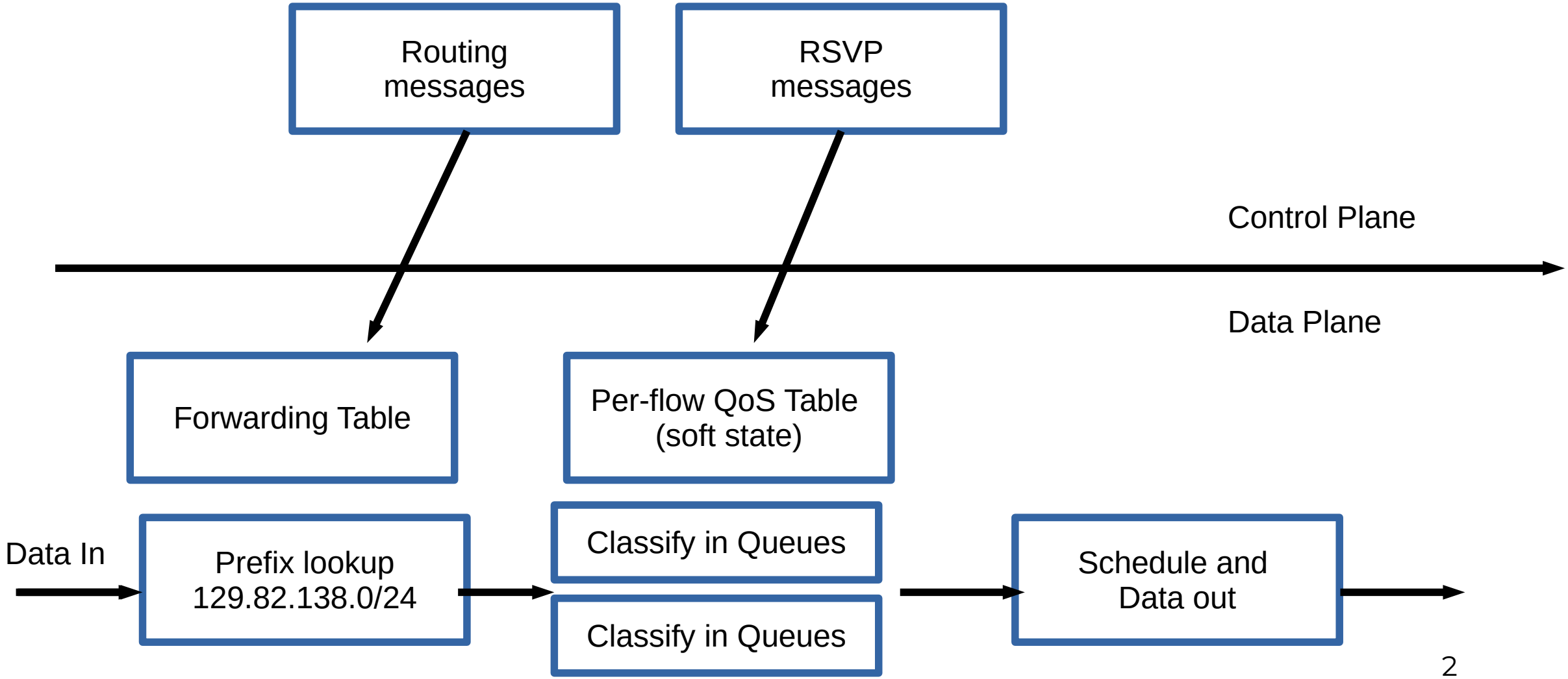
CSC7970 – NEXT-GENERATION NETWORKING

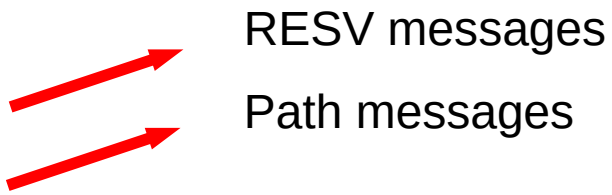
RESERVATION PROTOCOLS AND P2P

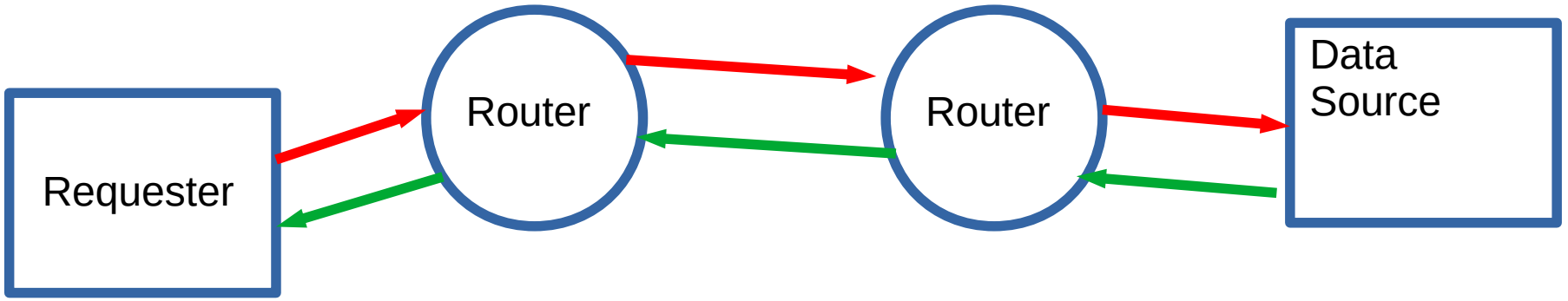
Instructor: Susmit Shannigrahi
sshannigrahi@tntech.edu

