COMP 790-088: Networked & Distributed Systems

Content Distribution Systems

Jasleen Kaur

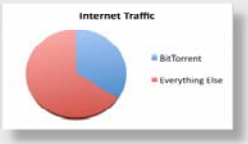
November 18, 2009

Content Distribution

- How to distribute content without requiring centralized, heavy-duty servers?
- Examples:
  - » Bittorrent
    - ❖ Peer-to-peer content distribution
  - » Akamai
    - ❖ Content distribution service

Bittorrent: Introduction

- Peer to Peer file transfer protocol
  - » Files are shared by many users
  - » Active participation of all users
- Transfer of large files
- Huge success in the file sharing domain
  - » 35% of internet traffic



Striking Characteristics

- A central server is not needed
  - » No single point of failure
  - » Allows distribution of content without straining budgets (low bandwidth, small computers)
- Not dependent on a single source
  - » Automatic replication
  - » Original source harder to trace
- More the number of users, greater will be the transfer rate
  - » Rate proportional to popularity of file
  - » Viral spreading of file throughout peers
- Data available in pieces, not as a single large file
- Tit for tat strategy
  - » Incentive for contributing resources

Architecture of BitTorrent

Components:
- Metainfo File
- Data
- Tracker
- Peers
- Client

HTTP → Peer – Tracker
TCP → Peer – Peer

Metainfo File

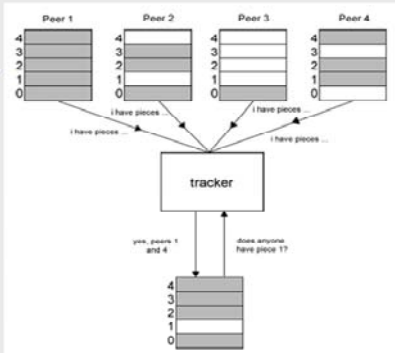
- Contains all information about a torrent
  - » File with a .torrent extension
- It has the following keys
  - » Info, tracker-info, creation date, comment, created by
  - » Keys are encoded before they are sent
- Hash of all the pieces are present in info field of metainfo file
- Files are uploaded in public sites by seeds
  - » Users download this file via HTTP and can participate in the torrent transfer

Data

- Data can be a single file or multiple files contained in a directory
- Data is split into many pieces of equal size
  - » Common piece sizes are 256 KB, 512 KB and 1 MB
  - » Each piece is further divided into many blocks
- A piece will have a hash, needed for data integrity check

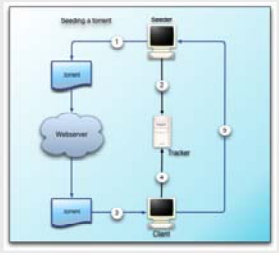
Tracker

- Manages users participating in a torrent
- Helps peers find each other
  - » Peers request tracker for other peers having the required pieces
  - » Tracker responds with a list of peers that have the requested piece
- A tracker can manage many torrents
- It is a HTTP service that works on port 6969



Working of BitTorrent

1. Seeder generates a torrent file and uploads torrent to a web server.
2. The seeder notifies the tracker that it is sharing the file described in the torrent file.
3. A leecher downloads the torrent file from the web server.
4. The leecher connects to the tracker specified in the torrent file.
5. The leecher connects to its peers to retrieve pieces of the files.



Peers

- Peers speak TCP
  - » Ports 6881 – 6889 are used by peers
- Following key strategies are used by peers while sharing files
  - » Rarest First
    - Ensures that peers have all pieces that *their* peers want (increased exchange opportunity)
    - Low likelihood that a currently uploading peer will later not have anything of interest to others
    - Implies only new pieces are downloaded from the original seed (no flash crowds)
  - » Random First Piece
    - Rarest may come from a slow peer (need a piece quickly)
  - » Endgame Mode
    - Ask all peers for last sub-pieces (prevent a slow peer from delaying your finish)
  - » Peer reciprocation
    - Upload to peers which upload to you (achieve pareto efficiency)
  - » Choking and Optimistic Unchoking
    - Allows to discover and tune-out peers that can offer better download rates

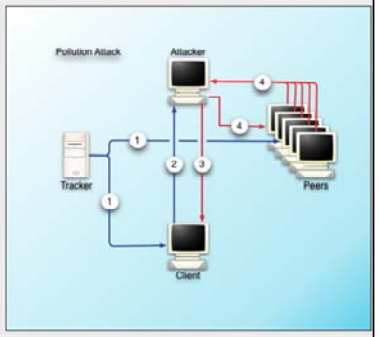
Client

- Executable program running on user's machine
- Co ordinates with OS to perform read write operations
- A .torrent file must be opened by the client
- Peers with same client perform better

Attacks on BitTorrent

Pollution Attack

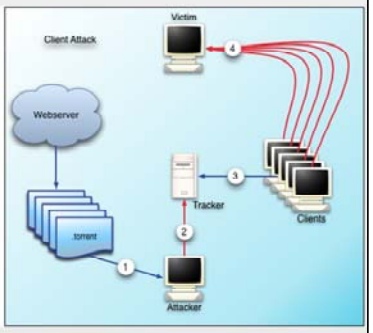
1. The peers receive the peer list from the tracker.
2. One peer contacts the attacker for a chunk of the file.
3. The attacker sends back a false chunk.
4. Attacker requests all chunks from swarm and wastes their upload bandwidth.



Attacks on BitTorrent

DDOS Attack

1. The attacker downloads a large number of torrent files from a web server.
2. Attacker spoofs IP address and port with that of victim and notifies the tracker
3. Tracker directs peers towards victim
4. Victim will be flooded with requests from other peers



Attacks on BitTorrent

- Bandwidth Shaping
  - » This is done by user's ISP
  - » Unencrypted BitTorrent packets are easily identified and filtered.
  - » Sophisticated filtering software can detect BitTorrent like behavior.
  - » Comcast has recently admitted to filtering BitTorrent traffic.

sandvine Intelligent Broadband Networks

comcast

Current Solutions

Pollution Attack

- Blacklisting
  - » Achieved using software such as Peer Guardian or moBlock.
  - » Blocks connections from blacklisted IPs which are downloaded from an online database.

Current Solutions

DDOS Attack

- Spoofing needs to be avoided in the first place
  - » This can be done by using filters
- Make the tracker validate a peer whether it has the torrent or not

Current Solutions

Bandwidth Shaping

- Encryption
  - » Most BitTorrent clients can encrypt the packets they send.
  - » Simple filters which simply look at the contents of the packet can easily be traversed
- Tunneling
  - » Using VPN software to connect to an unfiltered network.
  - » Such tunnels which are free from filters provide easy path to BitTorrent packets

Conclusion

- BitTorrent is a popular P2P technology deployed across the Internet.
  - » 27-55% of Internet traffic (measured at different locations)
- The protocol has found a niche as a preferred method for the decentralized distribution of large files.
- Pros:
  - » Lower cost to content provider (used even by organizations that want to distribute their own or licensed material)
  - » Higher redundancy (and availability)
  - » Greater resistance to "flash crowds"
- Cons:
  - » Downloads can take time to rise to full speed (it takes time for a node to become an effective uploaded)
  - » Non-contiguous download not suitable for "streaming" or "progressive downloads"
  - » Does not offer use anonymity (exposes users with insecure systems)
  - » Causes home routers to lock up (frequently contacts 300-500 servers per second, filling up NAT tables)