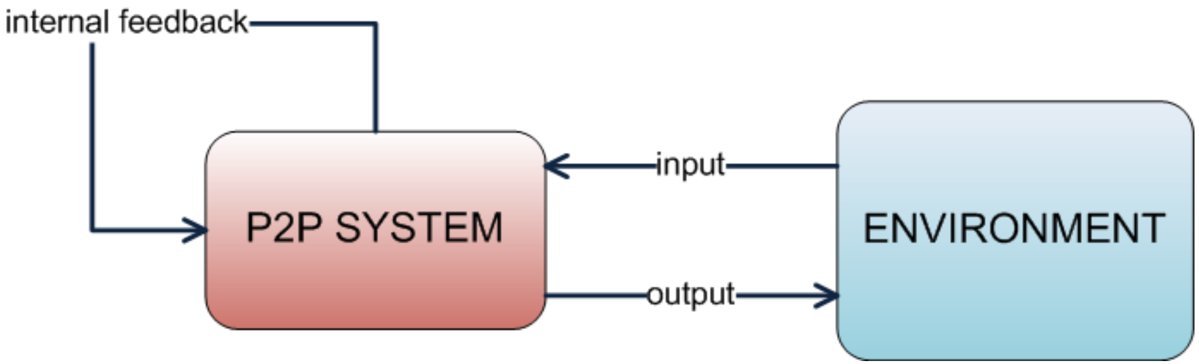
1) Peer-to-peer systems in which **each peer is owned by a user**, which interacts with it by means of a user interface.

2) Peer-to-peer systems in which **autonomous peers** manage resources, sensors and actuators to provide services to users.

3) Hybrid peer-to-peer systems in which some peers are owned and managed by users, while other peers are autonomous.

Peer-to-peer systems

The activities of a peer-to-peer system are driven by the environment and by internal feedbacks.



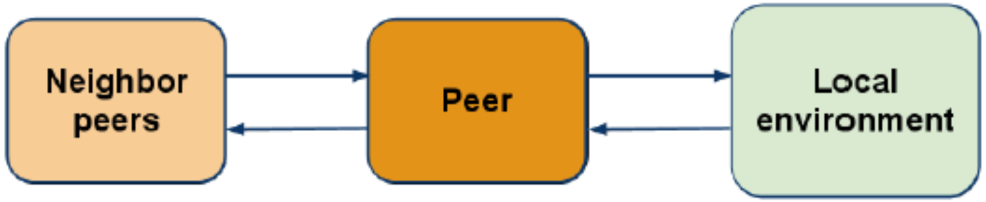
Environmental inputs usually target a very limited number of peers. When a peer receives an input (from the environment or from other peers), its internal structures maps the input to an output. The mapping process could require the peer to cooperate with other peers, exchanging messages in order to discover and eventually consume resources.

In any case, **localized reactions follow localized inputs**.

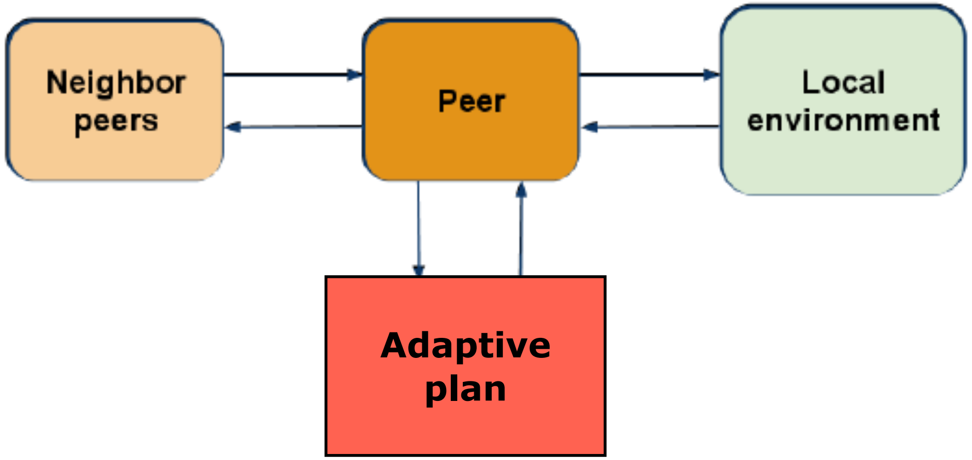
Peer-to-peer systems

The reaction of a peer to direct or indirect inputs from the environment is defined by its internal structure, which can be based

• either on static rules shared by every peer (protocols)



• or based on an adaptive plan which determines successive structural modifications in response to the environment.



