# BitTorrent

*Slides adapted from*

*Prof. Dah Ming Chiu, Chinese University of Hong Kong, Hong Kong*
*Dr. Iqbal Mohomed, University of Toronto, Canada*

# Content Distribution

- IP multicast
- CDN (Content Distribution Network)
- Application layer multicast
  - Overlay structures
    - Tree-based (push)
    - Data-driven (pull)
  - P2P swarming
    - BitTorrent, CoolStreaming

# BitTorrent

- Released in the summer of 2001
- Basic ideas from game theory to largely eliminate the free-rider problem
  - All precedent systems could not deal with this problem well
- No strong guarantees unlike DHTs
- Working extremely well in practice unlike DHTs ☺

# Basic Idea – Swarming Protocol

- A file is chopped into small pieces, called chunks
- Pieces are disseminated over the network
- As soon as a peer acquire a piece, it can trade it for missing pieces with other peers
- A peer hopes to be able to assemble the entire file at the end

# Basic Components

- Web server
- The .torrent file
- Tracker
- Peers

# Web Server

- Content discovery (i.e., file search) is handled outside of BitTorrent, using a Web server
  - To provide the “meta-info” file by HTTP
  - For example, http://bt.btchina.net
- The information about each movie or content is stored in a metafile such as “supergirl.torrent”

# The .torrent File

- Static file storing necessary meta information
  - Name
  - Size
  - Checksum
    - The content is divided into many “chunks” (e.g., 1/4 megabyte each)
    - Each chunk is hashed to a checksum value
    - When a peer later gets the chunks (from other peers), it can check the authenticity by comparing the checksum
  - IP address and port of the **Tracker**

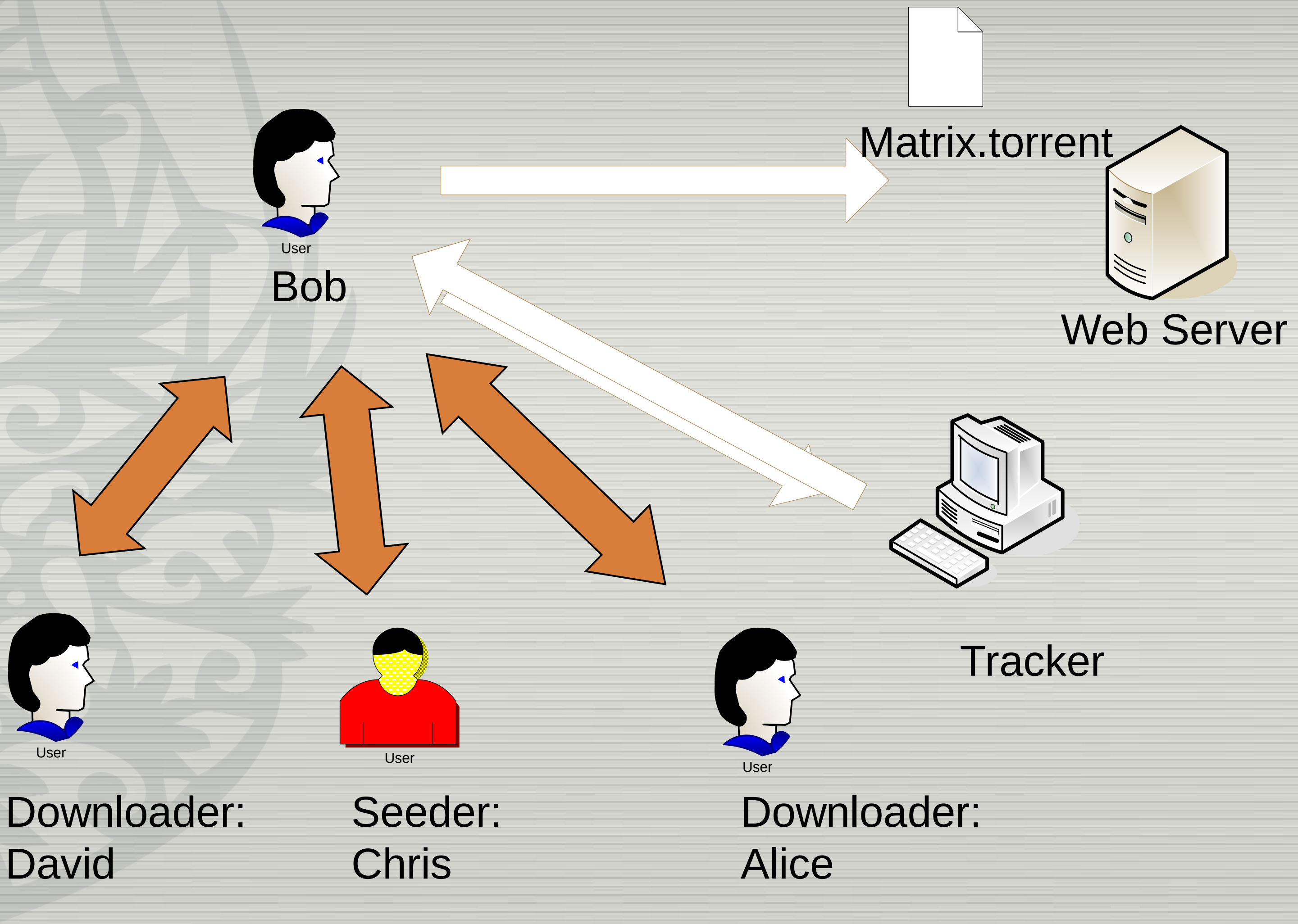
# Tracker

- Keeping track of peers
  - To allow peers to find one another
  - To return a random list of active peers

