# 6.033 Spring 2019
## Lecture #13

- **New Technologies on the Internet**
  - **File-sharing (BitTorrent, DHTs)**
  - **VoIP (Skype)**
  - **Video Streaming**

# Internet of Problems

How do we **route** (and address) scalably, while dealing with issues of policy and economy? ➡ **BGP**

How do we **transport** data scalably, while dealing with varying application demands? ➡ **TCP, in-network resource management**
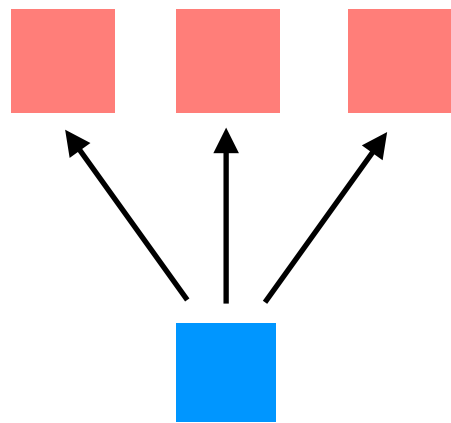
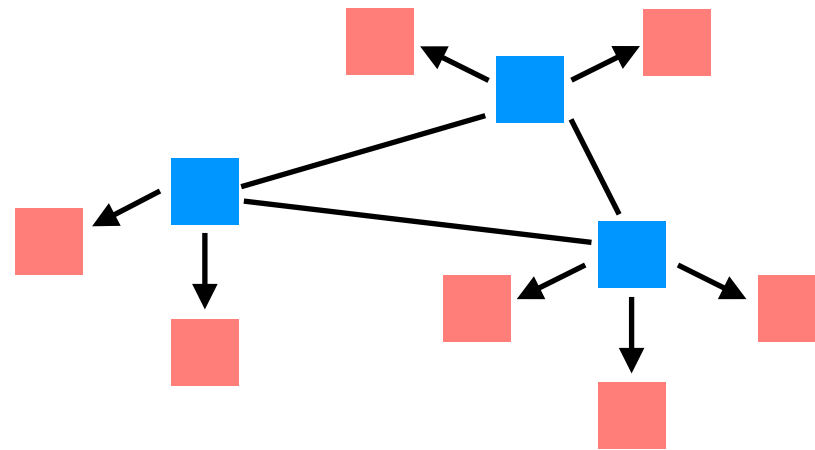How do we **adapt** new applications and technologies to an inflexible architecture? ➡ **P2P Networks, CDNs** (and more)
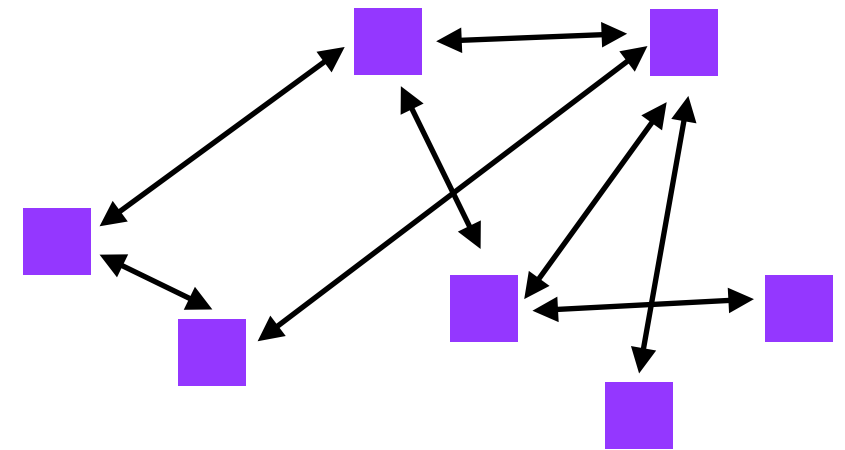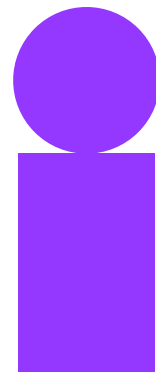
# File-sharing Techniques

**client-server**     **CDNs**     **P2P**



**scalability increases**
(in theory)

**problem:** how do we incentivize peers in a P2P network to upload?
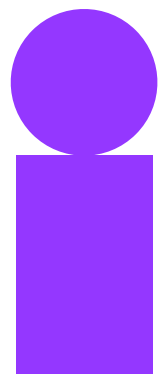
# Discovering the P2P Network
## (the "swarm")



**peer**

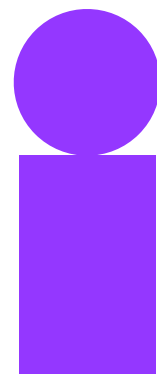# Discovering the P2P Network
## (the "swarm")

