P2P

Peer 2 Peer

No always on server, all end-systems connect to each other

they request and serve each other

Self-Scalibity - New peers bring in new capability, but also new demands

Complex Management - Changing IPs, churn

Bit Torrent - File sharing using 256 bit chunks

Server with a tracker, registar of peers

Request Chunks

Alice may ask for a list of chunk peers have

Downloading Rarest First

Alice sends chunks to 4 peers at the highest rate, chokes others

updates friends every 30s, reveals every 10s

Higher upload rate, find better partners, get file faster

Distribution Time

Distribute F-sized file to N clients

V_s - server upload

v_i - peer upload $\quad d_i$ - peer download

Client Server

For Server

Time to upload copies

NF/v_s

$$D_{cs} \geq \max\left\{\frac{NF}{v_s}, \frac{F}{d_{min}}\right\}$$

increases linear

For Client

Each Client must download a copy

d_{min} - min download rate

F/d_{min}

Lower Bound

$\frac{v_s}{N} \leq d_{min}$ - Server sends file in parallel $\quad D_{cs} = \frac{NF}{v_s}$

$\frac{v_s}{N} > d_{min}$ - Server sends file in parallel at rate d_{min} $\quad D_{cs} = \frac{F}{d_{min}}$

Too long if N is Big

P2P (lower) Bounded

Server need to upload a copy

F/v_s

Client downloads a copy

F/d_{min}

Aggregate copies - NF bits

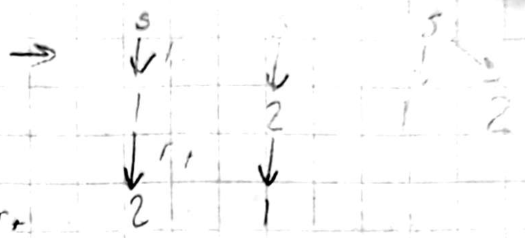
$v_s + \sum_1^N v_i$

$$D_{P2P} \geq \max\left\{\frac{F}{v_s}, \frac{F}{d_{min}}, \frac{NF}{v_s + \sum u_i}\right\}$$

Only has N

P2P lower bound

Trees



Find $\max \sum r_t$

$\sum r_t \le d_i$ ← transmission rate is dependent on d_i of nodes

$\sum c_t(j) r_t \le u_j$

↑ Children Nodes

$$u_s \le \min\left(d_{min}, \frac{u_s + \sum_i u_i}{N}\right) \rightarrow r_i = \frac{u_i}{\sum^k u_j} u_s$$

Download rate $r_i + \sum r_j = \sum_j r_j = u_s < d_{min}$

Upload rate $\sum_i r_s = u_s$

Total Upload $(N-1) r_i = \frac{u_i (N-1)}{N} u_s < u_i$

] $\frac{E}{u_s}$

$$\frac{u_s + \sum^N u_i}{N} \le \min(d_{min}, u)$$

$$r_i = \frac{u_i}{N+1}$$

$$d_{min} \le \min\left(u_s, \frac{u_s + \sum u_i}{N}\right) \qquad \frac{F}{d_{min}}$$