Antonio Liotta and Adetola Oredope
Department of Computing and Electronic Systems
University of Essex, UK
liotta.antonio@gmail.com; tolaoredope@gmail.com

P2P-SIP: A STANDARDISED APPROACH TO MULTIMEDIA COMMUNICATIONS

Acknowledgement

- EU PHOSPHORUS (results presented in this tutorial were achieved using the PHOSPHORUS Test bed infrastructure) - http://www.ist-phosphorus.eu/
- Distributed K –ary System (DKS) Slides - http://dks.sics.se/
- Gonzallo Camrillo - http://internetng.dit.upm.es/ponencias-jing/2005/SIP.pdf
- IBM Israel Software Lab (ILSL) - http://www.haifa.ibm.com/
- Notes from Prof Raoulf Boutaba, University of Waterloo, Canada

Other reference material

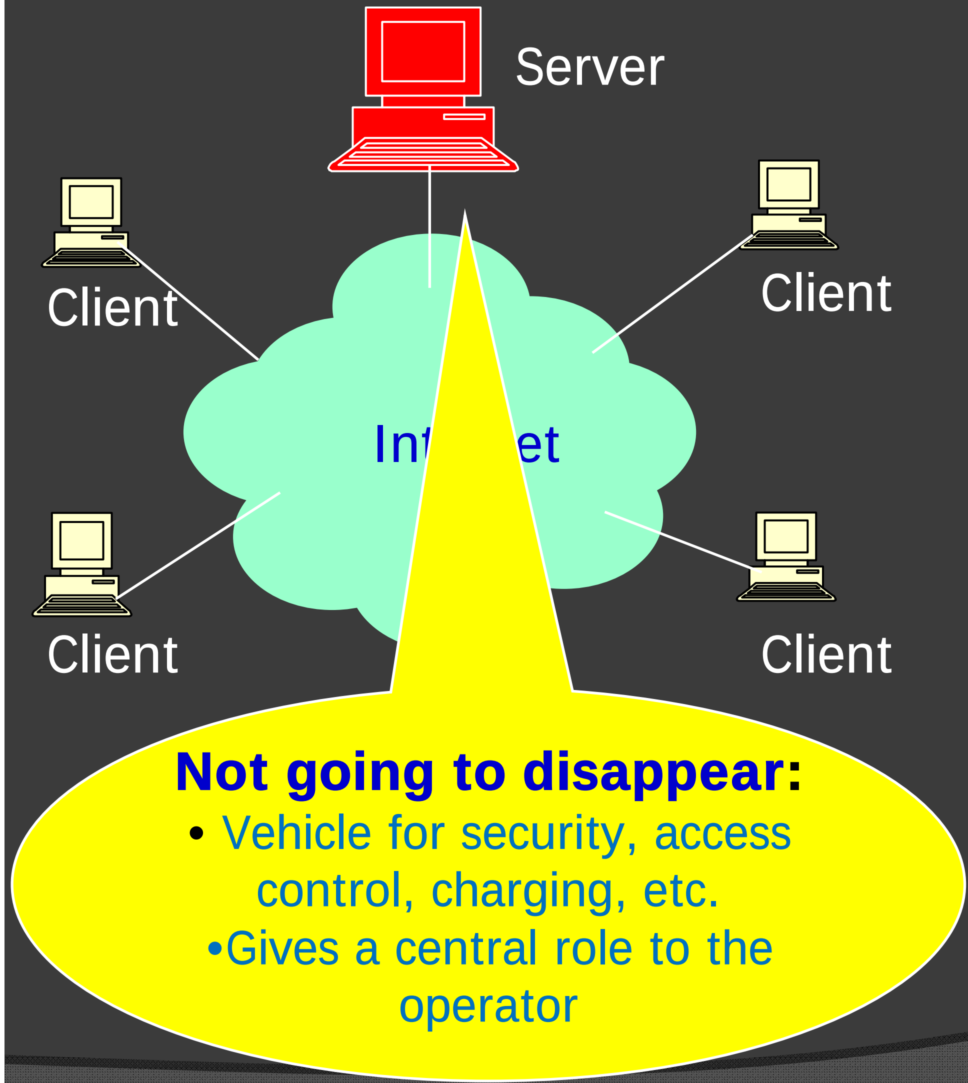
- Frans Kaashoek et al. “Peer-to-Peer computing research: a fad?” http://project-iris.net/talks/dht-toronto-03.ppt
- Vinod Muthusamy. “An Introduction to Peer-to-Peer Networks”

  http://www.eecg.toronto.edu/~jacobsen/mie456/slides/p2p-mie.pdf

Overview

- Peer-to-Peer (P2P) Technology
- Session Initiation Protocol (SIP)
- P2P-SIP: Decentralised SIP
- IETF Standardisation of P2P-SIP
- Early Trials and Implementations of P2P-SIP
- EU PHOSPHORUS (UEssex) P2P-SIP Test Bed

Client-Server computing



Client-server

- Server is data source
- Multiple clients per server
- Well-known, widely used (http, DNS, ftp, web services, etc)

Limitations

- Fault tolerance (single point of failure)
- Scalability hard to achieve
- Requires central administration
- Does not make the most of "Client" resources

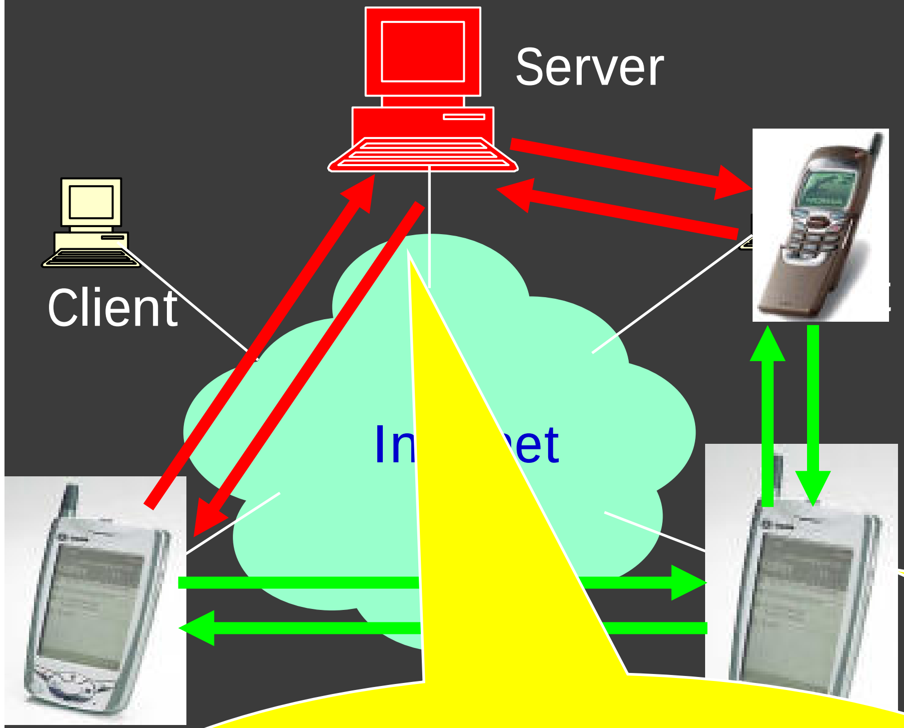
The Essence of Peer-to-Peer

- Distributed Nodes or Peers with similar interests to communicate using an overlay network
- The overlay network is made up of logical connections at the application level
- Nodes communicate directly without the intervention of a server

The three flavours of P2P

- P2P communication
- P2P networks
- P2P computing / applications

P2P communication



P2P communication

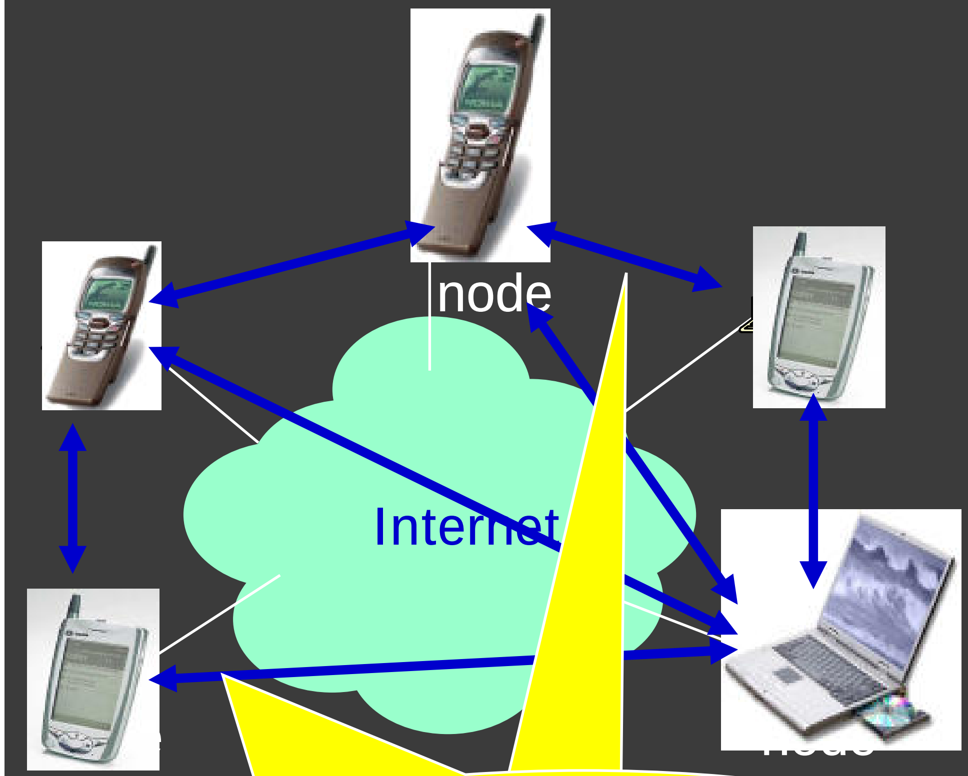
- Any node may initiate, manage, terminate session
- Person-2-person communication
- Machine-2-machine communication

Pure P2P communication does not rely on servers ... but on peer networks

Today the operator provides server-mediated P2P communication

- SIP, SMS, IM, etc.

P2P networks



In Essence P2P consists of:

- Structured aggregation of resources
- Efficient discovery of resources
- Direct interaction among nodes (non-server mediated)

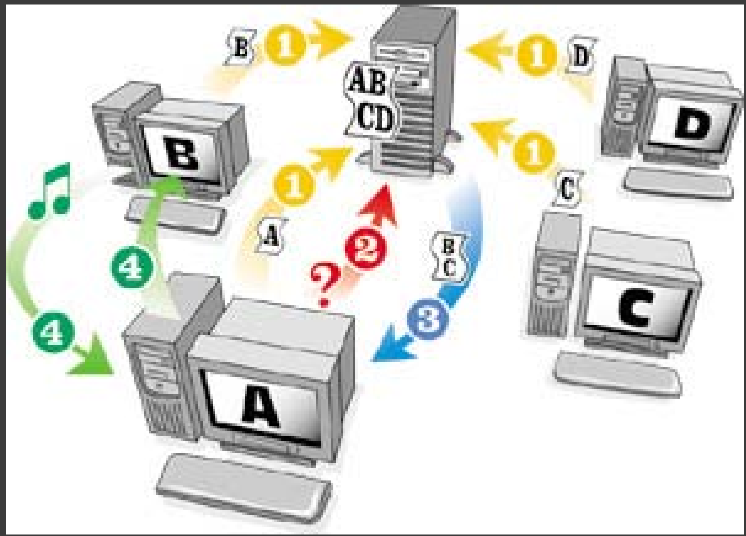
P2P networks

- Nodes are both clients and servers
- Nodes form a content/service network (application-level overlays)
- P2P networks are dynamic
- Autonomic self-managed (no centralised authority in theory!)

Limitations of pure P2P networks

- Hard to secure, charge etc
- Best-effort service
- Not mobile-friendly
- Not interoperable

P2P computing – enabler of innovative applications



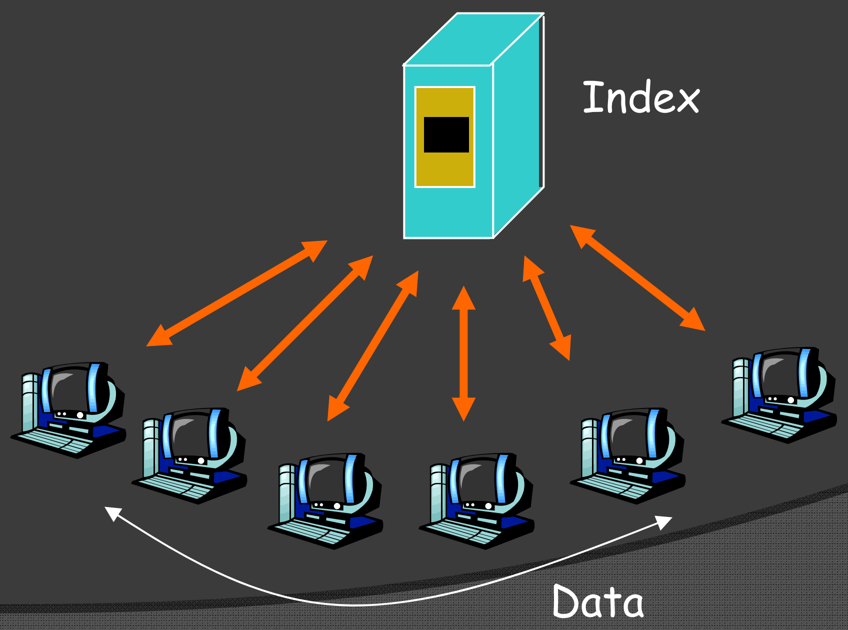
- File Sharing
- Communication
- Collaboration
- Computation
- Databases (distributed storage)

P2P Classification

- Degree of P2P decentralization
  - Hybrid decentralized P2P
  - Purely decentralized P2P
  - Partially centralized P2P
- Degree of P2P structure
  - Structured P2P
  - Loosely structured P2P
  - Unstructured P2P

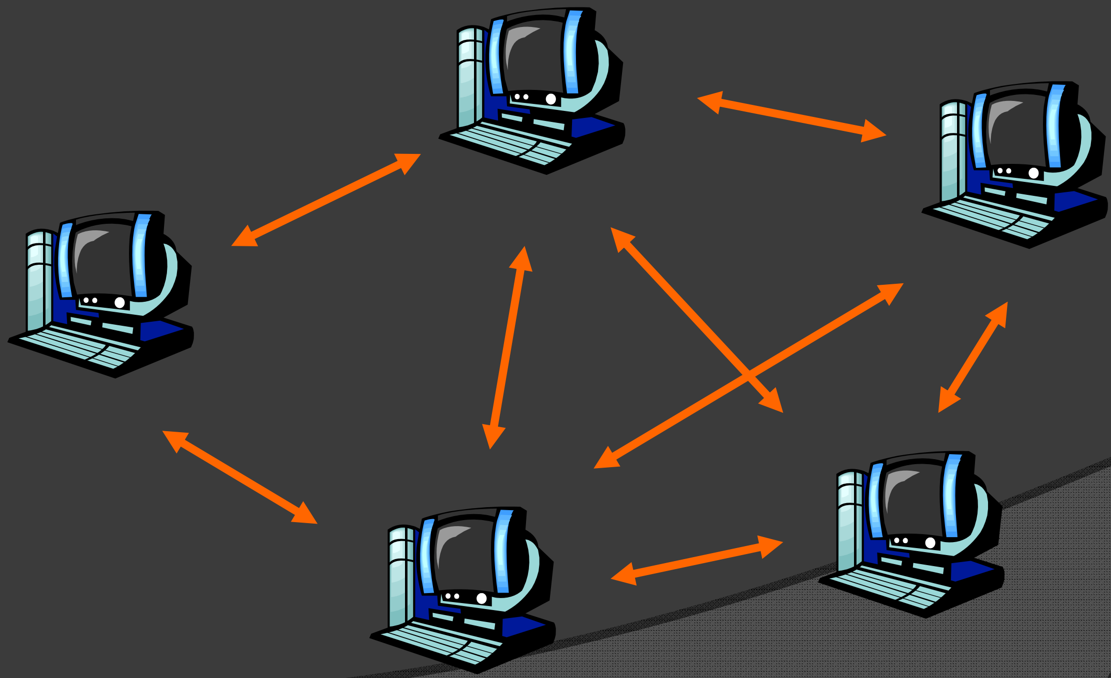
P2P Decentralization – Hybrid P2P

- Central server facilitates the interaction between peers
- Central server performs the lookups and identifies the nodes of the network.
- example: Napster
- (-) Single point of failure, scalability?, …