**peer** → **tracker**

# Discovering the P2P Network
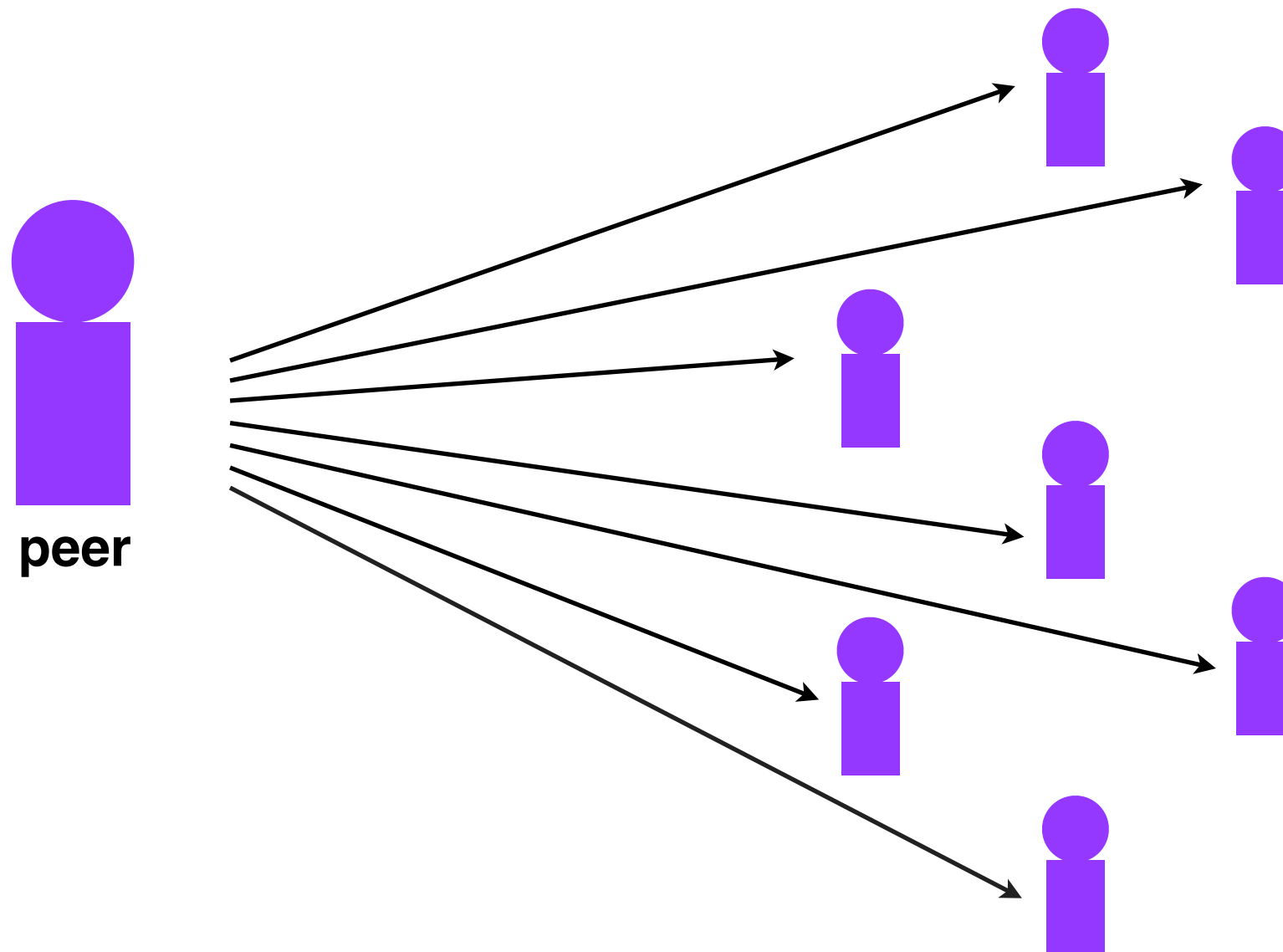## (the "swarm")



**peer**

130.136.254.21
130.136.254.22
171.66.3.182
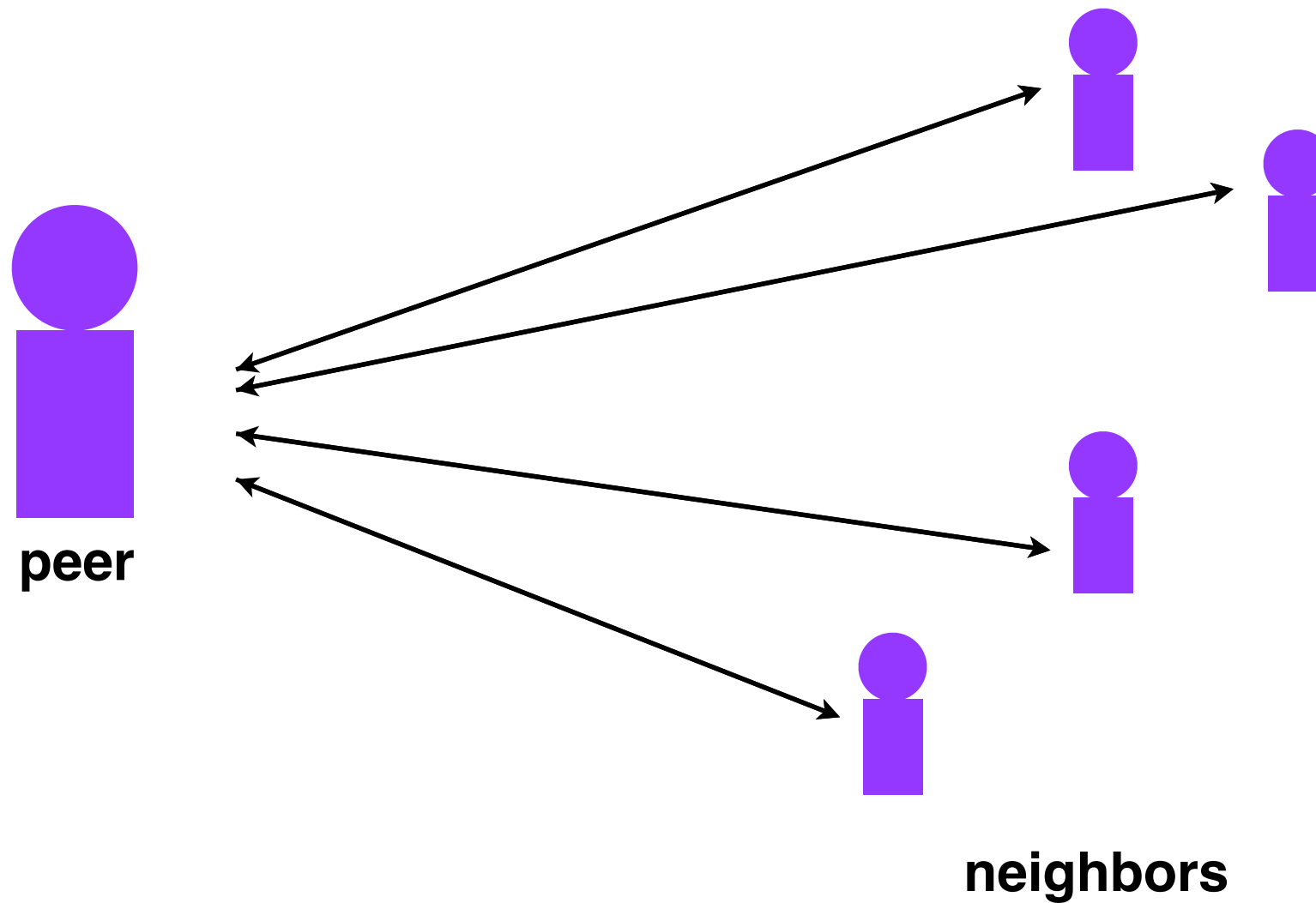128.31.1.11
128.83.122.180
128.232.103.202
155.98.35.4
...

