

CS 352 – Project 1

Java-Based BitTorrent Client

Robert Moore

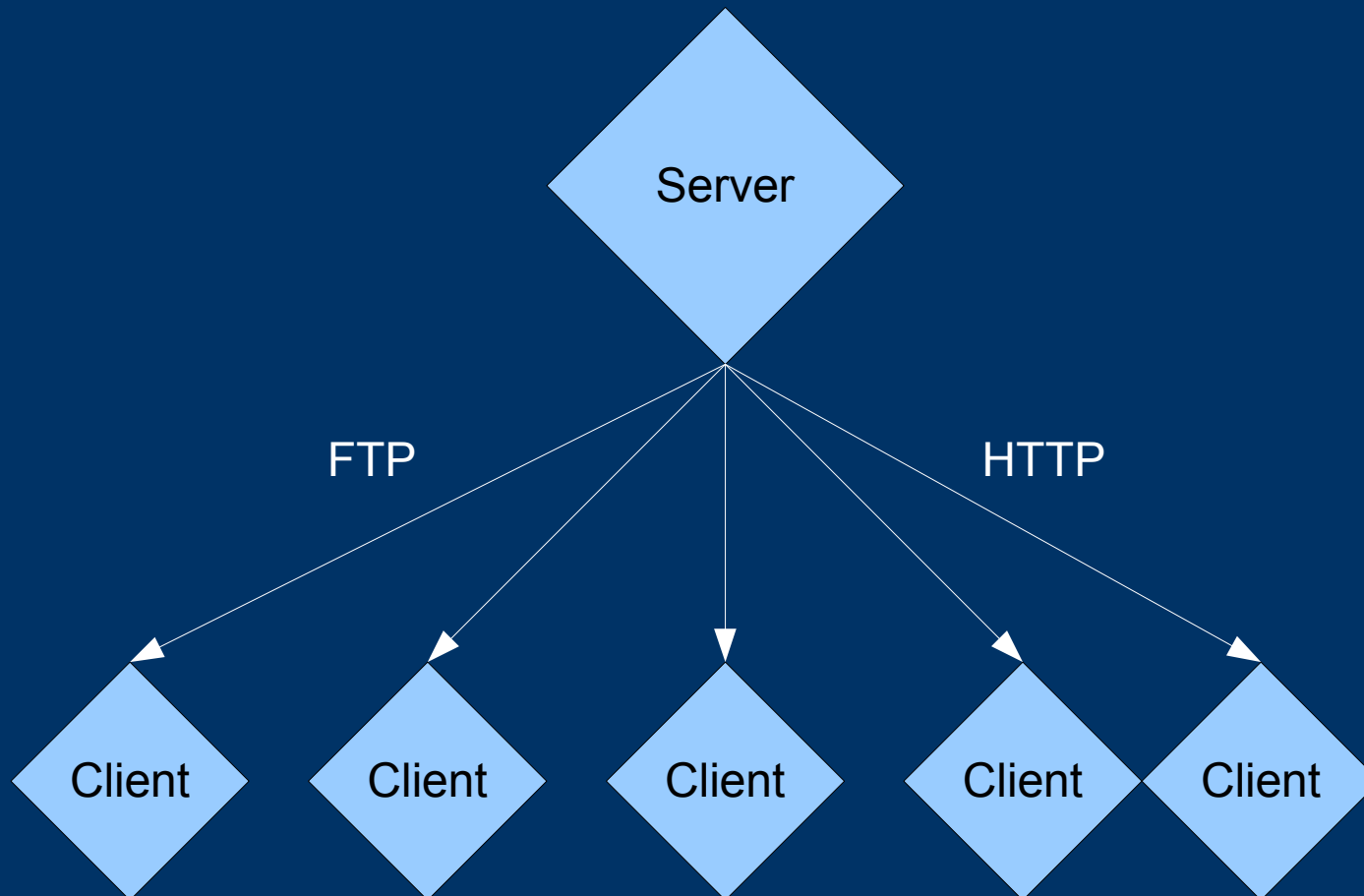
February 20, 2006

Presentation Outline

- Traditional File Sharing Methods
- The BitTorrent Idea
- The BitTorrent Protocol
- Project 1 Basics

