The bitTAKA Protocol

A modified BitTorrent protocol for anonymous peer-to-peer file sharing

Tural Badirkhanli, Amrik Kochhar, Albert Ni, Kuat Yessenov

Outline

- BitTorrent – Pros and Cons
- BitTAKA – Anonymous BitTorrent
- Simulation
- Possible Improvements

BitTorrent Pros

- Efficient because downloaders also act as uploaders
- Uploading is enforced via “tit-for-tat” based protocol

BitTorrent Cons

- Anonymity – users learn identifying information about their peers

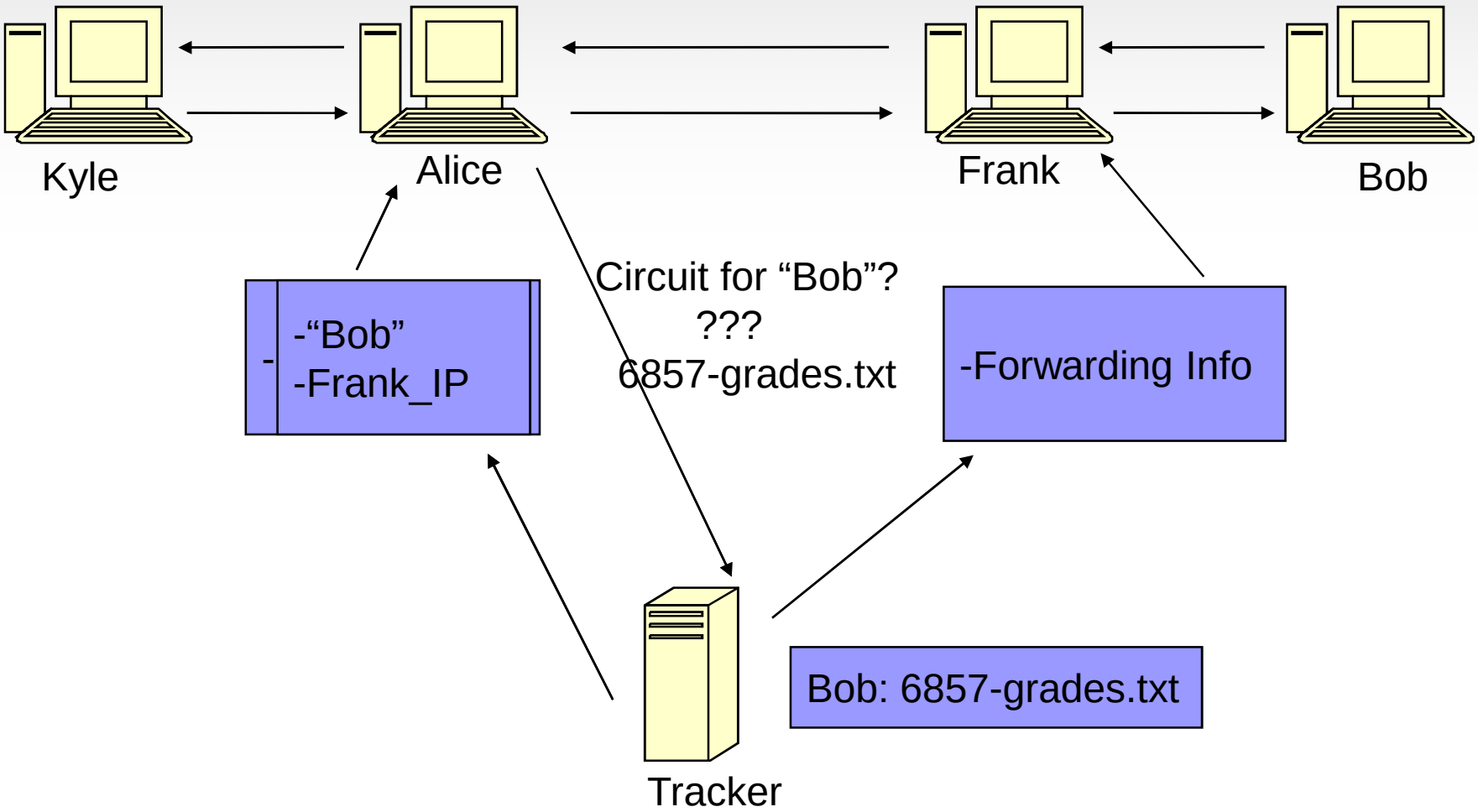
bitTAKA Goals

- Share files using a protocol fundamentally similar to BitTorrent's
- Provide anonymity to users of the bitTAKA network
  - In particular, users can share the files they want without learning anything about other users of the network
- Still be "reasonably" efficient

Protocol Outline

- Requires a trusted tracker to mediate traffic
- Traffic between peers sharing a file is forwarded through a third-party (called a “Forwarder”)

BitTAKA Protocol



Protocol Review

- Two-way IP to Name mapping maintained by the tracker
  - Any element of this mapping is NEVER learned by any user
- Incentive for forwarding is maintained by always having forwarding go both ways

Review of Goals I

- Still share files using something similar to the BitTorrent protocol
- In particular – how is tit-for-tat still enforced?

Review of Goals II

- Provide anonymity to users of the bitTAKA network
- In particular, can users share the files they want without learning anything about other users of the network?

Review of Goals III

- Still be “reasonably” efficient
- In particular, how do the extra hops affect performance?

Flaws

- Perhaps impractical due to issues with:
  - Scalability
  - Efficiency
- Still need to trust tracker – single point of “trust failure”

Potential Practical Improvements

- Cycle through forwarders before choking a peer
- Maintain data about each others' forwarding speeds
- Distributing tracker routing information
  1. Assumption of anonymity changes
  2. How to discover new routes?

Simulation

- Designed a basic testing framework in Java, able to simulate dozens of peers locally, and propagate messages
- Intend to model:
  - Choking algorithm
  - Network delays
  - Malicious peers and forwarders
- Performance statistics

Questions?