State variables

A P2P system is an **overlay network** of peers. When joining the network, each peer connects (i.e., is aware of, not necessarily opens a network connection) to other peers according to a predetermined strategy. Connections may change over time, depending on node departures and on other factors.

A peer shares **resources**: bandwidth, cache, CPU, storage, files, services, applications.

Sharing implies that resources are discoverable.

Some peers share a minimal amount of resources but consume a lot of resources, they are called **free riders**.

State variables

Replicable resources: can be moved or copied from one peer to another; examples are data, files.

Consumable resources: cannot be acquired (by replication), but may only be directly used upon contracting with their hosts; examples are very *basic* resources like storage space, bandwidth, CPU cycles, but also more complex ones, such as applications and services.

Services can be grouped in two categories:

- **distributed services**, whose execution involves many peers;

- **local services**, functional units exposed by peers to allow remote access to their local resources.

State variables

Each peer has a **lifetime** L which can be modeled considering the following factors: the role of the peer in the system, the availability of resources, and unpredictable events (e.g., hardware failures).

Variables that can be measured by the peer itself:
- usage of shared resources (e.g., cache, bandwidth)
- response time of neighbors
- query hit ratio (QHR)

Variables that can be set by the peer itself:
- size of the routing table (RT)
- RT filling strategy
- message routing algorithm
- upload bandwidth
- download bandwidth

State variables

We use stochastic models to study:

- ❑ **Resource distribution**
- ❑ **Resource popularity**

Dynamics of P2P systems

Nodes in a P2P network do not constitute a population in the traditional sense, because their growth in number is not the result of reproduction of existing nodes.

Nodes join or leave the network for many different unpredictable reasons, even if the presence of interesting resources is the fundamental motivation for users.

Some nodes are always connected, other nodes join to consume resources and then leave, etc.

Design issues

The **topology**, structure, and degree of centralization of the overlay network, and the message routing and location mechanisms it employs for messages and resources are crucial to the operation of the system, as they affect its scalability, security, fault tolerance, and self-maintainability.

In recent years, the research on P2P systems has focused on designing robust overlay schemes (defining the rules for bootstrapping, connectivity, message routing, caching), usually targeting specific applications.

In the following we briefly recall the most intriguing P2P design challenges, in order to have a base for the following categorization of overlay schemes.

Design issues

Effectiveness and Efficiency

- scalability
- bootstrapping
- connectivity management
- search performance
- consistency
- stability
- load balancing
- asymmetric bandwidth

Design issues

Security

- Passive attacks
  - eavesdropping
  - traffic analysis

- Active attacks
  - spoofing
  - man-in-the-middle
  - playback or replay
  - local data alteration
  - no forwarding
  - free riding
  - distributed denial of service (ddos)
  - network poisoning

Design issues

Security

- Trust management

How to enable a peer to decide whether to trust another peer, in the absence of a central trust managing authority?

Trust is important when sharing data or processing power, and crucial for e-commerce applications such as auctioning.

Trust is not transitive and subjective.

Design strategies for overlay schemes

In P2P systems, **information placement** plays an important role in the characterization of an overlay scheme. Information about shared resources can be

- published to a central server
- published to other peers
- locally stored by resource owners and not published

Overlay scheme models:

- ❑ **Hybrid Model (HM)**
- ❑ **Decentralized Unstructured Model (DUM)**
- ❑ **Decentralized Structured Model (DSM)**

Layered Model (LM): peers are grouped into layers, each one being organized according to a flat overlay scheme (HM, DUM or DSM).

