CSF641 – P2P Computing

點對點計算

Path-aware Multicast for Efficient File Distribution in Peer-to-Peer Overlay Networks

Department of Computer Science and Information Engineering
National University of Kaohsiung, Taiwan

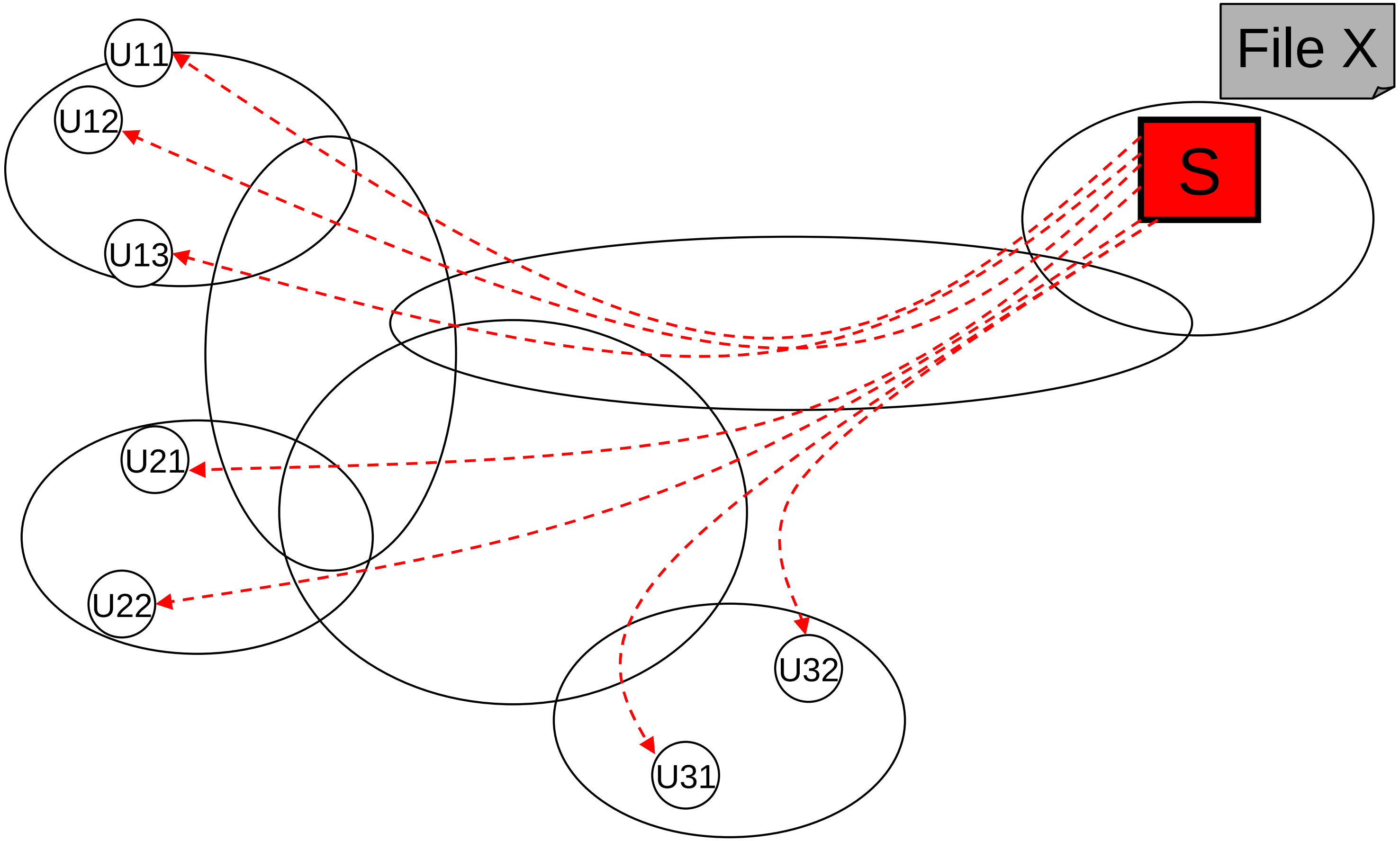
ICOIN 2006

Outline

1. Introduction
2. A Multicast Approach to File Distribution
3. Amplicast: Hybrid of Amplification and Multicast
4. PeerTop: Lightweight Network Probing
5. Experimental Results
6. Conclusion