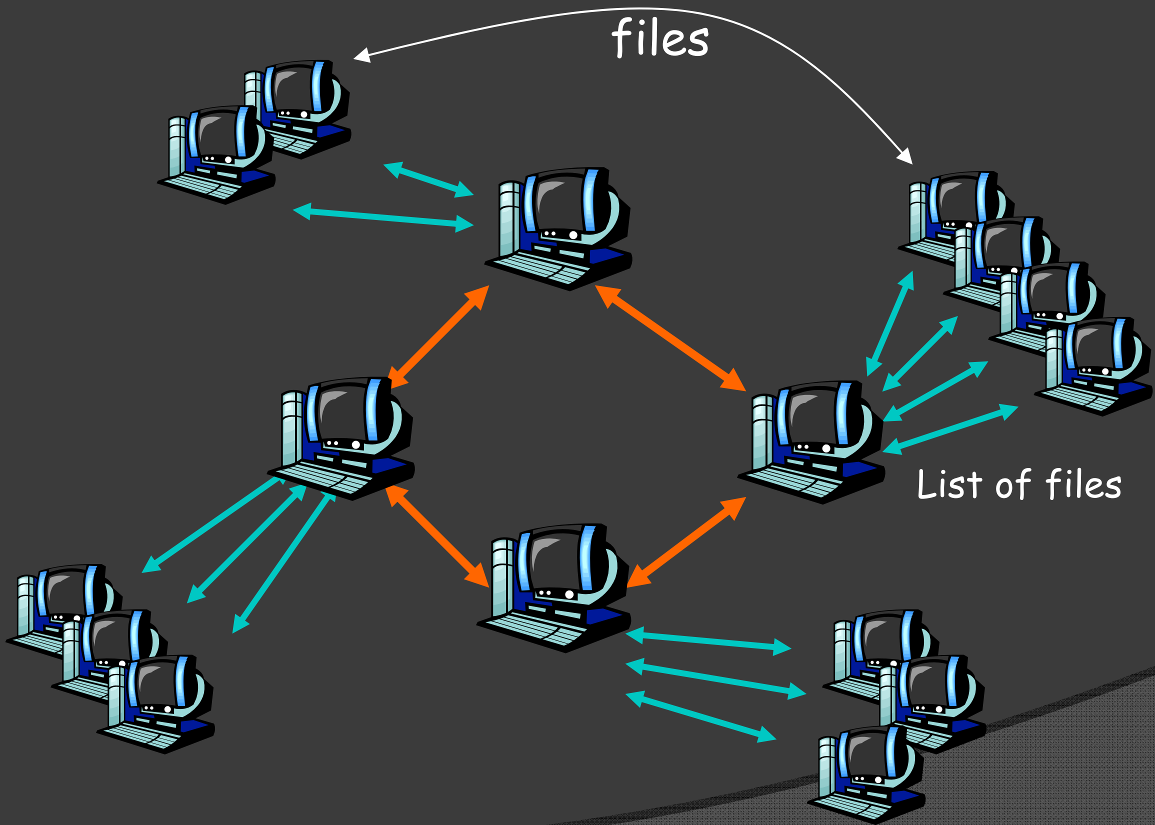
P2P Decentralization - Pure P2P

- network nodes perform the same tasks (Servents)
- no central coordination activity
- examples: original Gnutella, Freenet
- (-) data consistency?, Manageability?, Security?, Comm. overhead



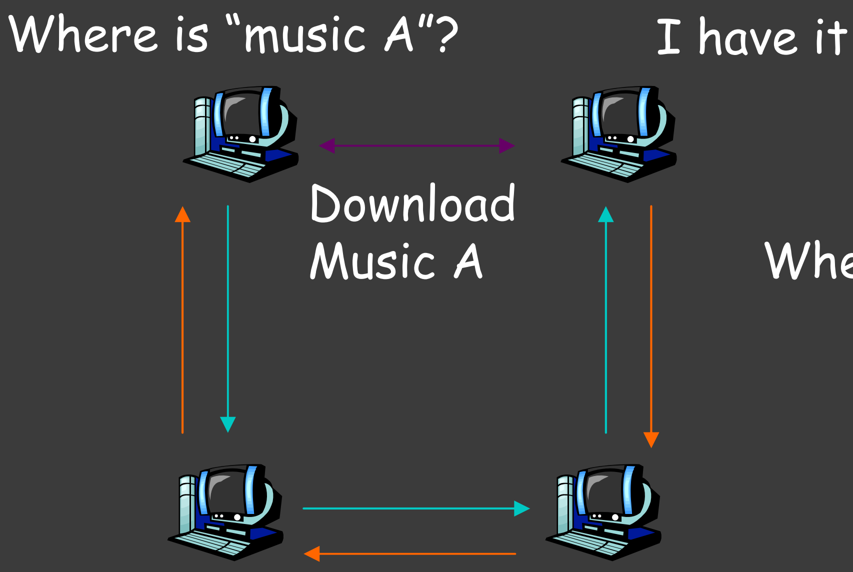
P2P Decentralization - Partially

- some of the nodes assume a more important role
- Supernodes act as local central indexes
- examples: Kazaa, recent Gnutella

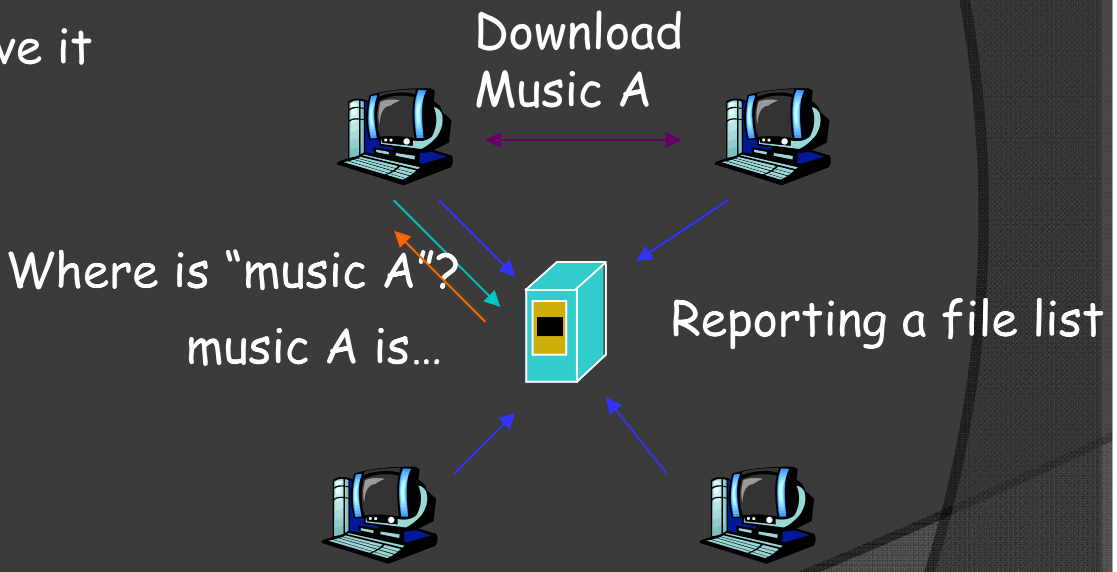


Unstructured P2P

- data is distributed randomly over the peers and broadcasting mechanisms are used for searching.
- placement of data is unrelated to the overlay topology.
- examples: Napster, Gnutella, KaZaa



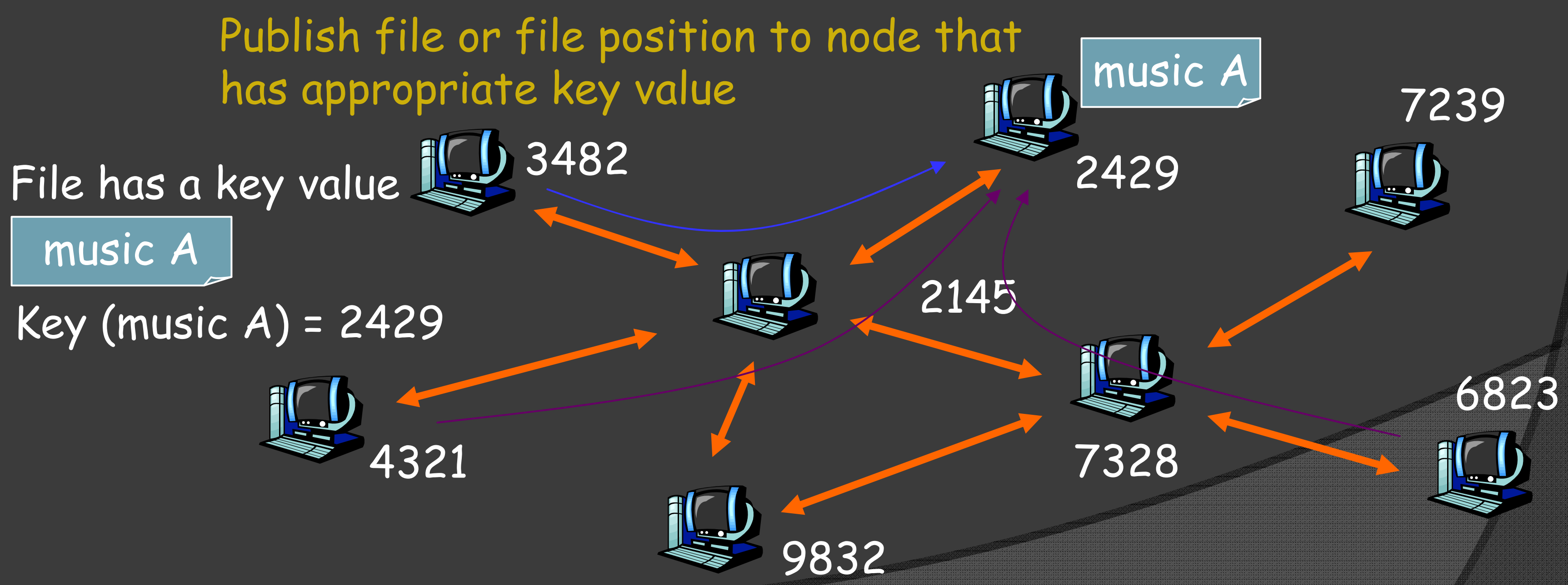
Unstructured & Purely decentralized
e.g. Gnutella



Unstructured & Hybrid decentralized
e.g. Napster (old version)

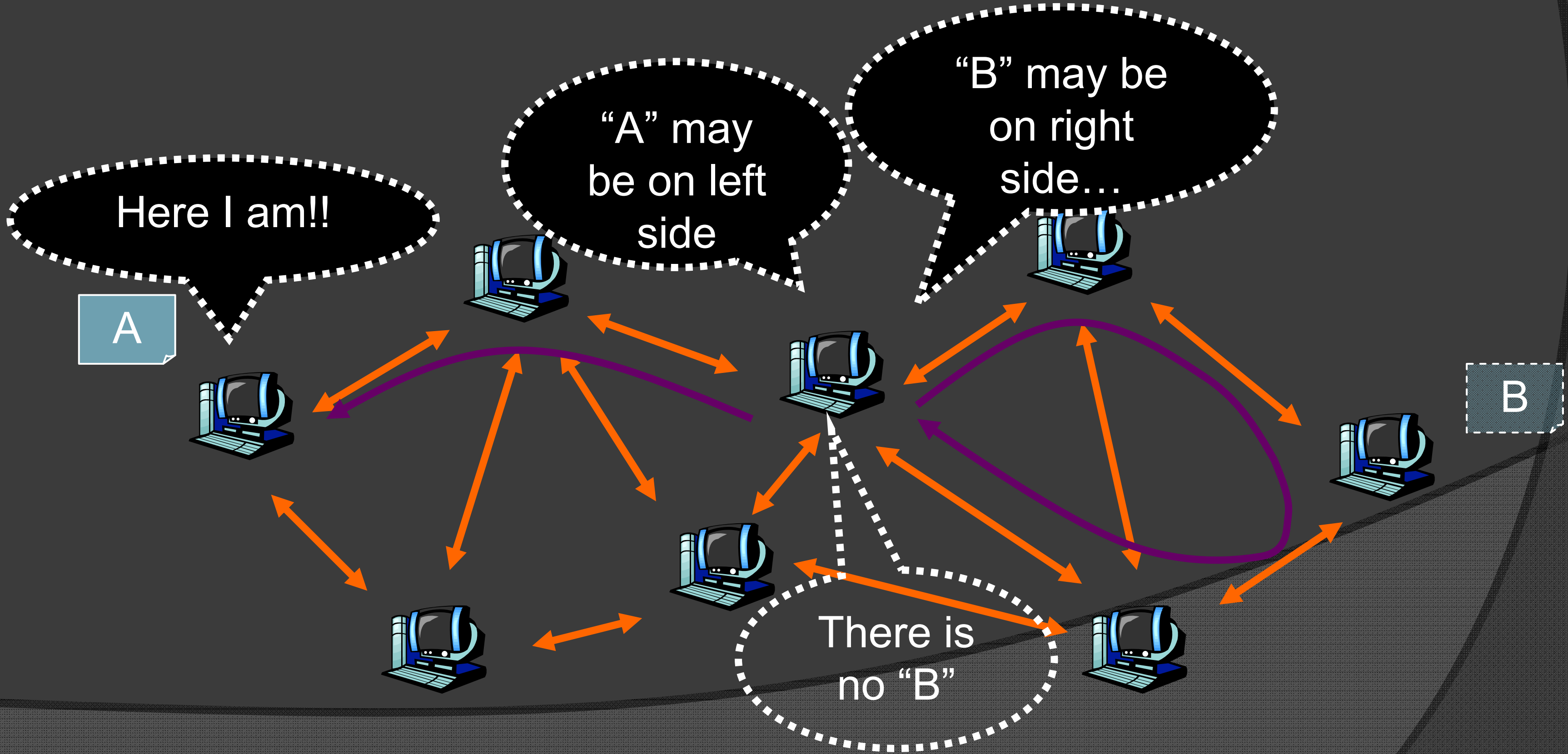
Structured P2P

- Network topology is tightly controlled and files are placed at precisely specified locations.
- Provide a mapping between the file identifier and location
- Examples: Chord, CAN, PAST, Tapestry, Pastry, etc.