# Peers

- Two types of peers:
  - *Downloader* (*leecher*) : A peer who has only ***a part (or none***) of the file.
  - *Seeder*: A peer who has the ***complete*** file, and chooses to stay in the system to allow other peers to download

# BitTorrent in Action



Matrix.torrent

Bob

Web Server

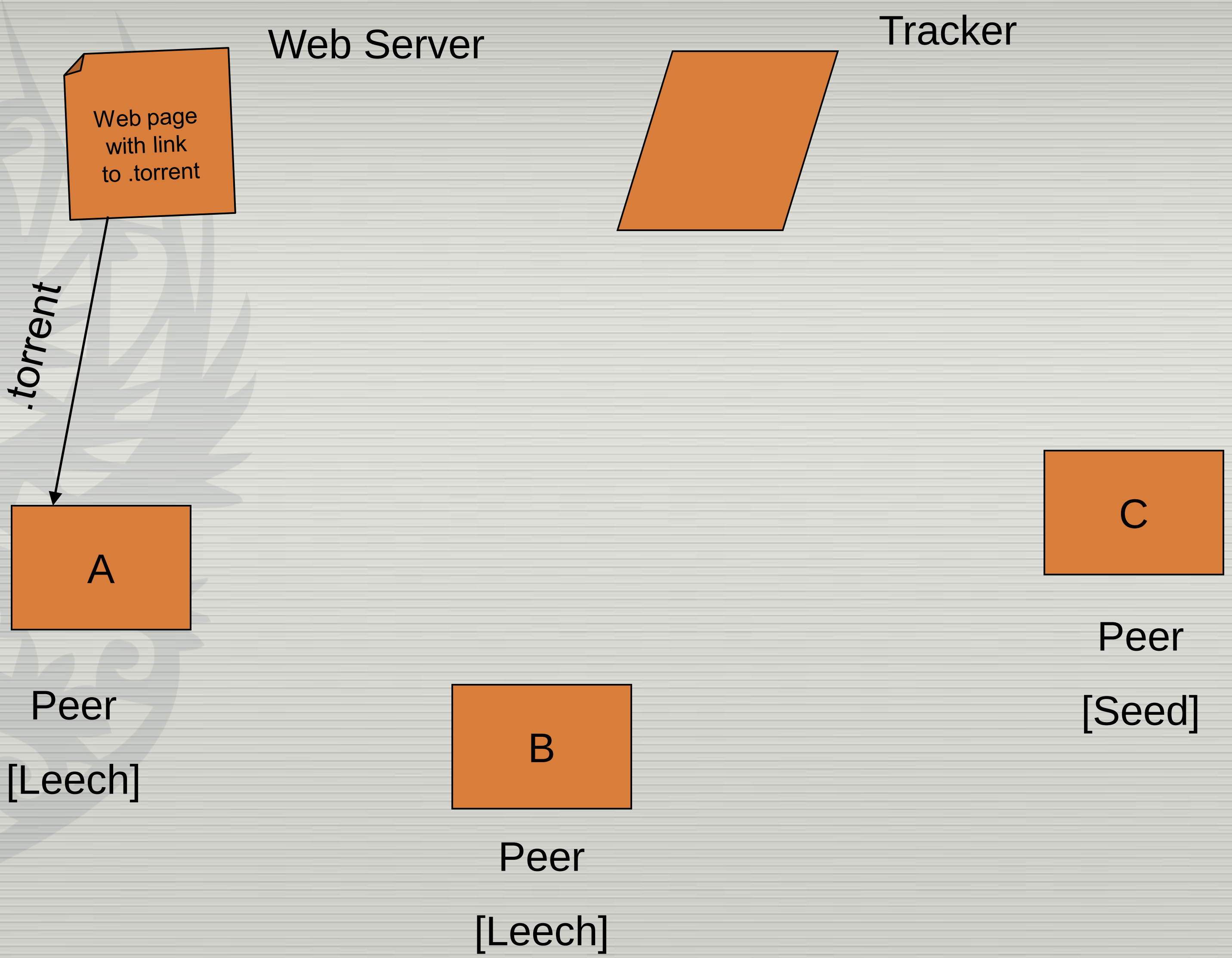
Tracker

Downloader:
David

Seeder:
Chris

Downloader:
Alice

# Overview – System Components



Web Server

Tracker

Peer

[Leech]