The bigger picture - IntServ



RSVP

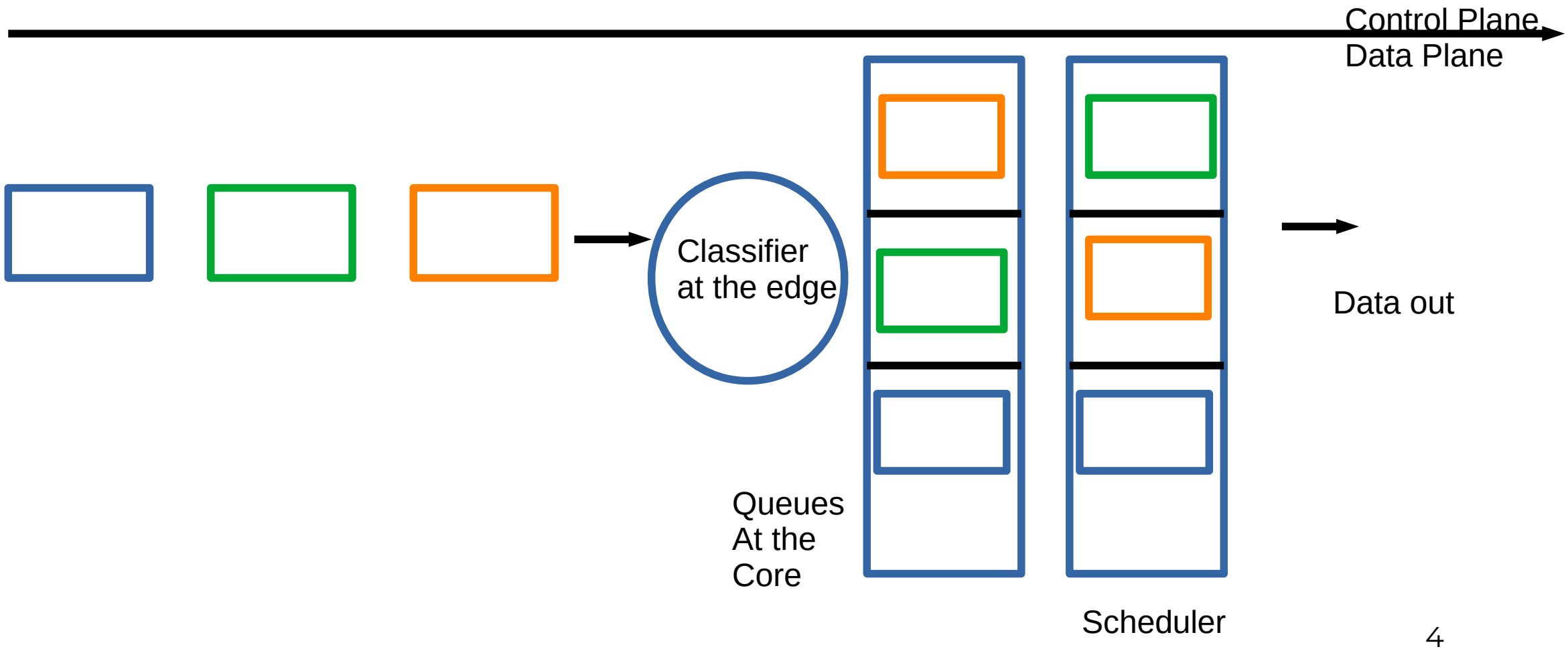




Need 5Gbps and 20 ms latency

Everyone needs to agree
Per-flow – lots of state
Should we create admission control?

The bigger picture - DiffServ



QoS

- Different applications can benefit from different handling
  - IP is best effort
  - If you look at the header, it used to have a type-of-service field
  - Not used in production, and later changed to DSCP

IPv4 Header Format — **wikipedia**

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

QoS

- What does DSCP stand for?
    - Differentiated Services Code Point
    - Used for DiffServ, upto the application to decide how to use it
  - You can say things like
    - Expedited Forwarding
    - Class selection
    - and so on..

QoS

- +-----+-----+-----+-----+-----+-----+-----+----------+
- | bit0 | bit1 | bit2 | bit3 | bit4 | bit5 | bit6 | bit7 |
- |-----+-----+-----+-----+-----+-----+-----+----------|
- | DSCP | ECN |
- +----------------+----------------------+------------+
- PHB
  - 000 000 → Default
  - Expedited Forwarding(EF) → 101 110
- RFC 2474, 2597, 2598

Drop Priority

- All sent packets get same average delay
  - Why?
- Drop packets early to avoid TCP congestion control
- You will spend a lot of time to recover than dropping a few packets

P2P

The Current State of the Internet