Loosely structured P2P

- Between structured and unstructured
- File locations are affected by routing hints, but they are not completely specified.
- example: Freenet



P2P classification summary

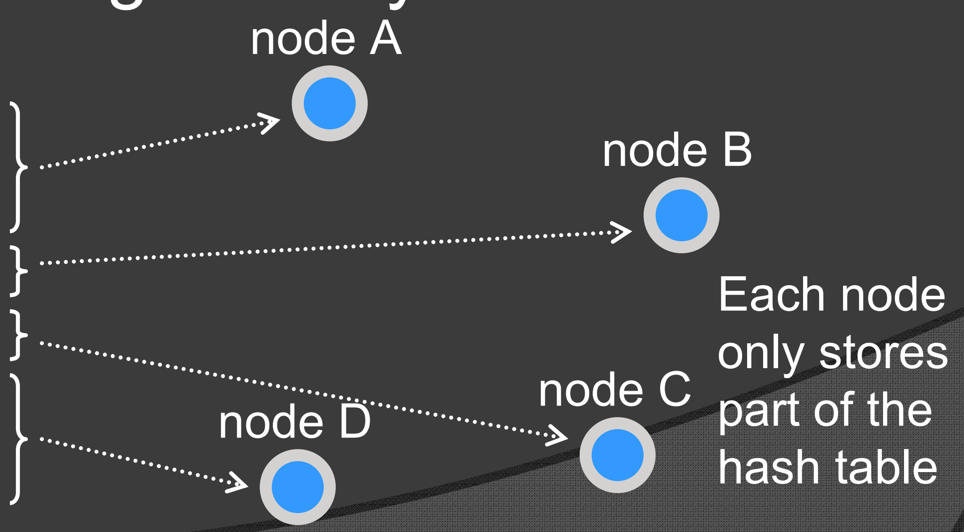
	Unstructured Networks	Loosely Structured Networks	Structured Networks
Hybrid Decentralized	Napster		
Pure Decentralized	Gnutella	Freenet	Chord, CAN, Tapestry
Partially Centralized	KaZaa, Gnutella		

The Structured P2P Approach: Detailed Descriptions and Principles

Structured P2P Description

- Most Structured approaches are based on the Distributed Hash Table (DHT)
- The DHT provides a basic *lookup service*, which allows *any* node to find the value associated with a given key

Key	Value
CS10	Algorithms
CS15	Networking
CS30	Distributed Sys.
CS100	Peer-to-Peer
CS250	Operating Sys.
CS310	Grid Computing

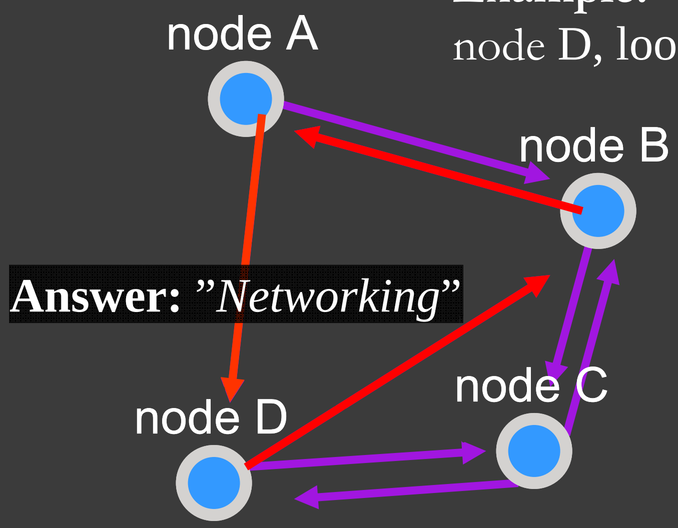


Structured P2P Description

- To provide the lookup service, the nodes must be interconnected

Key	Value
CS10	Algorithms
CS15	Networking
CS30	Distributed Sys.
CS100	Peer-to-Peer
CS250	Operating Sys.
CS310	Grid Computing

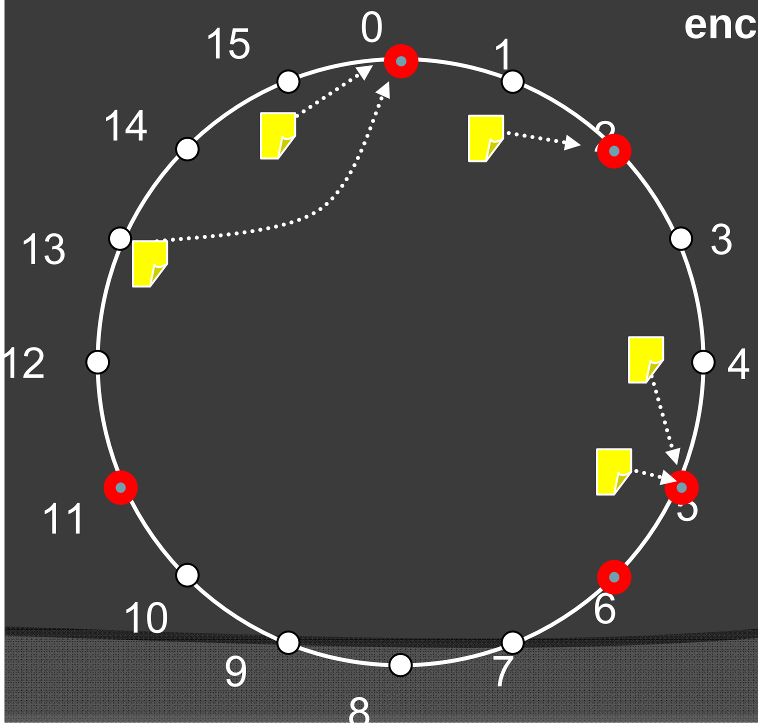
Example:
node D, lookup(*”CS15”*)



- **Each node maintains a *routing table* with pointers to some other nodes such that lookup requests can be routed to the node storing the requested key/value-pair (a.k.a. item)**

Structured P2P - CHORD

- Use a logical name space called the *identifier space*, consisting of identifiers {0,1,2,…, N-1}
- **The identifier space can be perceived as a logical ring modulo N**
- **Every node is assigned an identifier using a function H_1.**
- **Items are mapped to the identifier space using a function H_2, every node knows H_2**

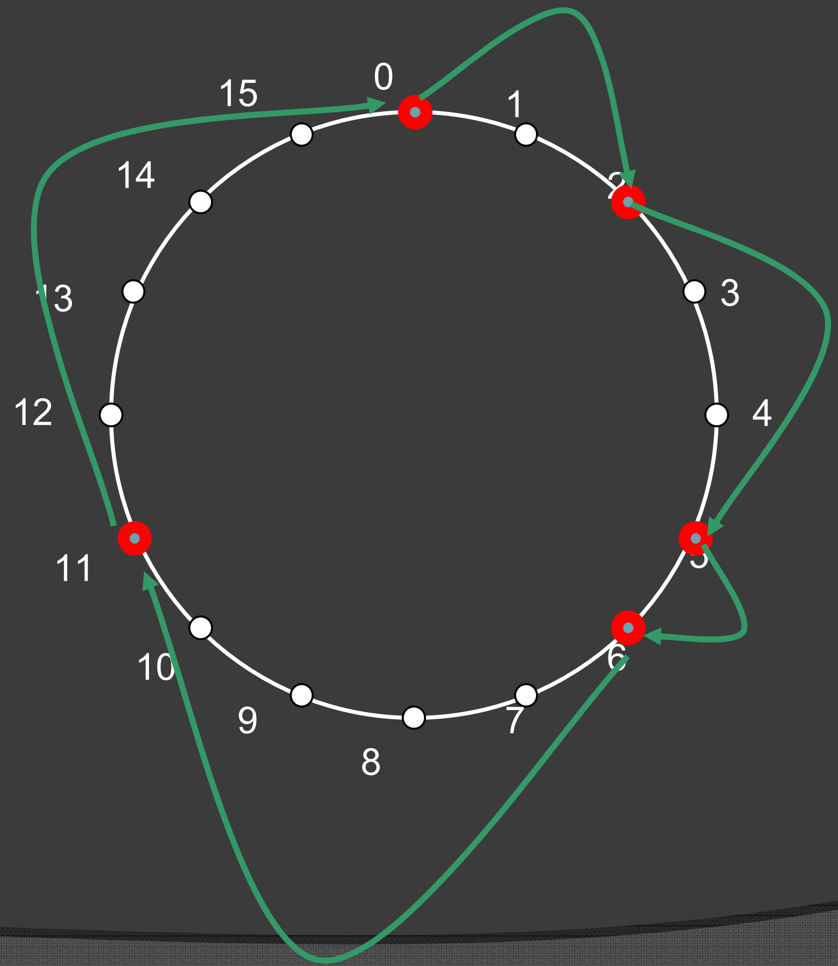


- **The items are stored at their *successor*, i.e. the first node encountered moving in the clockwise direction**

- Example: N=16, nodes {**a,b,c,d,e**}, and 5 items
- Node **a** gets identifier 0 since H_1(**a**)=0, the other nodes **b, c, d, e,** get identifiers 2, 5, 6, 11 the same way
- Item (”**cs15**”, ”**networking**”) is mapped to identifier 13 since H_2 (”**cs15**”) = 13, other items are similarly to 15, 2, 4, 5

CHORD Routing

- Each node maintains a *routing pointer* to the successor in the ring



- **The successor of a node *n* is the first node met going in clockwise direction starting at *n* +1**

Successor of Node 0 → Node 2

Successor of Node 2 → Node 5

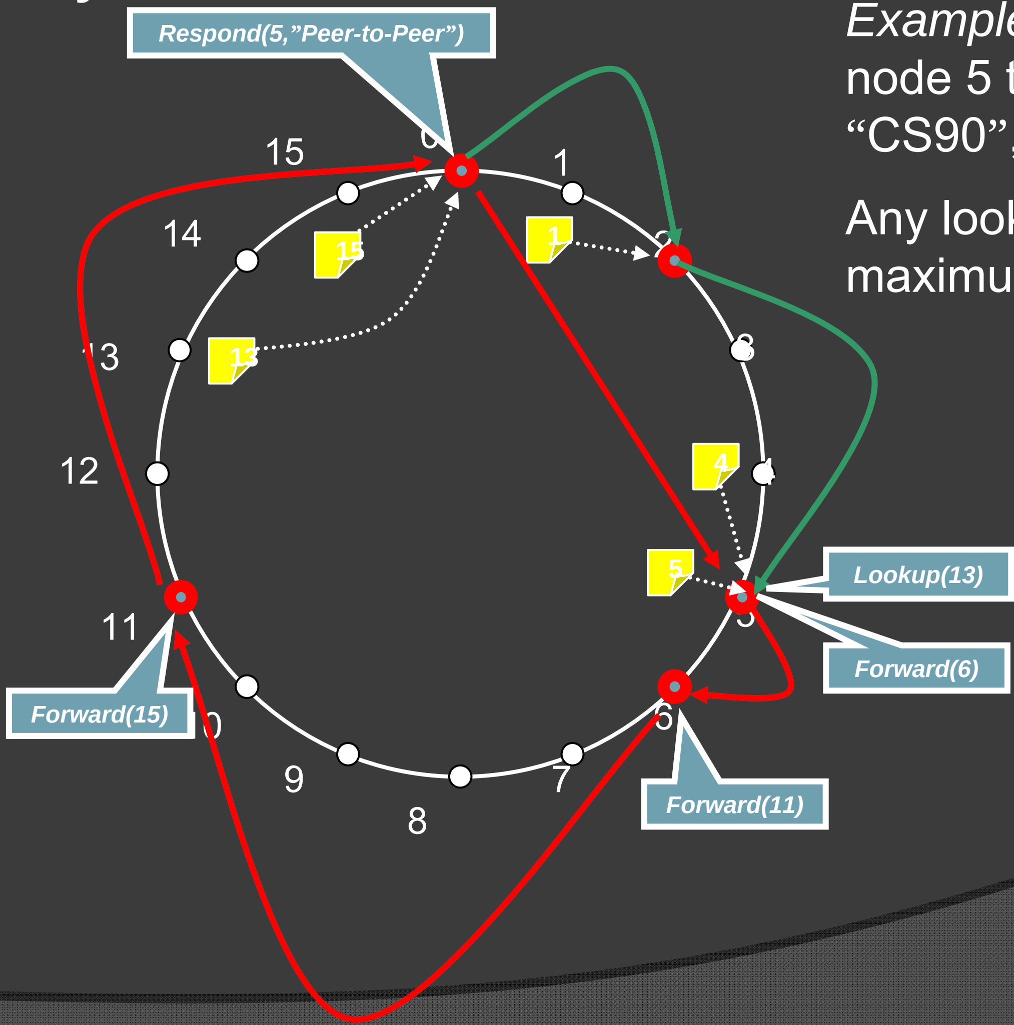
Successor of Node 5 → Node 6

Successor of Node 6 → Node 11

Successor of Node 11 → Node 0

CHORD Lookup

- **Lookups can be resolved in by following the successors sequentially**



Example: A lookup is made at node 5 to get the value of key "CS90", identifier H_2("CS90")=90

Any lookup can be resolved in maximum O(*N*) hops. ***Too slow!***

Key	Value
1 (CS10)	Algorithms
4 (CS15)	Networking
5 (CS30)	Distributed S.
13 (CS90)	Peer-to-Peer
15 (CS95)	Operating Sys.

CHORD Redundancy

- Each node, n, not only points to its successor but to the successors of

  $n+2^1, n+2^2, n+2^3, \ldots, n+2^L$ (all arithmetic operations modulo N)

- At each step in the routing, the distance between the currrent node and destination is halved (worst case).
- Yields $O(\log_2 N)$ hops at worst
  - N being the number of nodes