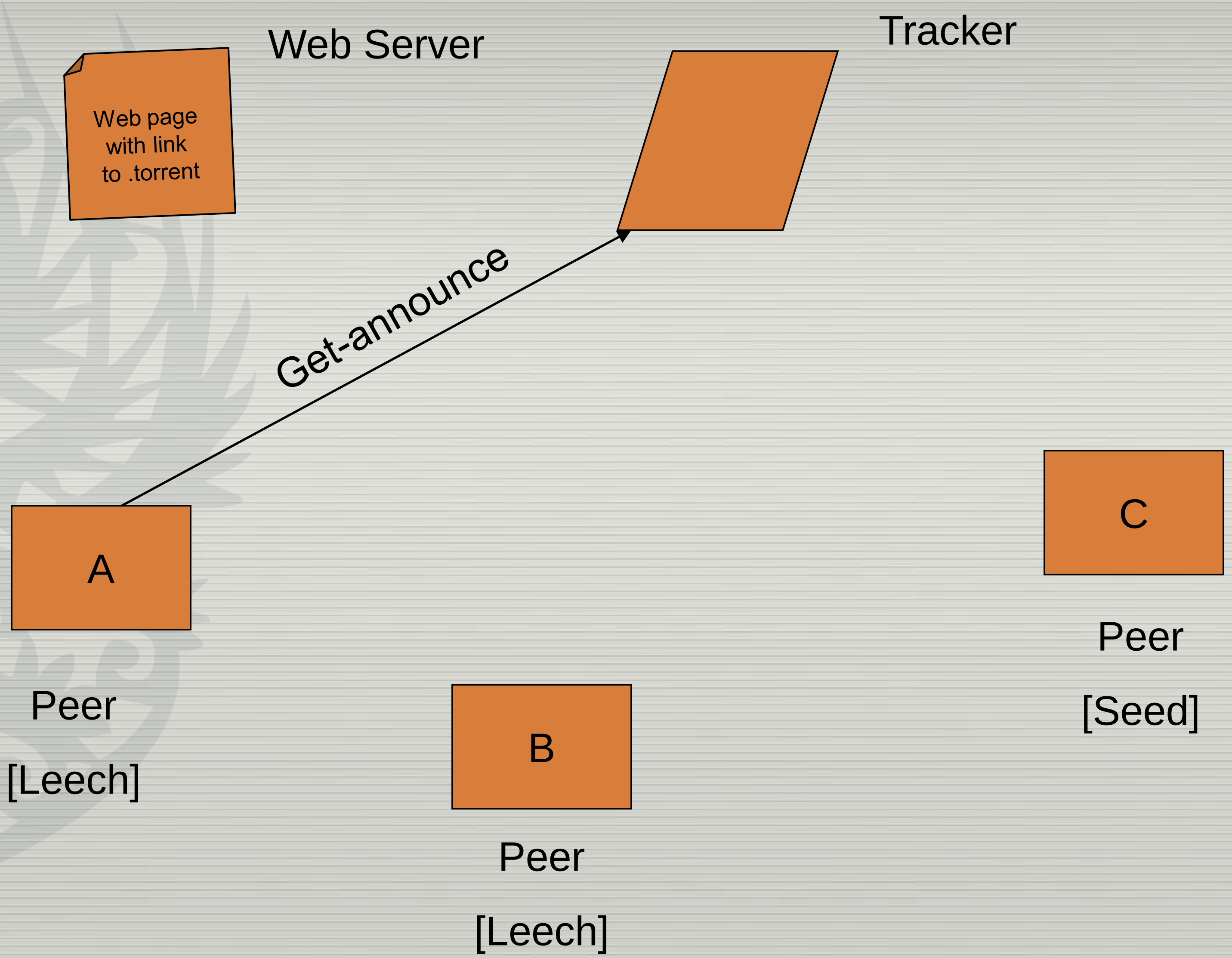
Peer

[Leech]

