HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Overlay and P2P Networks

Introduction and unstructured networks

Prof. Sasu Tarkoma

20.9.2010

Contents

- Overlay networks and intro to networking
- Unstructured networks
- BitTorrent

Overlay Networks

An overlay network is a network that is built on top of an existing network.

The overlay therefore relies on the so called underlay network for basic networking functions, namely routing and forwarding.

Today, most overlay networks are built in the application layer on top of the TCP/IP networking suite.

Overlay technologies can be used to overcome some of the limitations of the underlay, and at the same time offering new routing and forwarding features without changing the routers.

The nodes in an overlay network are connected via logical links that can span many physical links. A link between two overlay nodes may take several hops in the underlying network.

Requirements for Overlay Networks

1. Support the execution of one or more distributed applications by providing infrastructure for them.
2. Participate and support high-level routing and forwarding tasks. The overlay is expected to provide data forwarding capabilities that are different from those that are part of the basic Internet.
3. Deployed across the Internet in such a way that third parties can participate in the organization and operation of the overlay network.

Protocol Stack

- Layers are part of a network architecture
  - Provide services for layers above
  - Hiding the complexity of the current layer
- Multiple layers are needed in order to reduce complexity
  - Separation of network functions
  - distribution of complexity
  - OSI, TCP/IP
- Protocols are building blocks of a network design
  - Can exist independently of layering

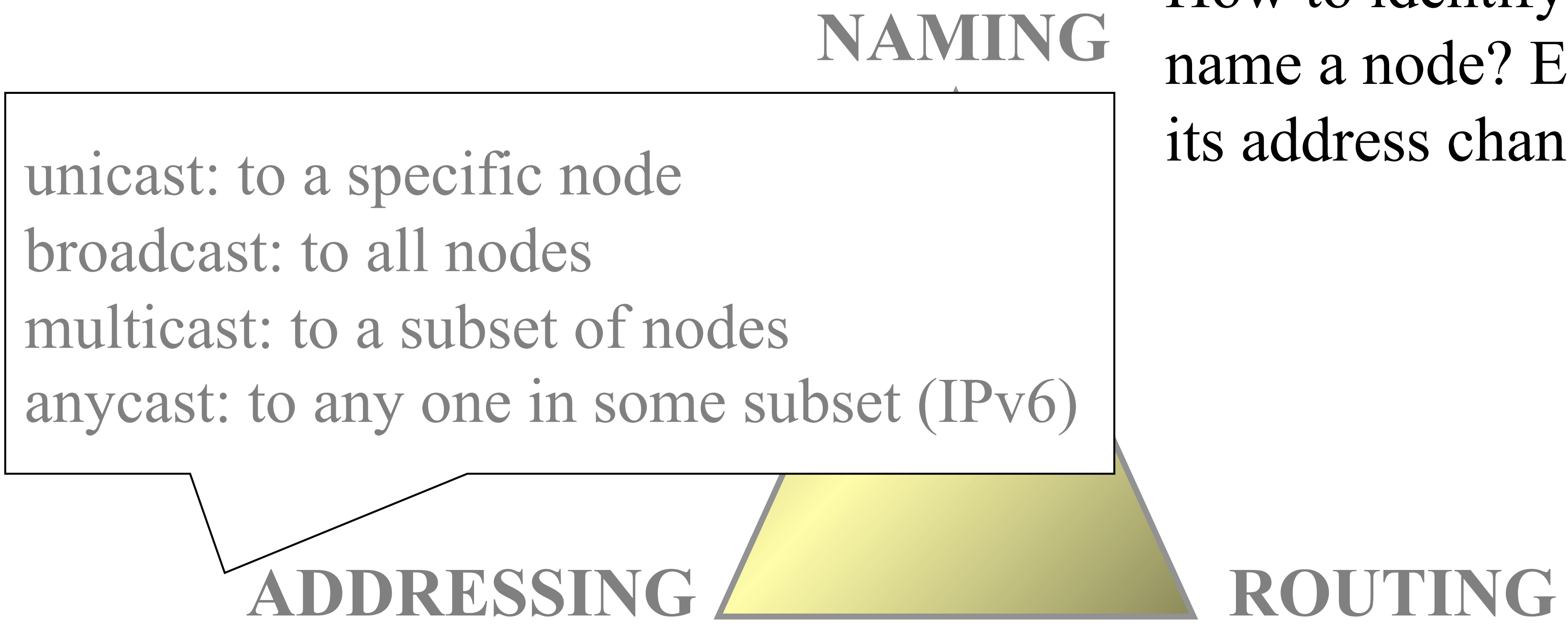
Background

- What is network architecture?
- Layered architecture
- The original requirements for IP
- Later requirements for IP
- Motivation for overlay networks

Network architecture

- A set of principles and basic mechanisms that guide network engineering
  - Physical links
  - Communication protocols
    - Format of messages
    - The way in messages are exchanged
    - Protocol stack
- Where is the state?

Naming, Addressing, and Routing



NAMING

How to identify and name a node? Even if its address changes.

unicast: to a specific node
broadcast: to all nodes
multicast: to a subset of nodes
anycast: to any one in some subset (IPv6)