- The Internet is becoming more and more centralized
  - Dominated by tech companies that want to control it
  - (Almost) any communication today has to go through a (cloud) server
  - Where it is mined, sold, resold, and used for various things

- A number of reasons to avoid centralization
  - Privacy
  - Dependability
  - Restriction of free speech

Decentralization through P2P

High-level idea

- Why having to go to a server to communication with other users?
- Let’s communicate with them directly!

- BitTorrent
  - still dominates Internet traffic
  - Apple/MS/RedHat use them for software distribution

Where BitTorrent is not so great?

- Implementing a data-centric design on top of TCP/IP
  - No way to reason about trust
  - Explicit peer discovery (which is the IP address of each peer?)
  - Have to bind data to a specific peer (aka IP address)
  - Does not do well with “flash-crowd” scenarios
  - Peers have no clue how “far” or close another peer is
  - Problems with that?

Flash crowd scenarios

BitTorrent flashcrowds only occur in very small fractions (0.3-2%) of the swarms but that they can affect over ten million users.

B. Zhang, A. Iosup, J. Pouwelse and D. Epema, "Identifying, analyzing, and modeling flashcrowds in BitTorrent," 2011 IEEE International Conference on Peer-to-Peer Computing, Kyoto, 2011, pp. 240-249.

How can we find torrent-files?