Example: node n=15

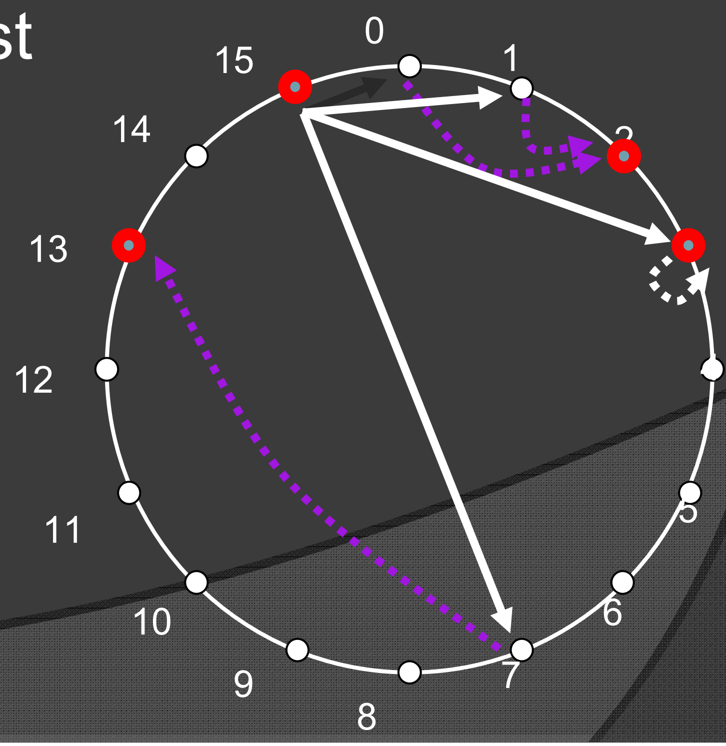
Start of the Intervals

Successor of $15+2^0=0$ → Node 2

Successor of $15+2^1=1$ → Node 2

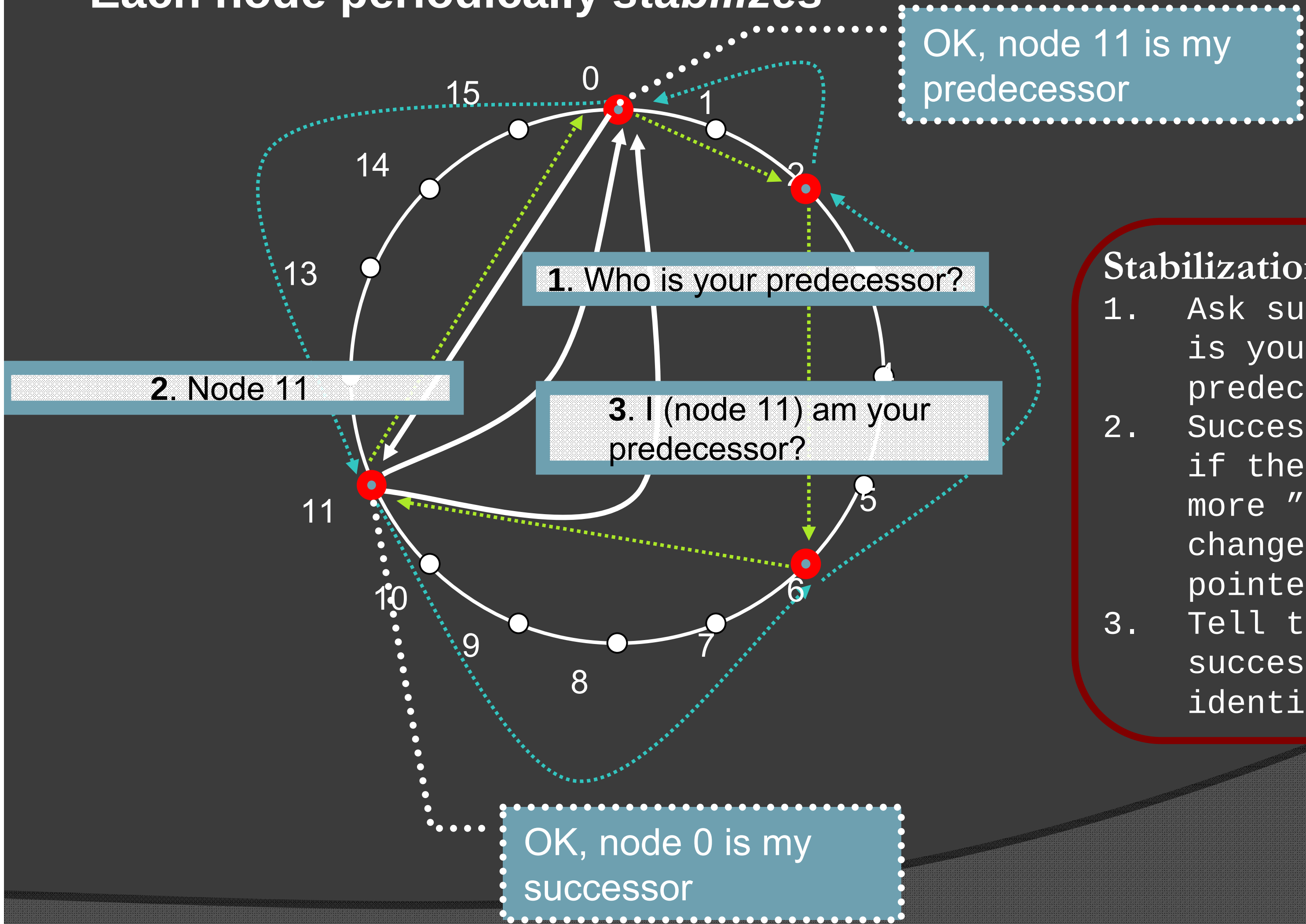
Successor of $15+2^2=3$ → Node 3

Successor of $15+2^3=7$ → Node 13



CHORD Stabilization

- **Each node periodically *stabilizes***

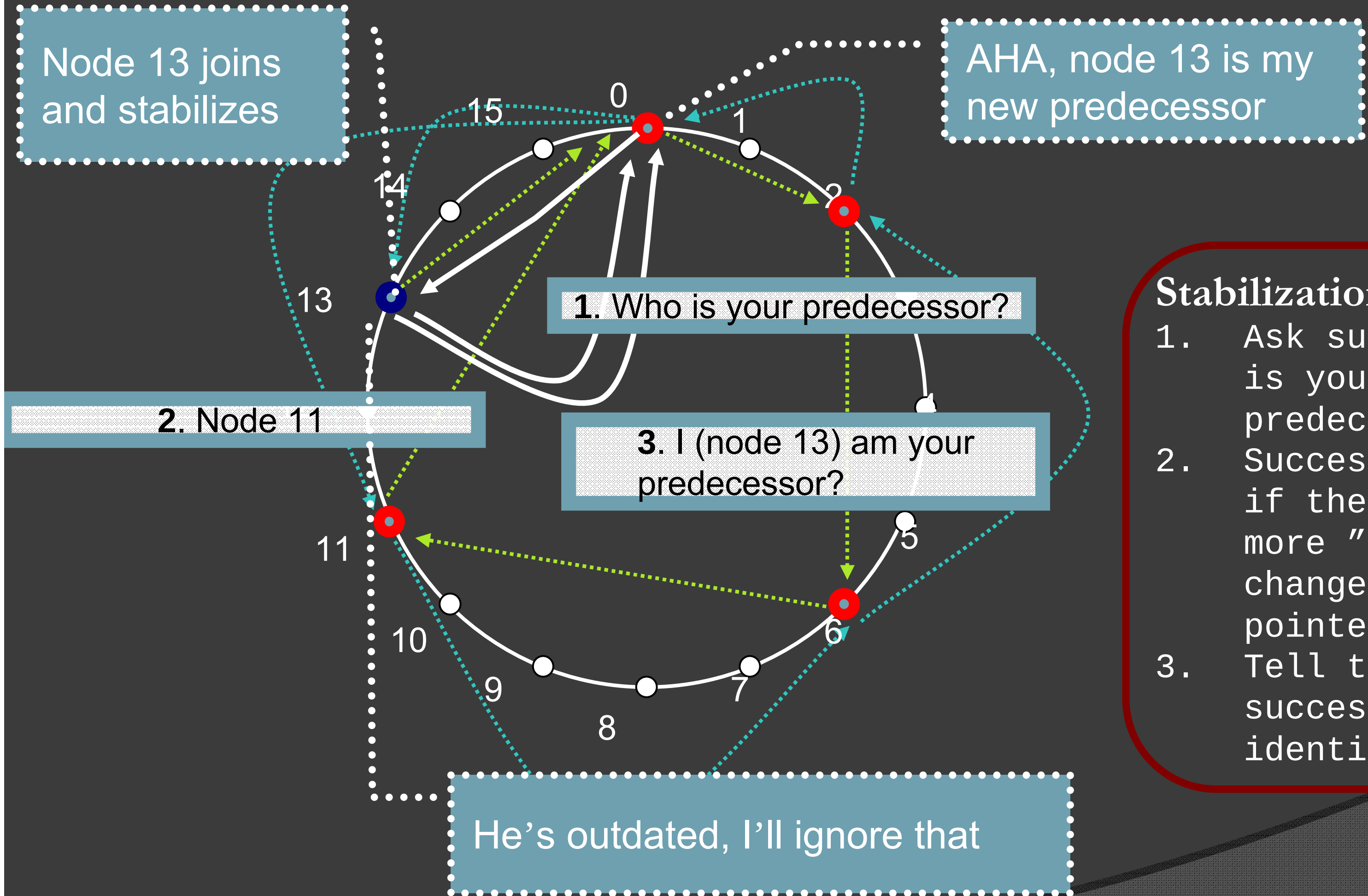


Stabilization Algorithm:

```
1.   Ask successor ”who
     is your
     predecessor”
2.   Successor answers,
     if the answer is
     more ”suitable”
     change my successor
     pointer
3.   Tell the current
     successor about my
     identifier
```

CHORD Peer Join

- **A joining node, only notifies its successor about its existence**

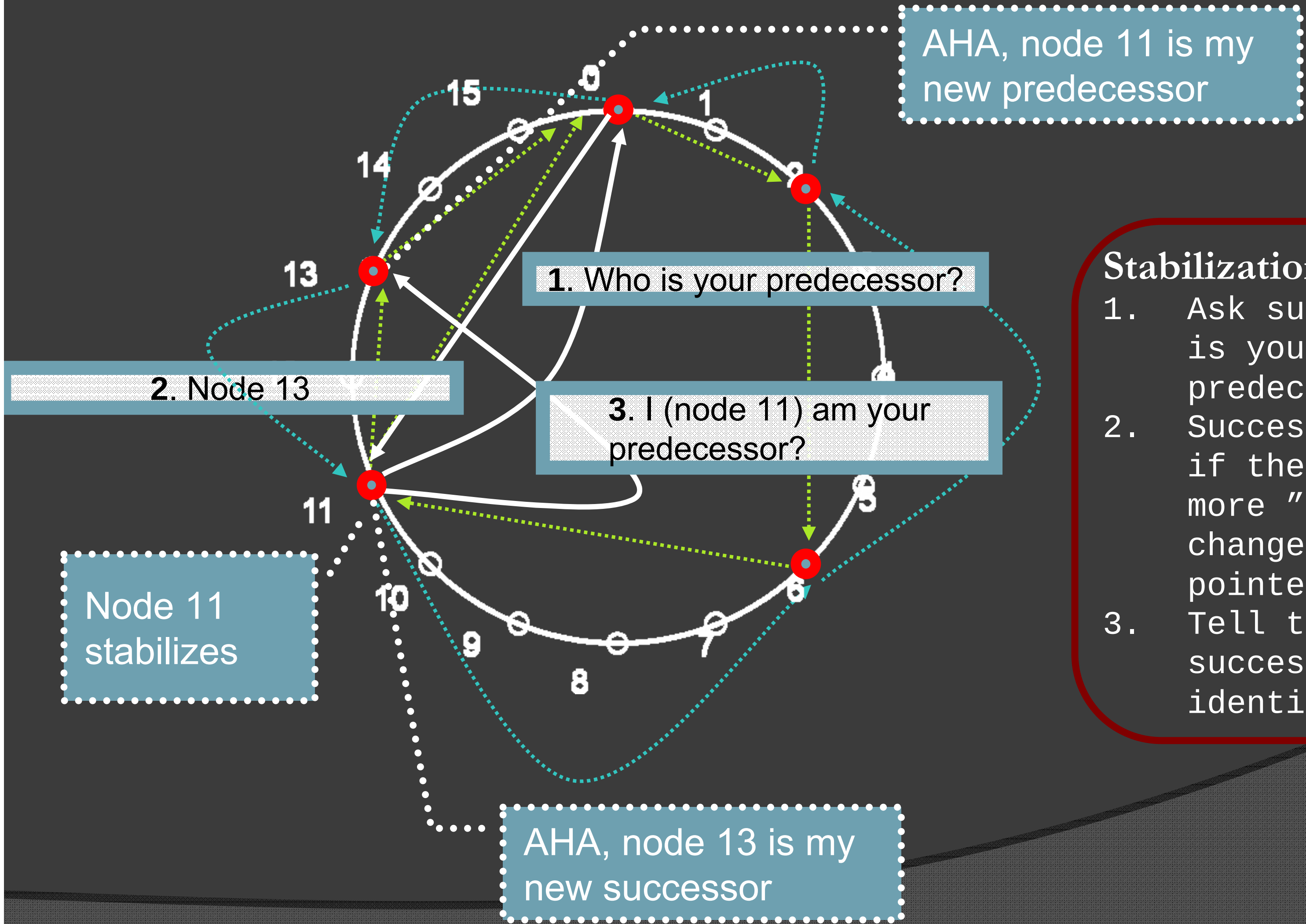


Stabilization Algorithm:

1. Ask successor "who is your predecessor"
2. Successor answers, if the answer is more "suitable" change my successor pointer
3. Tell the current successor about my identifier

CHORD Overlay Maintenance

- **A joining node, only notifies its successor about its existence**

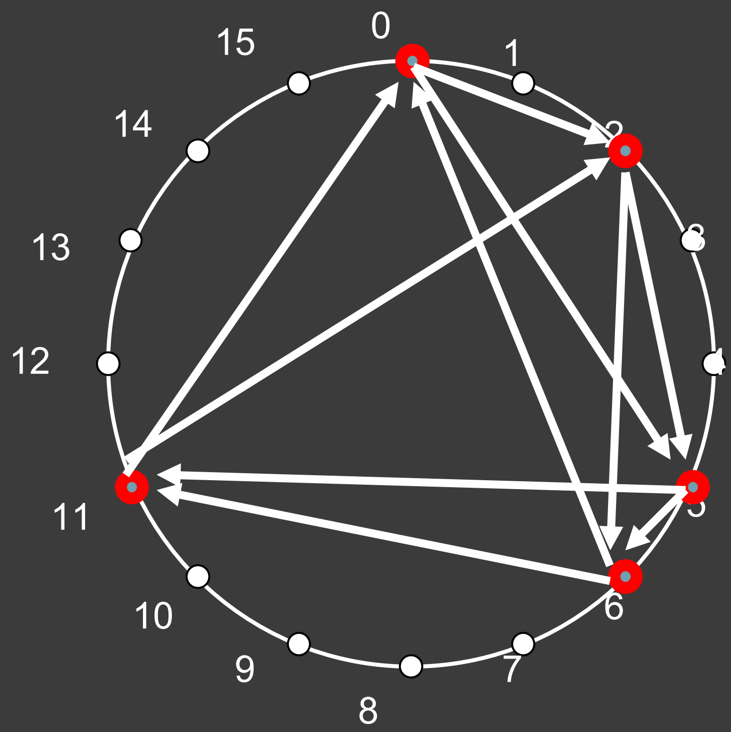


Stabilization Algorithm:

```
1.   Ask successor "who
     is your
     predecessor"
2.   Successor answers,
     if the answer is
     more "suitable"
     change my successor
     pointer
3.   Tell the current
     successor about my
     identifier
```

CHORD Overlay Maintenance

- **Every node in the system maintains additional routing pointers to its *f* successors**



