Peer to Peer Applications: Overview

Kameswari Chebrolu

Peer to Peer (P2P)

- Peer: Desktop/Laptops/Smart-phones managed by users
  - Contribute resources in the form of processing power, bandwidth, disk storage.
- Peer to Peer Architecture: Architecture that distributes application functionality across peers
  - Peers often equal; act as both client (consumer) and server (supplier)

Usage Scenarios

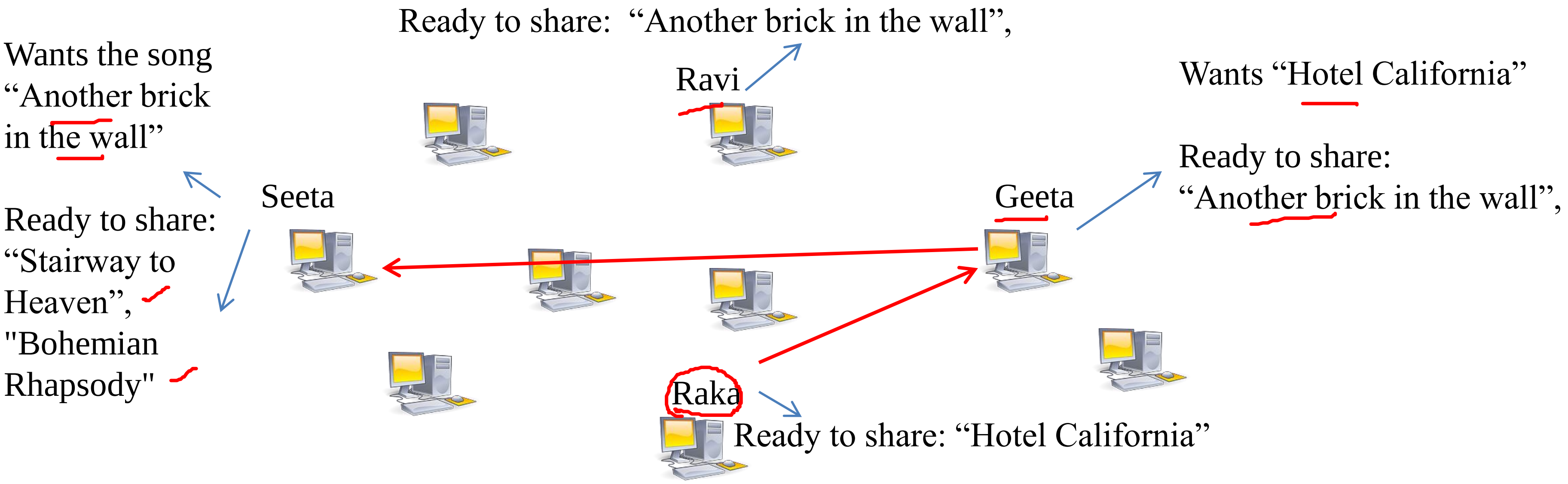
- **File Sharing:** movies, songs, software such as Linux distributions, software updates, games etc
- Multimedia: Voice calls (E.g. Skype) p2p
- Miscellaneous: currency (Bitcoin), web caches, distributed databases

Why P2P?

- Consider distribution of large files: If client-server architecture is used to serve a large client population
  - Server needs to be always-on
  - Server needs to be powerful → Processing ✓
  - Server needs to have high bandwidth ✓
  - Above imply high cost; difficult to scale
- P2P leverages peer resources → low cost, highly scalable
  - Challenge: Intermittent connectivity

Problem Statement

- Help users share files among themselves



Ready to share: “Another brick in the wall”,

Ravi

Wants the song “Another brick in the wall”

Seeta

Ready to share: “Stairway to Heaven”, "Bohemian Rhapsody"

Wants “Hotel California”

Geeta

Ready to share: “Another brick in the wall”,

Raka

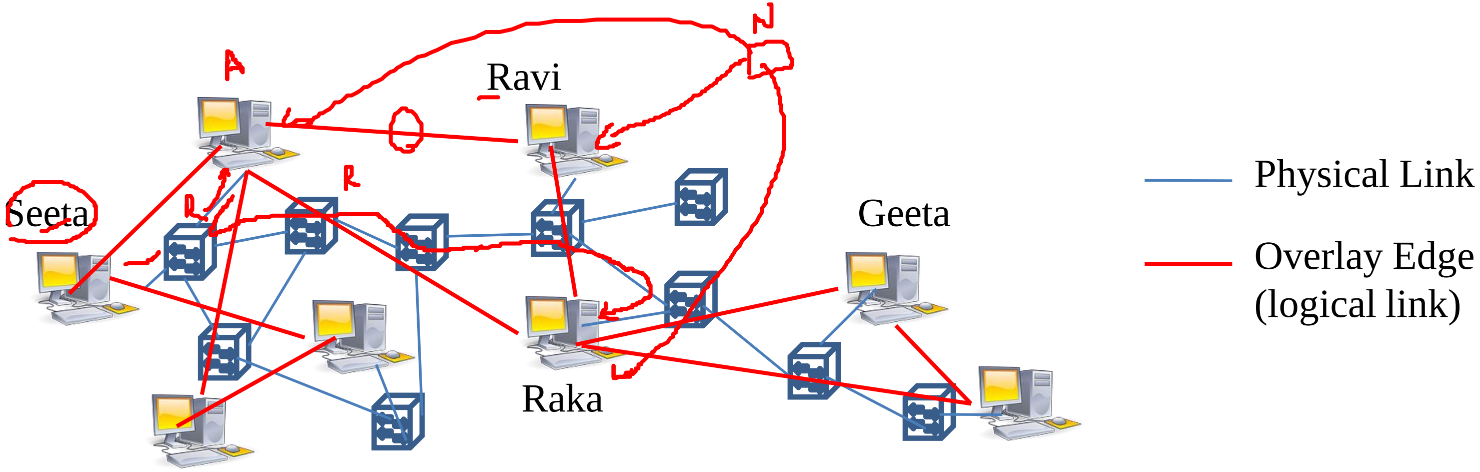
Ready to share: “Hotel California”

Need to put Geeta or Ravi in touch with Seeta; Raka in touch with Geeta

Geeta: Acts as server when uploading “Another brick in the wall” to Seeta and as client when downloading “Hotel California” from Raka

Solution Approach: Overlay Networks

- Overlay Network (red interconnections) formed fully at application layer
  - Nodes are end-users; links could be TCP connections



Application Functionality

- Join : how does a user join the peer-to-peer overlay network?
- Publish : how does a user advertise files willing to share?
- Search : how does a user find a file?
- Fetch : how does a user download a file?
- Challenge: Peers come and go (churn); Peer's IP address can change over time

Types of Overlays

- **Unstructured overlays:**
  - Edges between nodes are randomly formed (E.g. a new node randomly chooses three existing nodes in overlay as neighbors)
  - Easy to build, robust against churn; search however is inefficient
- Structured overlays
  - Edges arranged carefully to speed up search; rare files can also be found
  - Based on Distributed Hash Tables (DHT)
  - Complex, less robust against churn