- Usually through a website
  - Users go online, search for what they want, and download the torrent-file
- Do we need to download the torrent-file from a website?
  - Not necessarily…
- Magnet links as an alternative
  - Download a torrent-file directly from another peer

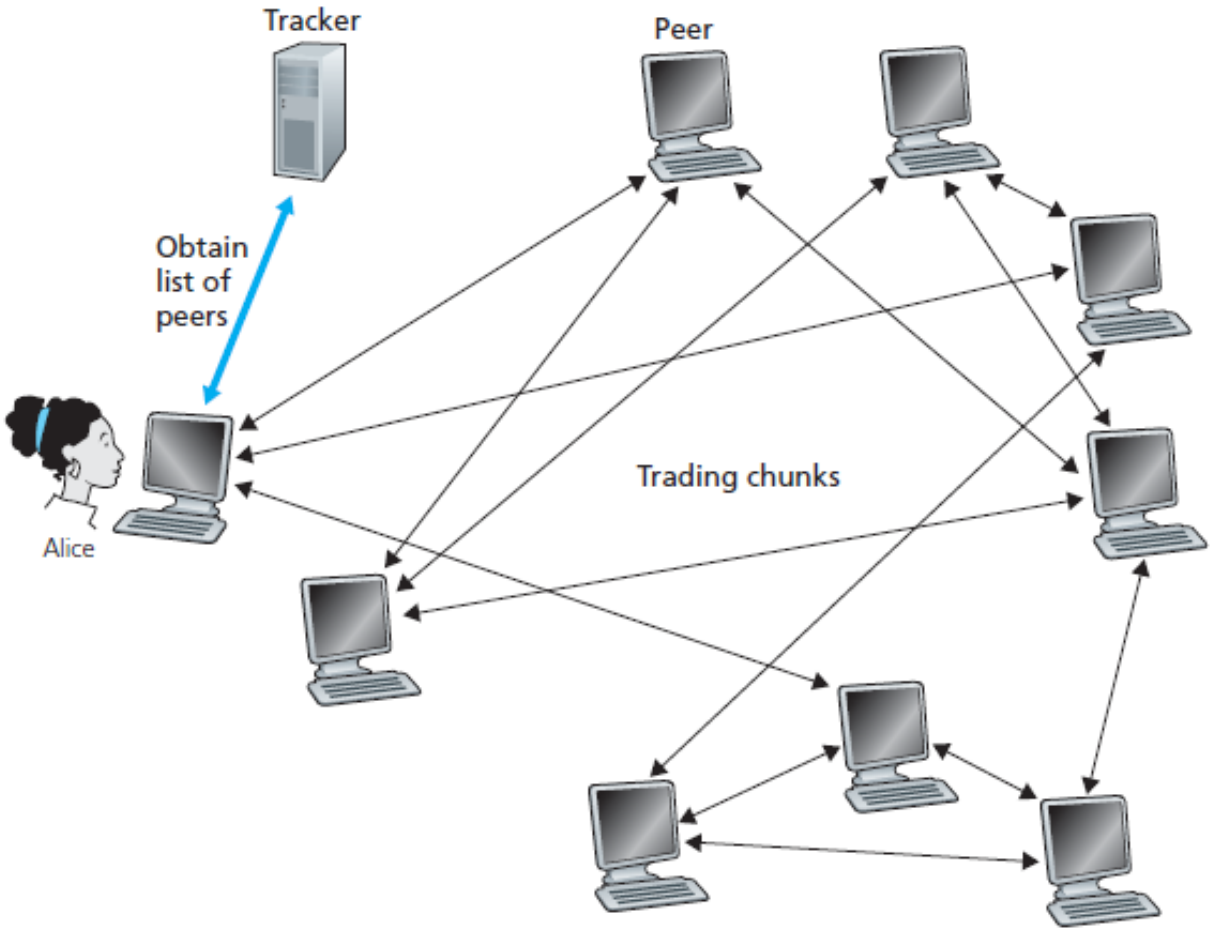
Peer Discovery

- Discovery of available peers to download a torrent from?
  - Operate on top of an IP network
  - We need to know the IP address of peers that have the data we want

- 2 ways to do it
  - Through a tracker – centralized approach
  - Through a Distributed Hash Table (DHT) – distributed approach

Example

What happens if the tracker fails?