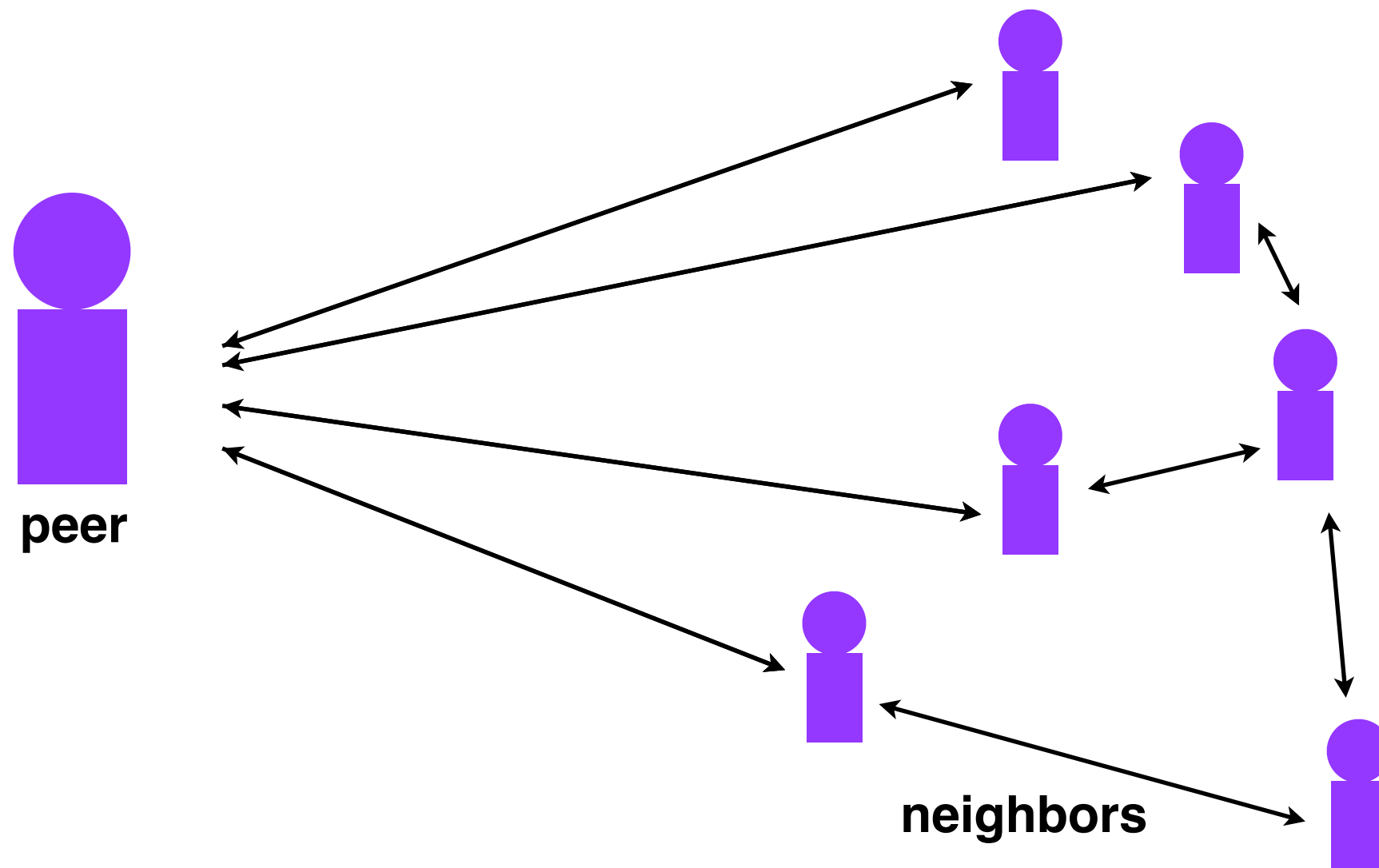**tracker**

# Discovering the P2P Network

(the "swarm")
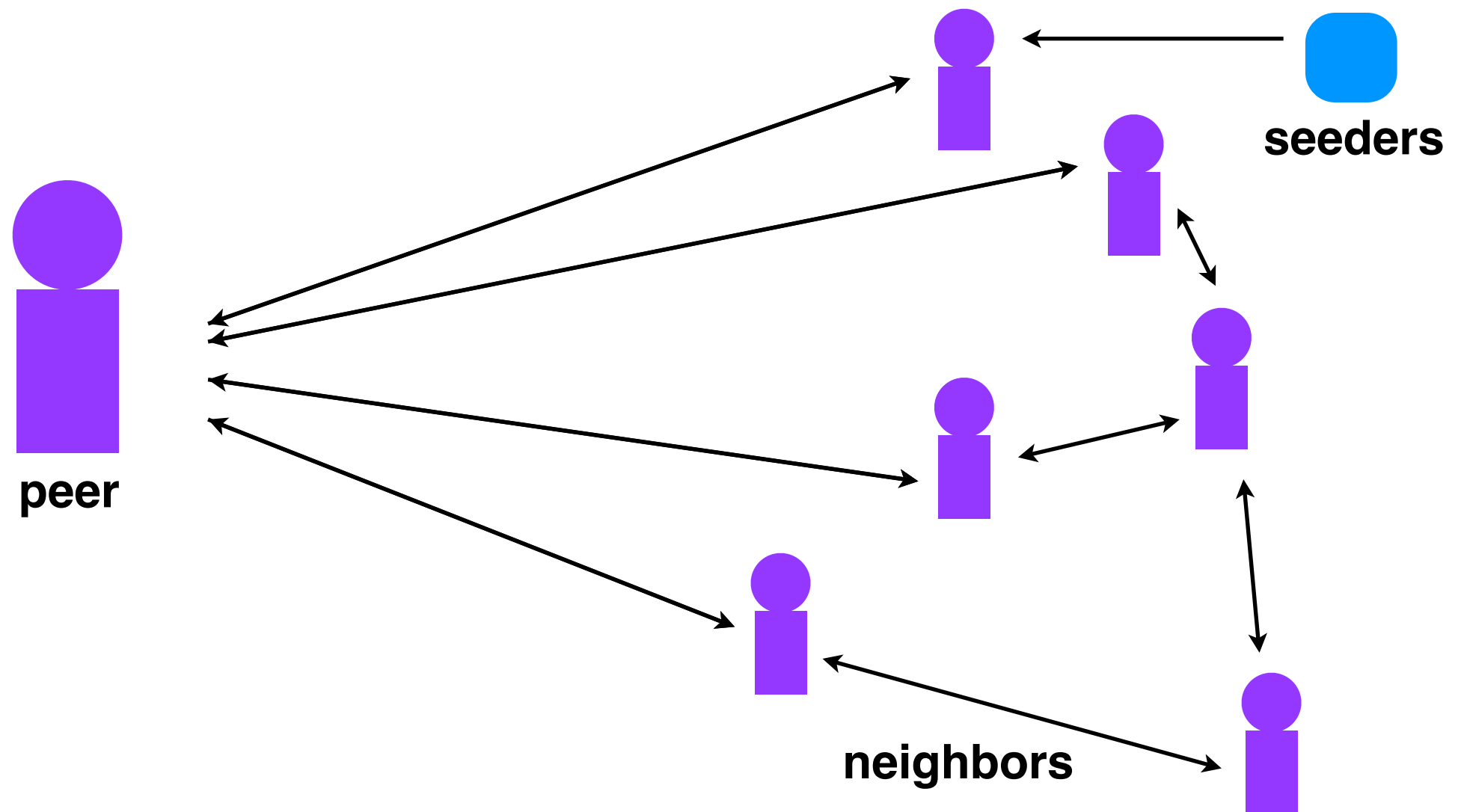


**peer**

# Discovering the P2P Network
## (the "swarm")