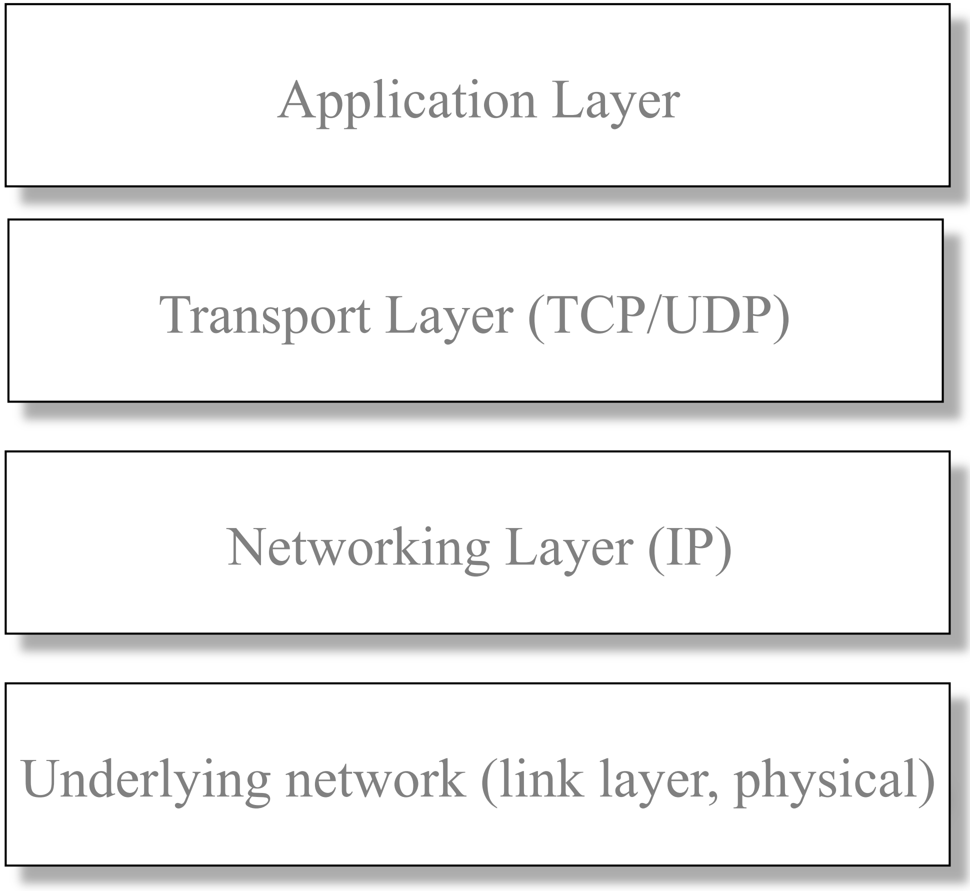
ADDRESSING

Where is the node located?

ROUTING

How to route information to the node’s address?

TCP/IP Network Stack



Application Layer

Transport Layer (TCP/UDP)

Networking Layer (IP)

Underlying network (link layer, physical)

Evolution of the Network

Video delivery has become one of the recent services on the Web

Estimates of P2P share of network traffic range from 50% to 70%

Cisco’s latest traffic forecast for 2009-2013 indicatesthat annual global IP traffic will reach 667 exabytes in 2013, two-thirds of a zettabyte.

The traffic is expected to increase some 40% each year

Much of this increase comes from the delivery of video data

According to the study, P2P traffic will continue to grow, but become a smaller component of Internet traffic in terms of its current share

Video is being delivered by a set of protocols, typically coordinated by overlay solutions and CDN solutions

We will cover these in during the course

CDNs

Content Delivery Networks (CDNs) are examples of overlay networks that cache and store content and allow efficient and less costly way to distribute data in massive scale

CDNs typically do not require changes to end-systems and they are not peer-to-peer solutions from the viewpoint of the end clients

Challenges for Overlay Networks

The Real World. In practice, the typical underlay protocol, IP, does not provide universal end-to-end connectivity due to the ubiquitous nature of firewalls and Network Address Translation (NAT) devices.

Management and administration. Practical deployment requires that the overlay network has a management interface.

Overhead. An overlay network typically consists of a heterogeneous body of devices across the Internet. It is clear that the overlay network cannot be as efficient as the dedicated routers in processing packets and messages. Moreover, the overlay network may not have adequate information about the Internet topology to properly optimize routing processes

Network Invariants and Metrics

The correctness and performance of a routing algorithm can be analyzed using a number of metrics

Typically it is expected that a routing algorithm satisfies certain invariant properties that must be satisfied at all times. The two key properties are *safety* and *liveness*

The former states that undesired effects do not occur, in other words the algorithm works correctly, and the latter states that the algorithm continues to work correctly, for example avoids deadlocks and loops

These properties can typically be proven for a given routing algorithm under certain assumptions

Important metrics: *shortest path*, *routing table size*, *path stretch*, *forwarding load*, *churn*

Terminology

- Peer-to-peer (P2P)
  - Different from client-server model
  - Each peer has both client/server features
- Overlay networks
  - Routing systems that run on top of another network, such as the Internet.
- Distributed Hash Tables (DHT)
  - An algorithm for creating efficient distributed hash tables (lookup structures)
  - Used to implement overlay networks
- Typical features of P2P / overlays
  - Scalability, resilience, high availability, and they tolerate frequent peer connections and disconnections

Peer-to-peer in more detail

- A P2P system is distributed
  - No centralized control
  - Nodes are symmetric in functionality
- Large faction of nodes are unreliable
  - Nodes come and go
- P2P enabled by evolution in data communications and technology
- Current challenges:
  - Security (zombie networks, trojans), IPR issues
- P2P systems are decentralized overlays

Characteristics of P2P systems

P2P can be seen as an organizational principle

- Applied in many different application domains

Characteristics

- Self-organization
- Lack of central coordination
- Resource sharing
- Based on collaboration between peers
- Peers are typically equal
- Large number of peers
- Resilient to certain kinds of attacks (but vulnerable to others)

P2P Volume

Estimates ranger from 40-70% of Internet Traffic

Latest estimates from Cisco suggest that video delivery is the growing and the share of P2P traffic is becoming smaller

P2P can be used for video delivery as well

.. And voice (Skype, P2PSIP)

Hundreds of millions of people use P2P technology today

Evolution of P2P systems

- ARPAnet had P2P like qualities
  - End-to-end communication, FTP, USENET,..
  - Today’s BGP is P2P
- Started from centralized servers
  - Napster
    - Centralized directory
    - Single point of failure
- Second generation used flooding (Gnutella v0.4)
  - Local directory for each peer
  - High cost, worst-case O(N) messages for lookup
  - Third generation use some structure (Gnutella v0.7)
- Research systems use DHTs
  - Chord, Tapestry, CAN, ..
  - Decentralization, scalability
- Some recent CDNs and content delivery systems exhibit P2P features (P2P assisted CDN)

Trend	Challenges	Solutions
P2P	Growth in traffic, upstream bottlenecks	P2P caching
Internet Broadcast	Flash crowds	P2P content distribution, multicast technologies
Internet Video-on-Demand	Growth in traffic, especially metropolitan area and core	Content Delivery Networks (CDNs), increasing network capacity, compression
Commercial Video-on-Demand	Growth in traffic in the metropolitan area network	CDNs, increasing network capacity, compression
High-definition content	Access network IPTV bottleneck, growth in VoD traffic volume in the metropolitan area network	CDNs, increasing network capacity, compression

