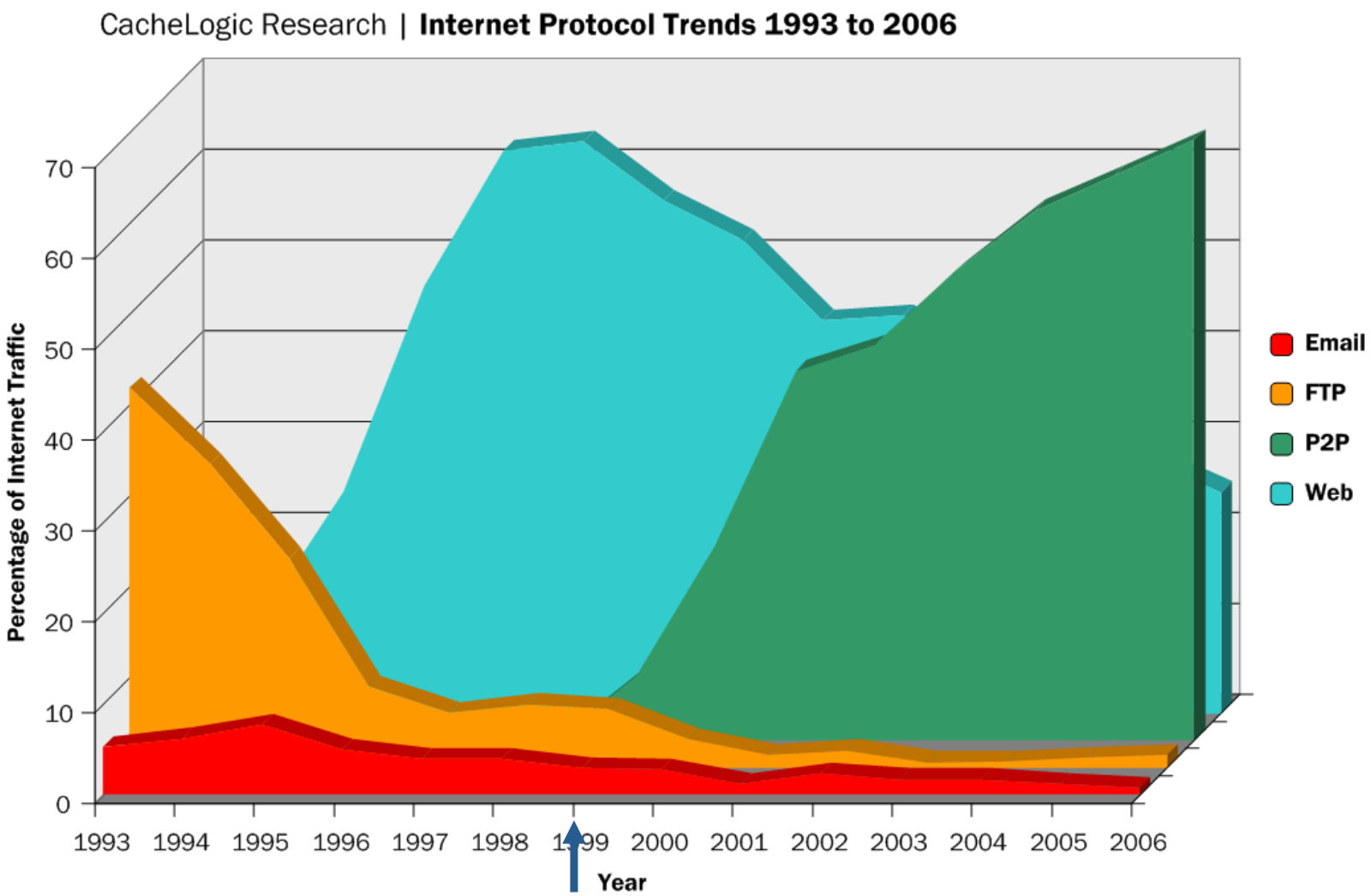
Overlays of Peer to Peer Systems

Franck Petit
Lip6, UPMC

Mainly based on materials by:
- Yang Guo and Christoph Neumann, Corporate Research, Thomson Inc.
- Jon Crowcroft, Univ. of Cambridge
- Jim Kurose, Brian Levine, Don Towsley, UMASS
- Matthew Allen, Univ. of California Santa Barbara
- Manan Rawal, Bhaskar Gupta
- Olivier Marin

- History, motivation and evolution
  - History: Napster and beyond
  - What is Peer-to-peer?
  - Why Peer-to-peer?
- Brief P2P technologies overview
  - Unstructured P2P-overlays
  - Structured P2P-overlays

History, motivation and evolution



P2P represented ~65% of Internet Traffic at end 2006

- 1999: *Napster*, first widely used P2P-application

1999: *Napster*, first widely used P2P application

The application:

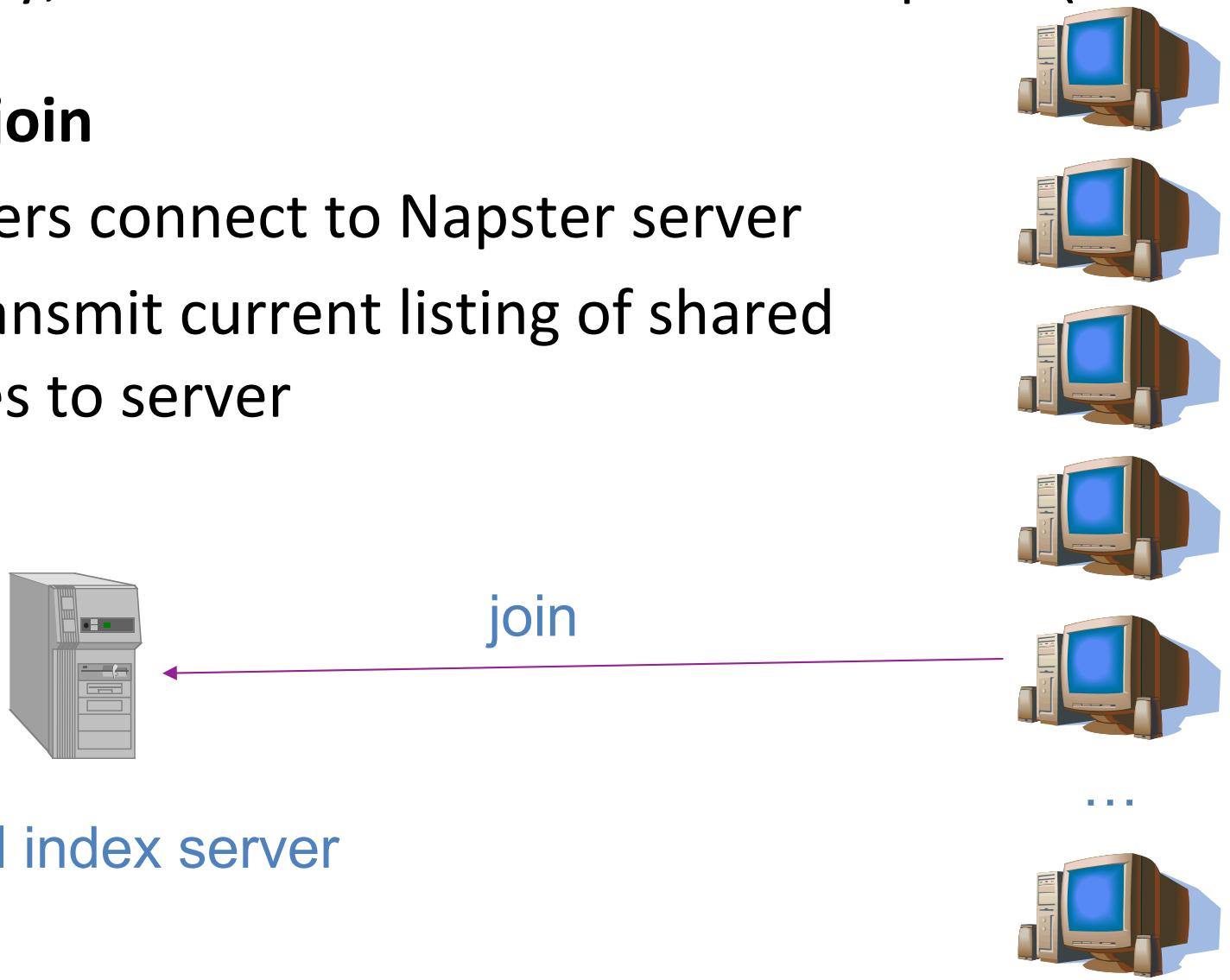
- A P2P application for the distribution of mp3 files
  - Each user can contribute its own content

How it works:

- Central index server
  - Maintains list of all active peers and their available content
- Distributed storage and download
  - Client nodes also act as file servers
  - All downloaded content is shared

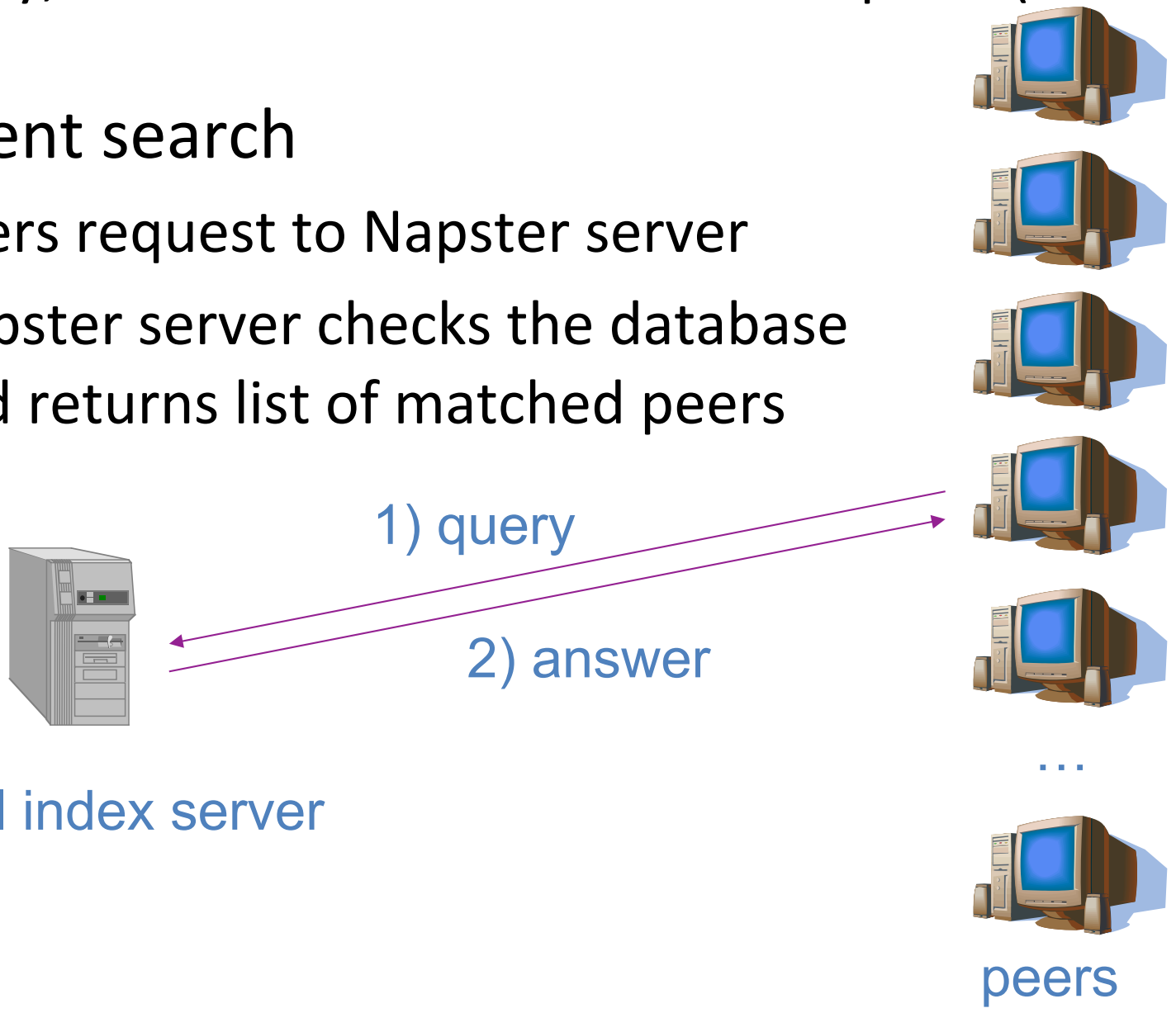
History, motivation and evolution - Napster (cont'd)