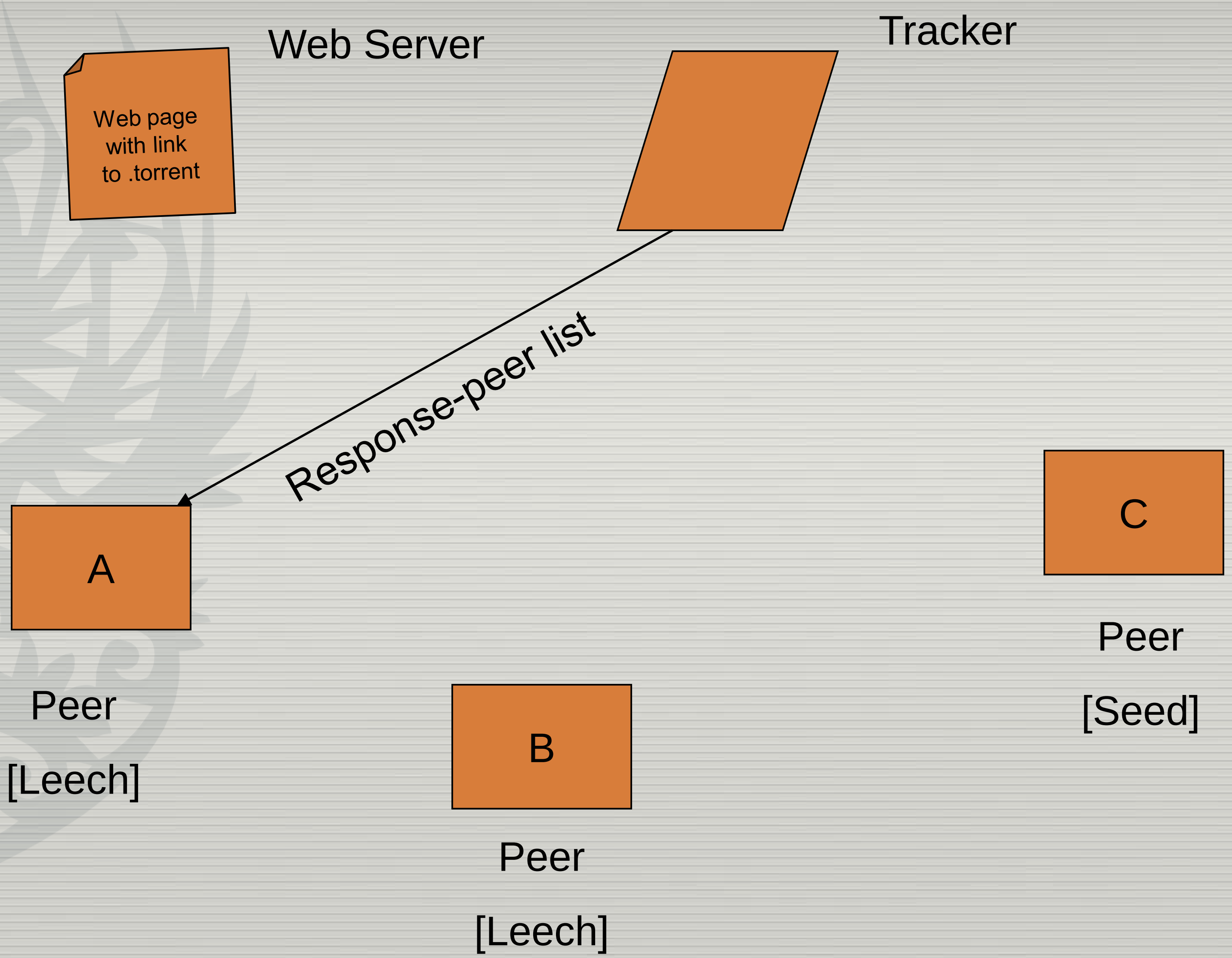
Peer

[Seed]