Design strategies for overlay schemes

	HM	DUM	DSM
Scalability	Low	Medium	High
Bootstrapping	Simple	Complex	Simple
Connectivity Mgmt	Simple	Complex	Complex
Time Complexity	*1*	*O(N)*	*O(logN)*
Space Complexity	*1*	*O(d)*	*O(logN)*
Consistency	High	Low	High
Stability	High	High	Low
Load Balancing	No	No	Yes

Design strategies for overlay schemes

	HM	DUM	DSM
Eavesdropping	No	Yes	Yes
Traffic Analysis	Yes	Yes	No
Spoofing	No	Yes	Yes
Man-in-the-middle	Yes	Yes	Yes
Playback or Replay	Yes	Yes	Yes
Local Data Alteration	Yes	Yes	Yes
No-forwarding	No	Yes	No
Free Riding	Yes	Yes	Yes
DDoS	No	Yes	No
Network Poisoning	No	Yes	No

Hybrid Model (HM)

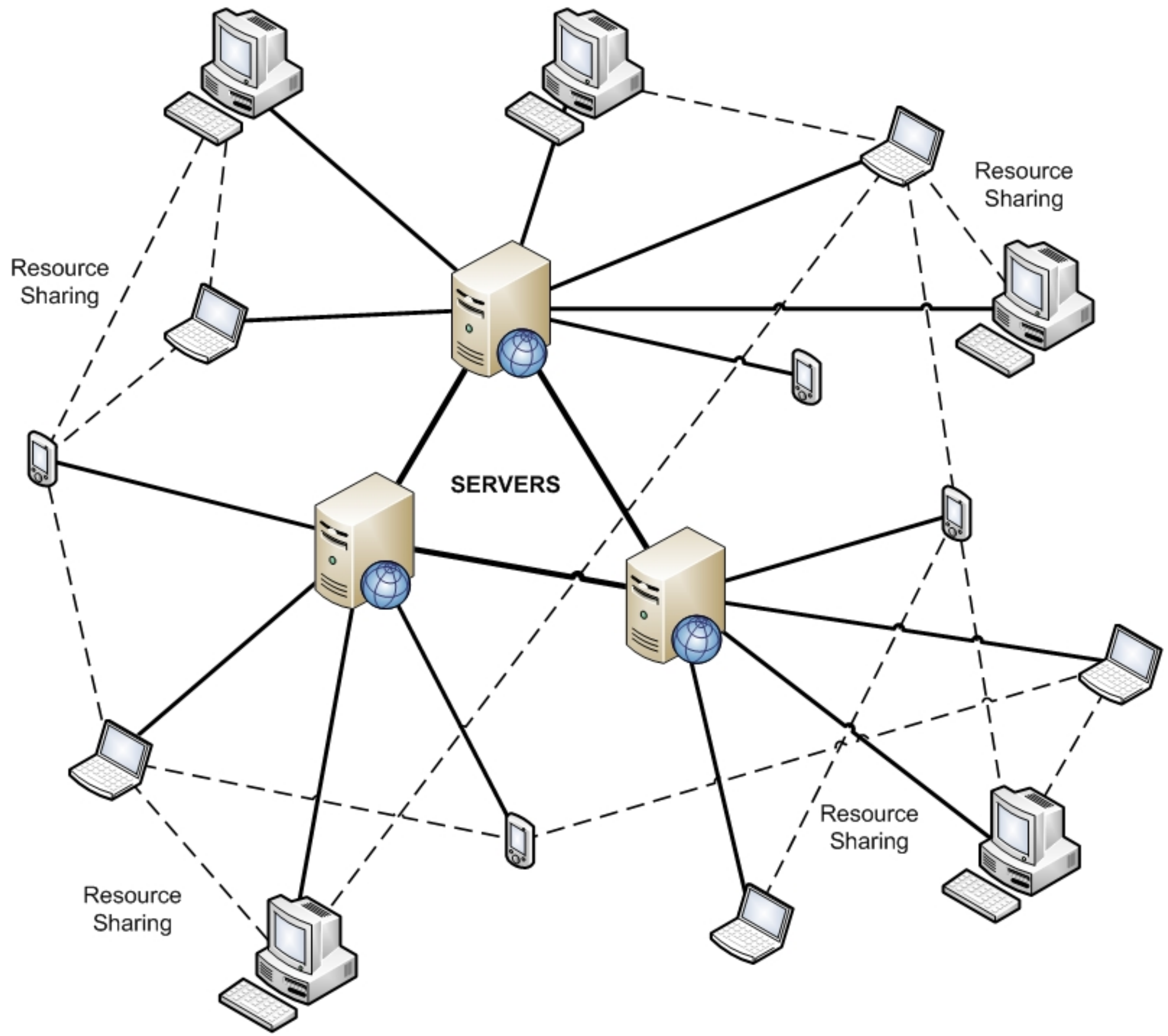
Peers connect to one or more central servers in which they publish information about the resources they offer for sharing. Each central server maintains, for each resource, the list of owners, in some case partially replicating the lists of other known servers.