Example: *f*=2

Every node knows its *f*=2 successors

Node **0** : Node 2 and Node 5

Node **2** : Node 5 and Node 6

Node **5** : Node 6 and Node 11

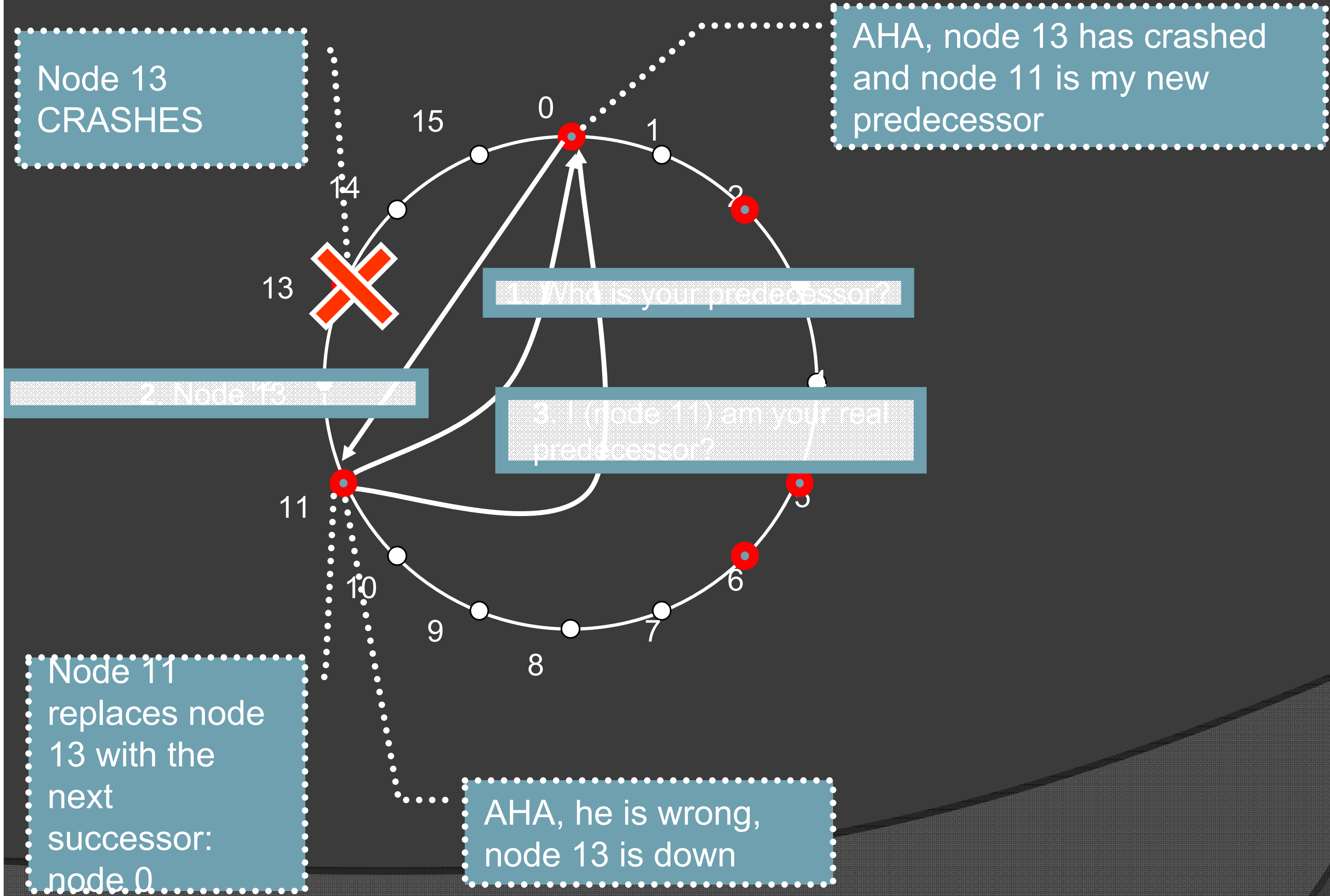
Node **6** : Node 11 and Node 0

Node **11** : Node 0 and Node 2

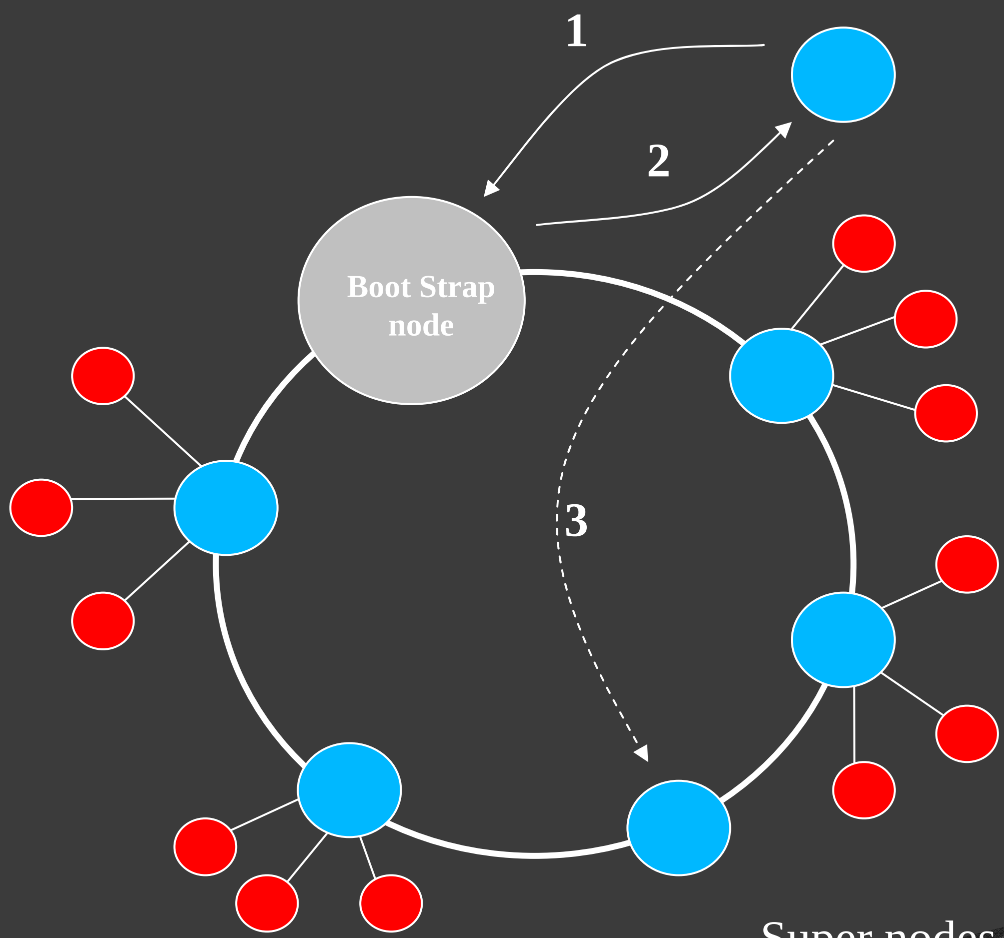
- **If node *n* detect that its successor has failed, it replaces it with the first alive successor node it knows**

CHORD Peer Leave

- **A joining node, only notifies its successor about its existence**



Hybrid Structured P2P



The Super Node is Authenticated by the Boot strap Node only if joining the overlay for the first time

The user profile of the super node is distributed on the overlay network.

The super node is responsible for the attached ordinary nodes SIP functionalities i.e. Proxy and Registrar.

Super nodes can also behave like servers to the ordinary nodes

Advantages of P2P Approaches

- Distributed
- Highly Scalable
- Highly Redundant
- Fault tolerant
- Self management
- Low cost

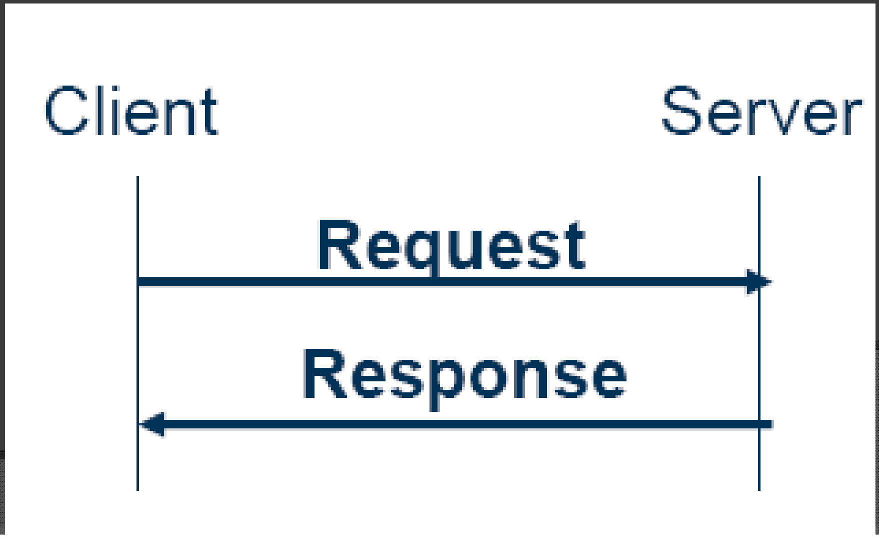
Limitations of P2P Approaches

- Limitations P2P
- Consumption of Extra resources
- Introduction of Extra Over heads
- Security
- Copyright Violations and Infringements
- Digital Rights Management

Session Initiation Protocol (SIP)

SIP Overview

- SIP is a end-to-end, client-server, extensible, text based protocol.
  - The design base was HTTP and SMTP
- SIP was originally used to establish, modify and terminate multimedia



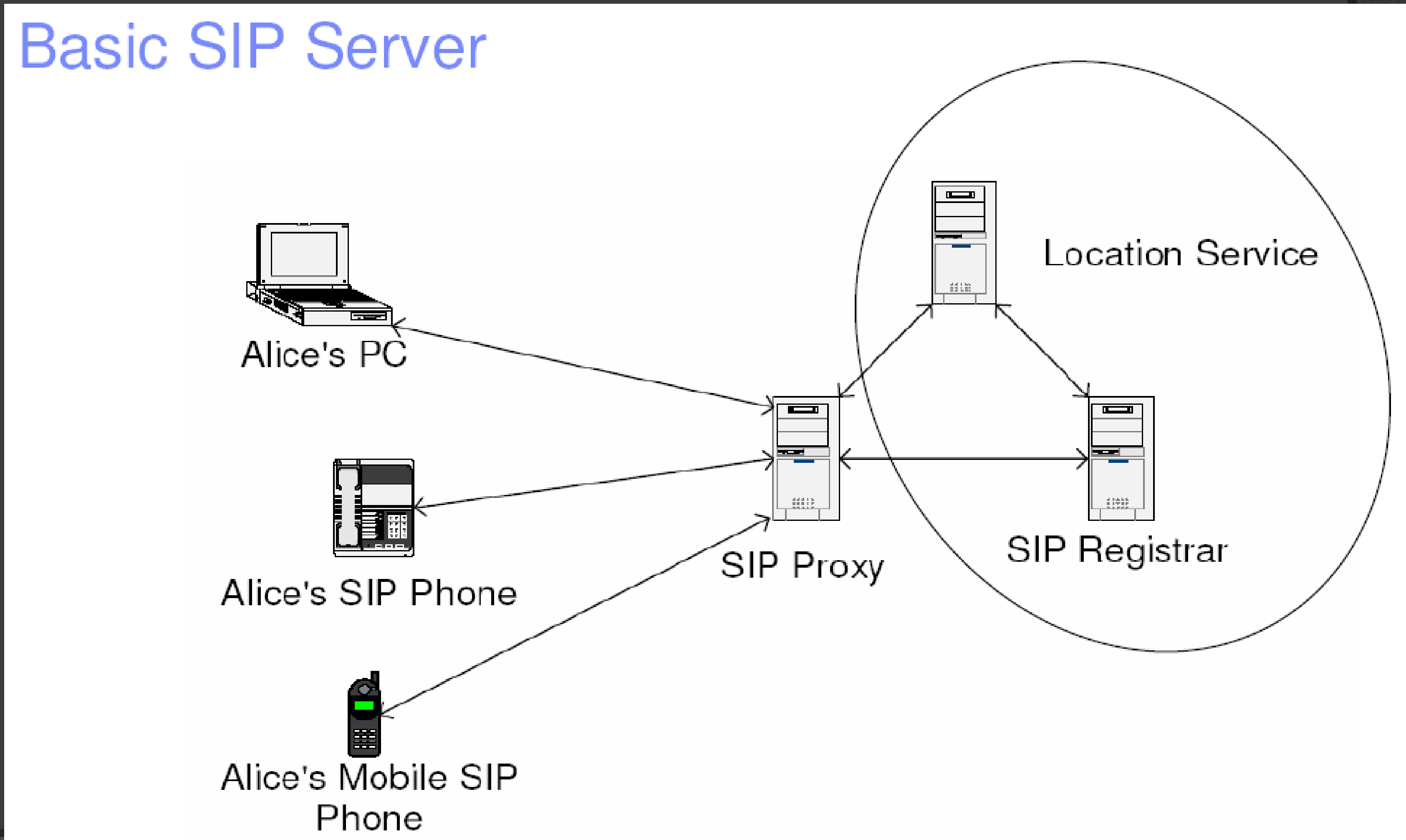
Session Initiation Protocol (SIP)

- Proposed by the Internet Engineering Task Force (IETF) as the protocol for handling call setup, routing, authentication and other feature messages to endpoints within an IP domain.
- SIP has evolved to be able to set-up a broad range of sessions:
  - Multimedia (e.g., voice, video, etc)
  - Gaming
  - Presence and Instant Messaging

SIP Properties

- SIP is a end-to-end, client-server, extensible, text based protocol. The design base was HTTP and SMTP
- SIP messages are either requests or responses.
- SIP messages carries zero or more “bodies”.
- SDP is the common body for session initiation.
- SIP runs on any transport protocol (UDP, TCP, TLS, SCTP)
  - The spec mandates UDP and TCP. Other transport protocols are optional

SIP Domain


Basic SIP Server
Location Service
Alice's PC
Alice's SIP Phone
SIP Proxy
SIP Registrar
Alice's Mobile SIP Phone

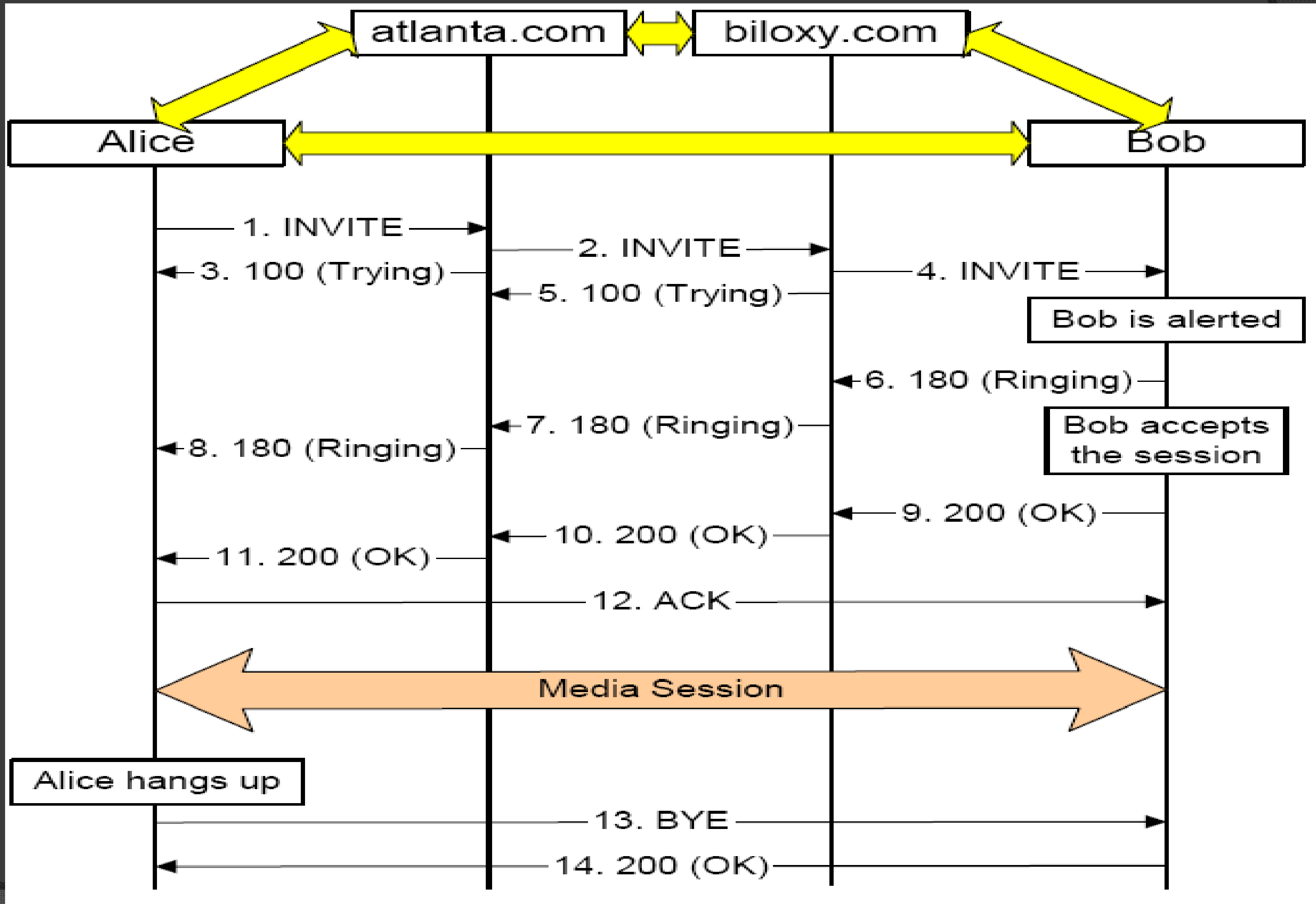
SIP Functionality

- SIP provides the following functionality:
  - User location (not geographical location)
  - User availability
  - User capabilities
  - Session set-up
  - Session management
- SIP does not provide services
  - But it enables the system to provide services
  - It has been demonstrated that it is easy to provide services with SIP

SIP Specifications

- The base SIP specification defines six SIP requests types (aka methods). They are INVITE, ACK, OPTIONS, BYE, CANCEL and REGISTER .
- SIP uses EXTENSIONS to extend other functionalities not supported by the core methods. This allows the creation of rich SIP services and also allows SIP to extended beyond multimedia services

SIP Trapezoid



Session Initiation Protocol (SIP)