Challenges

- Challenges in the wide-area
  - Scalability
  - Increasing number of users, requests, files, traffic
  - Resilience: more components → more failures
  - Management: intermittent resource availability → complex management
- Copyright issues in file sharing and content distribution

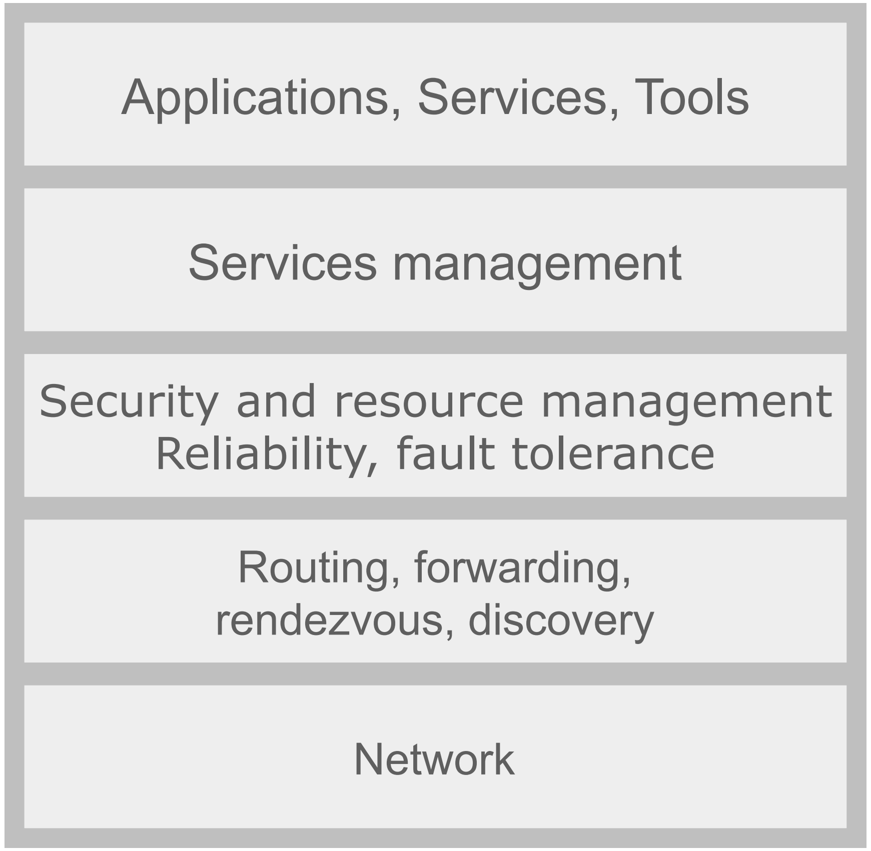
Evolution

- Should some features provided by P2P systems and overlay solutions be incorporated into the network infrastructure?
  - Content centric networking
  - In-network caching
  - Software-defined networking

Overlay Networks

- Origin in Peer-to-Peer (P2P)
- Builds upon Distributed Hash Tables (DHTs)
- Easy to deploy
  - No changes to routers or TCP/IP stack
  - Typically on application layer
- Overlay properties
  - Resilience
  - Fault-tolerance
  - Scalability