Upon request from a peer, the central directory provides the list of peers that match the request, or the best peer in terms of bandwidth and availability. Further interactions occur directly between the resource provider and the consumer.

If the central server is not unique, queries may also be forwarded to neighbor servers.

Examples: Napster, eMule and BitTorrent

Hybrid Model (HM)



Decentralized Unstructured Model (DUM)

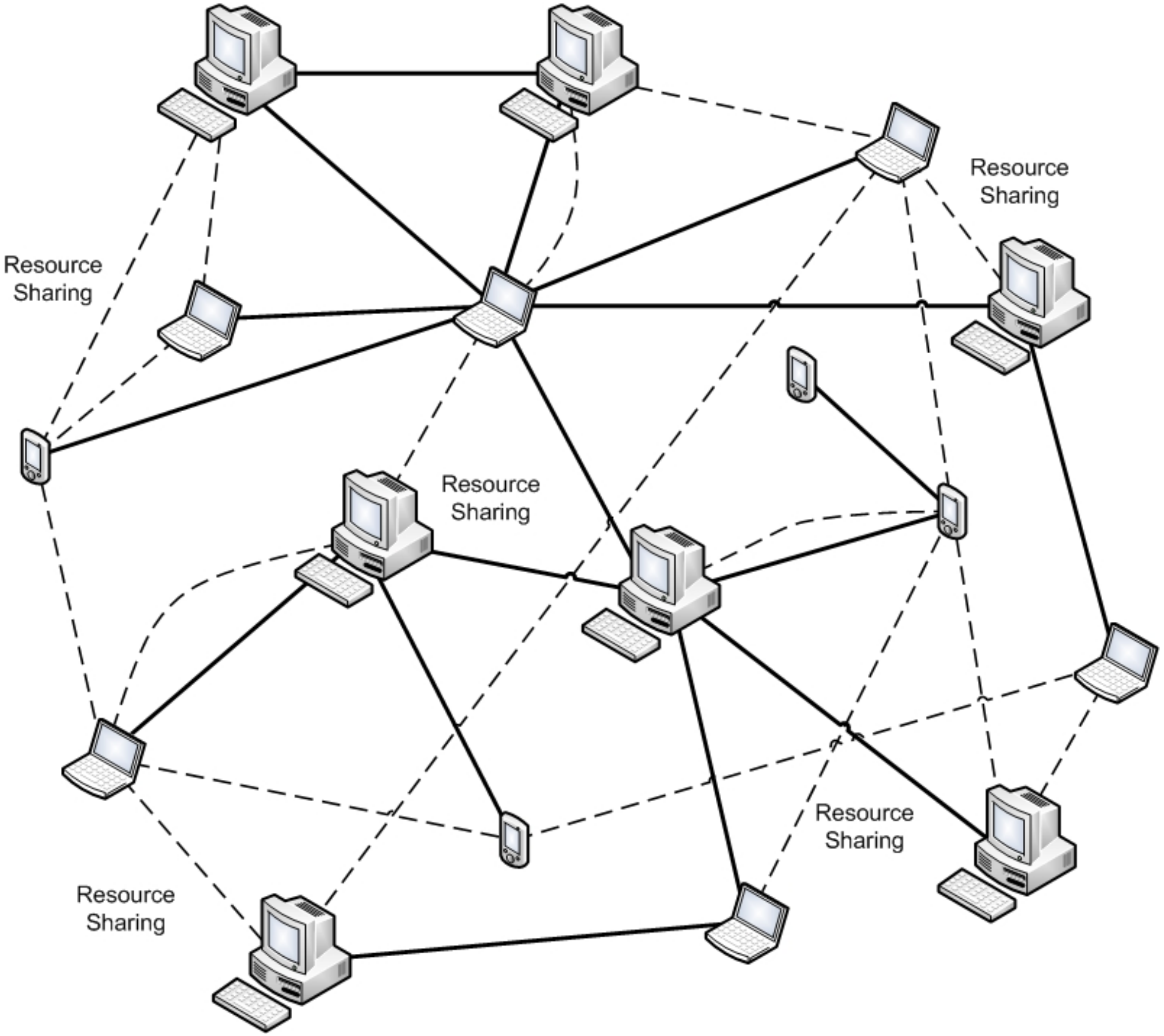
Each peer propagates requests to directly connected peers, according to some strategy, e.g., by **flooding** the network with messages, a strategy that consumes a lot of network bandwidth, and hence does not prove to be very scalable, but it is efficient in limited communities such as company networks.