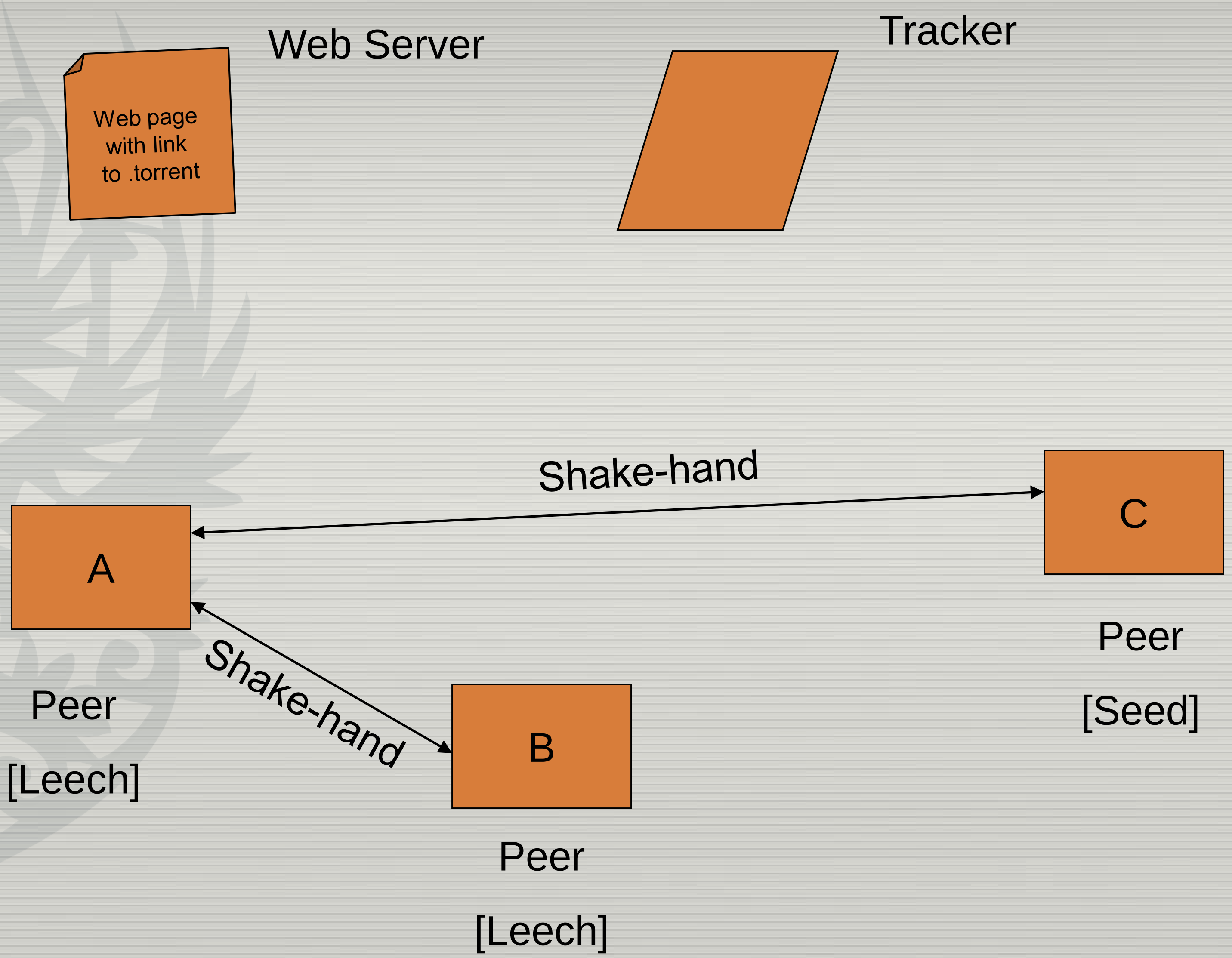
# Overview – System Components



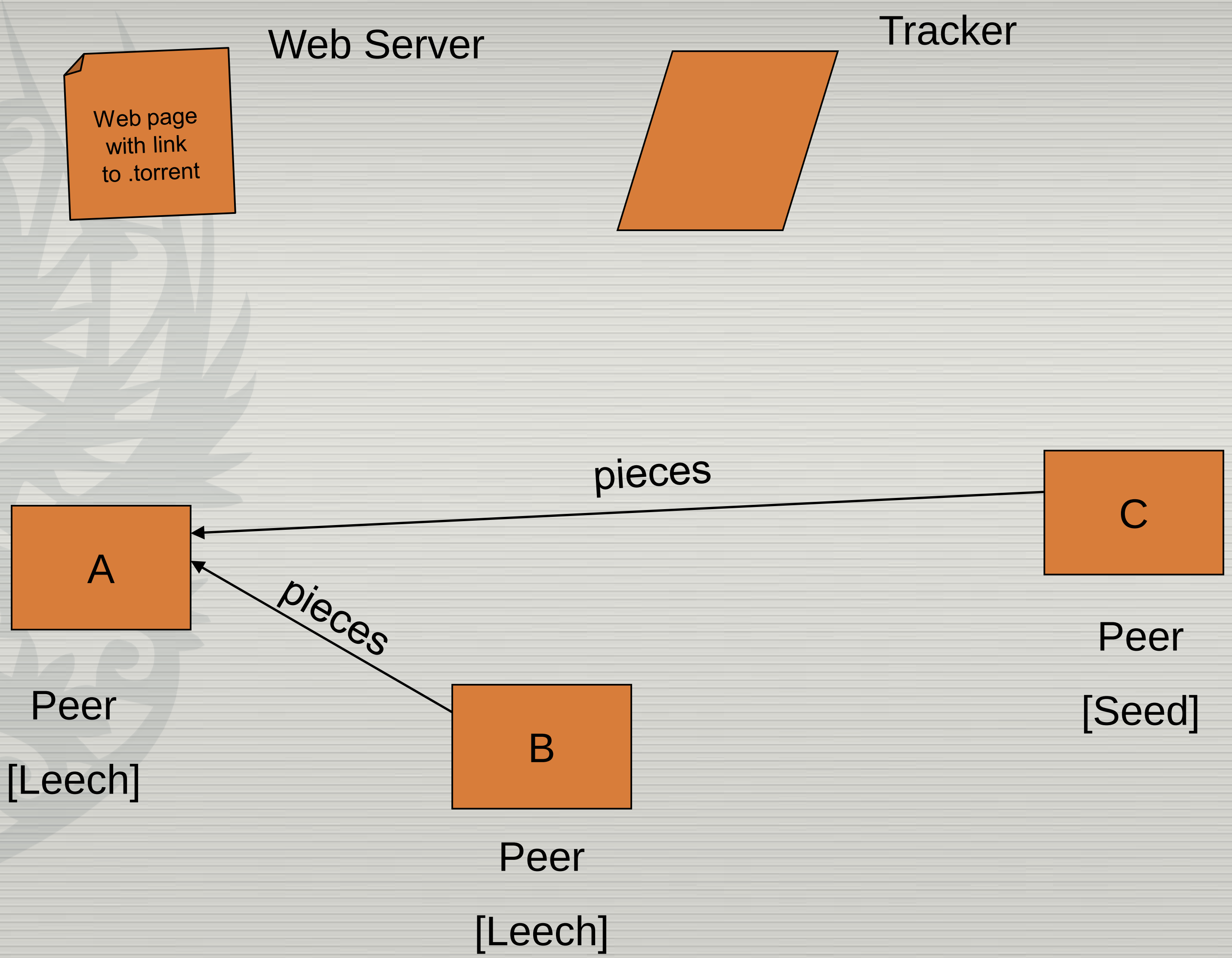
# Overview – System Components



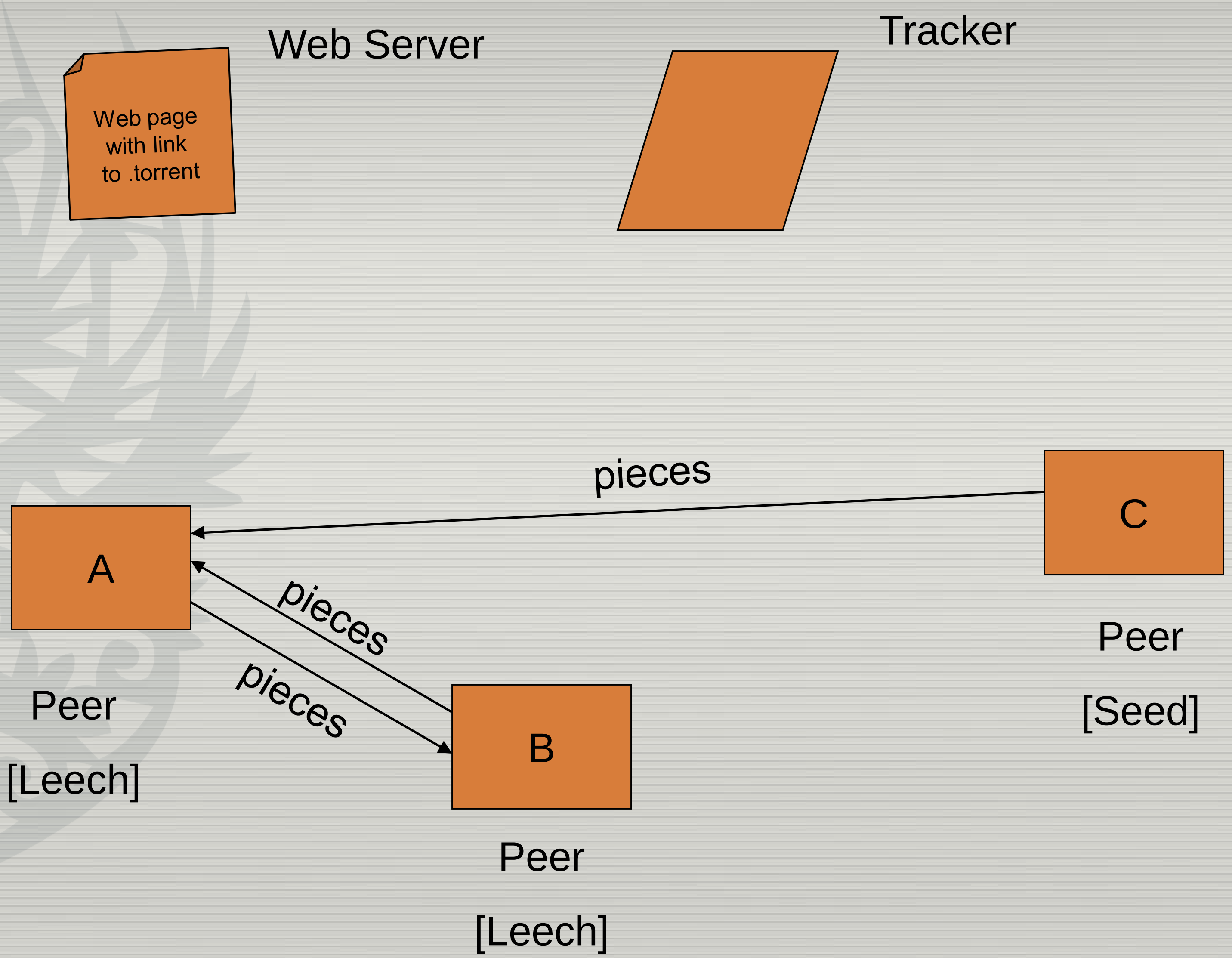
# Overview – System Components



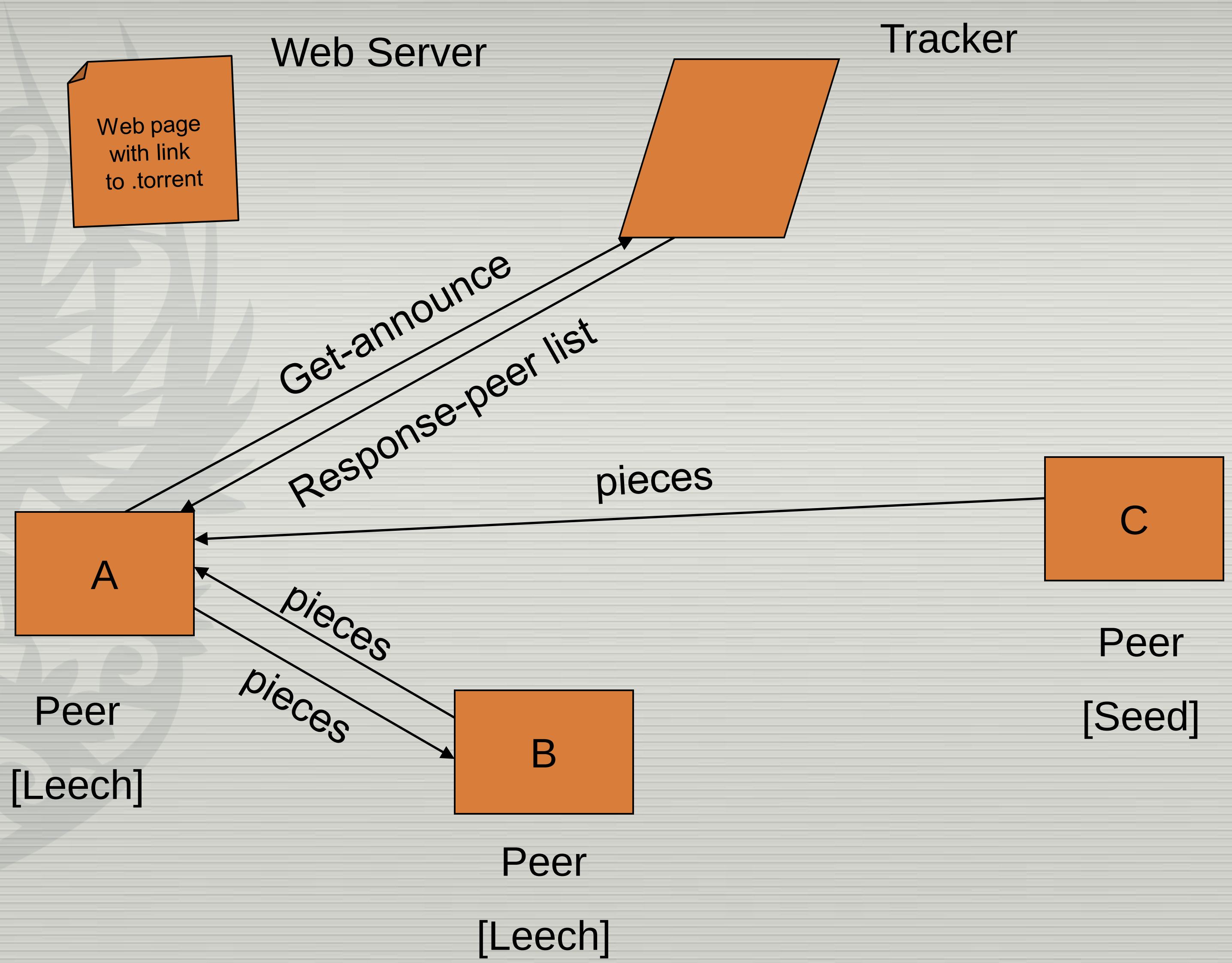
# Overview – System Components



# Overview – System Components



# Overview – System Components



Web Server

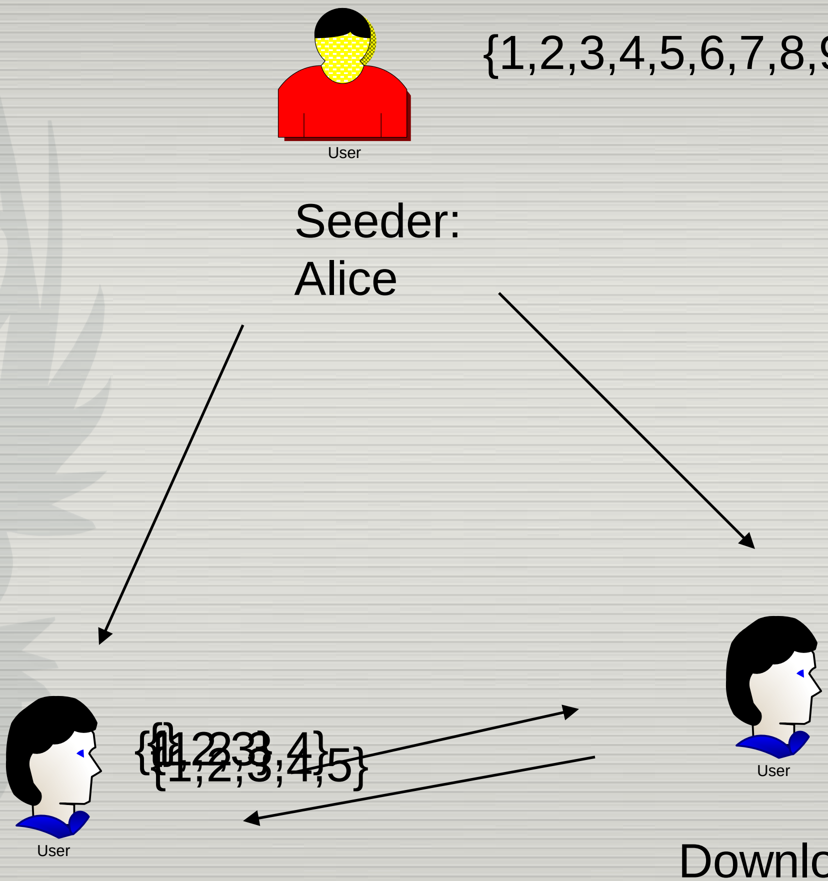
Tracker

# Chunks

- A file is split into chunks of fixed size, typically 256Kb
- Each peer maintains a bit map that indicates which chunks it has

  ↳ 어떤 chunk를 가지고 있는지 없는지에 대한 정보
- Each peer reports to all of its neighboring peers (obtained from tracker) what chunks it has
  - This is the information used to build the implicit delivery trees

# Swarming Example



{1,2,3,4,5,6,7,8,9,10}

Seeder:
Alice

{}
{1,2,3}
{1,2,3,4}
{1,2,3,4,5}

{1,2,3}
{1,2,3,4}
{1,2,3,4,5}

Downloader
Bob

{}
{1,2,3}
{1,2,3,5}

{1.2.3}
{1.2.3.5}

Downloader
Joe

# Rarest First

→ 제일 귀한 chunk를 먼저 요청하는 정책 사용
⇒ rarest first candidate

- Rarer pieces are given priority in downloading with the rarest being  the first candidate
- The most common pieces are postponed towards the end
- This policy ensures that a variety of pieces are downloaded from the seeder, resulting in quicker chunk propagation