File Distribution

Fundamental operation: transmitting a file from a source peer to a group of destination peers

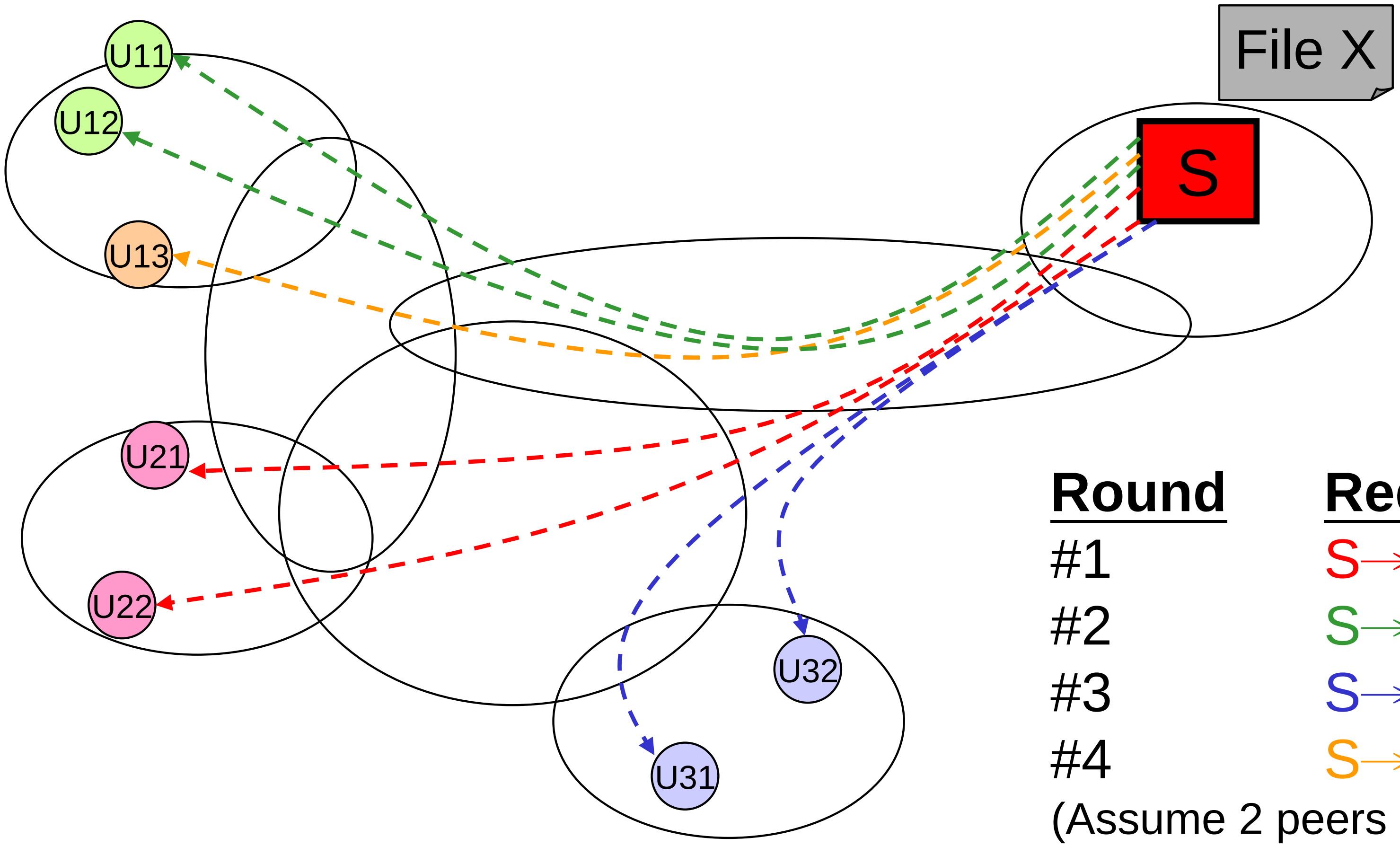


Applications

- ***P2P file-swapping***: a peer simultaneously receives multiple requests from other peers for the same file
- ***Content-push applications***: a source peer needs to replicate the same file to a specified group of peers
  - ***Uploading***: a developer wants to upload a new program to a cluster of machines worldwide like PlanetLab
  - ***Mirroring***: a content provider wants to replicate contents to a set of mirror sites
  - ***Remote Backup***: a company wants to duplicate data to a couple of backup sites
  - ***Publishing***: a publisher wants to distribute contents to subscribers
  - ***Upgrading***: a software house wants to push software patch or data to its customers

