Today
- Overlays networks
- P2P evolution
- Pastry as a routing overlay example
Network virtualization and overlays

- Different applications with a range of demands/needs
  - network virtualization
    - To provide a service tailored to a class of applications
      - DHTs, P2P file sharing, CDNs
    - To support more efficient operation in a given network environment
      - Wireless ad-hoc networks, DTNs
    - To add extra features such as multicast or secure communication
      - Multicast (overlay multicast), resilience, security

- And the costs ...
  - Additional level of indirection
  - Opacity of the underlying network
  - Complexity of the network services
Skype – an example overlay

- **Peer-to-peer VoIP**
  - Developed by Kazaa in 2003, acquired by Microsoft in 2011 for US$ 8.5 billions
  - 34% of the call market share, 50 million concurrent online users in January 2013

- **Notes on design**
  - Super-peer structure (super-peer selected based on availability, reachability, bandwidth, etc)
  - Users login through a well-known server, but connect to the network and others through super-peers
  - TCP for control, TCP or UDP for voice

*Baset & Schulzrinne’s studies*
Peer-to-peer

- A distributed system architecture
  - No centralized control
  - Nodes are symmetric in function
  - Larger number of unreliable nodes

- Promise of P2P
  - Reliability – no central point of failure
    - Multiple replicas
    - Geographic distribution
  - High capacity through parallelism
  - Automatic configuration
  - ...

- Placing and finding resources on an overlay
Three generations of P2P

- Enable by technology improvements in computing and networking
  - Many predecessors (DNS, Netnews/Usenet, Grapevine, …)

- Unstructured and centralized
  - Napster

- Unstructured and decentralized
  - Gnutella, Kazaa, …
  - Peers connect with other random peers
  - Semi-structured models (superpeers) for scalability

- Structured and decentralized
  - E.g. DHTs like Chord, Tapestry, Pastry, Kademlia, and CAN
Napster’s legacy

- Grassroots service for sharing digital music files
- Cluster of central servers
  - Maintain an index of files being shared
  - Monitor state of each peer in the system
  - Maintains metadata for peers (e.g. connection bandwidth)
  - Cooperate to process queries, return a list of matching files and locations
- Peers
  - Maintains a connection to one of the central servers
  - Issue query to servers, and server request from peers
Gnutella – Unstructured and decentralized

- All peers are equal and can connect to anyone (V0.4) or leaf-peers can only connect to super-peer (V0.6)
- No constraints on placement of data objects
- Flooding or random walk for search

\[ y = 4338.8x^{1.0607}, \quad R^2 = 0.9912 \]

Gnutella: Active probing, trying to initiate a handshake, over 7 days & 600K sessions

Common DHT approach

- Assign random (128/160-bit) ID to each node and object
- Define a metric topology on the 128/160-bit numbers, i.e., the space of keys and node IDs
- Each node keeps contact info to $O(\log n)$
- A simple interface – put(key, value), get(key) $\rightarrow$ value
  - A lookup algorithm which finds the node whose ID is closest to a given key
    - Need a metric that identify closest node uniquely
    - Store/retrieve a <key, value> pair at/from that node
- Supports a wide range of applications – no meaning assigned to keys
Pastry as an example

- Routing overlay, a substrate for PAST (distributed file system), SCRIBE (distributed pub/sub), SQUIRRELL (cooperative web caching), SplitStream …
- Hash table's key-space is considered circular (~Chord); node IDs are 128-bit unsigned
- Routing overlay is formed on top of the hash table by each peer discovering and exchanging state information consisting of
  - Leaf nodes – $L/2$ closest peers by ID in each direction around the circle
  - Neighborhood list – $M$ closest peers based on routing metric
  - Routing table – one entry per address block assigned to it
Pastry routing table

- To form the address blocks, divide 128-bit key into digits, each digit $b$ bits long, yielding a numbering system with base $2^b$
  - So addresses are partitioned into distinct levels from the viewpoint of the client

Routing table for node 65a1fc ($b=4$, so $2^b = 16$)
Routing in Pastry

- Whenever a peer receives a msg with key $D$ to route
- If $D$ within the leaf set or is the current node
  - Forward to $L_i$ with GUID closest to $D$ or to current node
- Else use the routing table
  - Find address of a node which shares a longer prefix (at least one digit or $b$ bits) with destination address than the peer itself
  - If this set is not empty
    - Forward to that node
  - Else, if the peer does not have any contacts with a longer prefix or the contact has died
    - Pick a peer from its contact list with same length prefix whose node ID is numerically closer to destination and forward it there
Pastry routing

- Properties
  - $\log_{16} N$ steps
  - $O(\log N)$ size routing table / node

Prefix routing

Route(d46a1c)

65a1fc

d13da3

d4213f

d46a1c

d467c4

d471f1

d462ba
Finding popular objects

- Need exact-match queries to find objects for download
  - Search 20k popular objects from Gnutella in Gnutella & Kademlia (Overnet)

Curves are normalized by number of successful queries (97.4% in Kademlia and 53.2% in Gnutella)

Gnutella: 50% finished in 47s
Kademlia: 50% finished in 17s!

Gnutella is much better for most popular objects

*Qiao and Bustamante, Structured and Unstructured Overlays Under the Microscope - A Measurement-based View of Two P2P Systems That People Use, In Proc. of USENIX ATC 2006*
Summary

- New applications with new demands on the underlying network
- Architectural changes are, at best, difficult
- Overlays both as a path to deployment and an experimental testbed
  - Deploying narrow fixes?
  - No demands on underlying network (to ensure deployment)
- From grassroots efforts and research labs to products
  - But much research to be done
- Future Internet and overlays?