# Peer Selection

↳ 주고 받을 친구를 정하는 전략

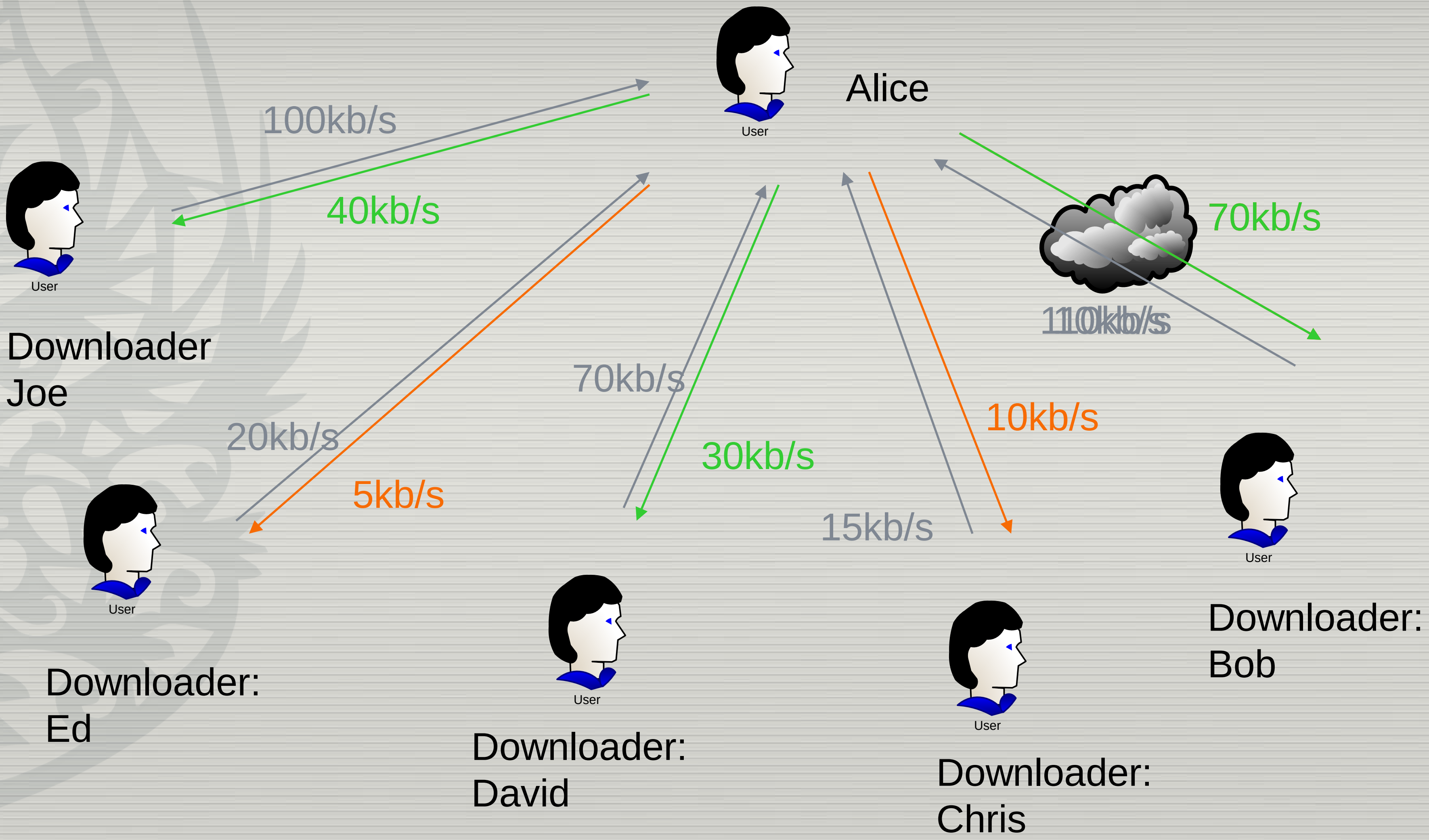
↱ 받은 만큼 돌려준다

Basic idea of **tit-for-tat** strategy in BitTorrent:

- Maintain 4-5 "friends" with which to exchange chunks
- If a friend is not exchanging enough chunks, get rid of him/her
  - data를 안보내는 느낌
  - Known as "choking" in BT
- Periodically, randomly select a new friend
  - Known as "optimistic unchoking" in BT
  - ↳ data 쭉쭉 잘보내주는 느낌
- If you have no friends, randomly select several new friends
  - Known as "anti-snubbing" in BT

# Example of Optimistic Unchoking

incentive machinism



User

Alice

100kb/s

40kb/s

User

Downloader Joe

70kb/s

110kb/s

20kb/s

70kb/s

30kb/s

10kb/s

5kb/s

15kb/s

User

Downloader: Bob

User

Downloader: Ed

User

Downloader: David

User

Downloader: Chris

# Questions?