Intuitive Approach: Root-Serve

- ***Successively*** serve all the requesting peers by the source peer
  - not simultaneously due to limited network bandwidth or server capability

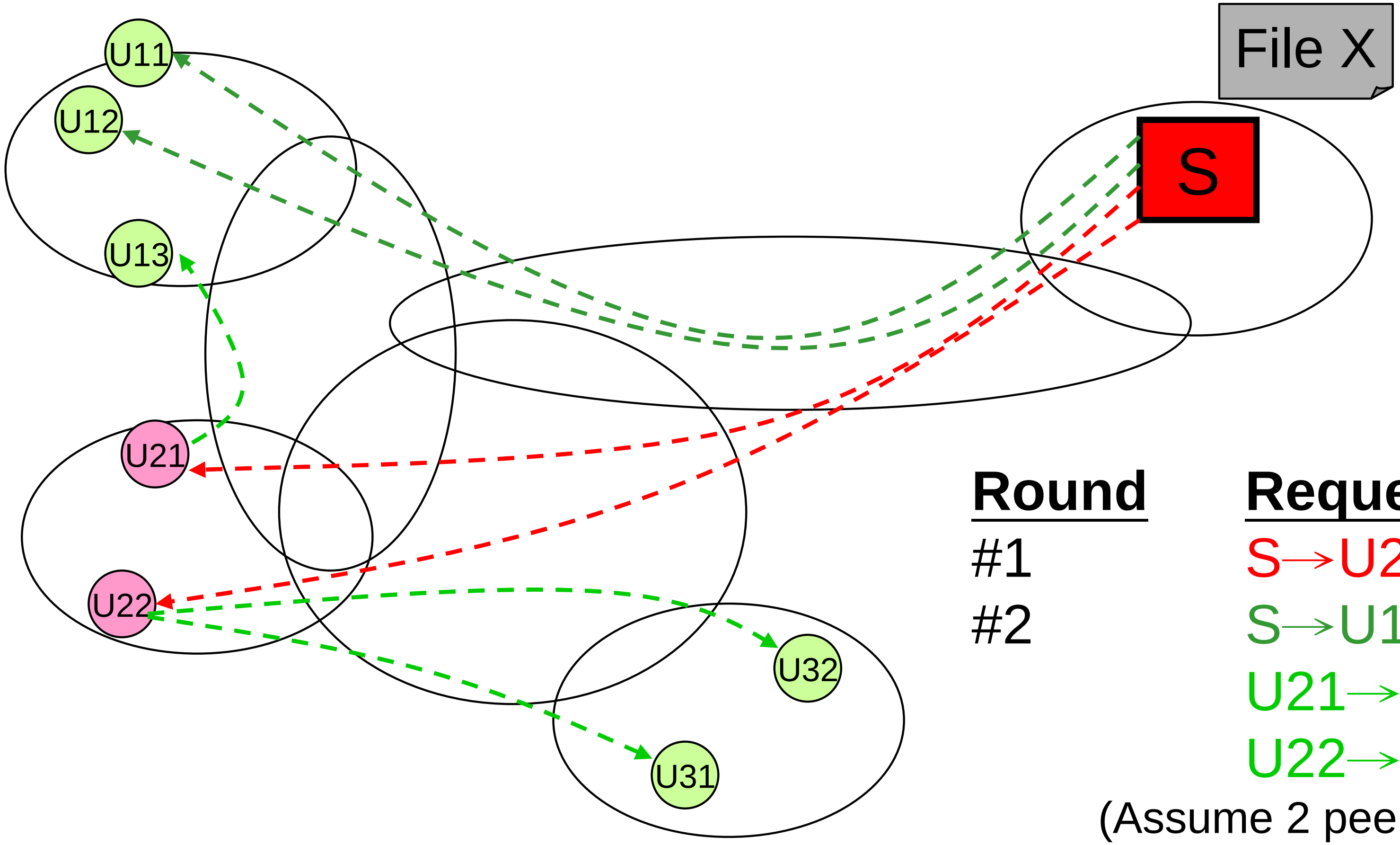


Round	Requestors
#1	S→U21, S→U22
#2	S→U11, S→U12
#3	S→U31, S→U32
#4	S→U13

(Assume 2 peers served per round)

