nTorrent: BitTorrent in Named Data Networking

Spyros Mastorakis

Internet Research Laboratory

UCLA

BitTorrent in TCP/IP

- BitTorrent is a popular peer-to-peer file sharing application
- BitTorrent aims to achieve:
  - Robust and efficient data dissemination among multiple parties (peers)
  - Authentication of individual data blocks
  - Data downloading from any peer (a peer does not care from whom they download data)
  - Data downloading in parallel from multiple peers to minimize the torrent downloading time

TCP/IP hurdles to BitTorrent

- BitTorrent needs to explicitly discover and select peers to retrieve data from:
  - pick specific IP addresses
- BitTorrent as an application layer protocol cannot be aware of the underlying network connectivity
  - such information is available at the network layer
  - ***result:*** massive amounts of long distance/inter AS traffic
- BitTorrent has to implement a data-centric logic at the application layer of TCP/IP

What is nTorrent?

- nTorrent is a proof-of-concept NDN application
  - similar functionality to BitTorrent (i.e., peer-to-peer file sharing)
- nTorrent leverages NDN
  - data-centric logic directly at the network layer

Why NDN though..?

- *Provide data-centric security per data packet* directly at the network layer
  - data integrity can be verified by both the network and the applications
  - **BitTorrent** uses hashes per piece, can only be verified by applications
- *Maximize download speed* directly at the network layer
  - e.g., parallel downloading, use the most efficient path first
  - **BitTorrent** has to do so at the application layer by explicitly selecting the "best" peers
- *Efficient data retrieval* directly at the network layer
  - e.g., traffic localization
  - **BitTorrent** uses ways external to the protocol (e.g., DNS "tricks", "local" trackers)

Design Challenges / Questions (1/2)

- How nTorrent data should be named?
  - *multiple files per torrent*
  - *multiple packets per file*
- How can we learn what Interest names to express?
- How peers bootstrap?
  - *“stable peers” with hardcoded prefixes*?
  - *trackers*?
- How to deal with routing scalability issues?
  - *torrent name routable across global Internet*?

Design Challenges / Questions (2/2)

- How peers interact with each other?
- How peers can learn routable prefixes?
- How peers can sign LINKs?
  - *opportunistic data dissemination*
  - *very dynamic environment*
  - *unsigned LINKs?*
- *BitTorrent is inherently liberal*
  - *Does not verify whether peers are legit*
  - *Just try... If the desired data comes back, assume the peer is legit..*
  - *Should we do the same?*
  - *Should we do more?*

Interested in Technical Details?

Please take a look at our poster later today!

Current Status

- Design almost finalized ***(still open to suggestions!)***
- Application implementation is underway: https://github.com/spirosmastorakis/nTorrent
- Poster: Later today and available online: *http://web.cs.ucla.edu/~mastorakis/nTorrent.pdf*
- Technical Report: ***Coming Soon!***

Q/A

Thank you!