- **Initial join**
  - Peers connect to Napster server
  - Transmit current listing of shared files to server



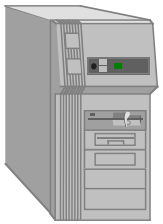
History, motivation and evolution - Napster (cont'd)

- Content search
  - Peers request to Napster server
  - Napster server checks the database and returns list of matched peers

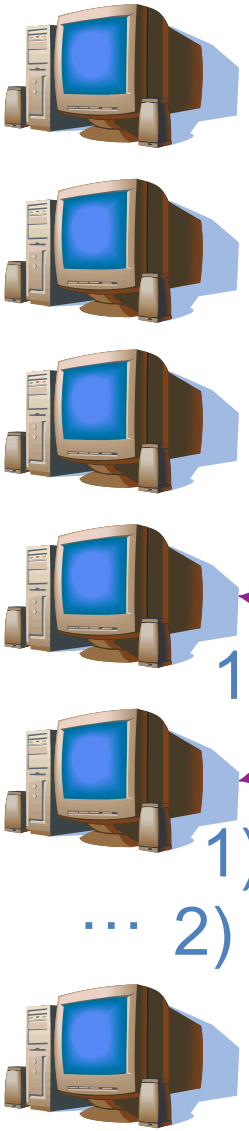


History, motivation and evolution - Napster (cont'd)

- File retrieval
  - The requesting peer contacts the peer having the file directly and downloads the it



Central index server



History, motivation and evolution - File Download

- Napster was the first simple but successful P2P-application. Many others followed…

P2P File Download Protocols:

- 1999: Napster (closed in 2001)
- 2000: Gnutella, eDonkey
- 2001: Kazaa (closed in 2012)
- 2002: eMule, BitTorrent

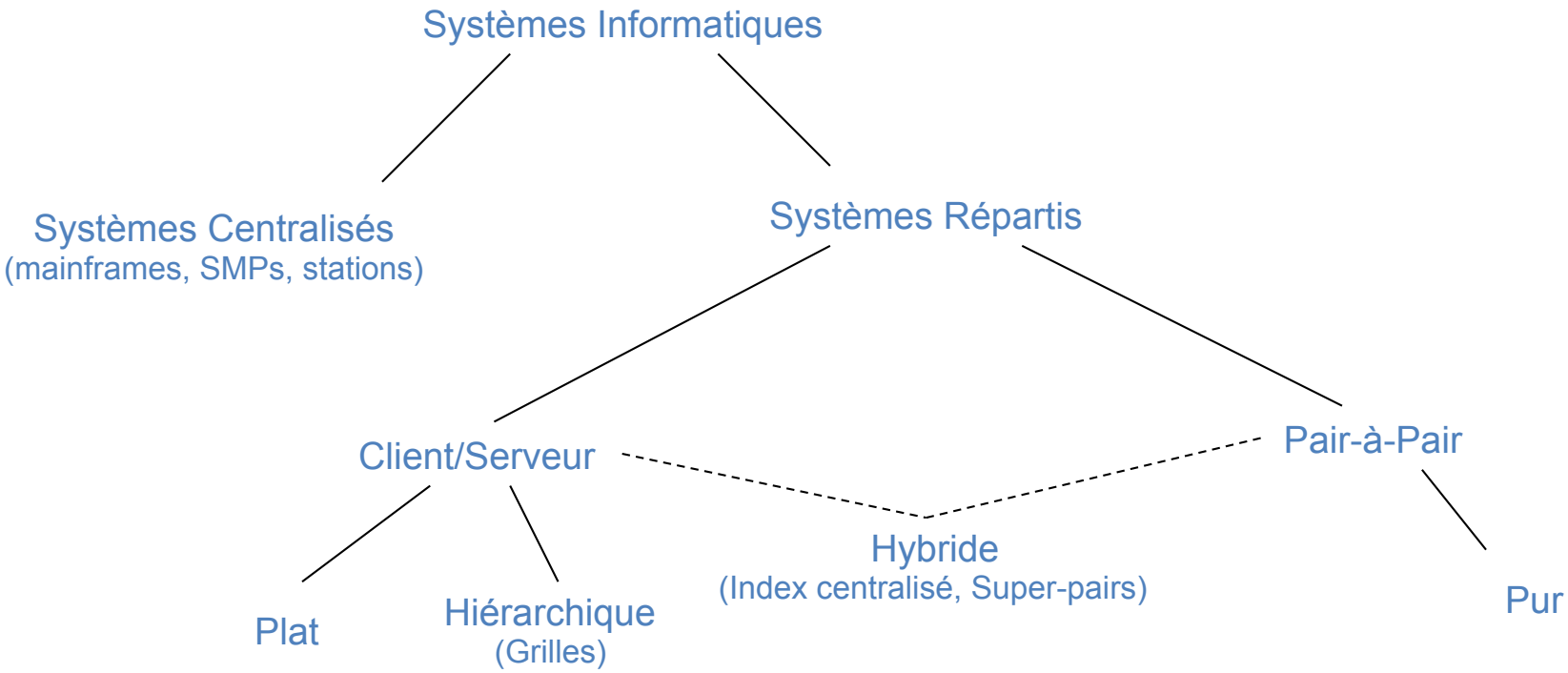
- History, motivation and evolution
  - History: Napster and beyond
  - What is Peer-to-peer?
  - Why Peer-to-peer?
- Brief P2P technologies overview
  - Unstructured P2P-overlays
  - Structured P2P-overlays

Definition of Peer-to-peer (or P2P)

- *DEFINITION 1* (early 2000s):
  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 small number of servers.
- *DEFINITION 2* (2009-):
  A peer-to-peer (or P2P) is any **distributed network architecture** composed of **participants** that make a portion of their resources (such as processing power, disk storage or network bandwidth) directly available to other network participants, without the need for central coordination instances.

Taken from the wikipedia free encyclopedia - www.wikipedia.org

The Big Picture

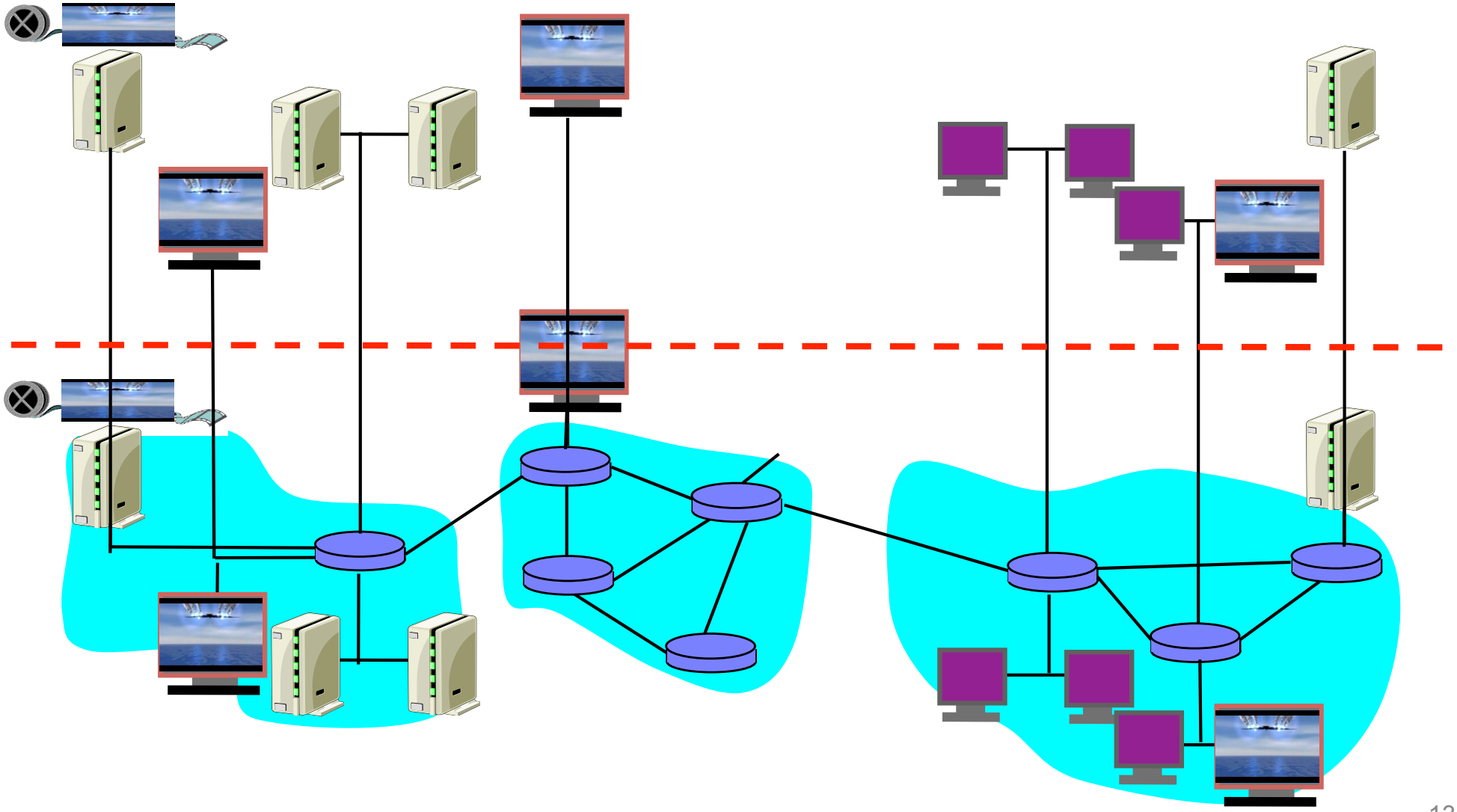


Peer to peer systems

- Assumed to be with **no distinguished role**
- So no single point of bottleneck or failure
- However, this means they need **distributed algorithms** for
  - Connection protocol
  - Service discovery (name, address, route, metric, etc)
  - Neighbour status tracking
  - Application layer routing (based possibly on content, interest, etc)
  - Resilience, handling link and node failures
  - etc

