Unit 5

Napster

One of the earliest popular P2P systems, Napster [25], used a server-mediated central index architecture organized around clusters of servers that store direct indices of the files in the system. The central server maintains a table with the following information of each registered client: (i) the client's address (IP) and port, and offered bandwidth, and (ii) information about the files that the client can allow to share. The basic steps of operation to search for content and to determine a node from which to download the content are the following:

1. A client connects to a meta-server that assigns a lightly loaded server from one of the close-by clusters of servers to process the client's query.
2. The client connects to the assigned server and forwards its query along with its own identity.
3. The server responds to the client with information about the users connected to it and the files they are sharing.
4. On receiving the response from the server, the client chooses one of the users from whom to download a desired file. The address to enable the P2P connection between the client and the selected user is provided by the server to the client.

Data indexing and overlays

The data in a P2P network is identified by using indexing. Data indexing allows the physical data independence from the applications. Indexing mechanisms can be classified as being *centralized*, *local*, or *distributed*:

- **Centralized indexing** entails the use of one or a few central servers to store references (indexes) to the data on many peers. The DNS lookup as well as the lookup by some early P2P networks such as Napster used a central directory lookup.
- **Distributed indexing** involves the indexes to the objects at various peers being scattered across other peers throughout the P2P network. In order to access the indexes, a structure is used in the P2P overlay to access the indexes. Distributed indexing is the most challenging of the indexing schemes, and many novel mechanisms have been proposed, most notably the *distributed hash table (DHT)*. Various DHT schemes differ in the hash mapping, search algorithms, diameter for lookup, search diameter, fault-tolerance, and resilience to churn.

  A typical DHT uses a flat key space to associate the mapping between network nodes and data objects/files/values. Specifically, the node address is mapped to a logical identifier in the key space using a consistent hash function. The data object/file/value is also mapped to the same key space using hashing. These mappings are illustrated in Figure 18.1.
- **Local indexing** requires each peer to index only the local data objects and remote objects need to be searched for. This form of indexing is typically used in unstructured overlays in conjunction with flooding search or random walk search. Gnutella uses local indexing.

Distributed indexing

Structured overlays

The P2P network topology has a definite structure, and the placement of files or data in this network is highly deterministic as per some algorithmic mapping. (The placement of files can sometimes be "loose," as in some earlier P2P systems like Freenet, where "hints" are used.) The objective of such a deterministic mapping is to allow a very fast and deterministic lookup to satisfy queries for the data. These systems are termed as *lookup systems* and typically use a hash table interface for the mapping. The hash function, which efficiently maps *keys* to *values*, in conjunction with the regular structure of the overlay, allows fast search for the location of the file.

An implicit characteristic of such a deterministic mapping of a file to a location is that the mapping can be based on a single characteristic of the file (such as its name, its length, or more generally some *predetermined* function computed on the file). A disadvantage of such a mapping is that arbitrary queries, such as range queries, attribute queries and exact keyword queries cannot be handled directly.

Unstructured overlays

Unstructured overlays: properties

Unstructured overlays have the serious disadvantage that queries may take a long time to find a file or may be unsuccessful even if the queried object exists. The message overhead of a query search may also be high.

The following are the main advantages of unstructured overlays such as the one used by Gnutella:

- Exact keyword queries, range queries, attribute-based queries, and other complex queries can be supported because the search query can capture the semantics of the data being sought; and the indexing of the files and data is not bound to any non-semantic structure.
- Unstructured overlays can accommodate high churn, i.e., the rapid joining and departure of many nodes without affecting performance.

The following are advantages of unstructured overlays if certain conditions are satisfied:

- Unstructured overlays are efficient when there is some degree of data replication in the network.
- Users are satisfied with a best-effort search.
- The network is not so large as to lead to scalability problems during the search process.

Gnutella

Message Types Used by Gnutella

- *Ping* messages are used to discover hosts, and allow a new host to announce itself.
- *Pong* messages are the responses to *Pings*. The *Pong* messages indicate the port and (IP) address of the responder, and some information about the amount of data (the number and size of files) that node can make available.
- *Query* messages. The search strategy used is flooding. *Query* messages contain a search string and the minimum download speed required of the potential responder, and are flooded in the network.
- *QueryHit* messages are sent as responses if a node receiving a *Query* detects a local match in response to a query. A *QueryHit* contains the port and address (IP), speed, the number of files found, and related information. The path traced by a *Query* is recorded in the message, so the *QueryHit* follows the same path in reverse.

Graph structures of complex networks

P2P overlay graphs can have different structures. An intriguing question is to characterize the structure of overlay graphs. This question is a small part of a much wider challenge of how to characterize large networks that grow in a distributed manner without any coordination [4]. Such networks exist in the following:

- Computer science: the WWW graph (WWW), the Internet graph that models individual routers and interconnecting links (INTNET), and the autonomous systems (AS) graph in the Internet.
- Social networks (SOC), the phonecall graph (PHON), the movie actor collaboration graph (ACT), the author collaboration graph (AUTH), and citation networks (CITE).
- Linguistics: the word co-occurrence graph (WORDOCC), and the word synonym graph (WORDSYN).
- The power distribution grid (POWER).
- Nature: in protein folding (PROT), where nodes are proteins and an edge represents that the two proteins bind together, and in substrate graphs for various bacteria and micro-organisms (SUBSTRATE), where nodes are substrates and edges are chemical reactions in which substrates participate.

Current empirical measurements show the following properties of some commonly occuring graphs:

WWW In-degree and out-degree distributions both follow power laws; it is a small world; and is a directed graph, but does show a high clustering coefficient.

INTNET Degree distributions follow power law; small world; shows clustering.

AS Degree distributions follow power law; small world; shows clustering.

ACT Degree distributions follow power law tail; small world (similar path length as ER); shows high clustering.

AUTH Degree distributions follow power law; small world; shows high clustering.

SUBSTRATE In-degree and out-degree distributions both follow power laws; small world; large clustering coefficient.

PROT Degree distribution has a power law with exponential cutoff.

PHON In-degree and out-degree distributions both follow power laws.

CITE In-degree follows power law, out-degree has an exponential tail.

WORDOCC Two-regime power-law degree distribution; small world; high clustering coefficient.

WORDSYN Power-law degree distribution; small world; high clustering coefficient.

POWER Degree distribution is exponential.

Small-world networks

- Real-world networks are small worlds, having small diameter, like random graphs, but they have relatively large clustering coefficients that tend to be independent of the network size.
- Ordered lattices tend to satisfy this property that clustering coefficients are independent of the network size.

1. Define a ring lattice with n nodes and each node connected to k closest neighbors ($k/2$ on either side). Let $n \gg k \gg ln(n) \gg 1$.
2. Rewire each edge randomly with probability p. When $p = 0$, there is a perfect structure, as in Figure 18.15(a). When $p = 1$, complete randomness, as in Figure 18.15(c).

Scale-free networks

- Semi-random graphs that are constrained to obey a power law for the degree distributions and constrained to have large clustering coefficients yield scale-free networks, but do not shed any insight into the mechanisms that give birth to scale-free networks.

- Rather than begin with a constant number of nodes *n* that are then randomly connected or rewired, real networks (e.g., WWW, INTERNET) exhibit *growth* by the addition of nodes and edges.
- Rather than assume that the probability of adding (or rewiring) an edge between two nodes is a constant, real networks exhibit the property of *preferential attachment*, where the probability of connecting to a node depends on the node degree.