Cooperative Approach: Amplification

- ***After*** a requesting peer receives a file, it becomes a supplying peer of the file at next rounds



Round	Requestors
#1	S→U21, S→U22
#2	S→U11, S→U12
	U21→U13
	U22→U31, U22→U32

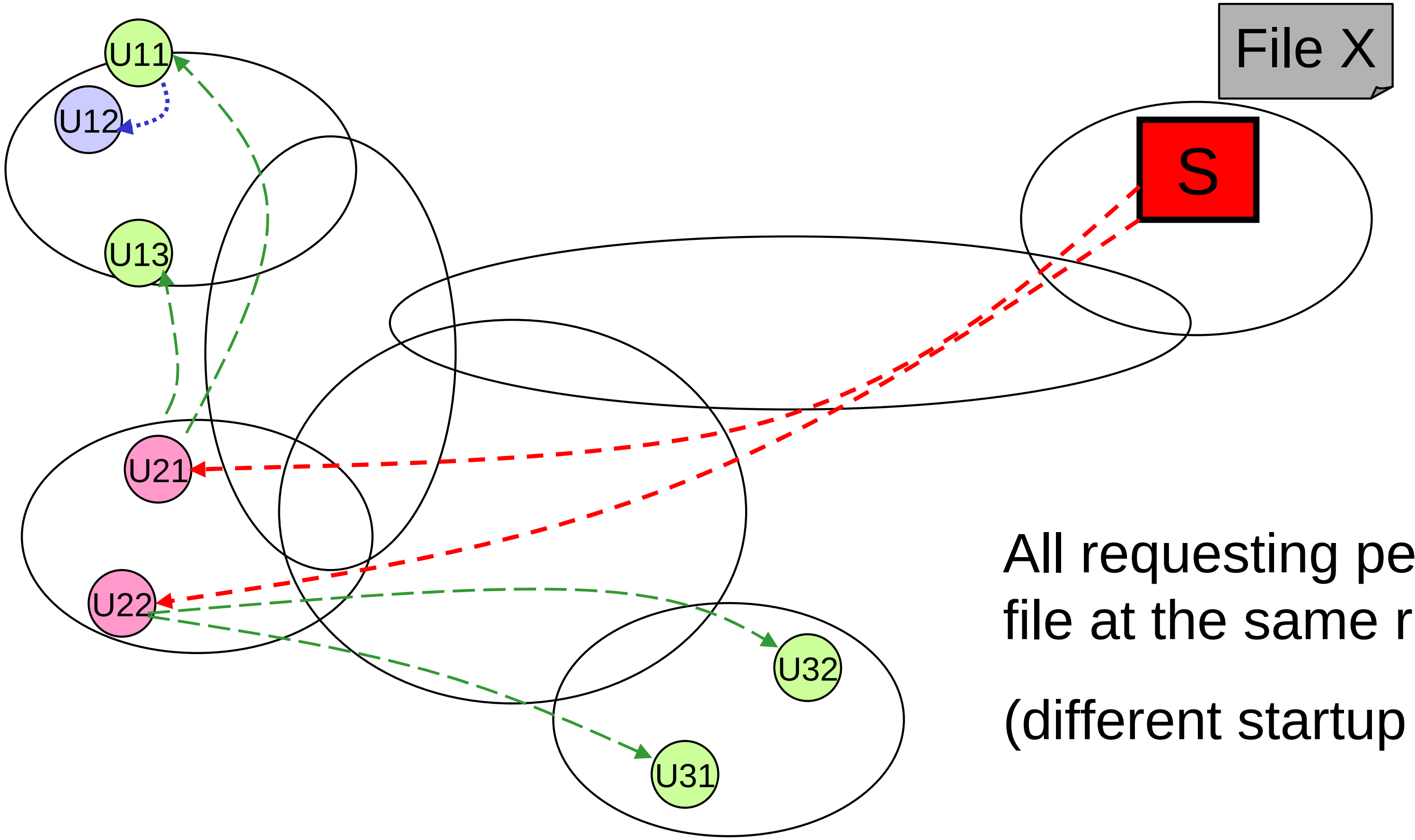
(Assume 2 peers served per round)