Credits: https://electronicspost.com/how-bittorrent-works

DHT-based discovery

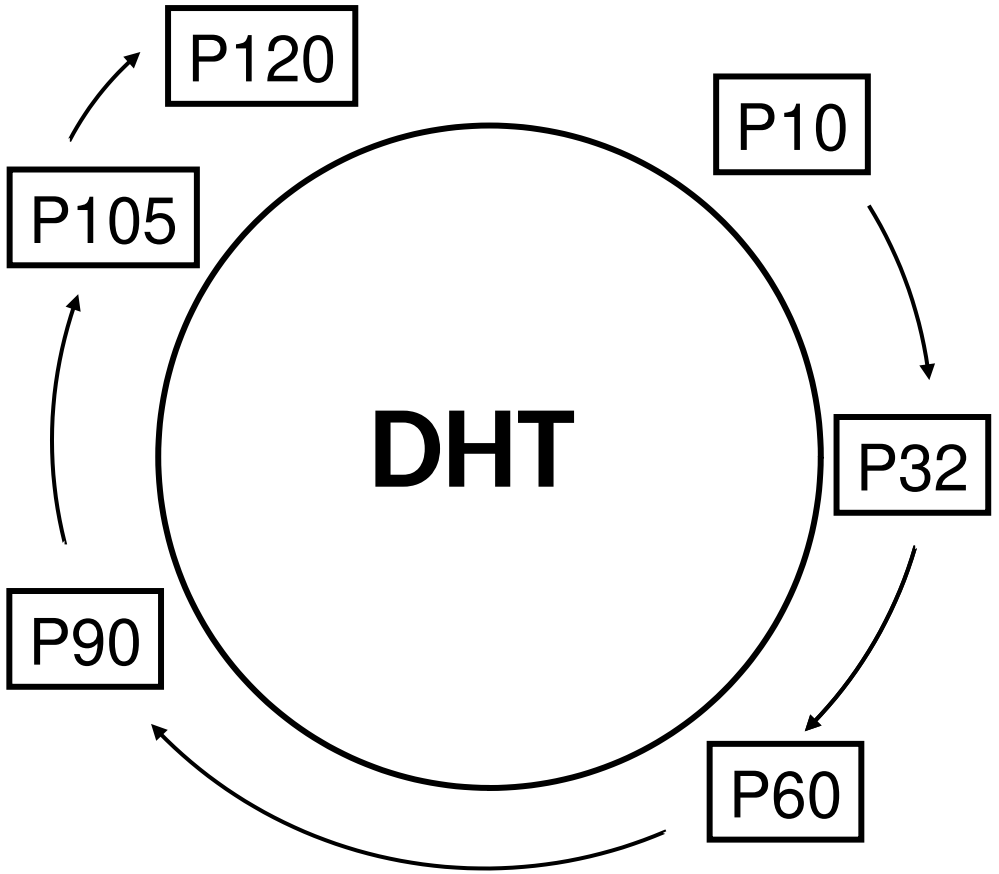
- How can we make this process fully distributed?
  - Instead of contacting a tracker, we can contact directly peers that may have the torrent we are interested in

- How to find such a peer?
  - Hash torrent and peer ids (e.g., IP addresses)
  - Map each hash value to a point on the DHT
  - Ask a number of peers on the DHT until you find someone that knows peers that download (have) the torrent

DHT Example

Hash of torrent (e.g., found on a torrent website)