Overlay Networks: Layering



Applications, Services, Tools

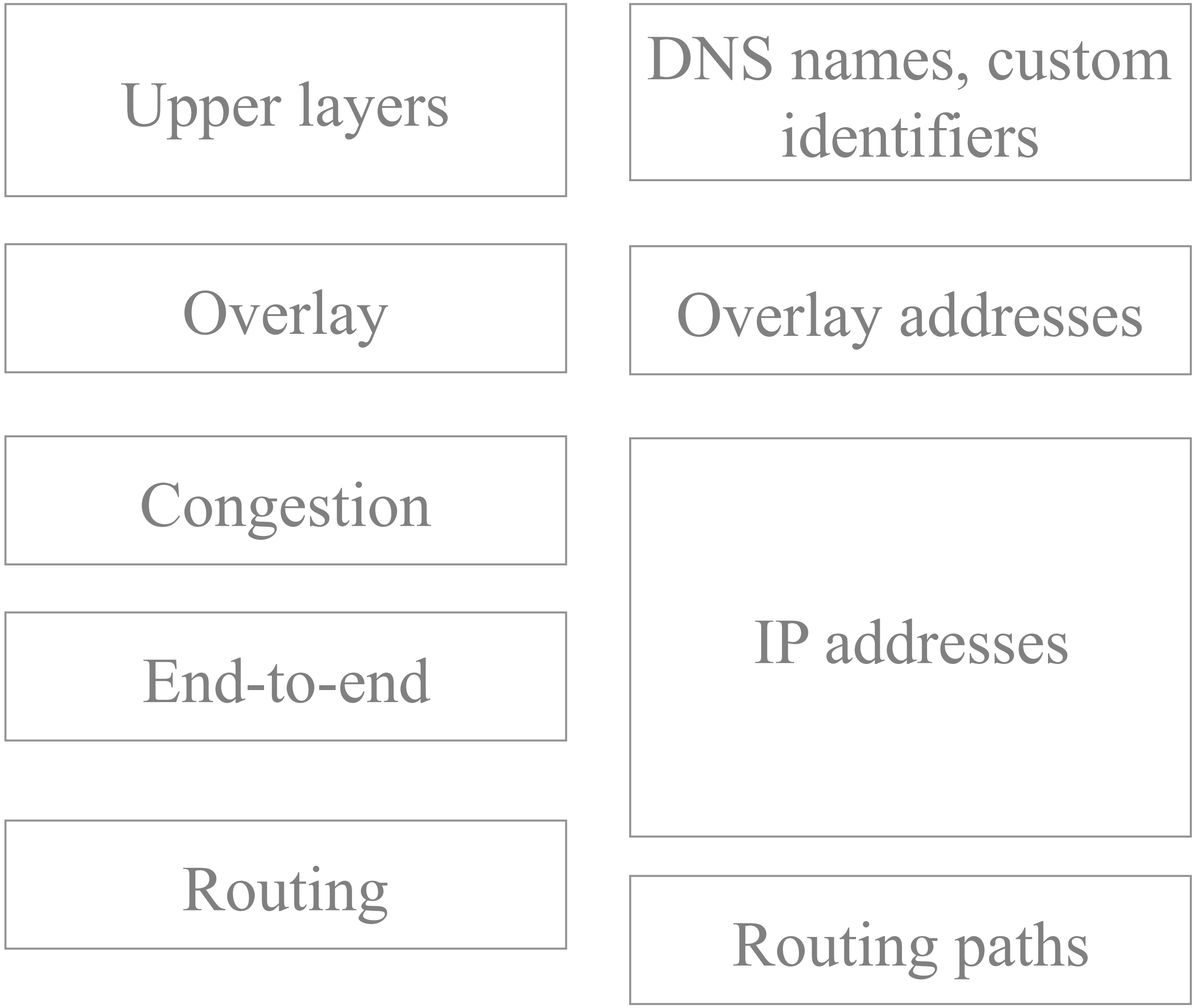
Services management

Security and resource management
Reliability, fault tolerance

Routing, forwarding,
rendezvous, discovery

Network

Overlay Networks: Layering



Upper layers

DNS names, custom identifiers

Overlay

Overlay addresses

Congestion

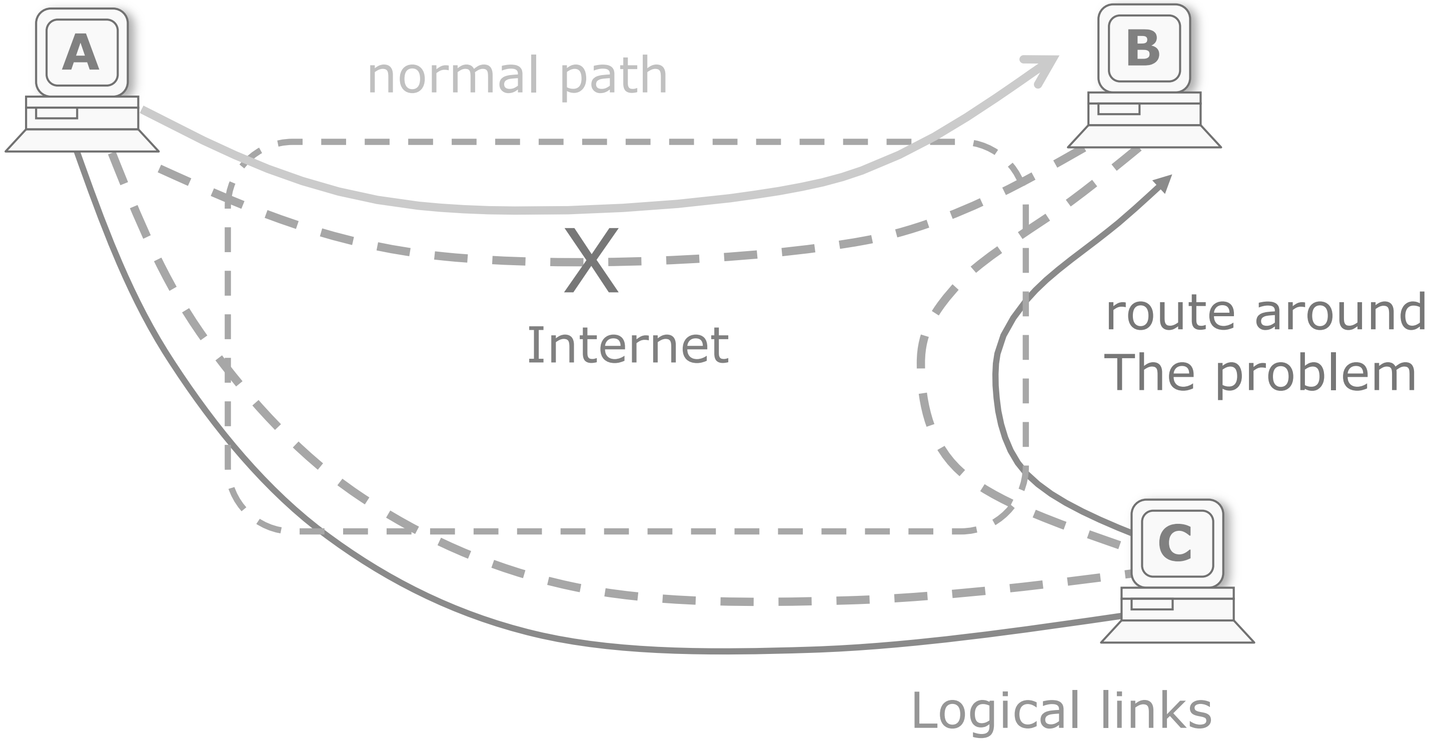
End-to-end

IP addresses

Routing

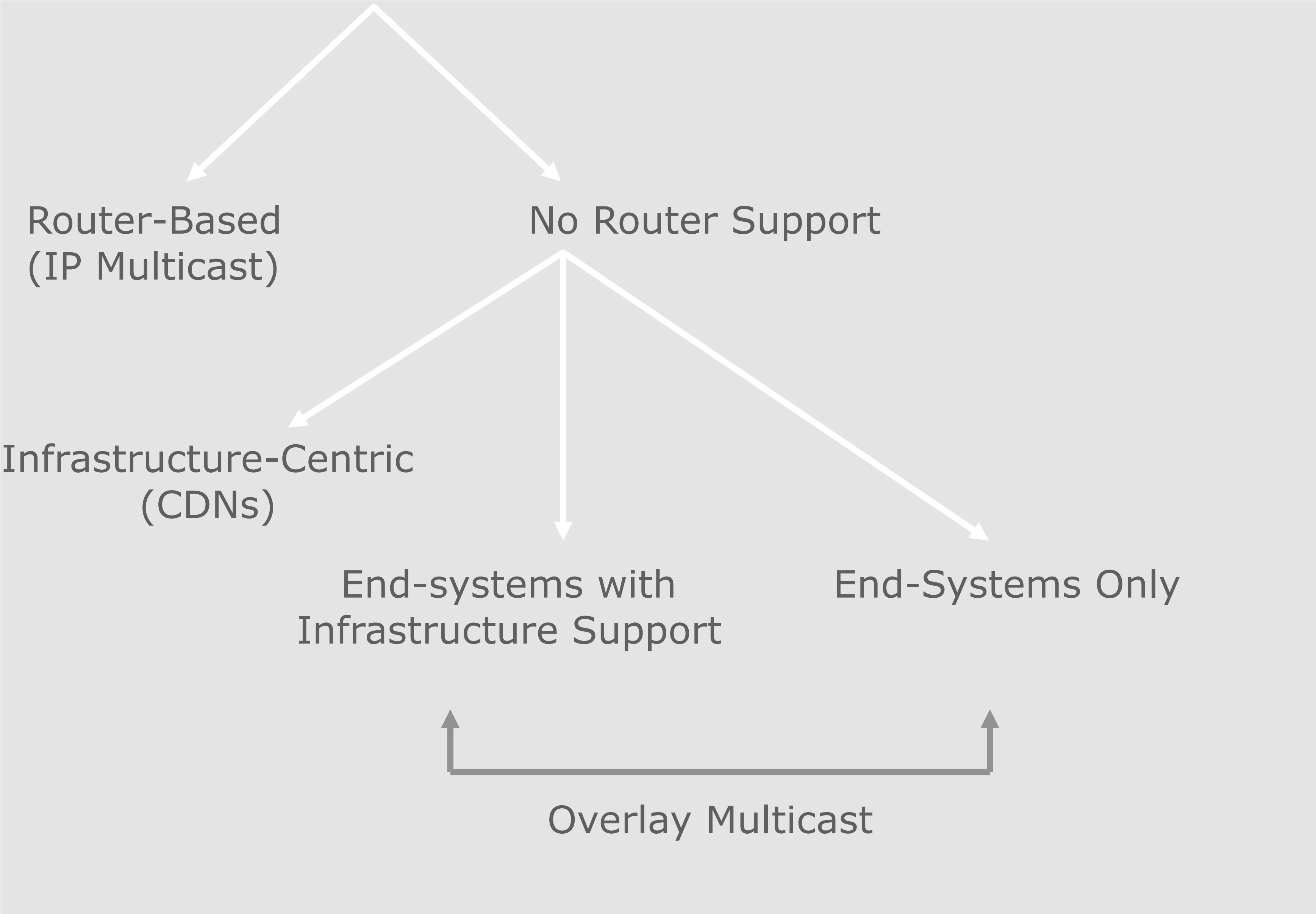
Routing paths

Resilient Routing with Overlay Networks



Some overlay and P2P applications

- File sharing
- Multicast and content delivery
- Web caching
- Censor-resistant data storage
- Event notification
- Naming systems
- Query and indexing
- Communication primitives
- Backup storage
- Web archive



Router-Based
(IP Multicast)

No Router Support

Infrastructure-Centric
(CDNs)

End-systems with
Infrastructure Support

End-Systems Only

Overlay Multicast

Comparison of Multicast Techniques

	IP multicast	Overlay multicast