**peer**

**neighbors**

# Discovering the P2P Network
## (the "swarm")

**peer**

**neighbors**

# Discovering the P2P Network
## (the "swarm")



**seeders**

**peer**

**neighbors**

# Incentivizing Peers to Upload

**round *t***

13

10

4

12

7

9

15

**round *t+1***
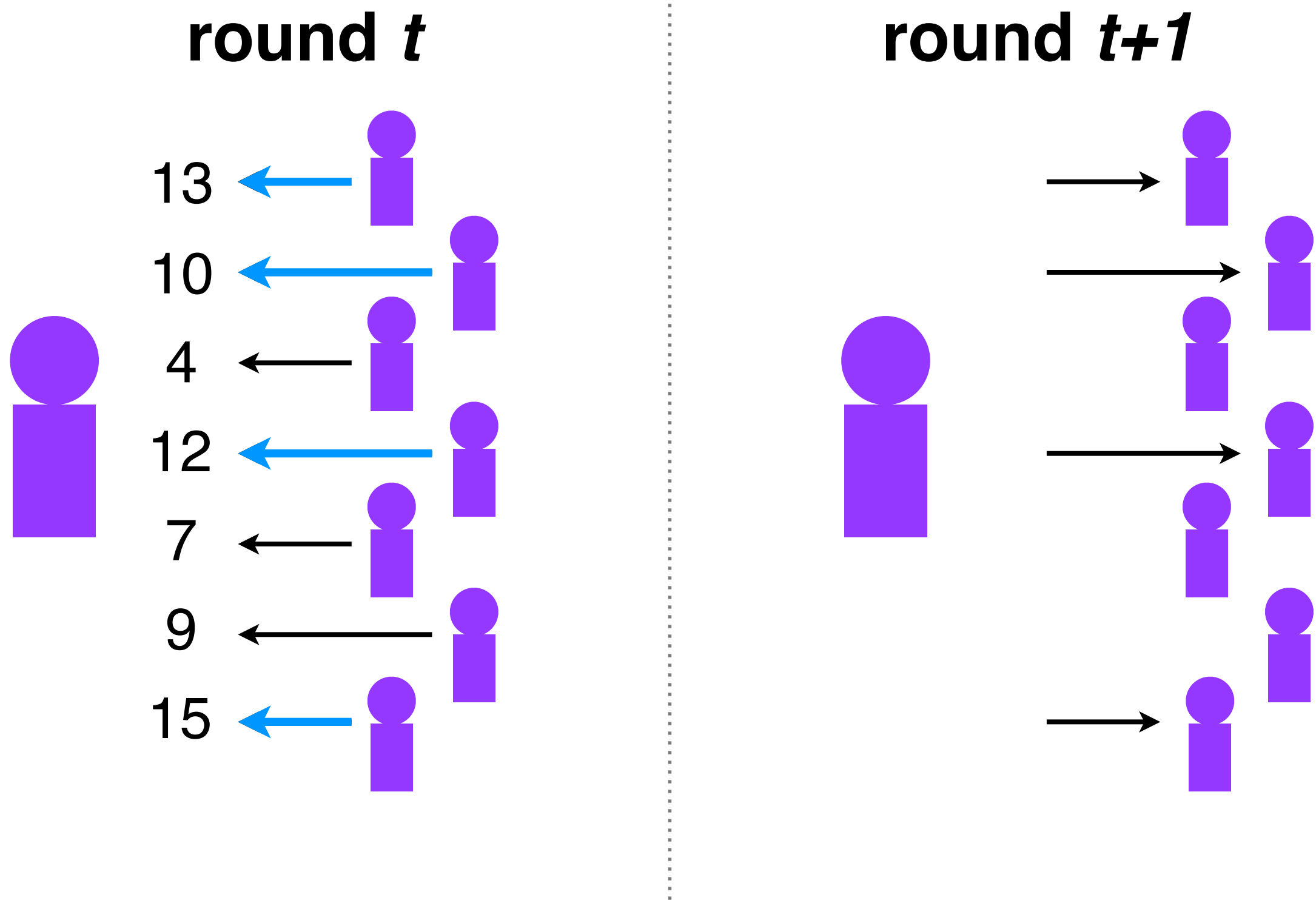
# problem: the tracker is a central point of failure
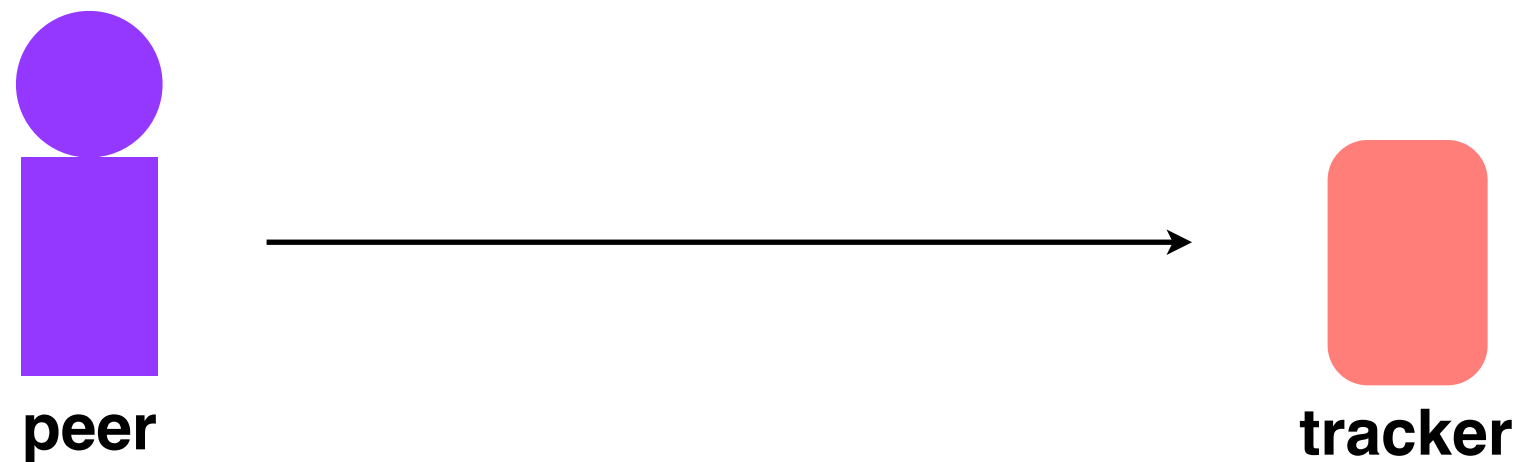