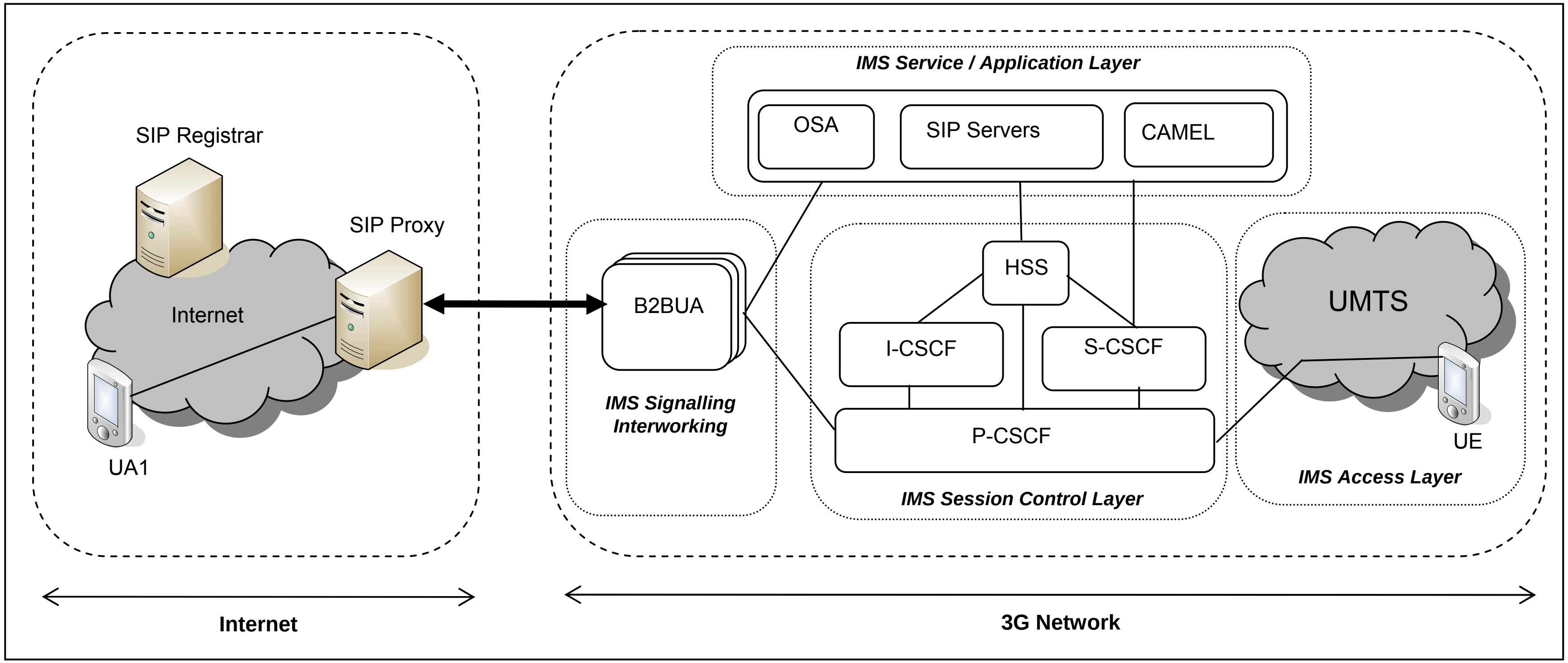
(1)INVITE sip:bob@biloxi.com SIP/2.0
(2)Via:SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
(3)To: Bob <bob@biloxi.com>
(4)From: Alice <alice@atlanta.com>;tag=1928301774
(5)Call-ID: a84b4c76e66710
(6)CSeq: 314159 INVITE
(7)Max-Forwards: 70
(8)Date: Thu, 21 Feb 2002 13:02:03 GMT
(9)Contact: <sip:alice@pc33.atlanta.com>
(10)Require:
(11)Supported:
(12)Content-Type: application/sdp
(13)v=0
(14)o= 53655765 2353687637 IN IP4 pc33.atlanta.com
(15)s=Session SDP
(16)t=0 0
(17)c=IN IP4 pc33.atlanta.com
(18)m=audio 3456 RTP/AVP 0 1 3 99
(19)a=rtpmap:0 PCMU/8000

SIP Message

SIP Extensions

- SIP is a modular extensible protocol.
  - Because of extensions
  - Or because of options in the core protocol.
  - Unlike ISUP, there are not different SIP flavours
  - But there are different functionalities, security mechanism, methods, headers, options, transport protocols, etc., that may or may not be implemented.
- SIP contains mechanisms to discover what is supported by a proxy or remote end.
  - Require, Supported, Proxy-Require, Allow headers
  - Contact header in registration

Session Initiation Protocol (SIP)



Internet and 3G Interworking using SIP

Advantages of SIP

- Lightweight Protocol
- Easily Extensible
- Can work with a wide range of protocol (TCP, UDP, STCP...)
- Standardised approach to Multimedia Signalling

Limitations of SIP

- Scalability
  - More servers and resources needs to be added to the SIP network as the number of clients increases
  - Server Management can be complicated
- Redundancy
  - Major disasters, in which the centralised infrastructure could potentially act as a single point of failure
- Flexibility
  - additional costs involved in the deployment of applications in smaller and adhoc environments

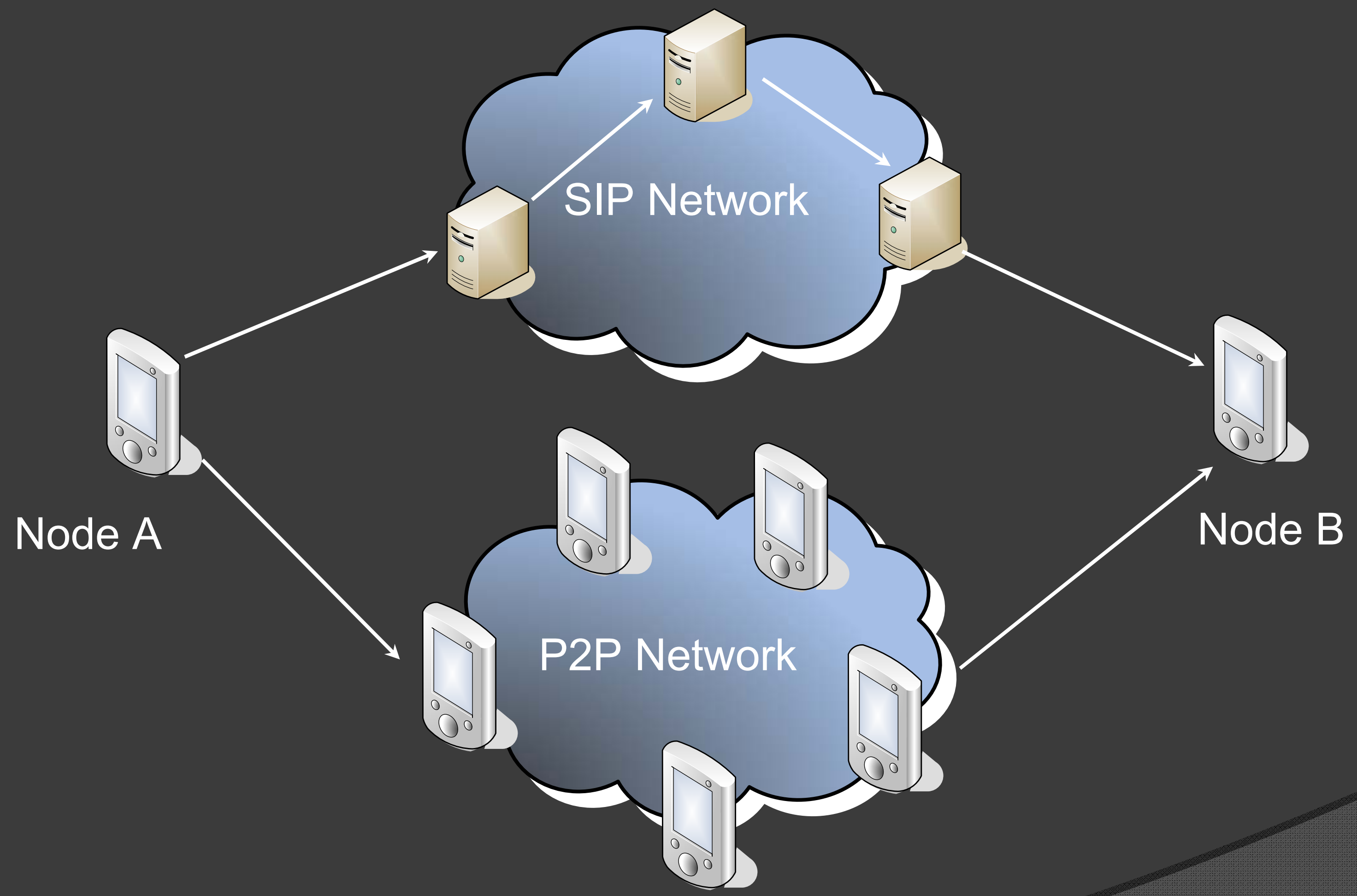
P2P-SIP:
Decentralised Approach to SIP

P2P-SIP Objectives

The main aim of P2P-SIP is to develop a Platform that

- Allows for a decentralized approach to deploy voice and multimedia Services
- incorporates the advantages of P2P into converged networks.
- Develop P2P Services specific to converged networks
- Integrate P2P Services as a service enabler for converged networks.
- Easy Creation of New Services.

P2P-SIP Architecture



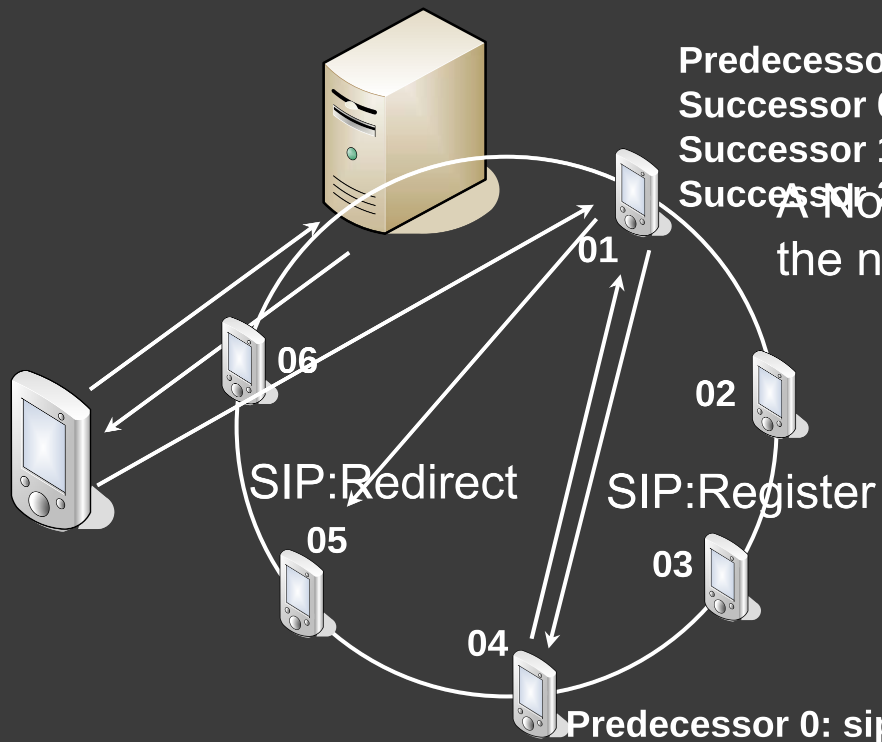
P2P-SIP Architecture

Bootstrap Node

01 wants to Locate 05

Predecessor 0: sip:06@192.168.1.3:5060;
Successor 0: sip:02@192.168.1.2:5060;
Successor 1: sip:03@192.168.1.3:5060;
Successor 2: sip:04@192.168.1.1:5060;

A NodeID is calculated based on the node's IP address



If each Node (N) has a routing table of size ***O(LogN)*** then it takes ***O(LogN)*** messages to locate a resource

Predecessor 0: sip:03@192.168.1.3:5060;
Successor 0: sip:05@192.168.1.2:5060;
Successor 1: sip:06@192.168.1.3:5060;
Successor 2: sip:01@192.168.1.1:5060;

P2P-SIP Register Message

Session Initiation Protocol

Request-Line: REGISTER sip:f3@192.168.1.2:5060;user=node SIP/2.0
    Method: REGISTER
    [Resent Packet: False]
  Message Header
    Via: SIP/2.0/TCP 192.168.1.1;branch=z9hG4bKZN11KpBD11X9g
    Max-Forwards: 70
    From: <sip:f2@192.168.1.1:5060;user=node>;tag=e6cNrcKNm5e3a
    To: <sip:f2@192.168.1.1:5060;user=node>
    Call-ID: e18e2b60-387c-122a-4586-0002a5b0f023
    CSeq: 79703170 REGISTER
    Expires: 3600
    Content-Length: 0
    DHT-NodeID: <sip:f2@192.168.1.1:5060;user=node>;algorithm=sum-8

P2P-SIP 200 OK Message

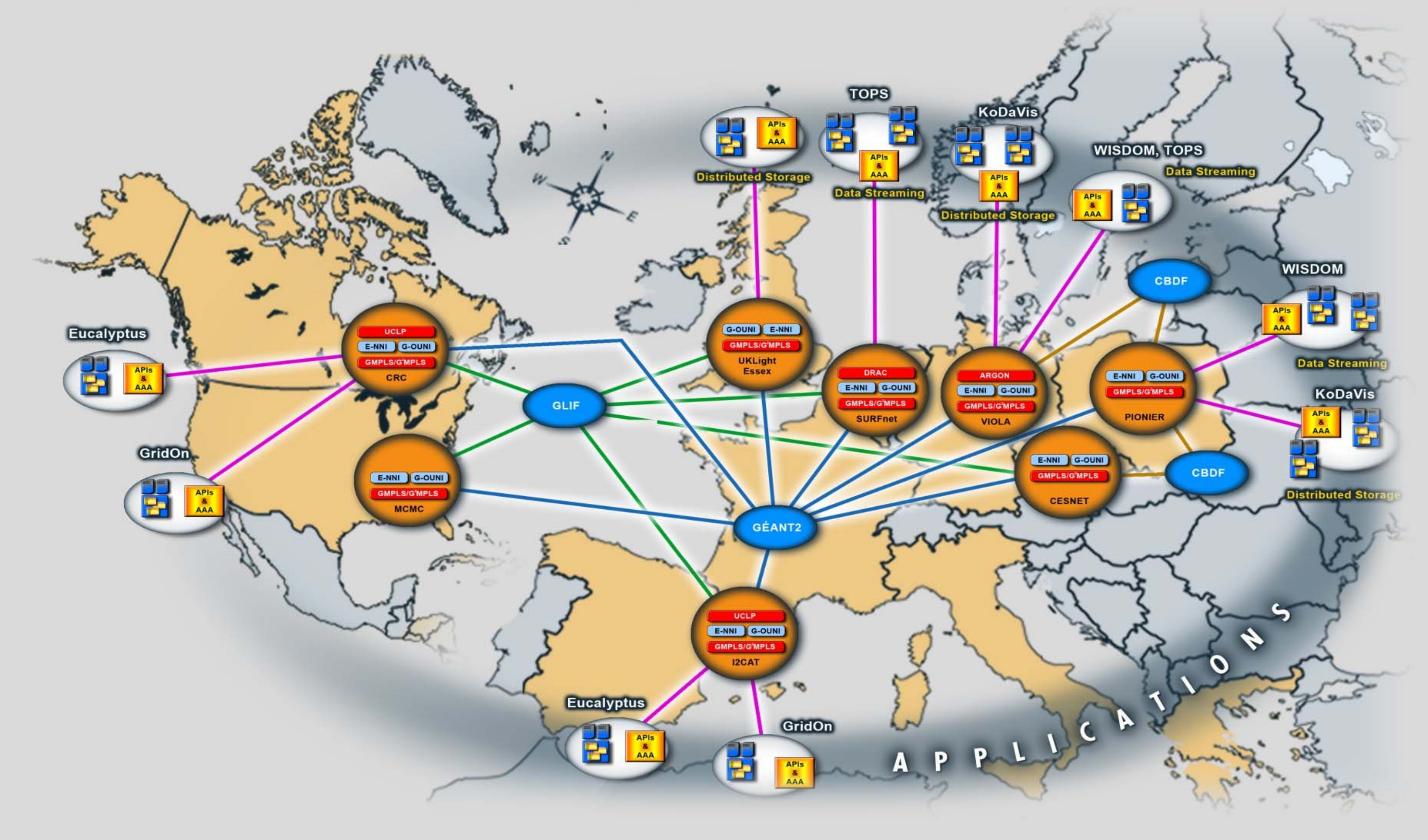
Session Initiation Protocol
Status-Line: SIP/2.0 200 OK
Status-Code: 200
[Resent Packet: False]
Message Header
Via: SIP/2.0/TCP 192.168.1.1;branch=z9hG4bKZN11KpBD11X9g;rport=43846
From: <sip:f2@192.168.1.1:5060;user=node>;tag=e6cNrcKNm5e3a
To: <sip:f2@192.168.1.1:5060;user=node>;tag=98r907mUZp68m
Call-ID: e18e2b60-387c-122a-4586-0002a5b0f023
CSeq: 79703170 REGISTER
Contact: <sip:f2@192.168.1.1:5060;user=node>
Content-Length: 0
DHT-Link: <sip:f2@192.168.1.1:5060;user=node>;link=P0
DHT-Link: <sip:f4@192.168.1.3:5060;user=node>;link=S0
DHT-NodeID: <sip:f3@192.168.1.2:5060;user=node>;algorithm=sum-8

