Peer-to-Peer

Apr 25 2016:

Overlay networks:

- Logical network implemented on top of a physical network
    - Subset of nodes from physical network exist in the logical network
    - There can be several overlay networks existing on top of the same physical network
    - The Internet itself is an overlay network!
    - Communication between nodes happens through the use of tunnels
        - A tunnel simply uses an inner-header for communication between nodes of the overlay. The outer headers are TCP/IP or UDP/IP.
        - For example, in P3, we had an inner header which gave identifiers to nodes and used the TCP/IP to provide communication between different users of the chatroom.
        - Another example is in tunneling IPv6 packets over IPv4-only networks. Tunnel server A on the edge of v6-network-1 will slap on the IPv4 header on a packet and send it to Tunnel server B attached to the destination, v6-network-2. Tunnrl server B will strip the v4 header and forward the packet to the correct v6 destination
        - Detour into Network Address Translation and how it is a stop-gap measure to fix IPv4 address-scarcity. Not ideal, since it makes it hard to run servers (requires port-forwarding to be able to run servers)

    - Example use(s) of overlays:
        - Alibi routing
            - Goal is to show definitely that a packet *did not* traverse a country (say)
            - Source ‘s’ wants to send a packet ‘p’ to destination ‘d’ avoiding region ‘f’
            - Show that event ‘x’ happened which conclusively proves that event ‘y’ did not happen, since ‘x’ and ‘y’ are mutually exclusive
            - Use relay (‘r’) with known locations in other countries to avoid the country you want to avoid.
            - Provide proof of ‘x’ (that packet traversed the relay). Show that ‘x’ implies ‘y’ did not happen (that packet did not traverse f)
            - R(s, d) = R(s, r) + R(r, d)
            - R(s, r) << R(s, f) + R(f, r)
            - R(r, d) << R(r, f) + R(f, d)

        - Use speed of light
        - Paper: http://www.cs.umd.edu/~dml/papers/alibi_sigcomm15.pdf
    - P2p networks (Covered next)

P2p networks:

- Community of users (peers) pool in their resources (storage, processing, network b/w) to obtain some service that's mutually beneficial
- Peers talk to each other over an overlay network
- Peer A wants to download a file. It needs to do two things:
    - Find peer B which has the file (the search problem)
    - Download from peer B (the transaction problem)
    - The common thing about most p2p is that the participants in the transaction problem are normal users (peers)
        - With the caveat that sometimes, hybrid systems can exist where some of the participants are 'special'

- How does peer A find who has the file it seeks (the search problem)?
    - Napster: Central server keeps track.
        - Simple but has problems
        - Could potentially experience high load
        - Single point of failure
    - Gnutella: Each node keeps track of a few others (neighbors)
        - Floods request with QID (Query identifier)
        - Also need a TTL so that queries don't keep living forever
        - Originally, Nodes keep track of which neighbor sent them what QID so that they can forward responses from downstream neighbors back to upstream ones.
            - This limited scalability
            - Today, they usually directly contact the source of the request (over UDP, not over the p2p protocol)
        - How would you bootstrap?
            - Gnutella can always run a few peers that are 'always on' at the same IP address, and ship this with the client code base
            - Connecting to one other node is sufficient because messages will start being flooded through the new client node and it will discover more peers
            - Subsequently, keep track of who you connected to last time so that you can reuse them next time
        - Problems
            - Lack of scalability due to flooding
            - Easy to find popular files but expensive to find files that aren't so

popular

- What are other alternatives? DNS scales, but it isn't fully decentralized.
- DHT (Let's do this in detail)