To improve scalability, caching of recent research requests can be used, together with probabilistic flooding (i.e., each peer propagates messages to a random number of neighbors).

Moreover, if a unique identity is added to the message, nodes can delete reoccurrences of the same message and stop loops from forming.

In large networks it may be necessary to include some kind of time-to-live (TTL) counter to stop the message flooding the network.

Decentralized Unstructured Model (DUM)



Decentralized Structured Model (DSM)

A globally consistent protocol ensures that any node can efficiently route a search to a peer that has the desired resource, even if the resource is extremely rare. Such a guarantee necessitates a more structured pattern of overlay links.

By far the most common type of structured P2P network is the **Distributed Hash Table (DHT)**. Each resource must be identified by a unique key and associated to a description (including a pointer to the resource owner): key and description form a <key,value> pair associated to the resource.

Decentralized Structured Model (DSM)

Each peer is assigned a random ID in same space of resource keys, and it is responsible for storing <key,value> pairs for a limited subset of the entire key space.

Publication: a resource description and the associated <key,value> pair are routed towards the peer with the ID that is most similar to the resource ID. This process is repeated until the nearest peer ID is the current peer's ID.

In DSM-based overlay schemes, the responsibility of storing information about shared resources is much more fairly distributed among peers than in DUM-based ones.