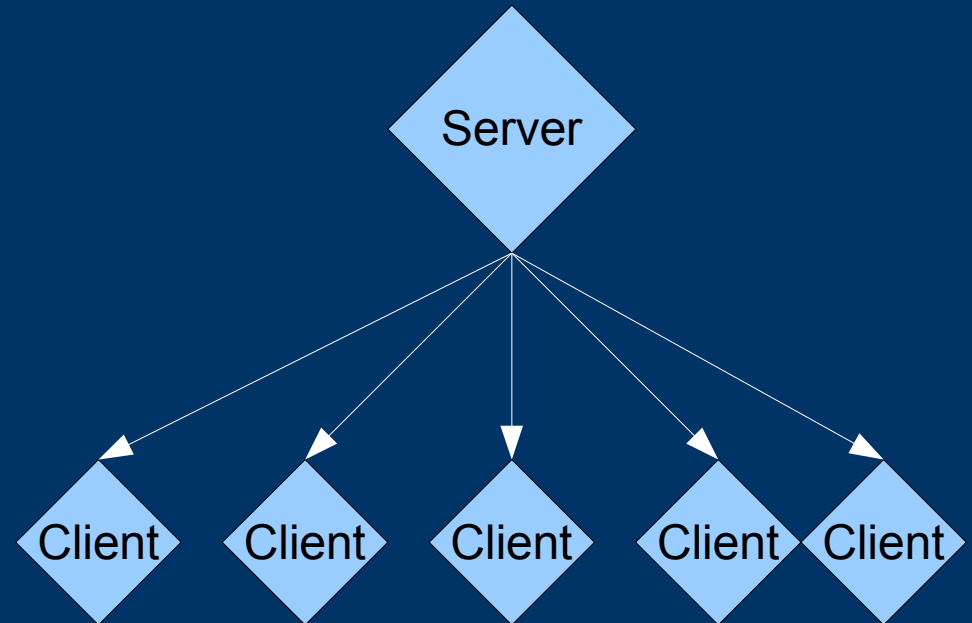
Traditional File Sharing

- Traditional File Transfer/Sharing Setup



Traditional File Sharing (cont'd)

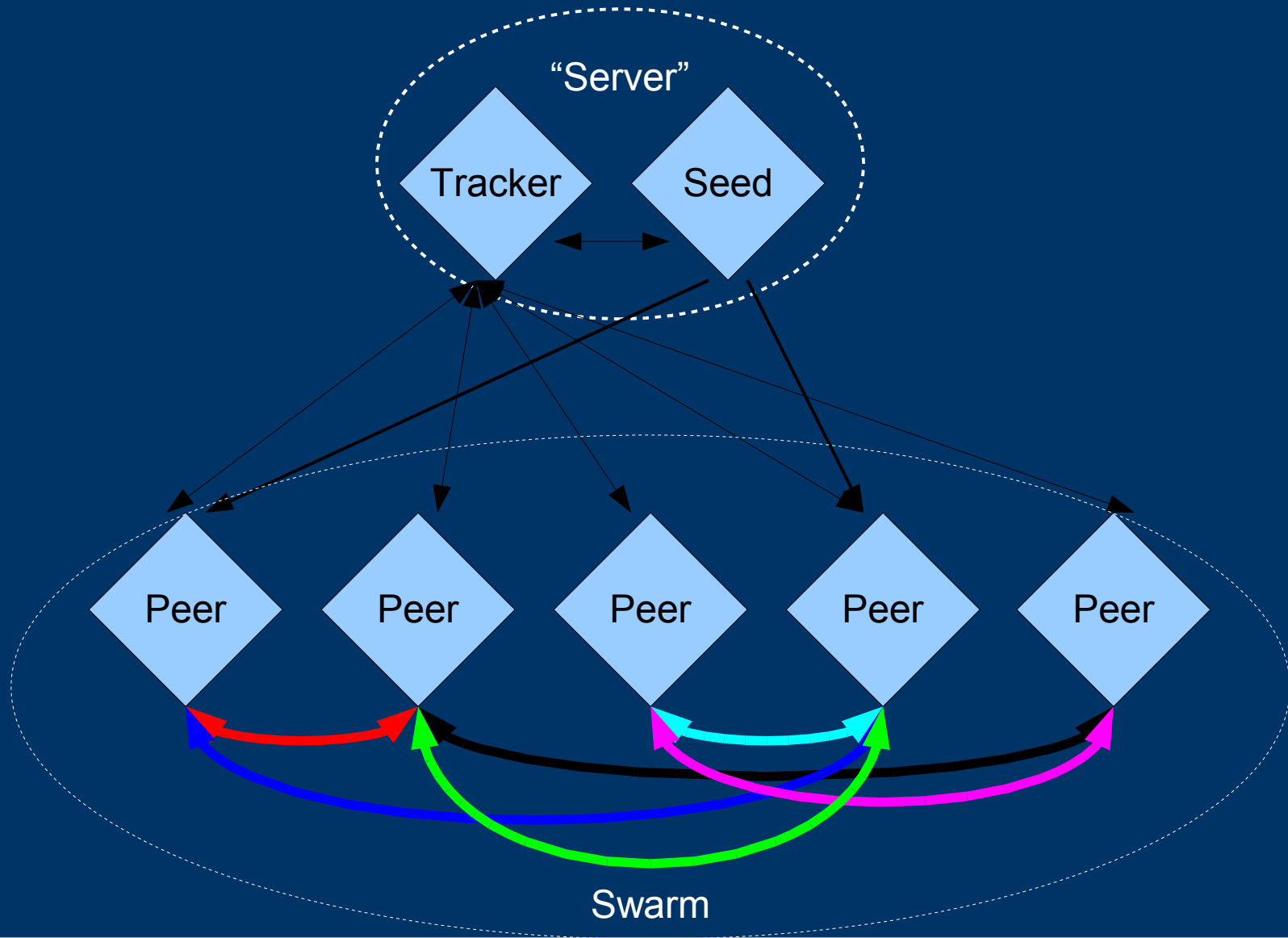
- Benefits
 - Simple
 - File Updates
 - Small Files
 - Few Clients
- Drawbacks
 - Saturated Server Upstream
 - Unused Client Upstream
 - File Size
 - Many Clients