peer           tracker

in practice, most BitTorrent clients are **trackerless**. The list of peers is stored (and replicated) across multiple machines in a distributed data structure
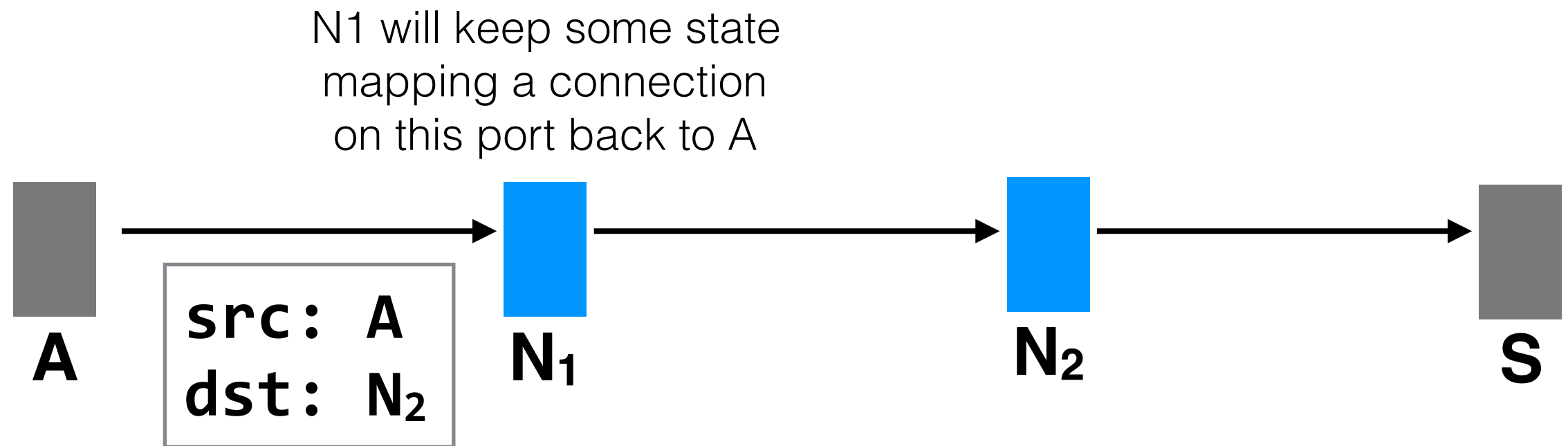
# VoIP
## (Voice over IP)

A

src: A
dst: ?

N₁        N₂        S

**problem:** S's IP is private
(can't route to it, and can't figure
out that it's "behind" N₂)

A

```
src: A
dst: N₂
```

N₁

N₂

S

**skype provides a directory, so assume we can get N₂'s IP**

N1 will keep some state
mapping a connection
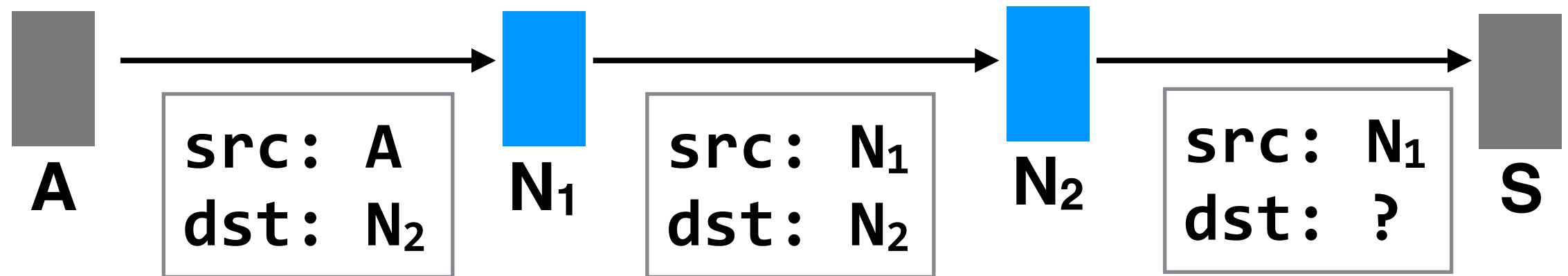on this port back to A