SIPDHT – Node 1

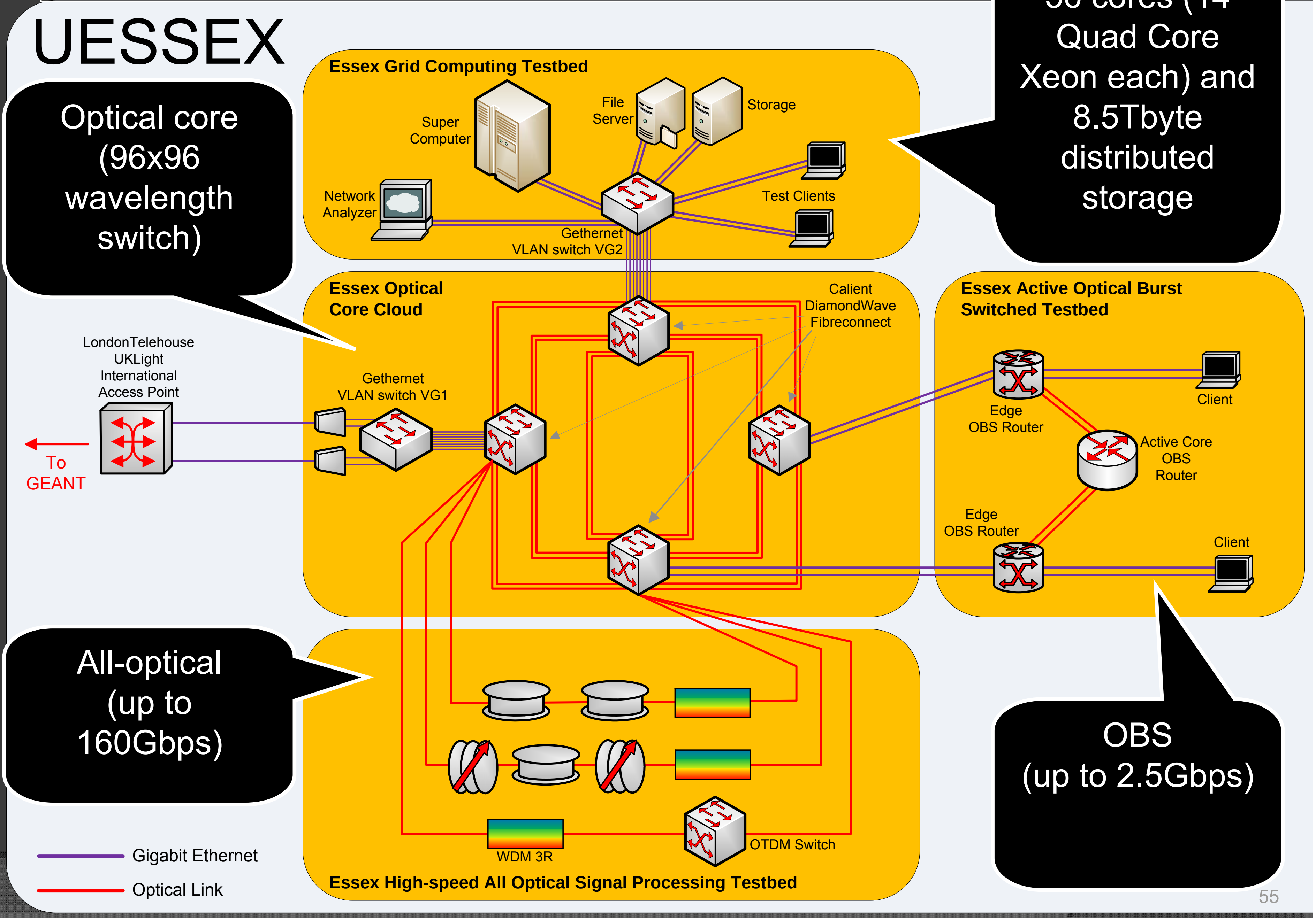
Node sip:b9@192.168.1.1:5060;user=node
- Predecessor 0: sip:bb@192.168.1.3:5060;user=node
- Successor 0: sip:ba@192.168.1.2:5060;user=node
- Successor 1: sip:bb@192.168.1.3:5060;user=node
- Successor 2: sip:b9@192.168.1.1:5060;user=node
- Finger 0 (0xba): sip:ba@192.168.1.2:5060;user=node
- Finger 1 (0xbb): sip:bb@192.168.1.3:5060;user=node
- Finger 2 (0xbd): sip:bb@192.168.1.3:5060;user=node
- Finger 3 (0xc1): sip:bb@192.168.1.3:5060;user=node
- Finger 4 (0xc9): sip:bb@192.168.1.3:5060;user=node
- Finger 5 (0xd9): sip:bb@192.168.1.3:5060;user=node
- Finger 6 (0xf9): sip:bb@192.168.1.3:5060;user=node
- Finger 7 (0x39): sip:bb@192.168.1.3:5060;user=node

Test Bed Demonstration

EU PHOSPHOSRUS Test Bed



UEssex PHOSPHORUS Local Test Bed



The UEssex P2P-SIP Test Bed

- Developed an IMS testbed based on IPv6, MIPv6, IPSec, and SIP
- Developed P2P extensions for the IMS and demonstrators
- Configured, deployed and tested SIP-DHT on our Local Test bed with SIP servers
- Deployed Peercast (P2P streaming application) on our local test bed of over 30 nodes with plans of integrating Peercast into the IMS test bed
- Currently working on the algorithm for Presence service using the SIP-DHT library with plans of also integrating it to with the IMS test bed.

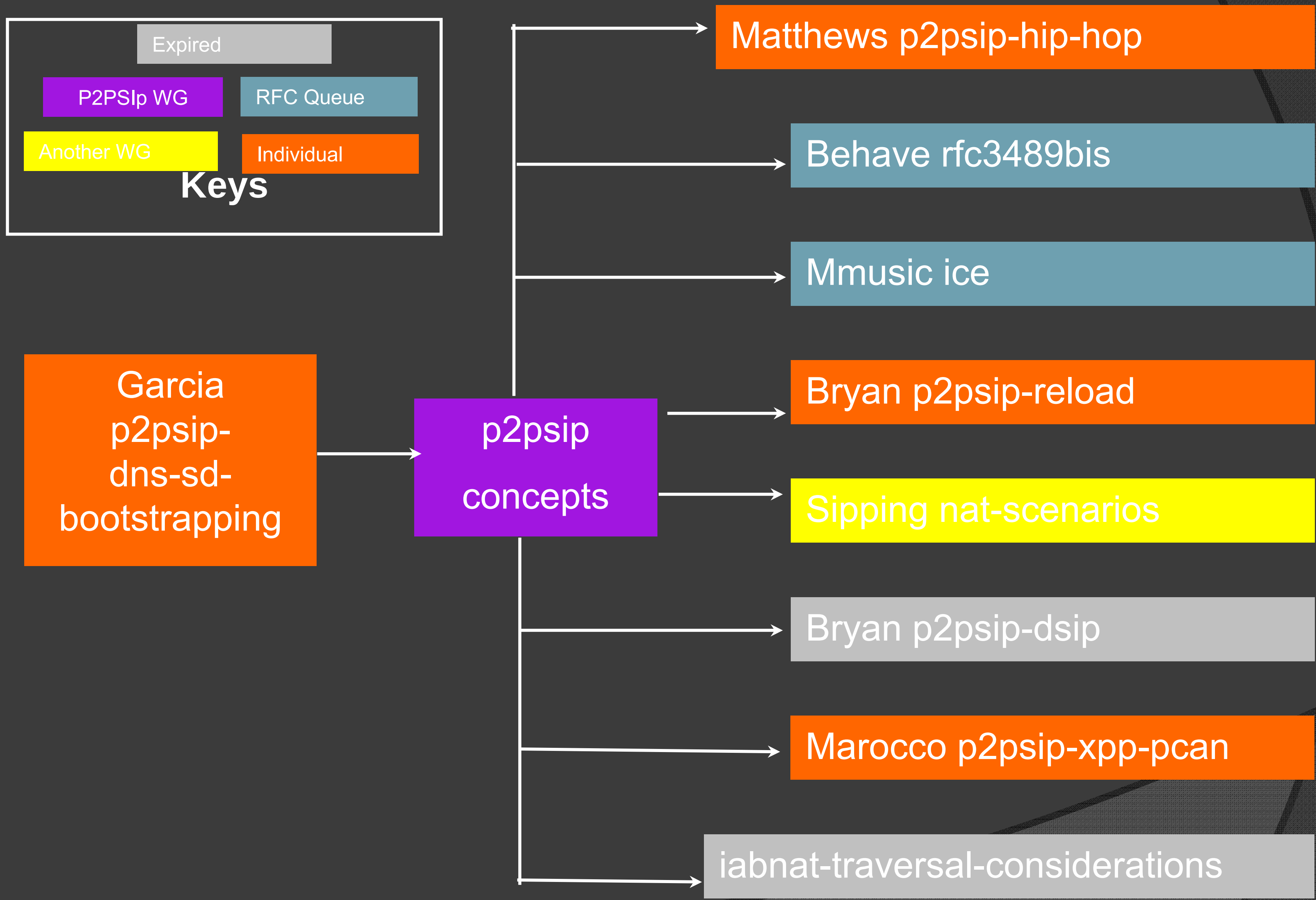
Research Objective

- Develop a test bed that supports SIP and P2P-SIP
- Carry out early trials of current P2P-SIP implementations
- Carry out tests to determine the amount of signalling generated when the DHT is stable
- Carry out tests to determine the amount of signalling generated when the DHT is not Stable
- Carry out tests to analyse amount of bandwidth consumed by individual peers and the overlay as a whole

Test Bed Properties

Tests	Supported		Results		Remarks
	Yes	No	Pass	Fail	
Convention SIP	✓		✓		Conventional SIP implementation is based on ReSIProcate [34]
P2P-SIP	✓		✓		P2P-SIPImplementation is based on SIPDHT [35]
Network Monitoring	✓		✓		This is achieved by using a traffic analyser
Peer Monitoring	✓		✓		This is collected by using tcpdump
Security		✓		✓	At no moment no implementation of P2P-SIPhas security functions
Parallel Processing	✓		✓		This was achieved by using the Sun Grid Engine (SGE)

Status of P2P-SIP in the IETF

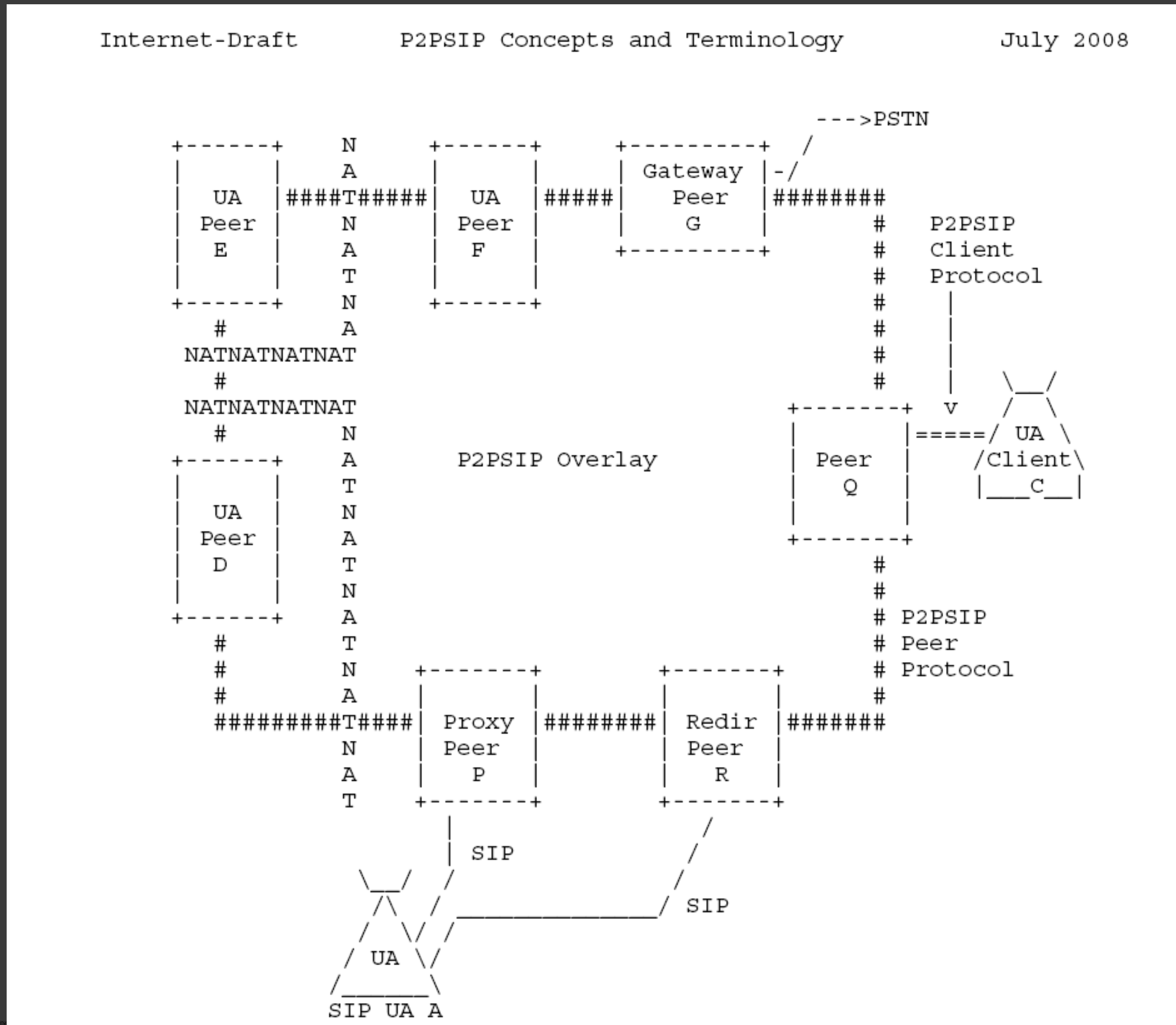


P2P-SIP Implementation

- **Open-Source Implementations**
  - Cisco P2PSIP Project *(draft-jennings-p2psip-asp)*
  - Columbia P2PP Project *(draft-baset-p2psip-p2pp)*
  - SIPDHT2 Project *(draft-marocco-p2psip-xpp, draft-marocco-p2psip-xpp-pcan)*
    - SIPDHT Project *(draft-bryan-sipping-p2p)* (no longer maintained)
  - University of Parma Kademlia dSIP Implementation *(draft-cirani-p2psip-dsip-dhtkademlia, draft-bryan-p2psip-dsip)*
  - Huawei's implementation of SEP peer and client protocols *(draft-jiang-p2psip-sep, draft-zheng-p2psip-client-protocol)*

- **Commercial Implementations**
  - SIPeerior Technologies P2PSIP Core and Endpoint Development Kits *(draft-bryan-p2psip-reload, draft-lowekamp-p2psip-reload-security, other supporting drafts)*