Traditional File Sharing (cont'd)

- History
 - First Proposed in 1971 (IETF RFC114)
 - Small File Sizes
 - Low Bandwidth
- Present
 - Large File Sizes (8GB+ DL-DVD ISO)
 - Improved Bandwidth
- How to Improve?
 - Increase Client Downstream
 - Utilize Client Upstream?
 - Decrease Server Upstream

The BitTorrent Idea



The BitTorrent Idea (cont'd)

- Main Ideas
 - Peers “Serve” Each Other
 - Server is Split Into Tracker & Peer (Seed)
- Benefits
 - Uses Peer Upstream
 - Increases Peer Downstream¹
 - Reduces Seed Upstream²
- Drawbacks
 - Multiple Simultaneous Connections
 - Client Upstream Use Can Saturate Pipe

The BitTorrent Idea (cont'd)

Connections

- Tracker/Peer
 - Metainfo Only
 - Low Bandwidth
 - Many Simultaneous Connections (Tracker)
- Peer/Peer
 - Communication/Data
 - Variable Bandwidth
 - Single Connection Between Peers
 - Multiple Peers Connected Simultaneously

The BitTorrent Protocol

- Tracker Requests are HTTP_GET
- Tracker Responses are Bencoded
- All Peer Traffic is Bencoded
- Files Broken into Pieces
- Pieces are Hashed to Verify Integrity

The BitTorrent Protocol (cont'd)

- HTTP_GET
 - Standard HTTP Request
 - Send “Info Hash” and Peer ID as Escaped Text
- Bencoding – Binary Encoding
 - Integers “i-123e”
 - Strings “12:Hello World!”
 - Lists “l<Bencoded Objects>e”
 - Dictionaries “d<Key><Value><K><V>...e”
 - Bencoder.java Available

The BitTorrent Protocol (cont'd)

- Simple Client (No Upload)
 1. Open .torrent Metainfo File
 2. Contact Tracker with Info Hash and Peer ID
 3. Receive List of Peers and Announce Interval
 4. Contact Peer(s)
 5. Request Pieces
 6. Verify Pieces (SHA-1 Hash)
 7. Save Data to Local Disk

Project 1 Basics

- General Outline
 1. Metainfo File is Parsed (TorrentFileHandler.java)
 2. Generate Random Peer ID
 3. Contact Tracker with Info Hash and Peer ID
 4. Select Peer at 172.23.26.98:63334
 5. Communicate With Peer
 6. Download and Verify All Pieces
 7. Save to File

Project 1 Basics (cont'd)

- Parsing .torrent Metainfo File
 - TorrentFileHandler.openTorrentFile(String)
 - TorrentFile object
 - String tracker_url
 - Vector<byte[]> piece_hash_values_as_binary
 - String info_hash_as_url
 - int file_length
 - int piece_length
- Generate Random Peer ID
 - Java.Random class
 - Generate Random int and Convert to Alphanumerics

Project 1 Basics (cont'd)

- Contact Tracker with Info Hash and Peer ID
 - HTTP GET Request
 - HTTP Escaped Strings, Key/Value Pairs
 - GET /announce?info_hash=<hash>&peer_id=<id>
- Tracker Response as Bencoded text/plain

Project 1 Basics (cont'd)

- Select Peer at 172.23.26.98:63334
 - Tracker Returns Bencoded List of Dictionaries
 - Peer ID, IP, Port
- Vector `Bencoder.unbencodeList(byte[])`
- HashMap `Bencoder.unbencodeDictionary(byte[])`
- `BencoderTester.java`

Project 1 Basics (cont'd)

- Communicate With Peer

- Handshake

- <pstrlen><pstr><reserved><info_hash><peer_id>

- General Messages

- <length prefix [4B dec]><message ID[1B char]><payload>

- Request Piece

- <0013><6><index><begin><length>

- Index of Piece
 - Offset of Block
 - Length of Block (2^{14} default)

Project 1 Basics (cont'd)

- Download and Verify All Pieces
 - SHA-1 Hash (160 bits)

```
MessageDigest.getInstance("SHA-1");  
byte[] hash = sha.digest(bytes);
```

- Save to File
 - `FileOutputStream.write(byte[])`

Project 1 Basics (cont'd)

- Info Hash
 - SHA-1 Hash of Bencoded Info (TorrentFileHandler)
- Piece Hash
 - Must Generate SHA-1 Hash of Each Piece
- Peer ID
 - Must Generate Randomly
- Bencoding
 - Encode/Decode with Bencoder.java

Resources

- Classmates
- Internet
 - <http://remus.rutgers.edu/cs352/S06/project1.html>
- Email
 - romoore@cs.rutgers.edu
- Office Hours
 - Wed 5-6pm Hill 488
- Due March 5, 5:00pm – CS Handin