Deployment	Multicast-capable routers	Deployed over the Internet
Multicast structure	Tree, interior nodes are routers, leaves are hosts	Typically a tree, both interior nodes of the structure and leaves are hosts
Transport layer protocol	UDP	TCP or UDP
Scalability	Limited	High (depends on solution)
Congestion control / recovery	No	Various, can utilize unicast (TCP) for node-to-node reliability
Efficiency	High	Low (varies), can suffer from high stretch and unoptimal interdomain routing
Example protocols	Protocol Independent Multicast (PIM), Core-based Trees (CBT), …	BitTorrent variants, Scribe, SplitStream, OverCast, …

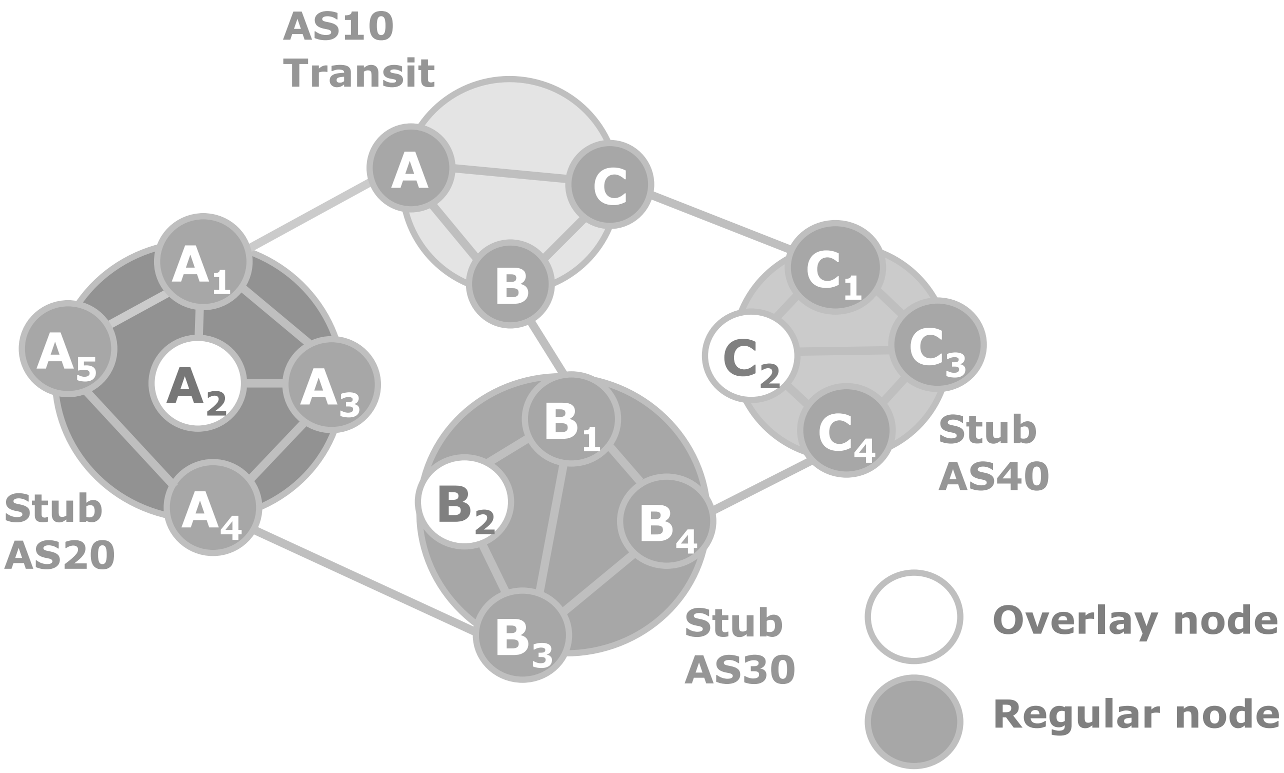
Cluster vs. Wide-area

- Clusters are
  - single, secure, controlled, administrative domains
  - engineered to avoid network partitions
  - low-latency, high-throughput SANs
  - predictable behaviour, controlled environment
- Wide-area
  - Heterogeneous networks
  - Unpredictable delays and packet drops
  - Multiple administrative domains
  - Network partitions possible