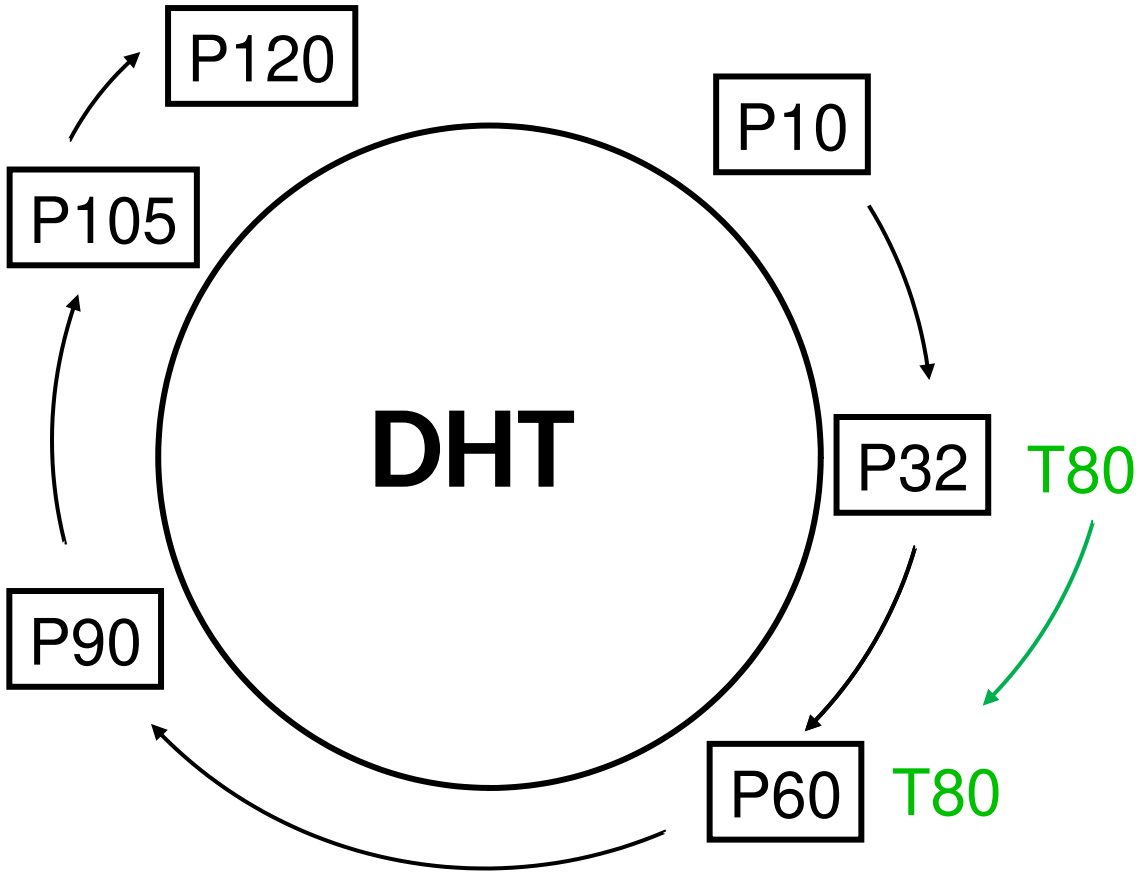
Example:
hash(linux_16.04) = T80



Example: Bootstrap DHT through P32

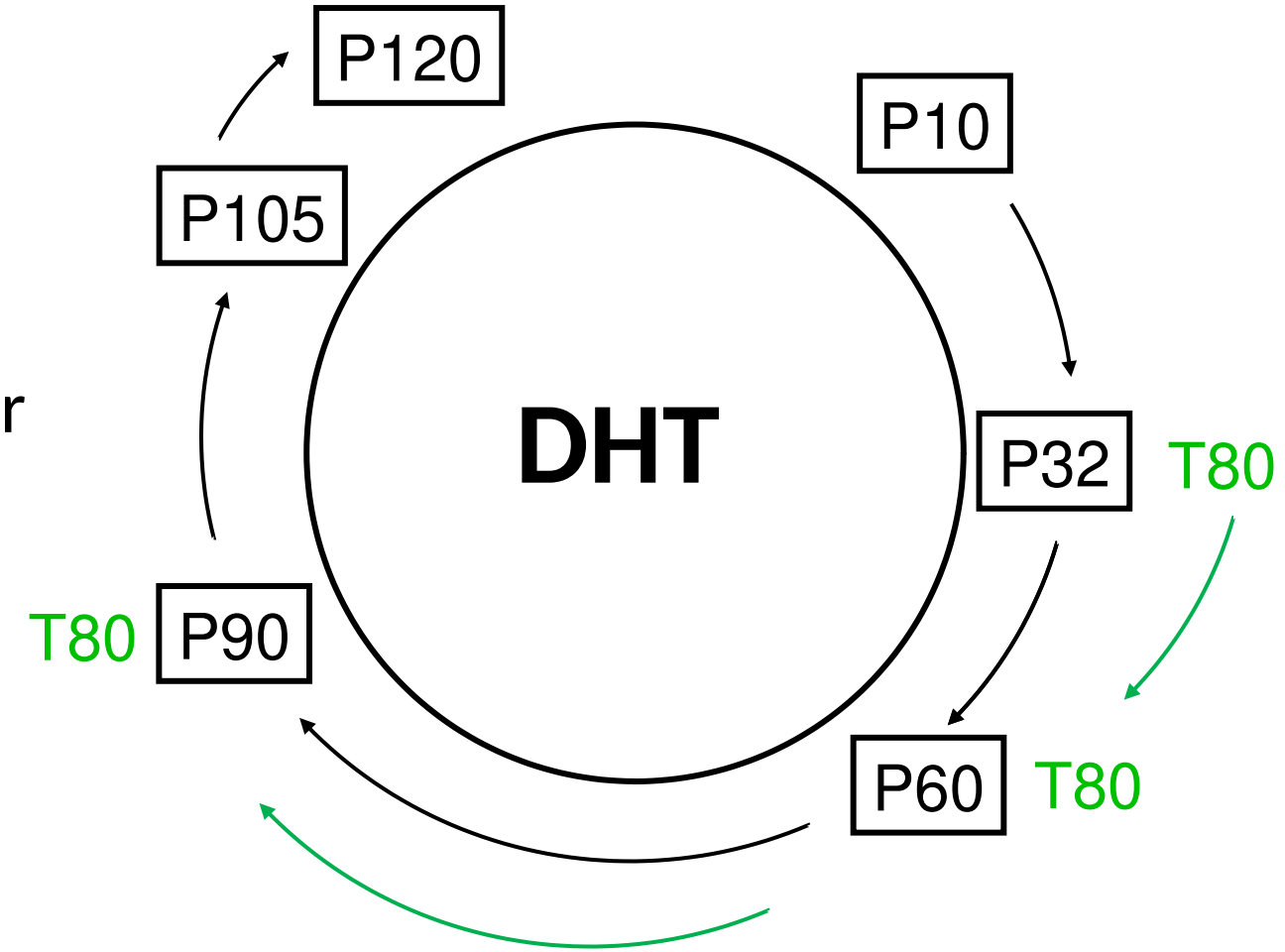
DHT Example

Query successor peer(s) on the DHT until a peer that is responsible for the torrent is found



DHT Example

Peer responsible for T80 found!



Discovering what data a peer has

- Peers maintain a bitmap for each torrent
  - Each bit indicates whether a peer has a specific torrent piece (each piece consists of multiple network-layer packets)
- Peers exchange bitmaps with each other
  - Let each other know what data they have

Conclusion

- P2P aims to offer the means to decentralization on the current Internet
  - Peers communicate with each other directly
  - Peers focus on the data to share taking data integrity into account
- BitTorrent is a well-designed system for peer-to-peer communication
  - Hard to trick BitTorrent!
  - Does its best to offer data-centricity on top of a point-to-point network
  - Dominated traffic, slowed down, and picking up again