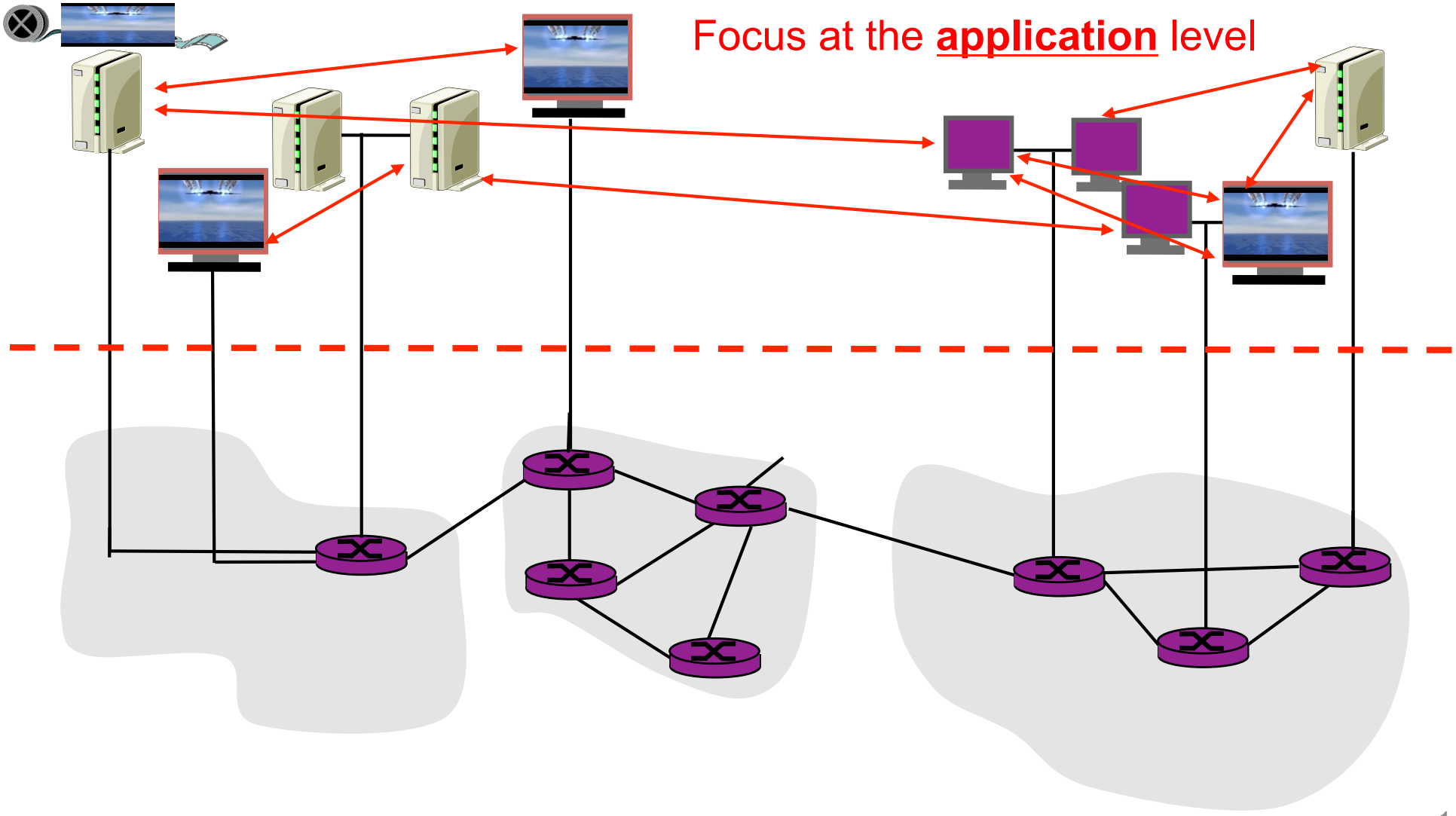
Overlays and peer 2 peer systems

- P2P systems rely on **overlays** structure

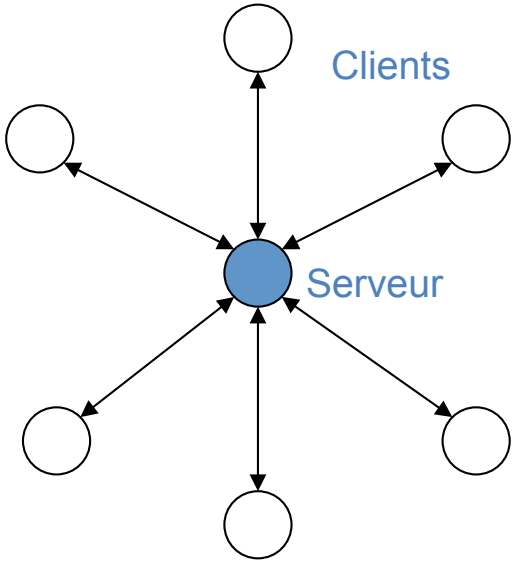


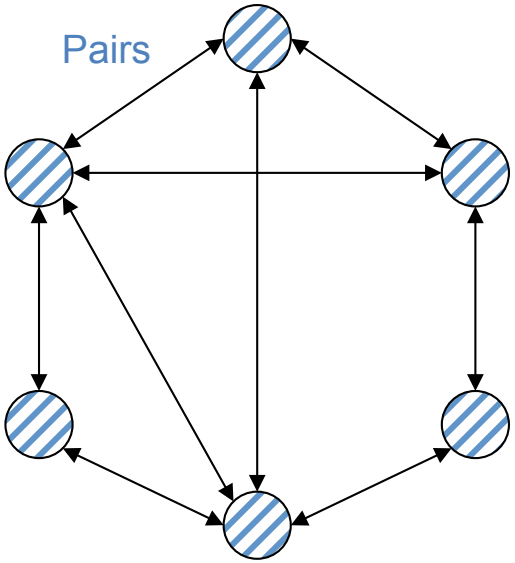
Overlays and peer 2 peer systems

- P2P systems rely on **overlays** structure



C/S vs. P2P





C/S vs. P2P

- Gérés
- Configurés
- Recherche de services
- Hiérarchique
- Ressources statiques
- Cycle de vie lié au serveur
- Centré IP
- Nommage basé sur le DNS
- Communications type RPC/RMI
- Synchrone
- Asymétrique
- Axé sur des modèles de liaison et d'intégration du langage de programmation (stub IDL/XDR, compilateurs, etc...)
- Sécurité de type Kerberos : acl, crypto

- Auto-Gérés
- Ad-hoc
- Découverte de services
- Maillage
- Ressources volatiles
- Cycle de vie autonome
- Non restrictif à IP
- Nommage spécifique
- Communication par messages
- Asynchrone
- Symétrique
- Axé sur la localisation de services, localisation du contenu, routage applicatif
- Anonymat, haute disponibilité
- Plus difficile à maîtriser

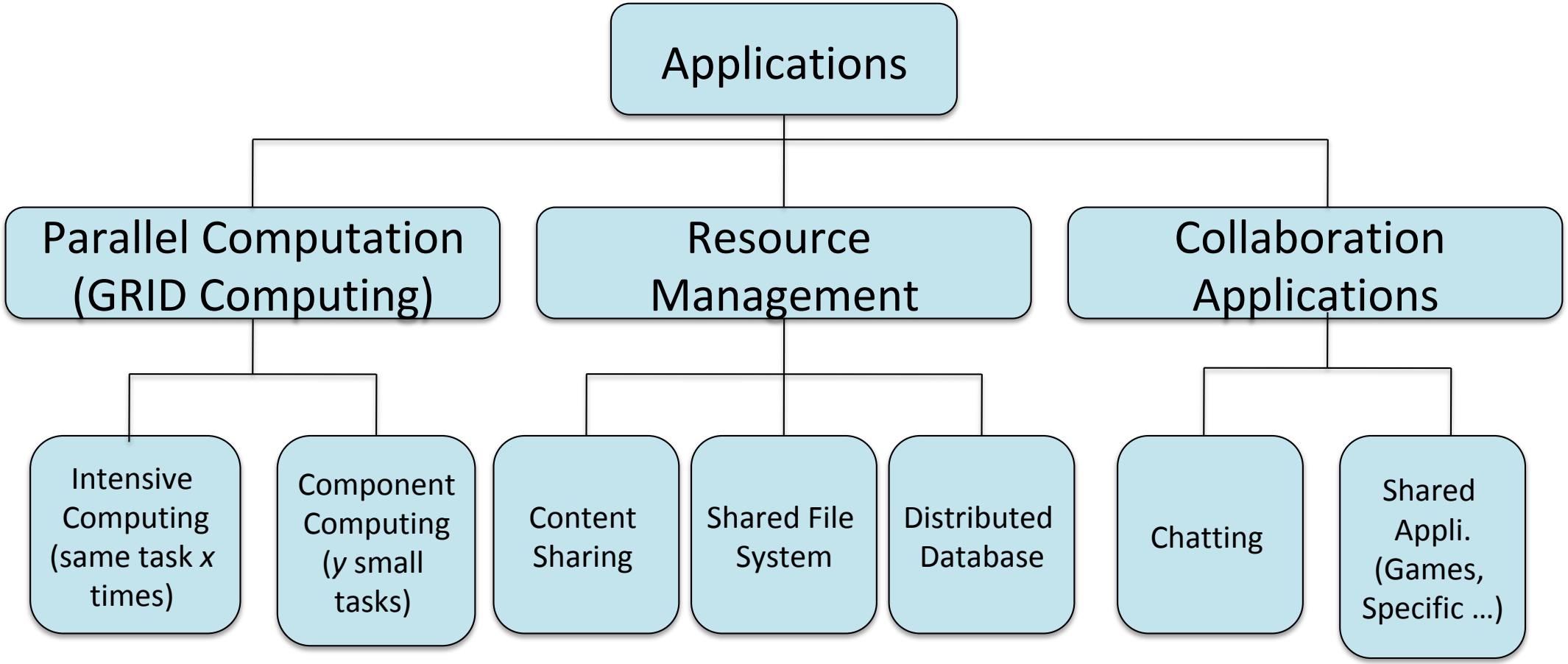
P2P : Nouveaux besoins algorithmiques

- Découverte de services (nom, adresse, route, métrique, ...)
- Recherche de voisins
- Routages spécifiques, de niveau applicatif
- Rémanence, récupération sur faute de liaison ou d'exécution

P2P : Contraintes algorithmiques

- Structure du système
  - totalement décentralisée
  - dynamique
- Diffusion à redéfinir
  - Pas d'ensemble identifié de processus : nombre inconnu et variable
  - Topologie d'interconnexion peut aussi être inconnue et dynamique
  - Symétrie entre sites (client **et** serveur)
- Modèle de panne **avec** fautes

Taxonomy of Applications



Taxonomy of Applications

- P2P-File download
  - Napster, Gnutella, KaZaa, eDonkey, Bittorrent…
- P2P-Communication
  - VoIP, Skype, Messaging, …
- P2P-Video-on-Demand
- P2P-Computation
  - GRID, scientific computation (seti@home, XtreemOS, BOINC,…)
- P2P-Streaming
  - PPLive, End System Multicast (*ESM*),…
- P2P-Gaming
  - WOW, City of Heroes,…

History, motivation and evolution - Applications

- P2P is not restricted to file download!