Wide-area requirements

- Easy deployment
- Scalability to millions of nodes and billions of data items
- Availability
  - Copes with routine faults
- Self-configuring, adaptive to network changes
- Takes locality into account
  - Overlay routing on top the Internet

Overlay Routing



Advantages in overlay rerouting in the multi-domain scenario.
The double-circled nodes represent the overlay node

Current Unstructured P2P Systems

In an unstructured P2P network content can be placed on any peer (the network algorithm does not mandate that a specific node maintains the content)

- BitTorrent
- Napster
- Skype
- Gnutella
- Freenet

Most systems target file sharing and content distribution

Discussion

Unstructured P2P algorithms have been called *first generation* and the structured algorithms have been called *second generation*

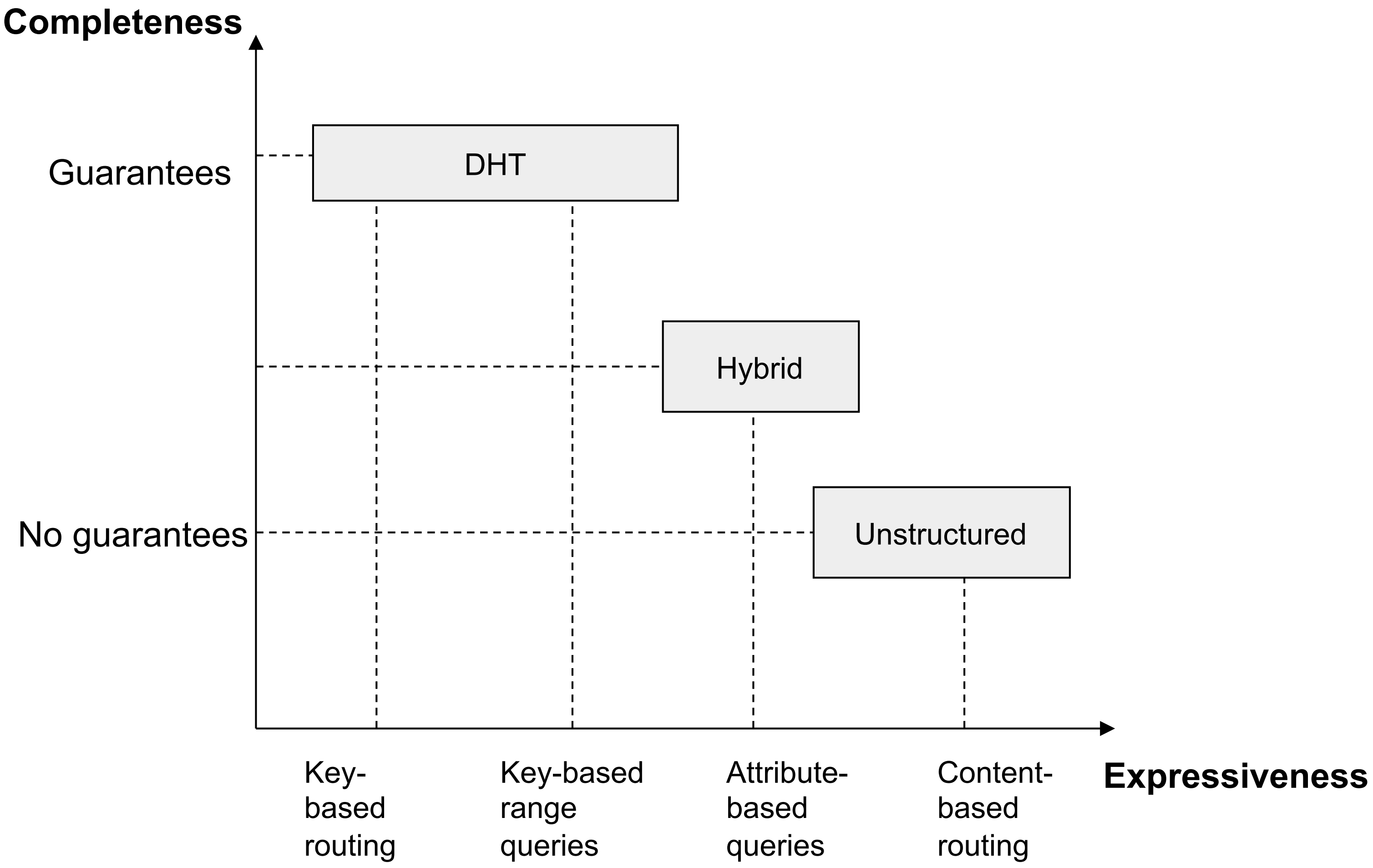
They can also be combined to create *hybrid* systems

The key-based structured algorithms have a desirable property, namely that they can find data locations within a bounded number of overlay hops

The unstructured broadcasting-based algorithms, although resilient to network problems, may have large routing costs due to flooding, or unable to find available content

Another view to P2P system is to divide them into two classes, *pure* and *hybrid* P2P systems. In the former, each peer is simultaneously a client and a server, and the operation is decentralized. In the latter class, a centralized component is used to support the P2P network

Structured vs. Unstructured



Data and content centricity

A large part of the research and development on P2P systems has focused on data-centric operation, which emphasizes the properties of the data instead of the location of the data

Ideally, the clients of the distributed system are not interested where a particular data item is obtained as long as the data is correct. The notion of data-centricity allows the implementation of various dynamic data discovery, routing, and forwarding mechanisms

In content-based routing systems hosts subscribe to content by specifying filters on messages. The content of messages defines their ultimate destination in the distributed system