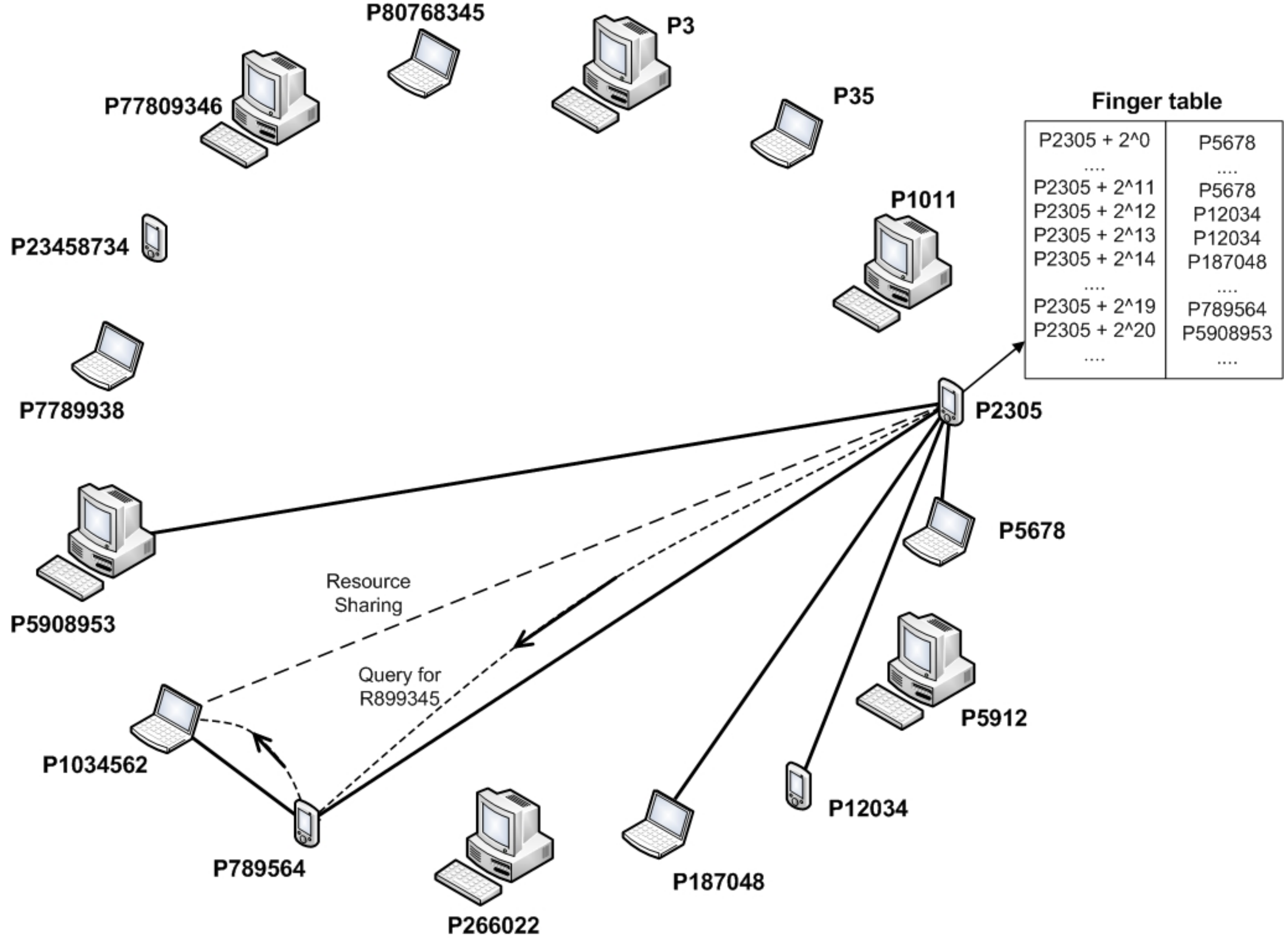
Lookup: the query is propagated towards the peer with the ID which is most similar to the resource ID.

Decentralized Structured Model (DSM)

Some common choices for maximum degree and route length are as follows, where N is the number of nodes in the DHT, using Big O notation:

- degree $O(1)$, route length $O(N)$
- degree $O(\log N)$, route length $O(\log N/\log\log N)$
- degree $O(\log N)$, route length $O(\log N)$
- degree $O(\sqrt{N})$, route length $O(1)$

Decentralized Structured Model (DSM)