P2P Protocols:

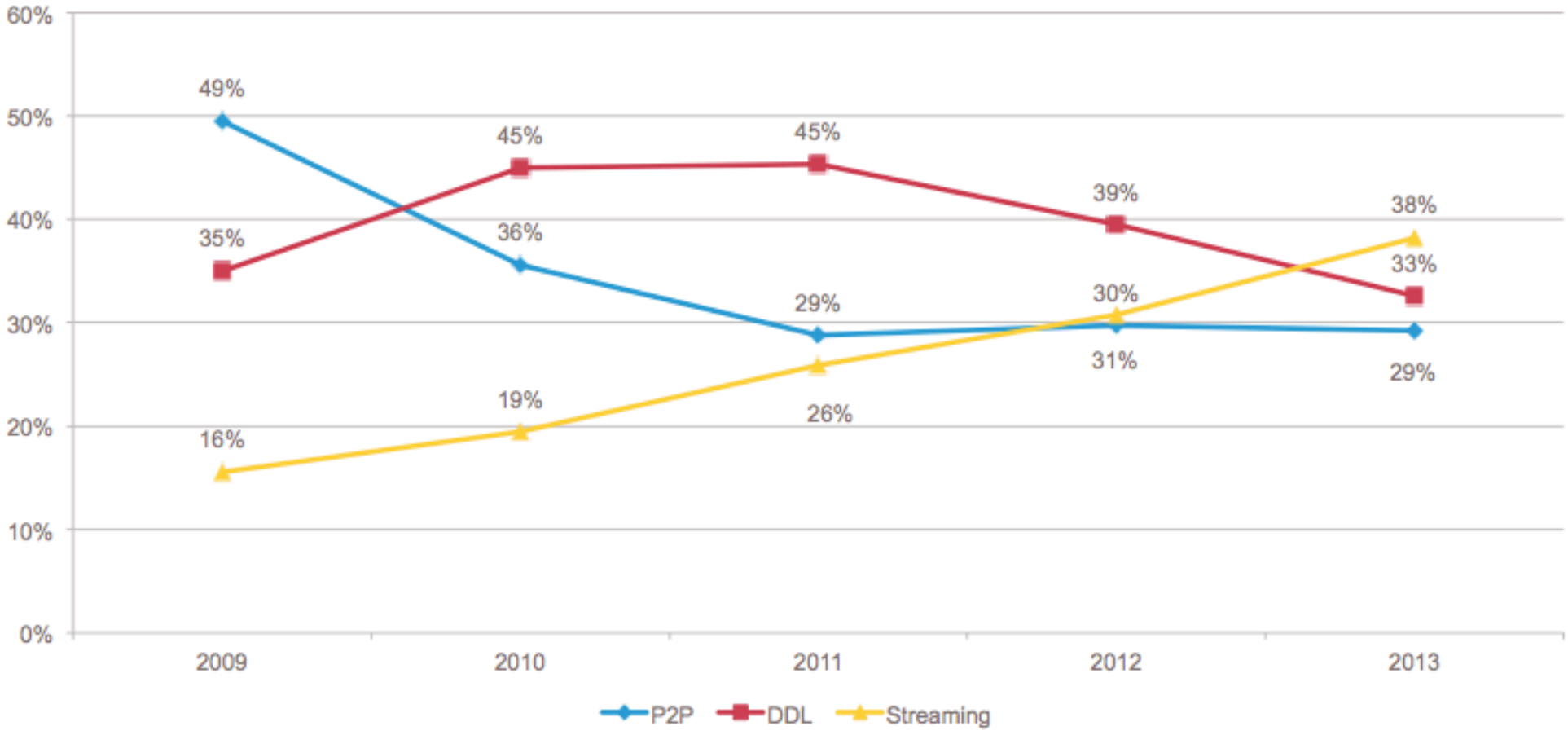
- 1999: Napster, End System Multicast (ESM)
- 2000: Gnutella, eDonkey
- 2001: Kazaa
- 2002: eMule, BitTorrent
- 2003: Skype
- 2004: PPLive
- 2006: TVKoo, TVAnts, PPStream, SopCast, Video-on-Demand, Gaming

Application type:

File Download
Streaming
Telephony
Video-on-Demand
Gaming

Evolution

Évolution de la répartition des usages (nombre de pages vues) par protocole depuis 2009

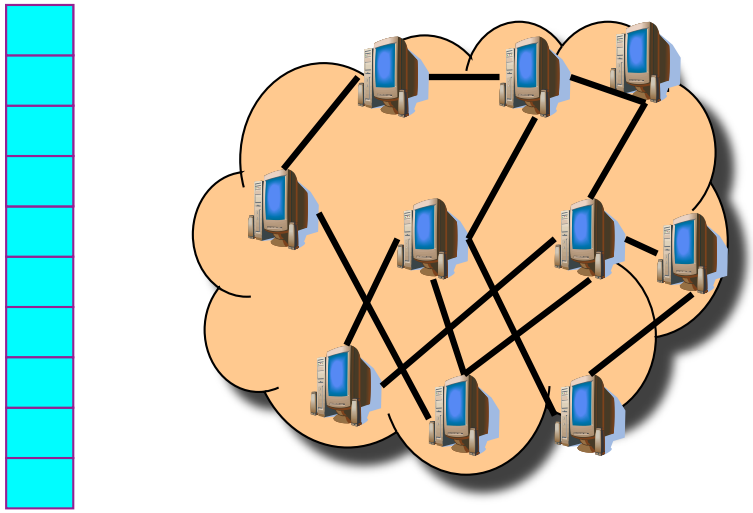


- History, motivation and evolution
  - History: Napster and beyond
  - What is Peer-to-peer?
  - Why Peer-to-peer?
- Brief P2P technologies overview
  - Unstructured P2P-overlays
  - Structured P2P-overlays

Why is P2P so successful?

- Scalable – It's all about sharing resources
  - No need to provision servers or bandwidth
  - Each user brings its own resource
  - *e.g.* resistant to flash crowds
    - flash crowd = a crowd of users all arriving at the same time

capacity



Resources could be:

- Files to share;
- Upload bandwidth;
- Disk storage;…

Why is P2P so successful? (cont'd)

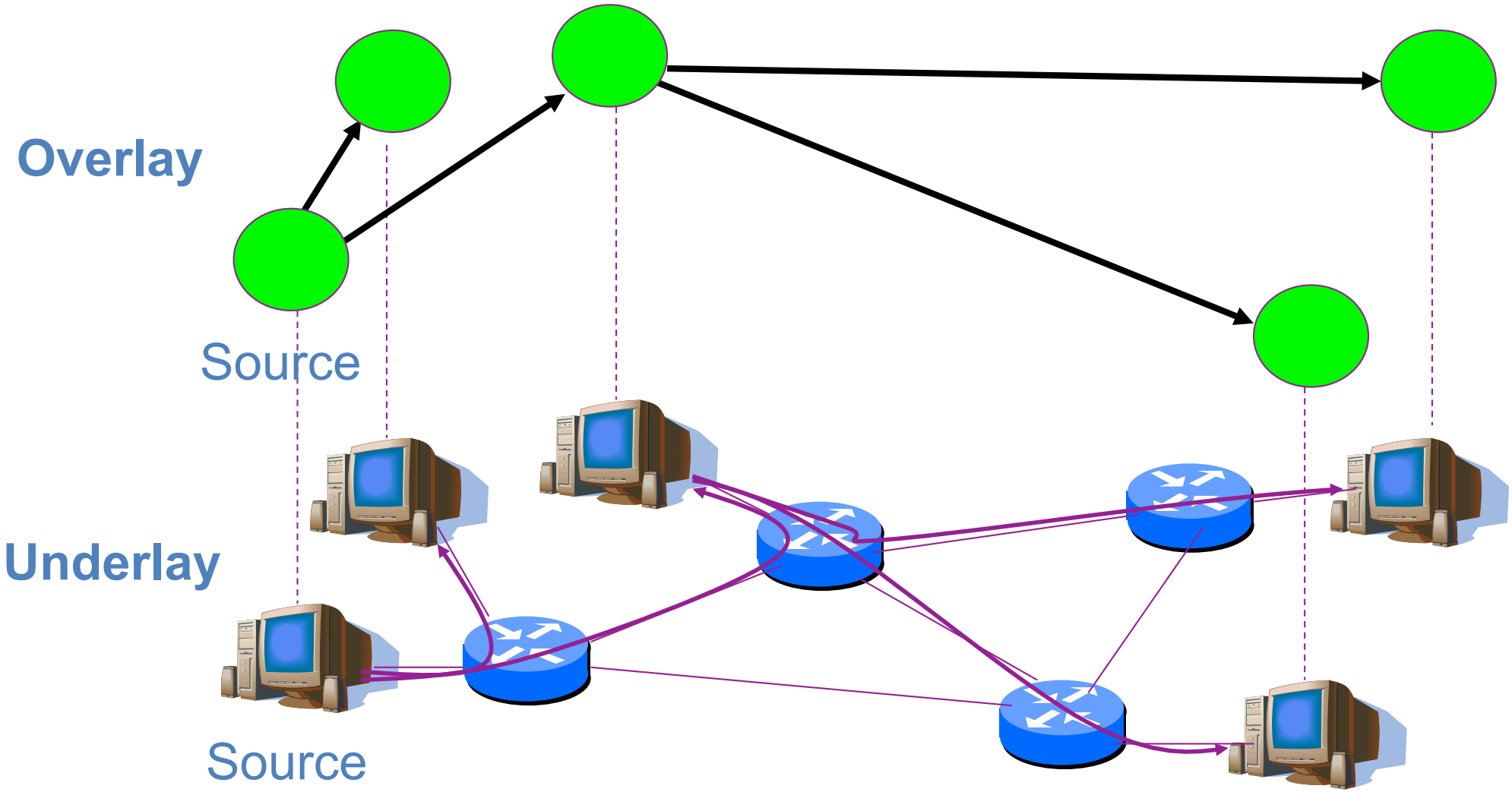
- Cheap - No infrastructure needed
- Everybody can bring its own content (at no cost)
  - Homemade content
  - Ethnic content
  - Illegal content
  - But also *legal* content
  - …
- High availability – Content accessible most of the time (replication)

- History, motivation and evolution
  - History: Napster and beyond
  - What is Peer-to-peer?
  - Why Peer-to-peer?
- Brief P2P technologies overview
  - Unstructured P2P-overlays
  - Structured P2P-overlays