Self-certified names

Self-certifying data is data whose integrity can be verified by the client accessing it

A node inserting a file in the network or sending a packet calculates a cryptographic hash of the content using a known hash function

This hashing produces a file key that is included in the data

The node may also sign the hash value with its private key and include its public key with the data

This additional process allows other nodes to authenticate the original source of the data

When a node retrieves the data using the hash of the data as the key, it calculates the same hash function to verify that the data integrity has not been compromised

Free-riding and tragedy of the commons

Users of P2P file sharing networks, such as Gnutella, face the question of whether or not to share resources to other peers in the community

They face essentially a social dilemma of balancing between common good and selfish goals

The selfish behaviour often encountered in P2P networks in which peers only download files and do not make resources available on the network is called *free-riding*

Free-riding occurs because the peers have no incentives for uploading files. Free-riding becomes a major problem when significant numbers of peers consume network resources while not contributing to the network. In the context of P2P this is often referred to as *tragedy of the digital commons*

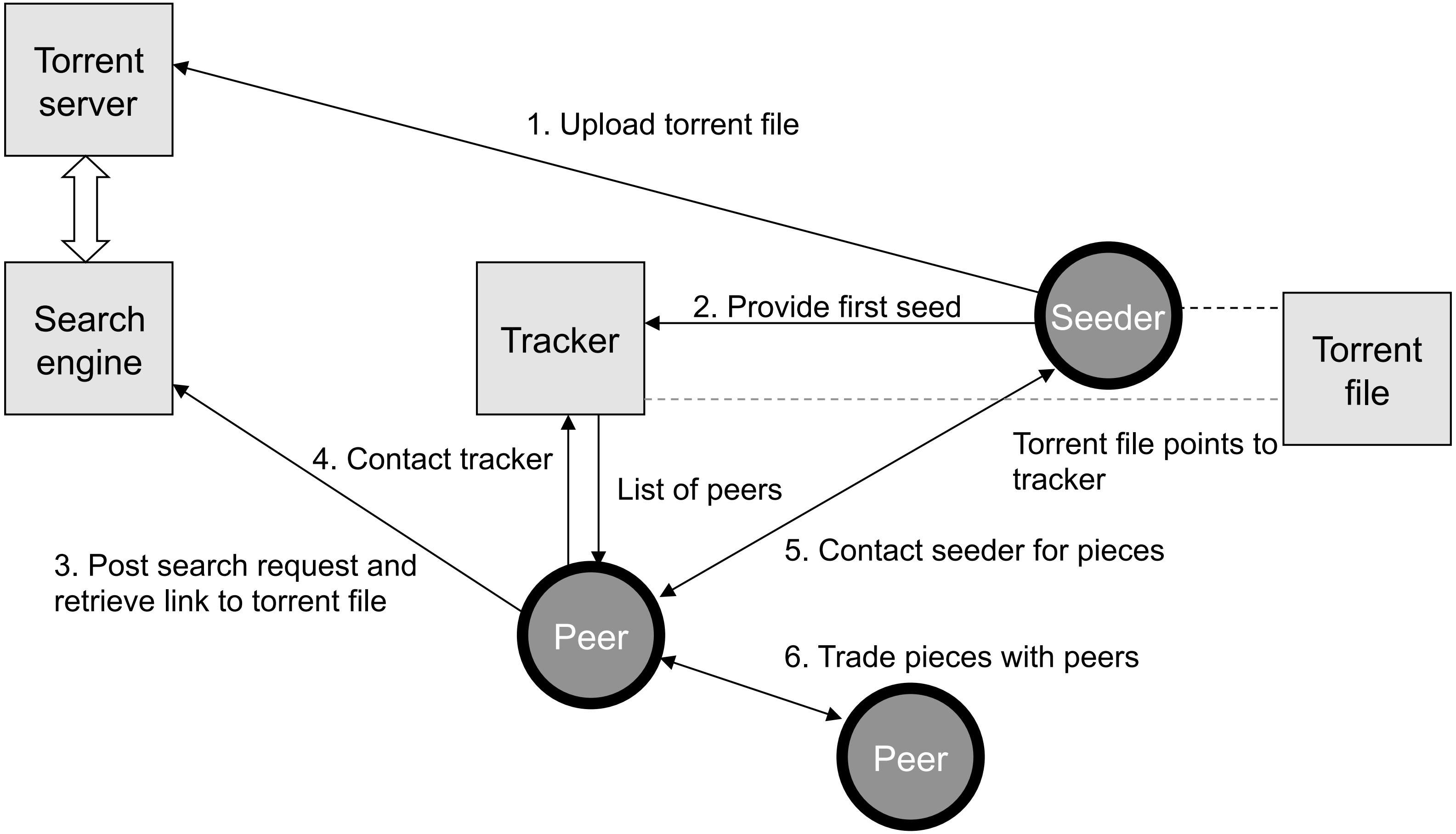
BitTorrent

BitTorrent is based on the notion of a **torrent**, which is a smallish file that contains metadata about host, the tracker, that coordinates the file distribution and files that are shared

A peer that wishes to make data available must first find a tracker for the data, create a torrent, and then distribute the torrent file. Other peers can then using information contained in the torrent file assist each other in downloading the file

The download is coordinated by the tracker. In BitTorrent terminology, peers that provide a complete file with all of its pieces are called **seeders**

BitTorrent: Downloading Files



Torrent server
1. Upload torrent file
Search engine
Tracker
2. Provide first seed
Seeder
Torrent file
Torrent file points to tracker
4. Contact tracker
List of peers
3. Post search request and retrieve link to torrent file
5. Contact seeder for pieces
Peer
6. Trade pieces with peers
Peer

Difference to HTTP

A BitTorrent file download differs from an HTTP request in the following ways:

- BitTorrent uses multiple parallel connections to improve download rates, whereas Web browsers typically use a single TCP Socket to transfer HTTP requests and responses
- BitTorrent is peer-assisted whereas HTTP request is strictly client-server
- BitTorrent uses the random or rarest-first mechanisms to ensure data availability, whereas HTTP is incremental

A solution to the broadcasting problem

BitTorrent attempts to solve the broadcasting problem, which has the goal of disseminating M messages in a population of N nodes in the shortest time

In an environment in which the nodes have bidirectional communications and the same bandwidth, the lower bound on download time (rounds) is given by $M + \log_2 N$, the unit is the time it takes for two nodes to exchange a message