**A**

```
src: A
dst: N₂
```

**N₁**

**N₂**

**S**

**skype provides a directory,
so assume we can get N₂'s IP**

N1 will keep some state
mapping a connection
on this port back to A

**A**

$$\text{src: A}$$
$$\text{dst: } N_2$$

**$N_1$**

$$\text{src: } N_1$$
$$\text{dst: } N_2$$

**$N_2$**

**S**

**skype provides a directory,
so assume we can get $N_2$'s IP**

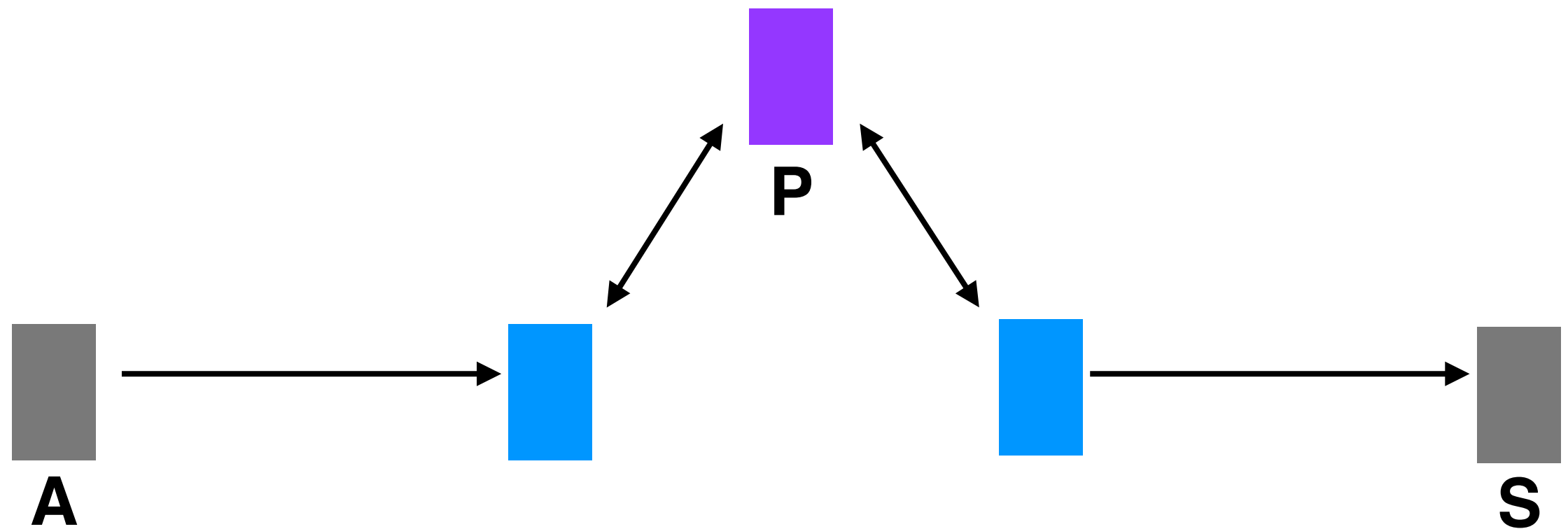N1 will keep some state
mapping a connection
on this port back to A

**A**

src: A
dst: N₂

**N₁**

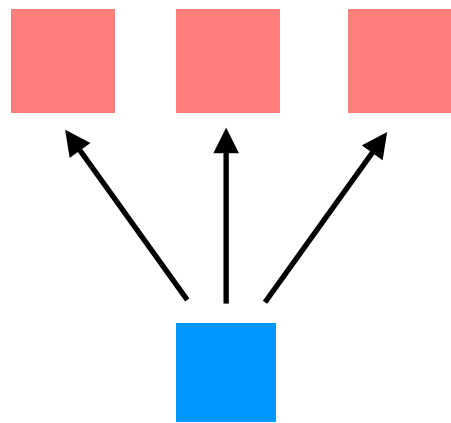src: N₁
dst: N₂

**N₂**

src: N₁
dst: ?

**S**
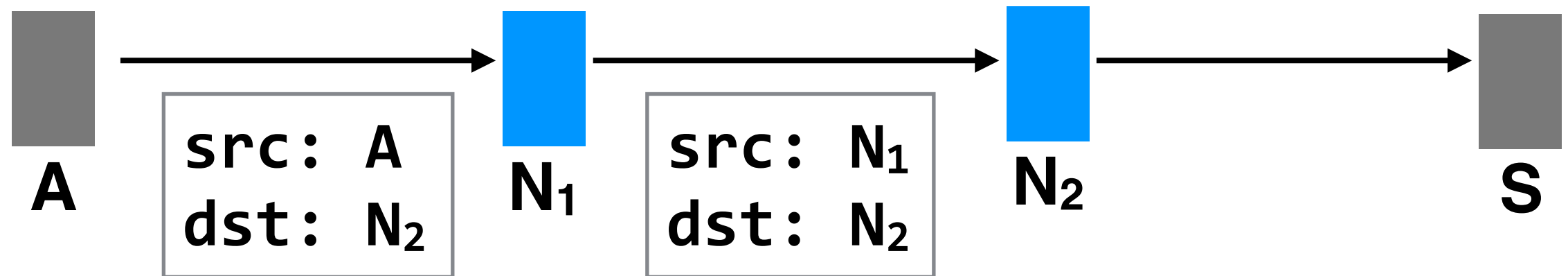
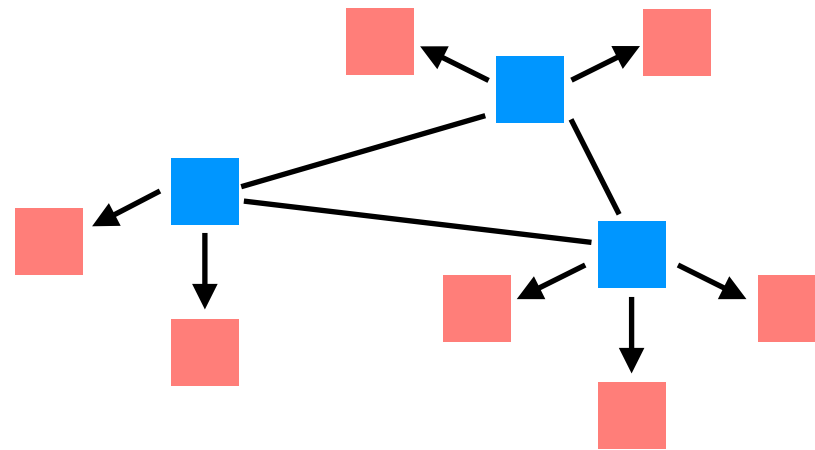**problem:** N₂ has no idea who
this packet is meant for