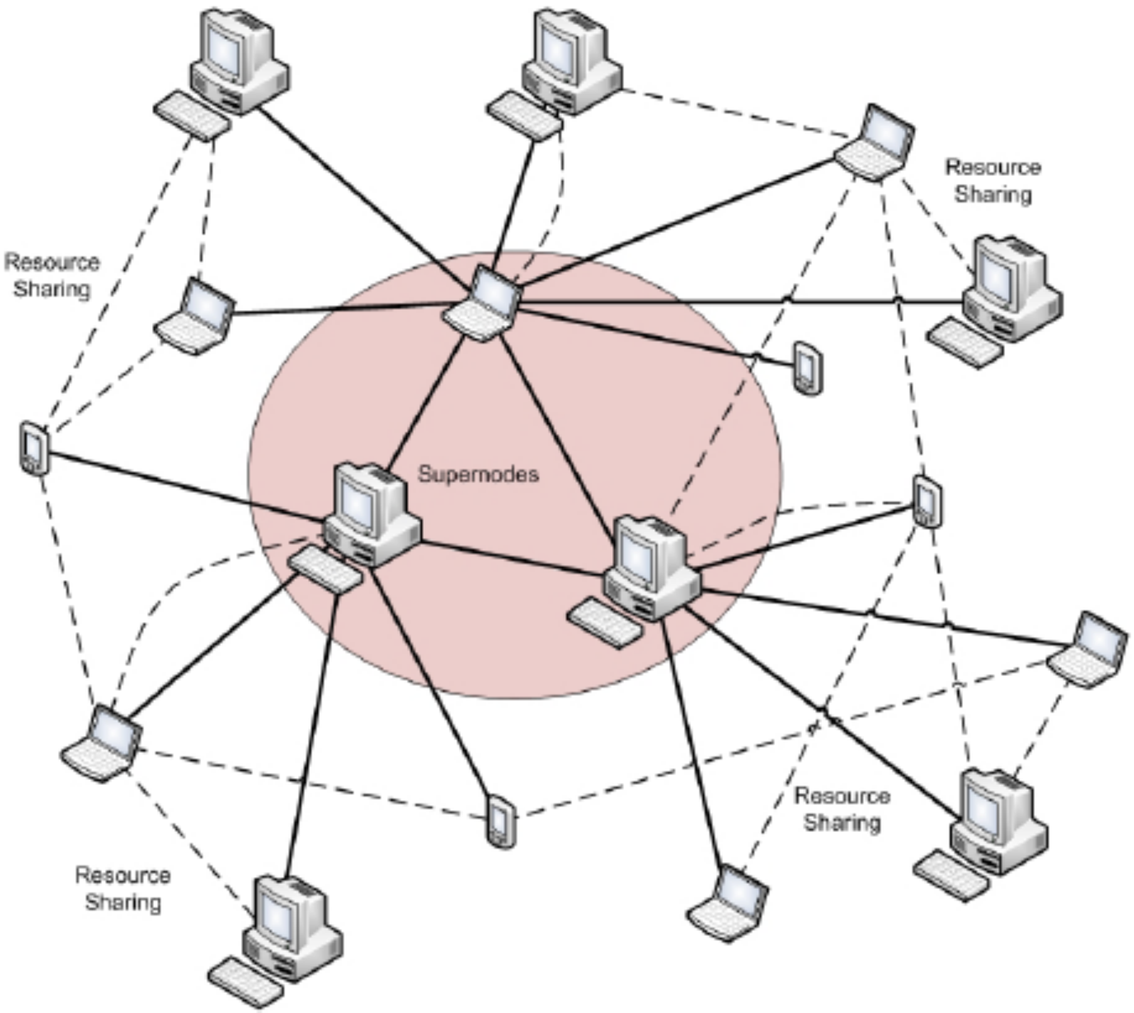


Layered overlay schemes

In layered architectures, peers are grouped into layers, each one being organized according to a flat overlay scheme (HM, DUM or DSM). The interaction among layers is usually defined by an application-specific protocol.

A typical example is the 2-layered architecture, in which peers with higher bandwidth and process capacity act as **supernodes**, assuming the responsibility of propagating messages, while peers with less capacities (leaf nodes) are only resource providers and consumers, and need to connect to the supernode layer in order to publish/discover shared resources.

Layered overlay schemes



Standardization Processes

During the last ten years, many different P2P architectural blueprints and protocols have been proposed. Unfortunately each developer promoted his/her brainchild targeted to a specific application and with a restricted set of functionalities, without any sort of significant coordination and standardization effort.

- ❑ **JXTA** https://github.com/chaupal https://github.com/chaupal/jxta-soap
- ❑ **IETF P2P Working Groups**
  - ❑ Application-Layer Traffic Optimization (ALTO)
    http://tools.ietf.org/wg/alto/
  - ❑ P2P Session Initiation Protocol (P2PSIP) *[concluded WG]*
    http://tools.ietf.org/wg/p2psip/
- ❑ **REsource LOcation And Discovery (RELOAD)**
  https://tools.ietf.org/html/rfc6940