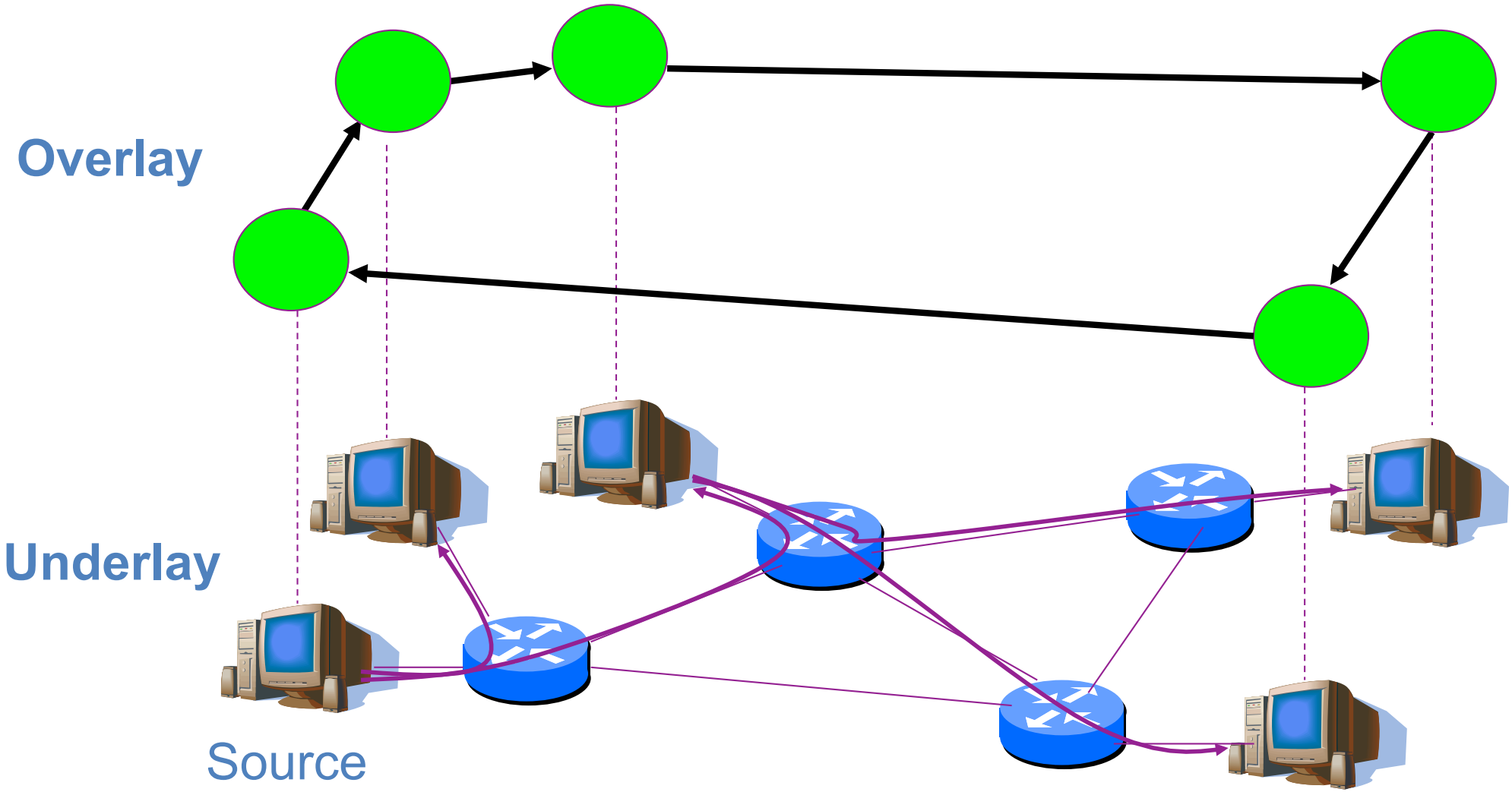
P2P-Overlay

- Build graph at application layer, and forward packet at the application layer
- It is a *virtual* graph
  - Underlying physical graph is transparent to the user
  - Edges are TCP connection or simply a entry of an neighboring node's IP address
- The graph has to be continuously maintained (e.g. check if nodes are still alive)

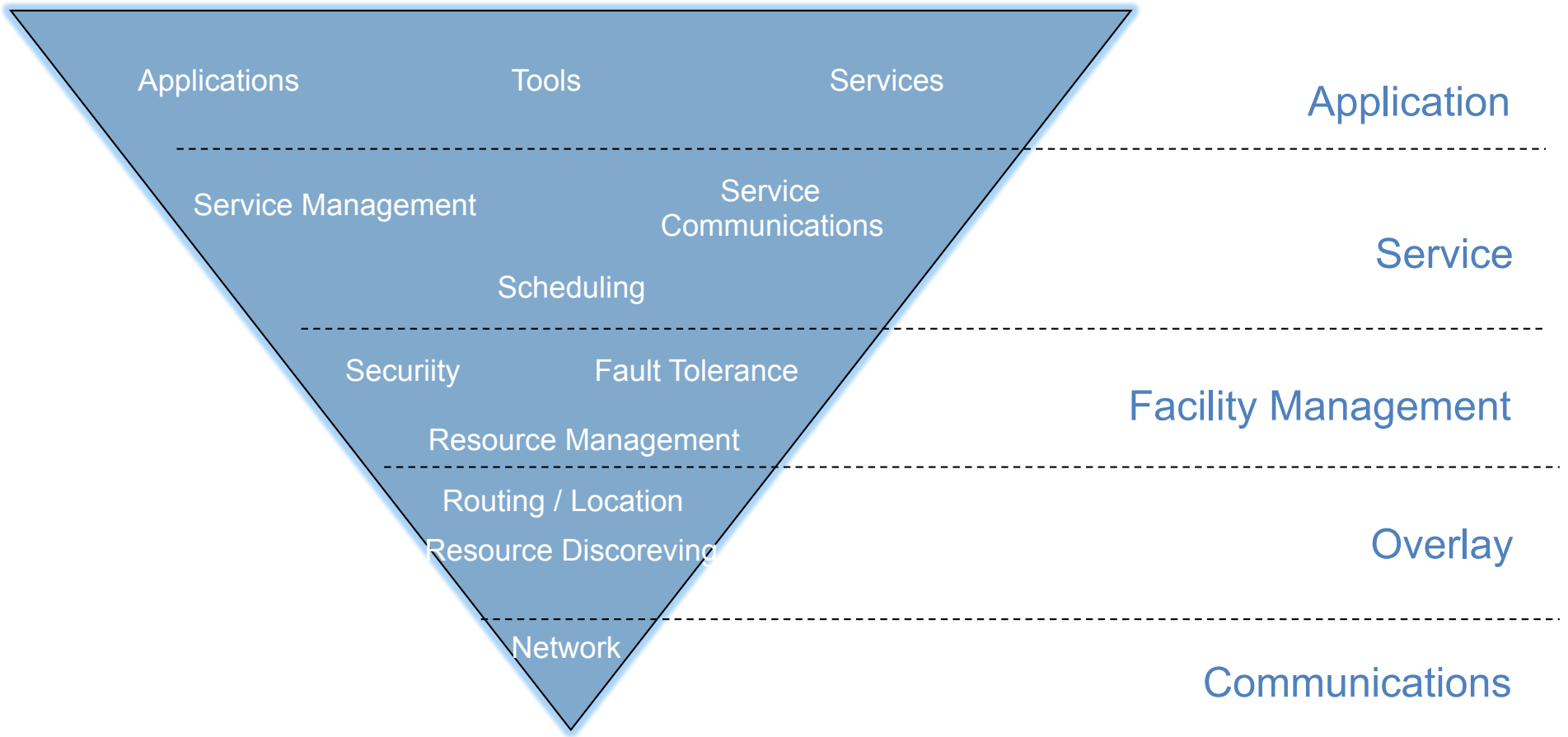
P2P-Overlay (cont'd)



P2P-Overlay (cont'd)



P2P-Overlay Abstration



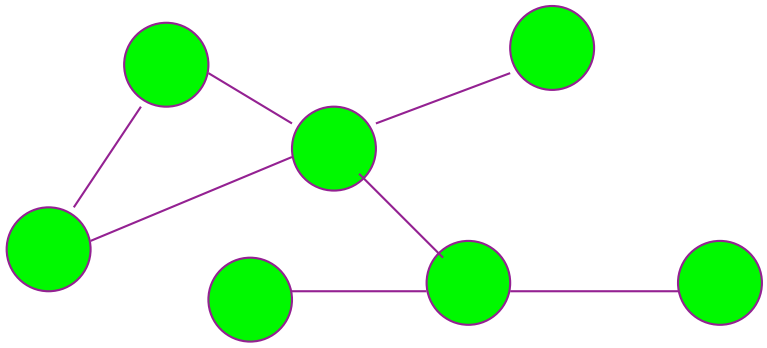
The P2P enabling technologies

- Unstructured P2P-overlays
  - Generally random overlay
  - Used for content download, telephony, streaming
- Structured P2P-overlays
  - Distributed Hash Tables (DHTs)
  - Used for node localization, content download, streaming

- History, motivation and evolution
  - History: Napster and beyond
  - What is Peer-to-peer?
  - Why Peer-to-peer?
- Brief P2P technologies overview
  - Unstructured P2P-overlays
  - Structured P2P-overlays

Unstructured P2P-overlays

- Unstructured P2P-overlays do not really care how the overlay is constructed
  - Peers are organized in a random graph topology
    - *e.g.*, new node randomly chooses three existing nodes as neighbors
    - Flat or hierarchical
  - Build your P2P-service based on this graph
- Several proposals
  - Gnutella
  - KaZaA/FastTrack
  - BitTorrent



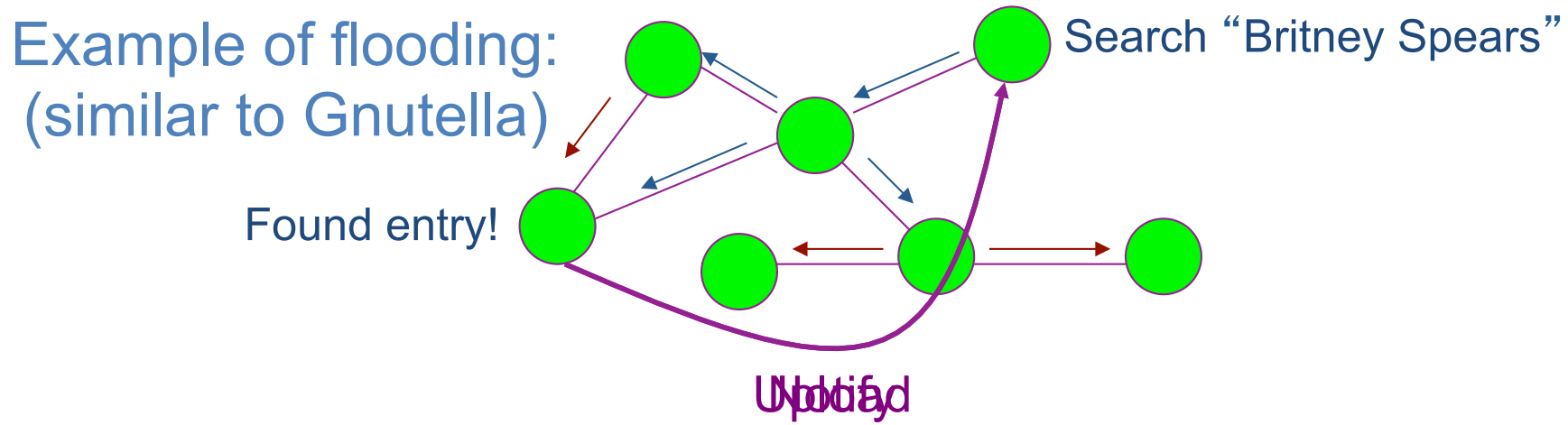
Unstructured P2P-overlays (cont'd)