Discussion of Amplification Approach

- Some requesting peers may download the file from peers other than the source peer
  - Reduce the load of the source peer
  - Reduce the waiting times of the peers
    - not necessary to wait the source peer available
    - probably the link to other peers is faster
- However, most requesting peers still need to wait several rounds before being served
  - Issues: which waiting peers to be served by which replicate peer at which round
- Client-side enhancements
  - parallel download
  - file splitting (swarming)

Pipelining Approach: Multicast

- ***As soon as*** a requesting peer receives something, it forwards the received part to downstream peers



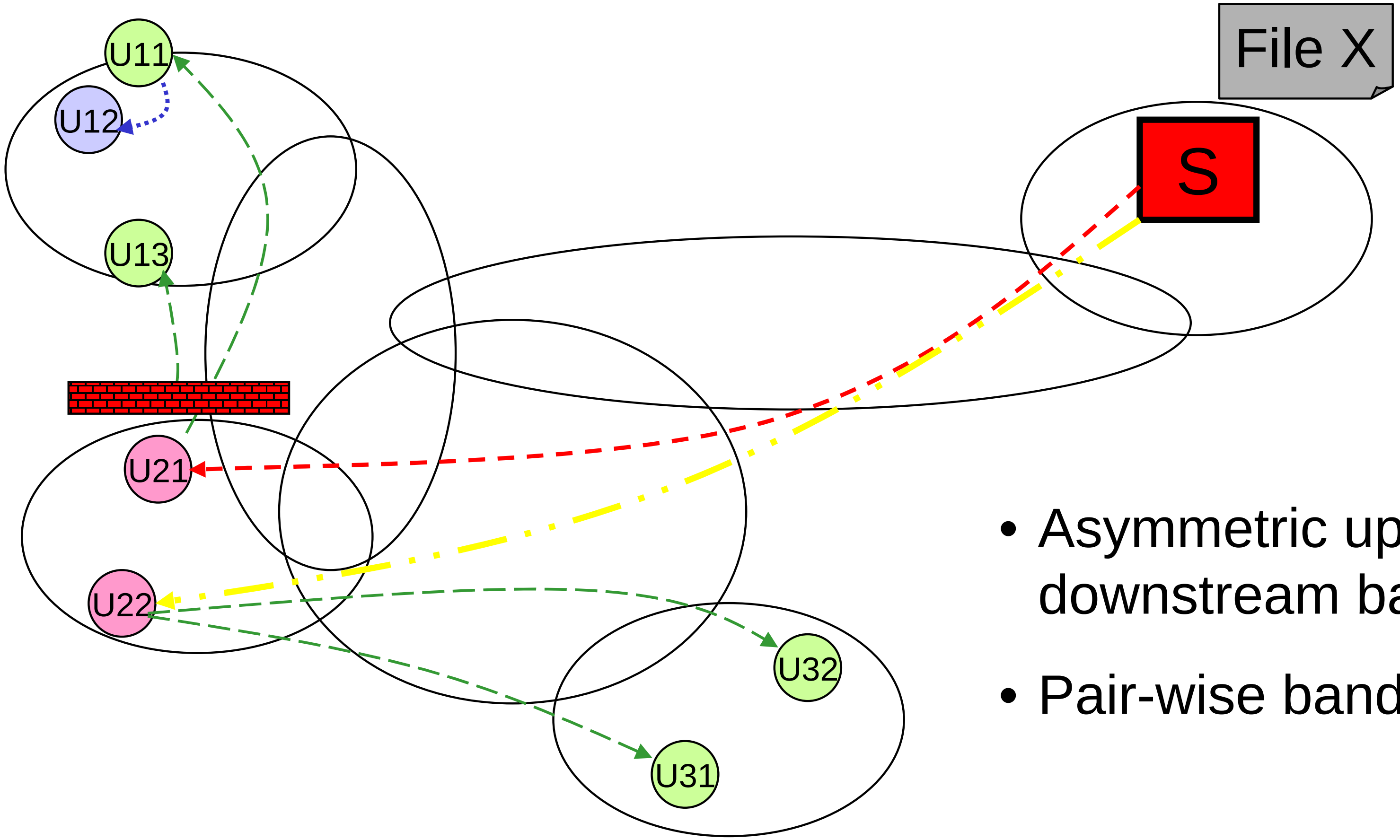
All requesting peers receive the file at the same round