This problem can be solved optimally with a centralized scheduler; however, BitTorrent lacks this centralized component and furthermore it does not have a completely connected graph as well

BitTorrent therefore has a heuristic approach to solving this problem that works very well in practice

Characteristics of the BitTorrent protocol I/II

- **Peer selection** is about selecting peers who are willing to share files back to the current peer
  - **Tit for tat** in peer selection based on download-speed.
  - The mechanism uses a **choking/unchoking** mechanism to control peer selection. The goal is to get good TCP performance and mitigate free riders
- **Optimistic unchoking**
  - The client uses a part of its available bandwidth for sending data to random peers
  - The motivation for this mechanism is to avoid bootstrapping problem with the tit for tat selection process and ensure that new peers can join the swarm

Characteristics of the BitTorrent protocol II

- **Piece selection** is about supporting high piece diversity
  - Local Rarest First for piece selection
  - BITFIELD message after handshake, then HAVE messages for downloaded pieces
- **End game mode**
  - To avoid delays in obtaining the last blocks the protocol requests the last blocks from all peers
  - Sends cancel messages for downloaded blocks to avoid unnecessary transmissions
  - When to start the end game mode is not detailed in the specification

Biased neighbor selection

A technique called **biased neighbor selection** has been proposed for reducing cross-ISP traffic

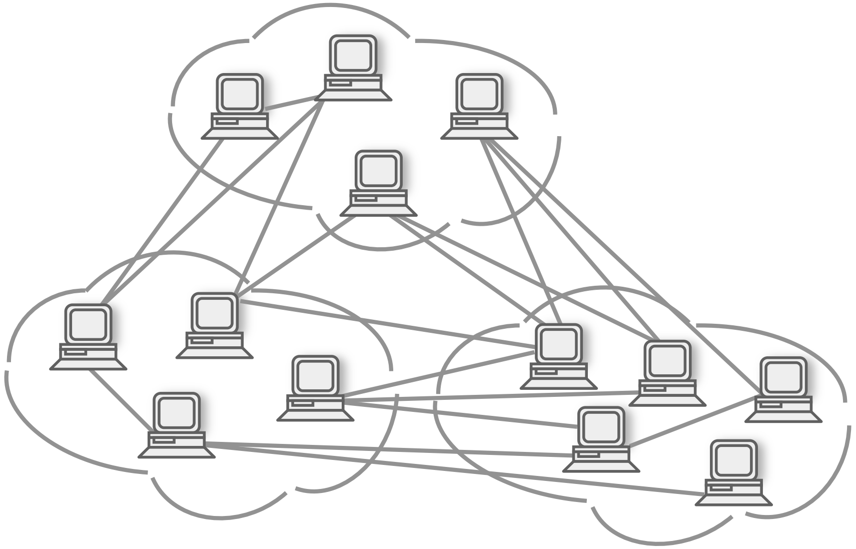
A BitTorrent peer chooses most of its neighbors from the local ISP, and only a few peers from other ISPs.

Essentially, the peer selection is biased towards local peers. A parameter *k* represents the number of external peers from other ISPs. The tracker is modified to select 35 − *k* internal peers and *k* external peers that are returned to the client requesting a peer list for a torrent.

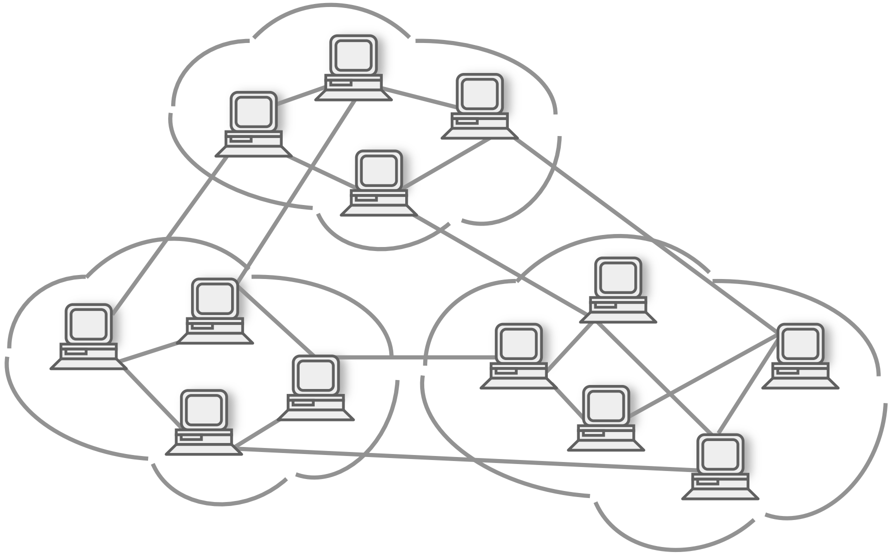
If there are less than 35 − *k* internal peers, the client is notified by the tracker to try again later.

The biased neighbour selection technique works well with the rarest first replication algorithm of BitTorrent; however, other piece selection algorithms, such as random selection, may not lead to optimal performance

BitTorrent: Effects of Network Topology



Uniform random neighbor selection



Biased neighbor selection

Modelling BitTorrent

BitTorrent performance has been analyzed in the literature using analytical models, including stochastic and fluid models, extensive simulation experiments, experiments on distributed testbeds (PlanetLab), and by obtaining traces from real clients

Both analytical and empirical evaluation and estimation are needed to dimension deployments to meet the service capacity demands

Fluid models can be used to analytically estimate the protocol performance and understand the time evolution of the system by using differential equations

Arrival processes

Various different arrival processes for new peers have been considered in the literature. The three key scenarios are as follows:

- The steady flow scenario used above assumes that new peers appear with a constant rate
- The flash crowd scenario, considers the case where a (large) number of peers appear at the same time after which no new peers arrive
- In a third scenario, the arrival rate is high in the beginning but smoothly attenuates as time passes

	BitTorrent
Decentralization	Centralized model
Foundation	Tracker
Routing function	Tracker
Routing performance	Guarantee to locate data, good performance for popular data
Routing state	Constant, choking may occur
Reliability	Tracker keeps track of the peers and pieces

Tomorrow

Tomorrow we have a guest lecture on modelling P2P and BitTorrent performance