**solution:** A and S route their communication through P (who has a public IP)
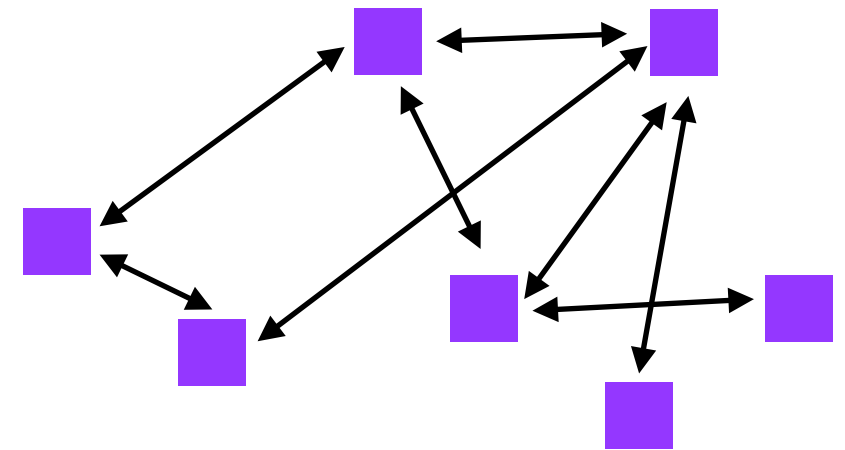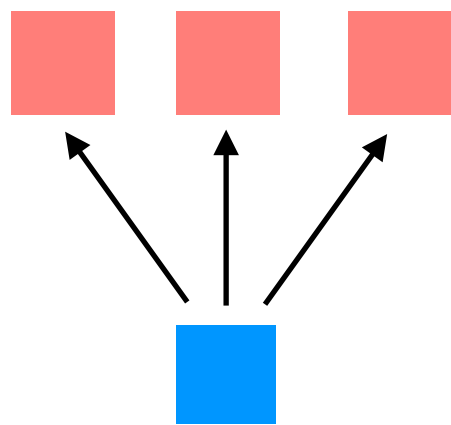
# File-sharing Techniques

**client-server**　　　　**CDNs**　　　　**P2P**
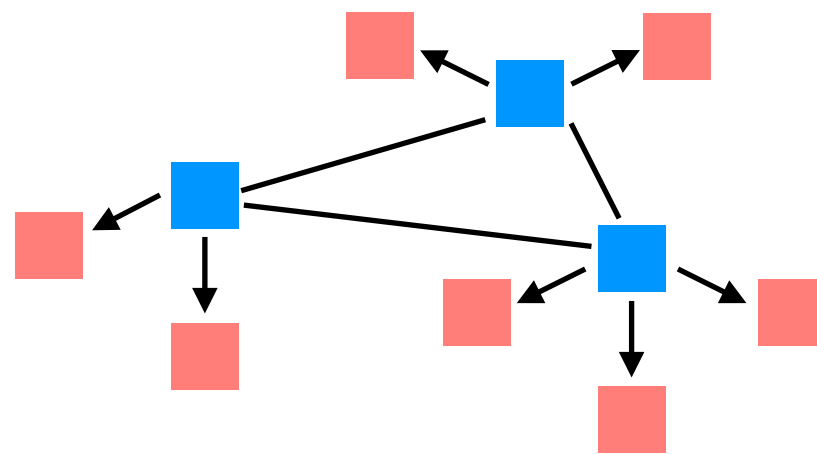


**scalability increases**
(in theory)

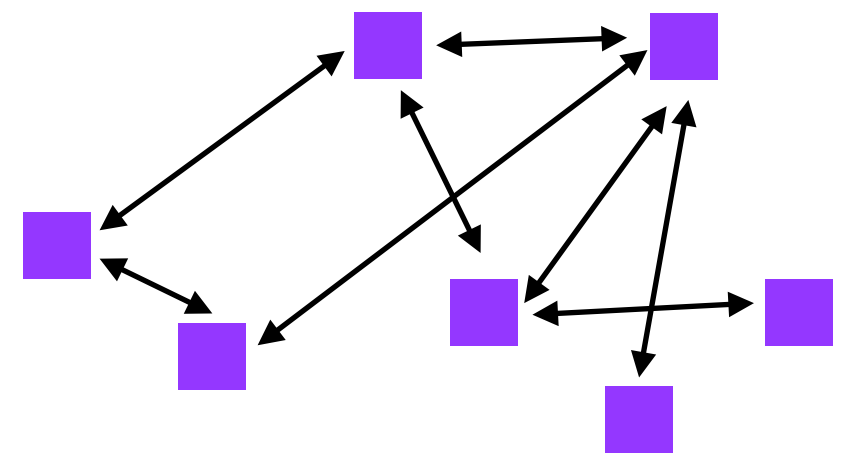# Video-streaming Techniques?

**client-server**          **CDNs**          **P2P**



→ **scalability increases**
(in theory)

- **P2P Networks** are, in theory, infinitely scalable. They can improve performance for some applications, and provide a way to overcome certain aspects of the Internet's architecture. **Incentivizing** peers to behave is an important problem.

- **CDNs** don't scale in the same way that P2P networks do, but are more appropriate for some applications, and provide some features that a P2P network can't (more on that in Thursday's recitation).