- Unstructured P2P-overlays are just a framework, you can build many applications on top of it
- "Pure" P2P

- Unstructured P2P-overlays pros & cons
  - Pros
    - Very flexible: copes with dynamicity
    - Supports complex queries (conversely to structured overlays)
  - Cons
    - Content search is difficult: There is a tradeoff between generated traffic (overhead) and the horizon of the partial view

One Example of usage of unstructured overlays

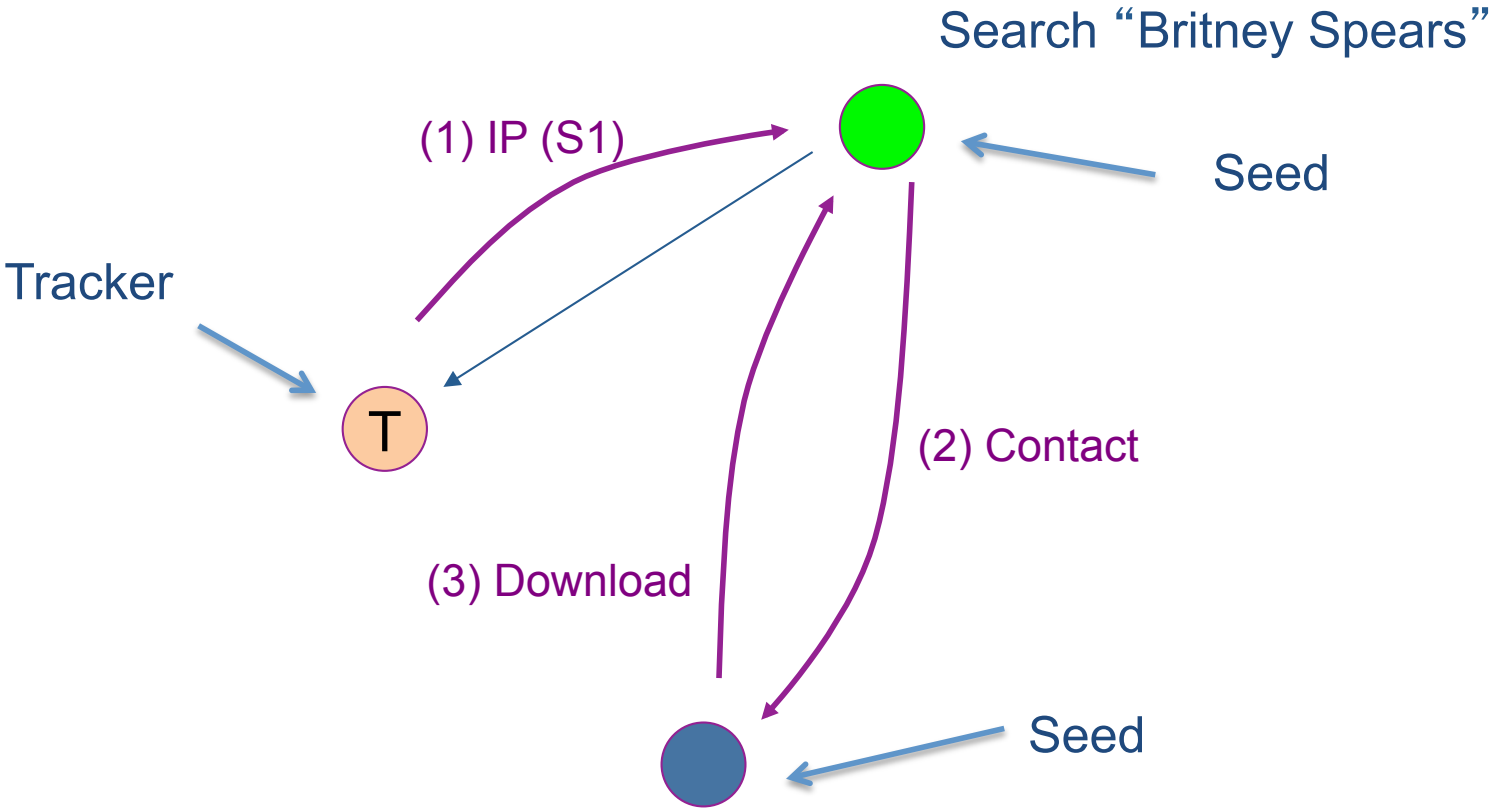
- Typical problem in unstructured overlays: How to do content search and query?
  - Flooding



  - Limited Scope, send only to a subset of your neighbors
  - Time-To-Live, limit the number of hops per messages

Another Example of usage of unstructured overlays

- Bittorent

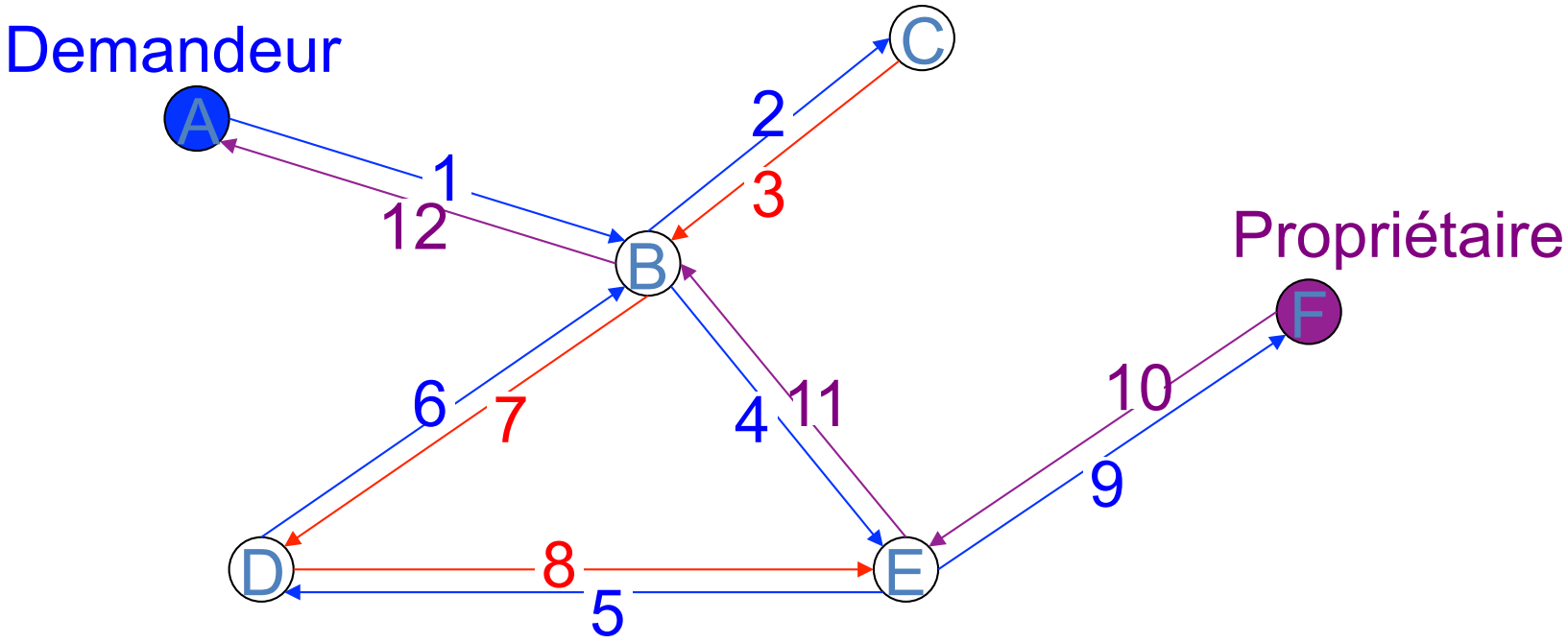


Semi-structuration : Freenet

Stockage persistant de données et services [Freenet00]

- Nœuds/Données identifiés par une clé binaire (fonction hash)
  - Identifiant de nœud : NodeID(utilisateur) = hash(@IP)
  - Identifiant (*clé*) de fichier : FileID(fichier) = hash(contenu)
- Gestion de tables de routage
  - Construction de voisinages
    - ⇒ Connaissance approximative du contenu des nœuds proches

Semi-structuration : Freenet



Séquence typique de routage de requête

- Dissémination de proche en proche
- Gestion de cul-de-sac (3) et de boucle (7)

Gnutella *vs.* FreeNet

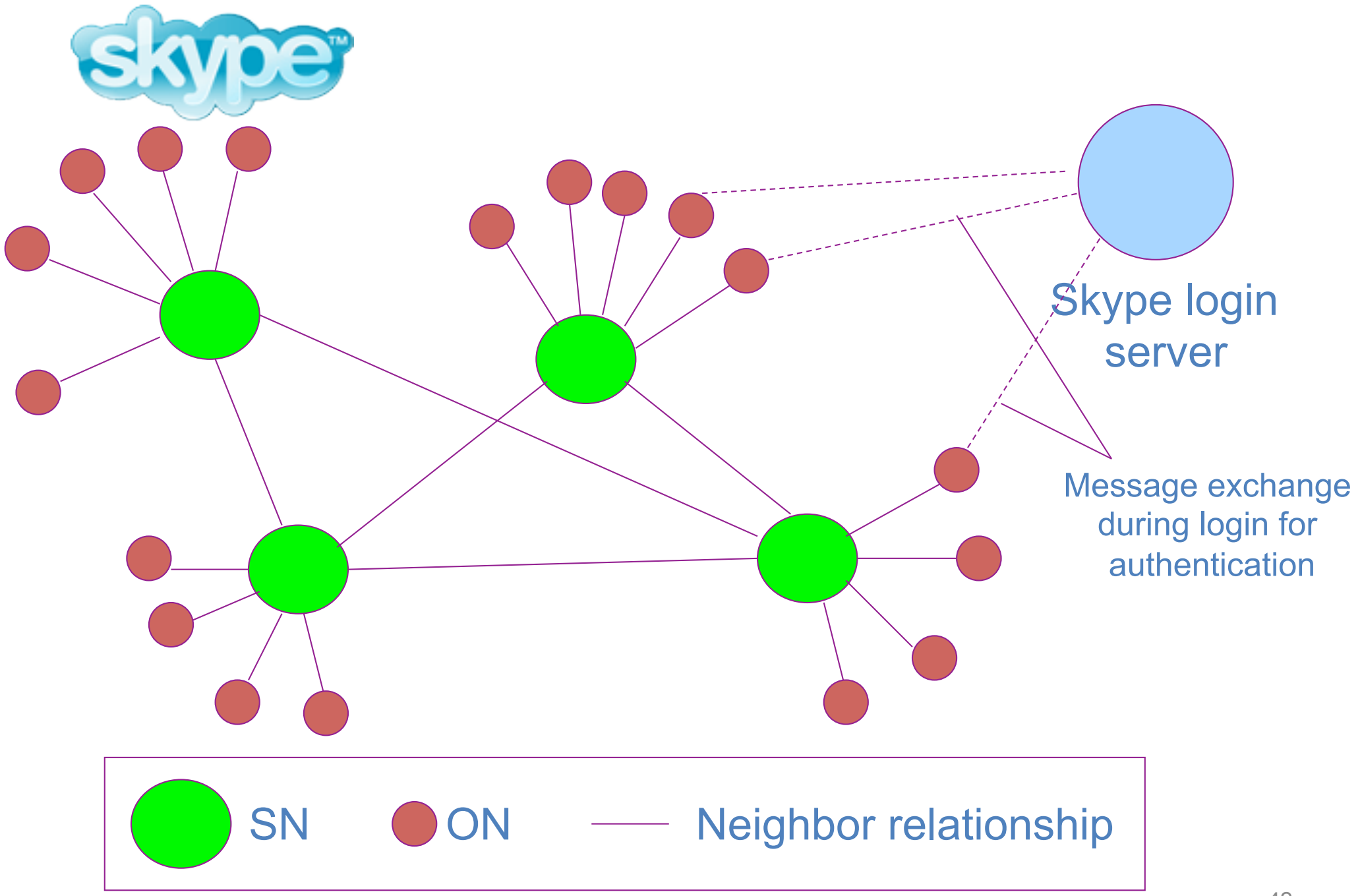
- Routage basé sur la diffusion (flooding)
- Aucune mémoire du trafic véhiculé
- Read-only
- Système non sécurisé