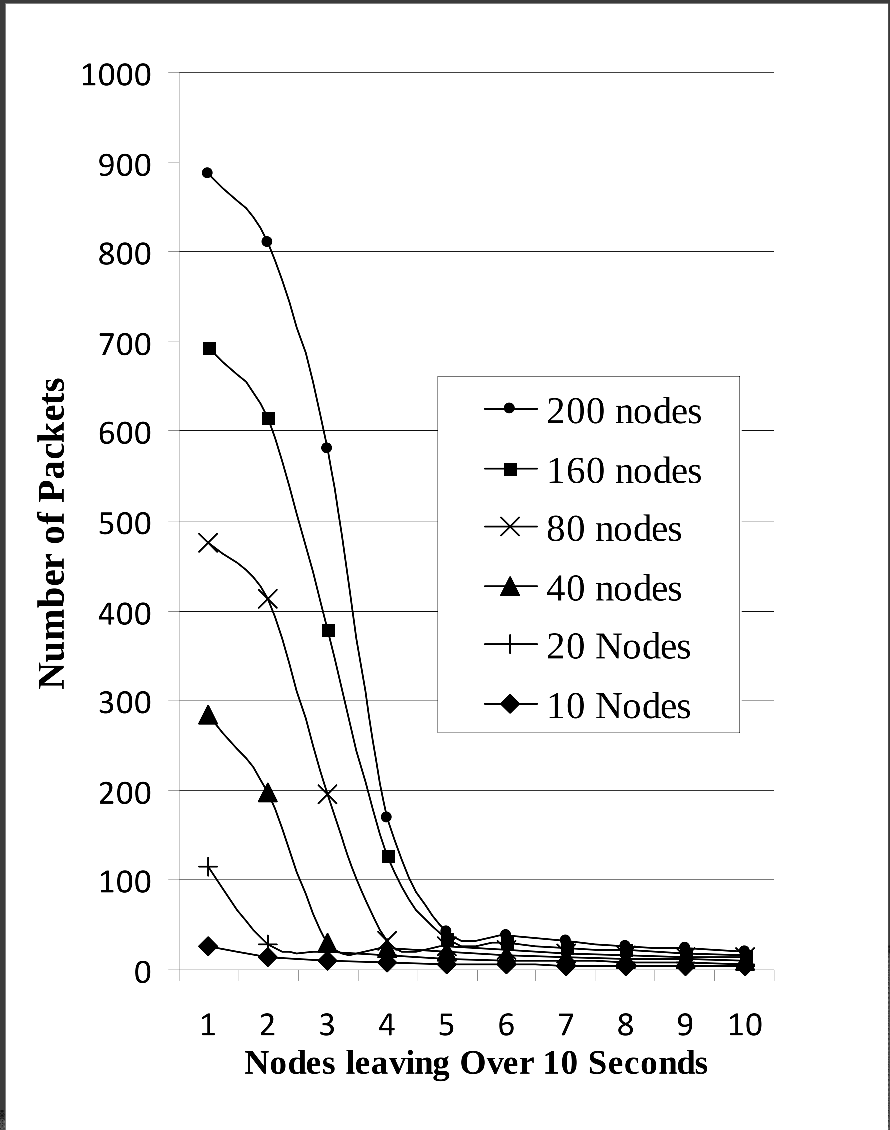
IETF P2P-SIP Reference Model



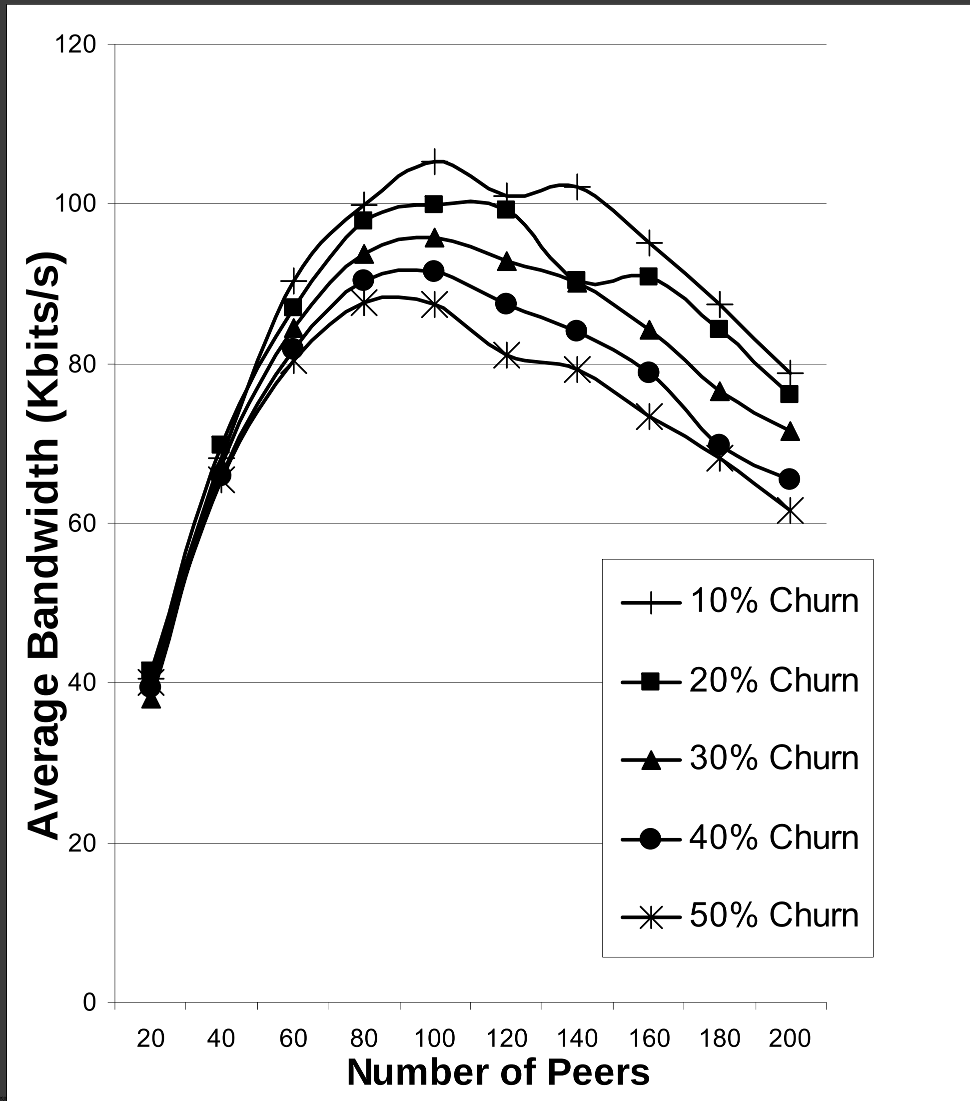
Source : http://tools.ietf.org/pdf/draft-ietf-p2psip-concepts-02.pdf

Case study:
P2P-SIP in Mobile Environments

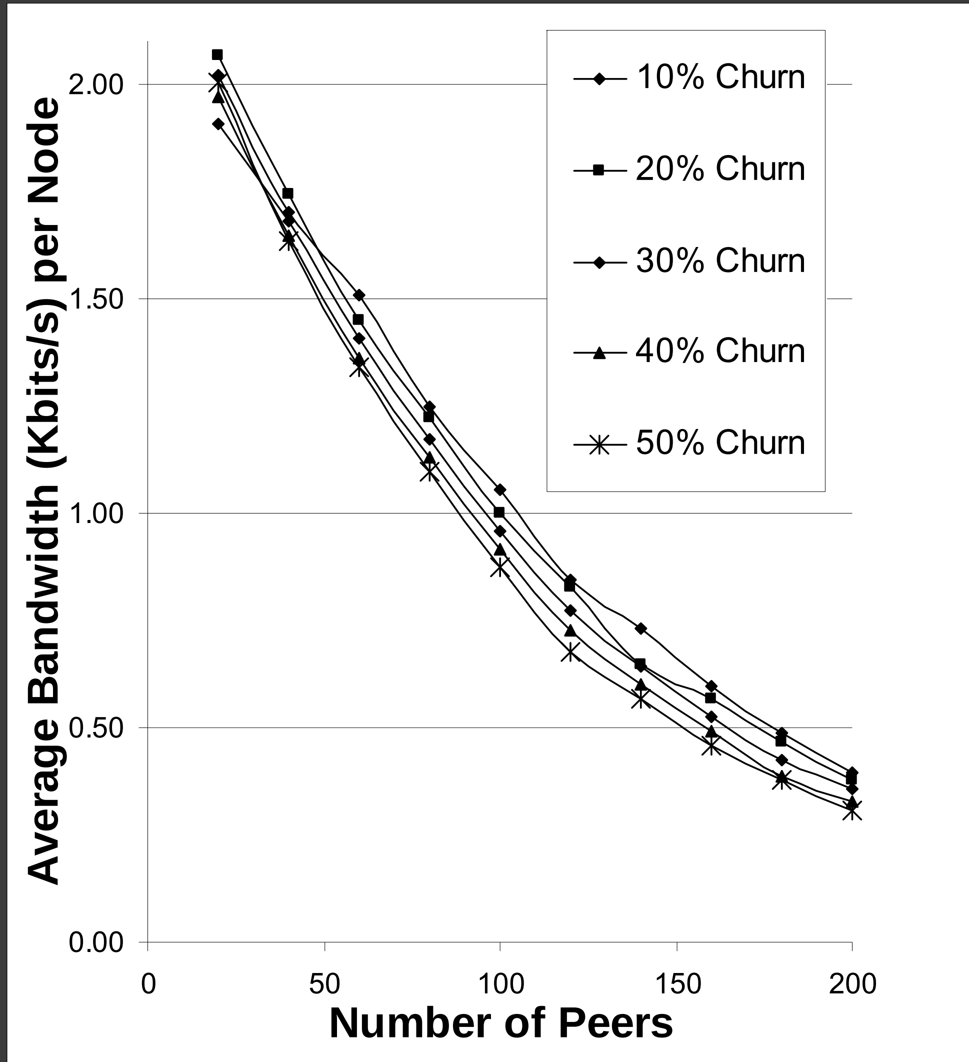
The Effects of Nodes Leaving on the Overlay



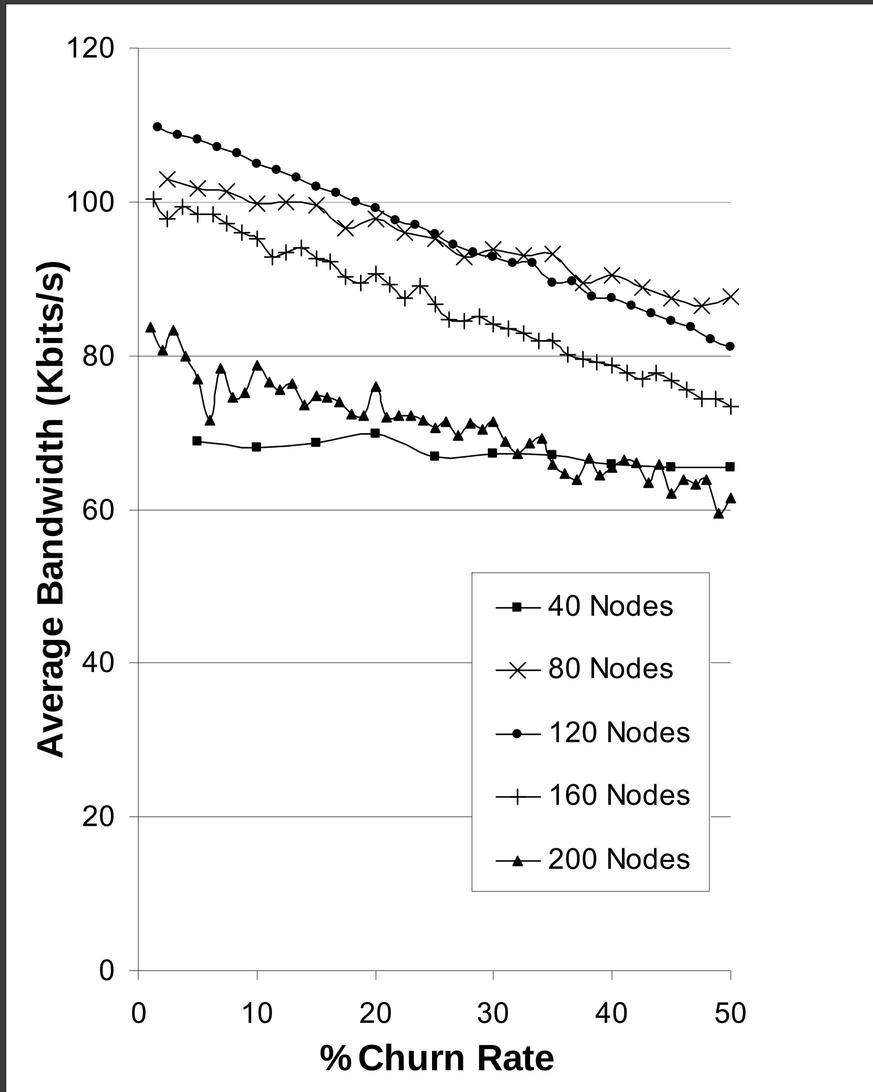
Average Bandwidth in the Overlay



Average Bandwidth Per Peer



Average Bandwidth in terms of Churn Rate



Concluding remarks

- A standardised approach as it is now an IETF Working Group
- It is based on Open Standards such as SIP
- Clients with available resources can host additional services reducing the amount of traffic directed to the provider network
- This allows for the creation of localised and personalised services. i.e offices, schools and hospitals
- Provision of converged services in ad-hoc environments i.e. emergency services and conferences.
- Provides an open interface in which services can be easily deployed.
- Creation of Additional Revenues based on the P2P Service model.

References

- K.W. Ross and Dan Rubenstein, "Tutorial on P2P Systems " Infocom 2003, http://cis.poly.edu/~ross/p2pTheory/P2Preading.htm.
- Dejan Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins and Zhichen Xu "Peer-to-Peer Computing", HP Labs Technical Report, HPL-2002-57 http://www.hpl.hp.com/techreports/2002
- S. Androutsellis-Theotokis and D. Spinnellis "A survey of Peer-to-Peer Content Distribution Technologies" ACM computing surveys, vlo. 36, No. 4, December 2004.
- Karl Aberer, Philippe Cudré-Mauroux, Anwitaman Datta, Zoran Despotovic, Manfred Hauswirth, Magdalena Punceva, Roman Schmidt "P-Grid: A Self-organizing Structured P2P System" ACM SIGMOD Record, 32(3), September 2003
- Athens Univ. of Economics and Bussiness, White Paper : A Survey of Peer-to-Peer File Sharing Technologies. http://www.eltrun.aueb.gr/whitepapers/p2p_2002.pdf
- Matei Ripeanu and Ian Foster, "Mapping the Gnutella Network: Macroscopic Properties of Large-Scale Peer-to-Peer Systems," Proceedings of the 1st International Workshop on Peer-to-Peer Systems, March 2002.
- J. Liang, R. Kumar, K.W. Ross, "Understanding KaZaA", http://cis.poly.edu/~ross/papers/.
- A. Crespo and H. Garcia-Molina, "Semantic overlay networks for p2p systems", Technical report, Computer Science Department, Stanford University, October 2002.
- Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications", IEEE/ACM Transactions on Networking, February 2003.

References (cont.)

- S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network.” In SIGCOMM, August 2001
- Antony Rowstron and Peter Druschel, “Pastry : Scalable, decentralized object location and routing for large-scale peer-to-peer systems”, Proc. Of 18th IFIP/ACM Interna-tional Conference on Distributed Systems Platforms, Heidelberg, Germany, November 2001.
- B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph., "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," Technical Report UCB/CSD-01-1141, UC Berkeley, April 2001.
- Bernard Traversat, Ahkil Arora, Mohamed Abdelaziz, “Project JXTA 2.0 Super-Peer Virtual Network”, http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf
- Ekaterina Chtcherbina, Thomas Wieland, “Project JXTA-Guide to a peer-to-peer framework”, http://www.drwieland.de/jxta-tutorial-part1a.pdf, http://www.drwieland.de/jxta-tutorial-part2a.pdf
- KaZaA, http://www.kazaa.com/.
- “eBay Feedback,” http://pages.ebay.com/help/feedback/reputation-ov.html.
- K. Aberer and Z. Despotovic, “Managing Trust in a Peer-2-Peer Information System,” in The 9th International Conference on Information and Knowledge Management, Atlanta, USA, November 2001, pp. 310–317.
- F. Cornelli, E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati, “Choosing Reputable Servents in a P2P Network,”, in The 11th International World Wide Web Conference, Honolulu, USA, May 2002, pp. 376–386.

References (cont.)

- S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," in *The 12th International World Wide Web Conference*, Budapest, Hungary, May 2003, pp. 640–651.
- M. Gupta, P. Judge, and M. Ammar, "A Reputation System for Peer-to-Peer Networks," in ACM 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video, Monterey, USA, June 2003, pp. 144–152.
- S. Marti and H. Garcia-Molina, "Limited Reputation Sharing in P2P Systems," in *ACM Conference on Electronic Commerce*, New York, USA, May 2004, pp. 91–101.
- L. Mekouar, Y. Iraqi, and R. Boutaba, "A Reputation Management and Selection Advisor Schemes for Peer-to-Peer Systems," in The 15th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM), Davis, USA, November 2004.
- L. Mekouar, Y. Iraqi, R. Boutaba: "Detecting Malicious Peers in A Reputation-Based Peer-to-Peer System," in The IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, USA, January 2005.
- L. Mekouar, Y. Iraqi, R. Boutaba: " Peer-to-Peer Most Wanted: Malicious Peers ," to appear in the Computer Networks Journal, Special Issue on Management in Peer-to-Peer Systems: Trust, Reputation and Security, 2005.
- W. Liu, R. Boutaba and J. W-K. Hong: "pMeasure: A Tool for Measuring the Internet". In Proceedings of the 2nd Workshop on End-to-End Monitoring Techniques and Services (E2EMON), San Diego, CA (USA), October 2004.

Questions