- Routage dynamique basé sur la similarité (distance) des clés
- Tables de routage + Cache
- Read/Write
- Système sécurisé

Collaboration

- Transfert de communications (VoIP)
- Pairs partagent leur bande passante
- Annuaire totalement décentralisé
- Routage au moyen de super-nœuds
- 246 millions d'utilisateurs (2014)
- Gros problèmes de sécurité

- Super Nodes (SN)
  - Connect to each other, building a flat unstructured overlay (similar to the Gnutella overlay)
- Ordinary Nodes (ON)
  - Connect to Super Nodes that act as a directory server (similar to the index server in Napster)
- Skype login server
  - Only central component
  - Stores and verifies usernames and passwords
  - Stores the buddy list


skype
Skype login server
Message exchange during login for authentication
SN
ON
Neighbor relationship

- Each node keeps a host cache with a list of Super Nodes IP-addresses
  - Up to 200 entries
- Some Super Nodes IP-addresses are hard-coded
  - Super Nodes provided by Skype

- These lists are used to locate a nodes Super Node at login

How is the overlay constructed? - Super Node Lists

- Each node keeps a host cache with a list of Super Nodes IP-addresses
  - Up to 200 entries
- Some Super Nodes IP-addresses are hard-coded
  - Super Nodes provided by Skype
- These lists are used to locate a nodes Super Node at login

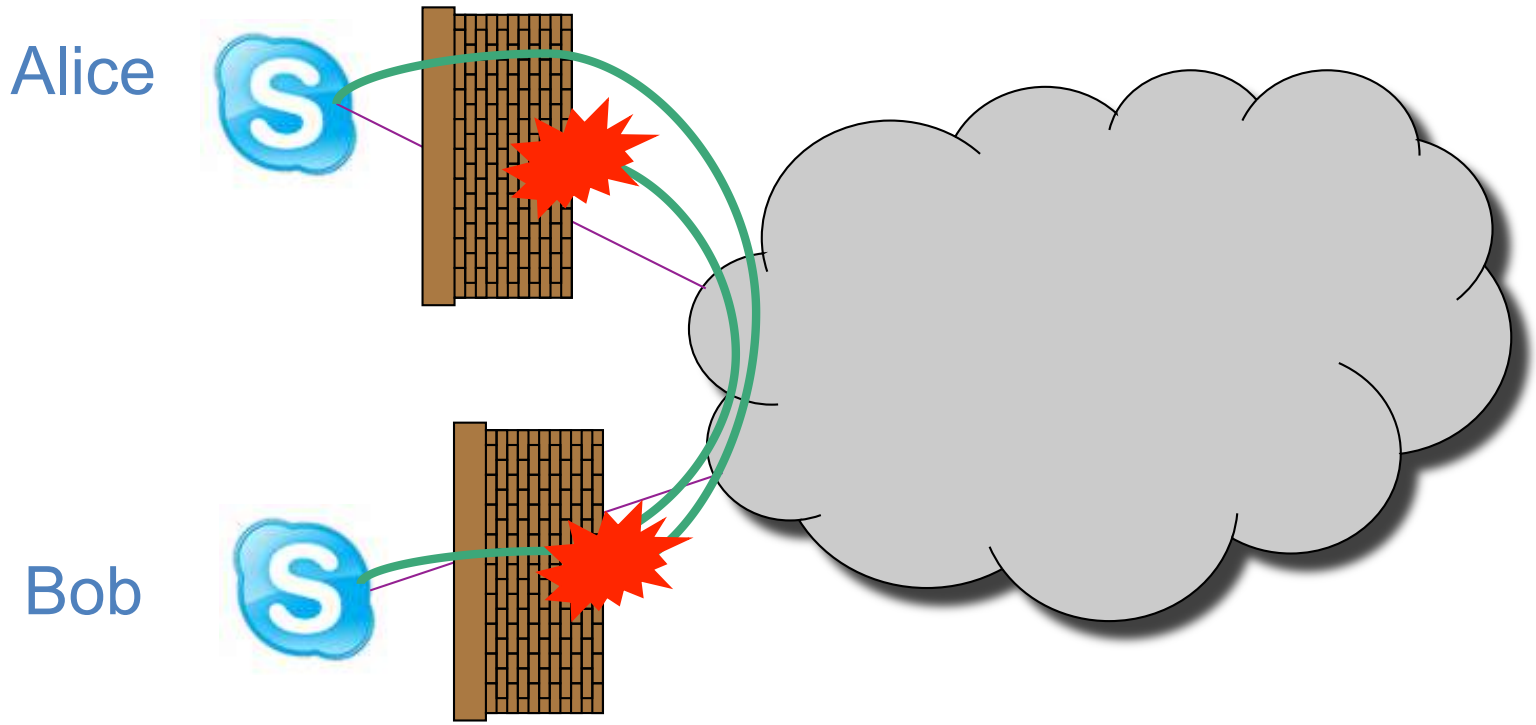
How is the overlay constructed? - Super Node Lists

- Super Nodes are index servers
  - I.e. index of locally connected Skype users (and their IP addresses)
- If buddy is not found in local index of a Super Node
  - Spread node search to neighboring Super Nodes … by flooding (not clear the details, probably similar to Gnutella)

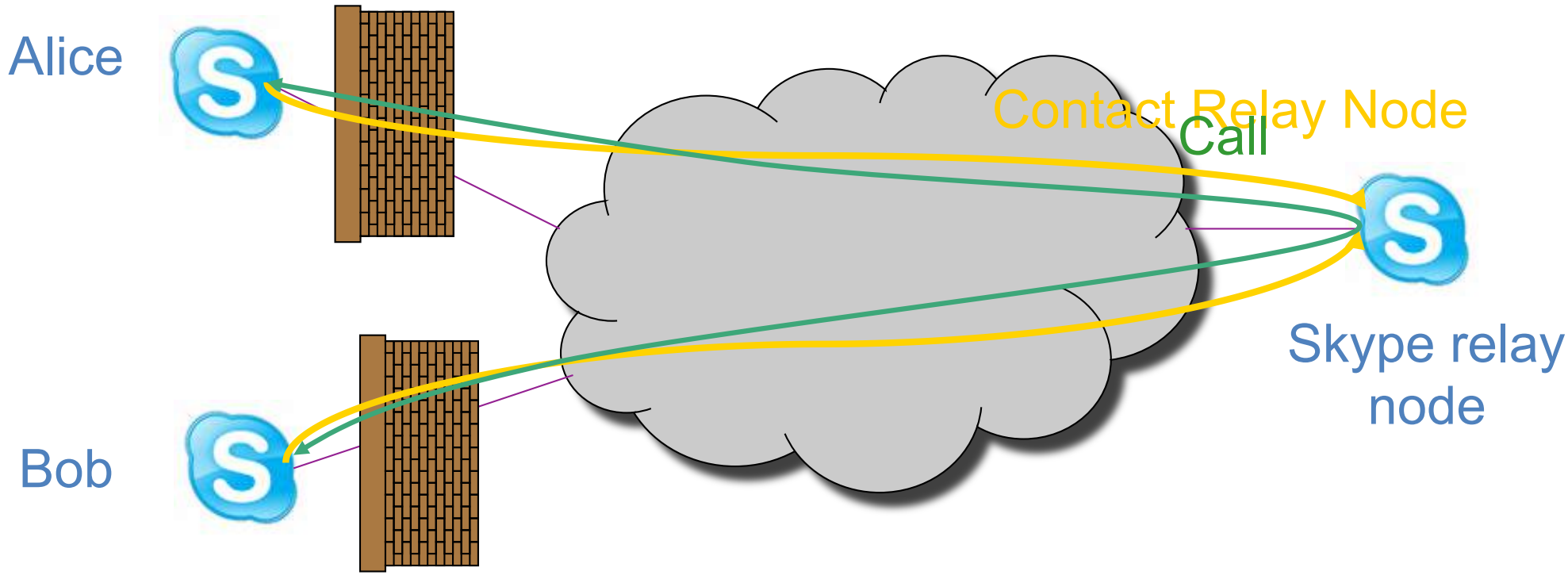
How is the overlay constructed? - Super Node Lists

- Contact login server and authenticate
- Advertise your presence to other peers
  - Contact a Super Node
  - Contact your buddies (through Super Node), and notify your presence

- Alice would like to call Bob (or inversely)



- Alice would like to call Bob (or inversely)

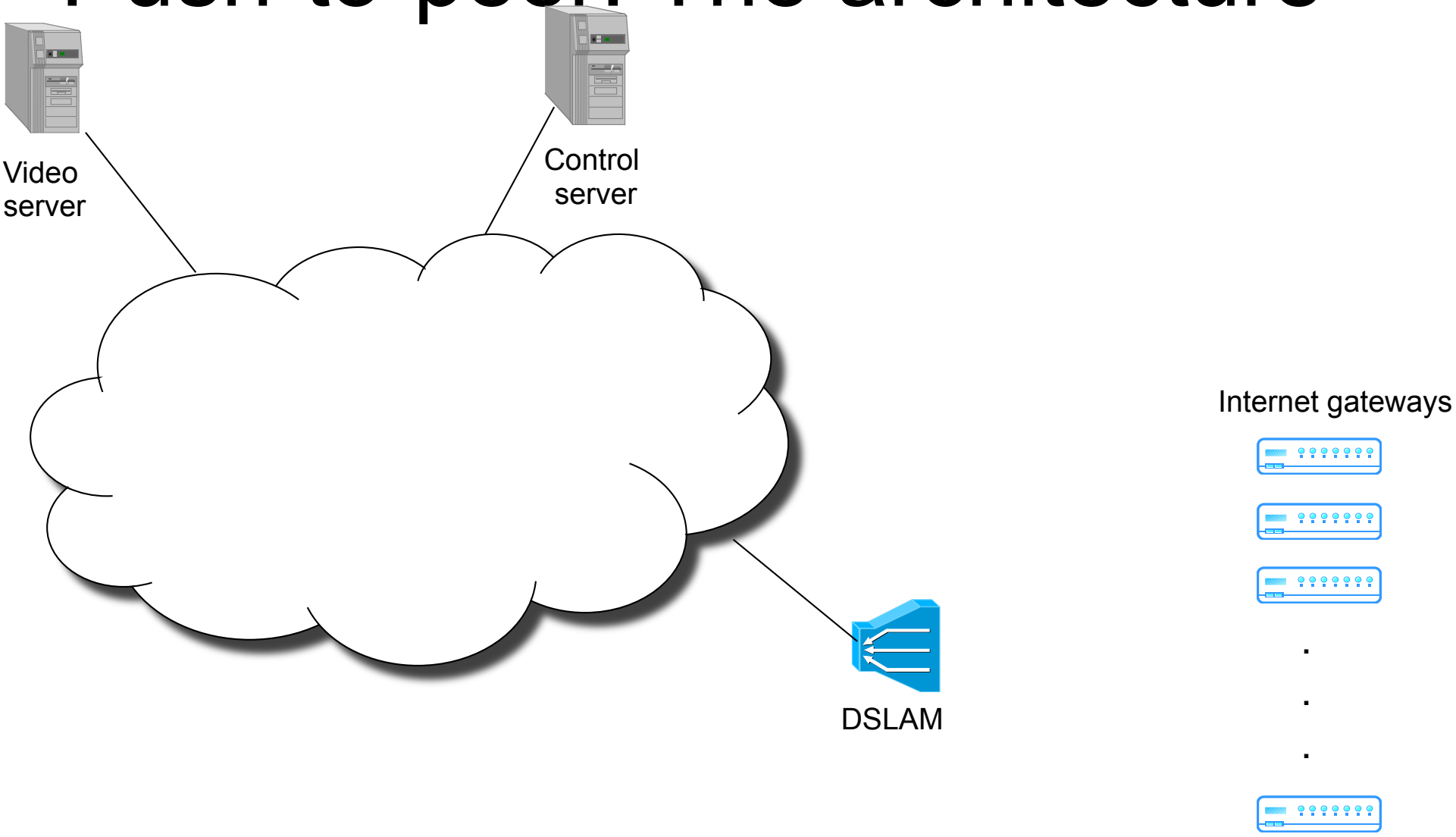


Project: Push-to-peer

Goal

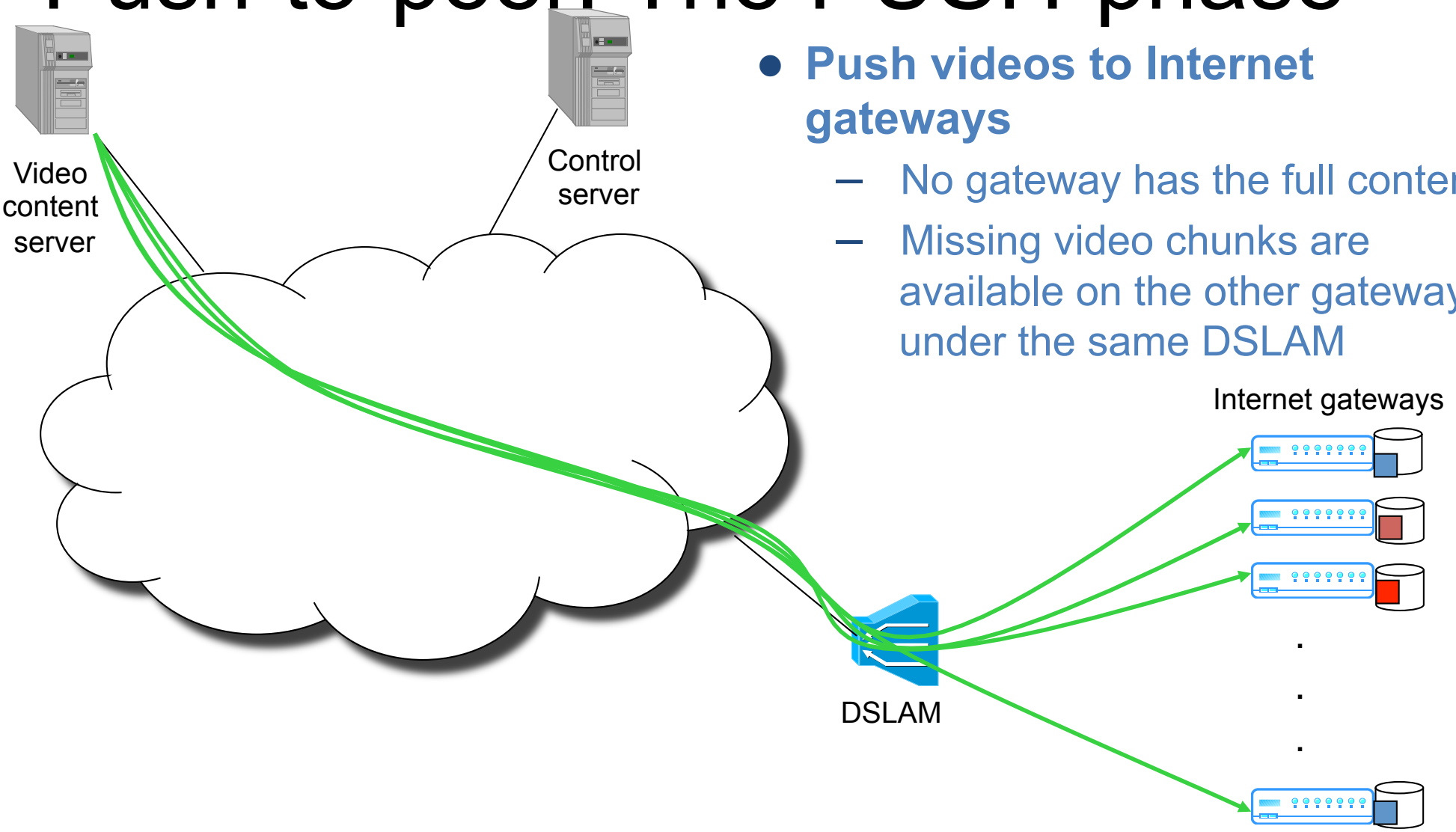
- Provide a Video-on-Demand service to Internet gateways and Set-Top-Boxes

Push-to-peer: The architecture



50

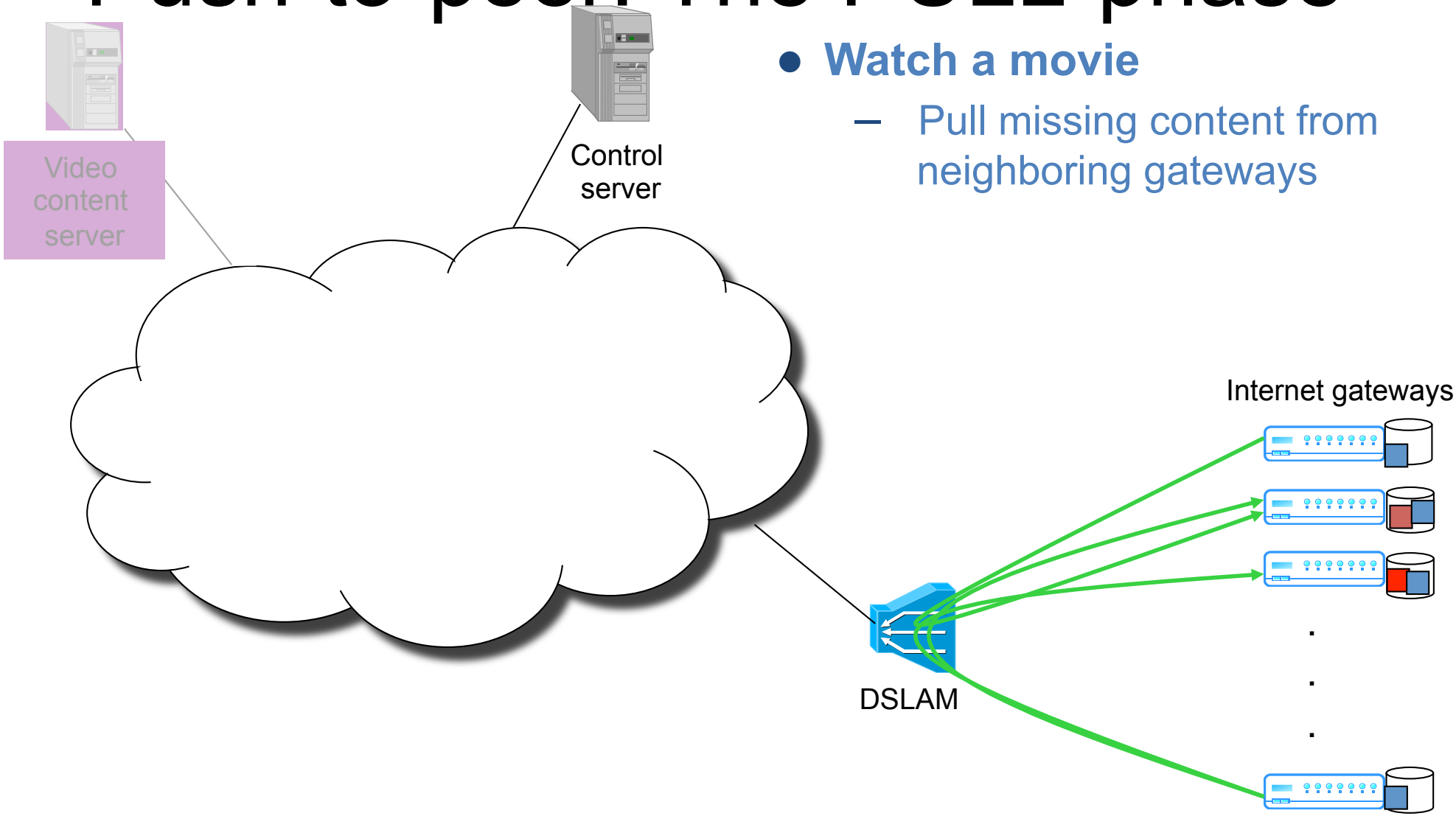
Push-to-peer: The PUSH-phase



- **Push videos to Internet gateways**
  - No gateway has the full content
  - Missing video chunks are available on the other gateways under the same DSLAM

Push-to-peer: The PULL-phase

- **Watch a movie**
  - Pull missing content from neighboring gateways



Why Push-to-peer?

- No ISP bandwidth consumption beyond DSLAM
- Retains advantages of content server-based solution:
  - Under full control of ISP;
  - Guaranteed content safety;
  - Short playback delays;
- But at a lower cost:
  - More robust (content server: single point of failure);
  - No need to provision content server uplink b/w;
  - Uses Internet gateway's storage

Push-to-peer: Quick technical overview

Assumptions

- A centralized control server is available
  - Needed anyhow for billing
  - Coordinates all gateways
  - Knows where each content is located
- The video server is not used at all in pull phase
  - It is owned by the content owner
  - We don't have any guarantee with respect to the performances of that server

- History, motivation and evolution
  - History: Napster and beyond
  - What is Peer-to-peer?
  - Why Peer-to-peer?
- Brief P2P technologies overview
  - Unstructured P2P-overlays
  - Structured P2P-overlays