(different startup times)

Challenges

The construction of the multicast tree should consider

- Bandwidth: avoid choosing a slow-link peer near the source
- NAT and Firewall Issues: avoid choosing a leaf peer too soon

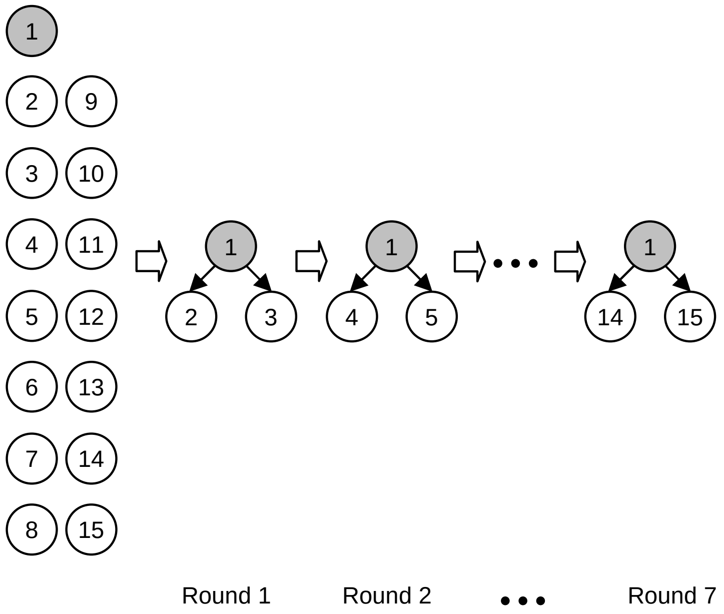


- Asymmetric upstream and downstream bandwidth
- Pair-wise bandwidth differentiation

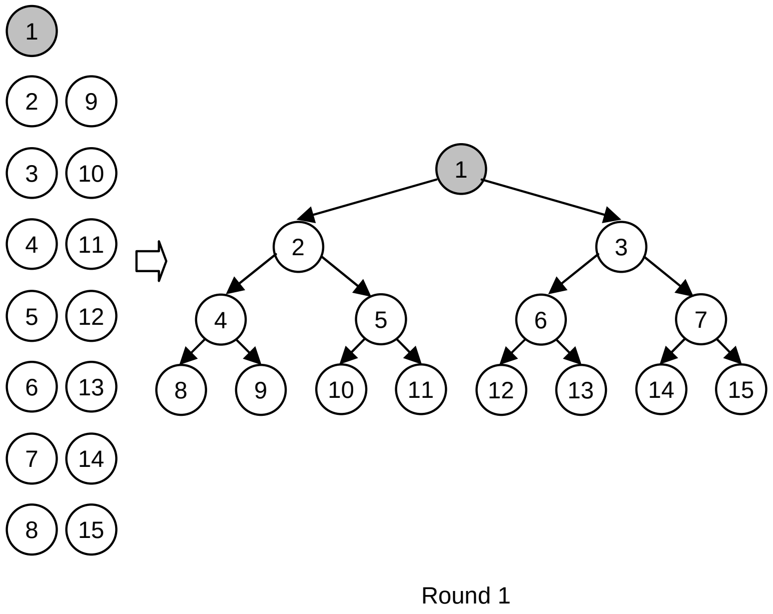
Stream Distribution vs. File Distribution with Multicast Trees

	Stream Distribution	File Distribution
Goal	Every tree node *smoothly* plays the live stream that lasts the same duration	Every tree node receives an *intact* copy of the file
Slow link	Slow-link nodes would buffer more data before starting to play. All descending peers inherit the delayed start	All peers descending below a slow link take longer times to receive the file
Pkt loss	Tolerable (worse video quality)	*Retransmission* is necessary
Pkt delay	Like a lost packet	OK or retransmitted
Properties	Different startup time Same duration time	Different startup time Different duration time
Tree construction	Usually construct a single tree to connect as many nodes as possible	Waiting for a fast-link peer is probably quicker than joining a slow-link tree

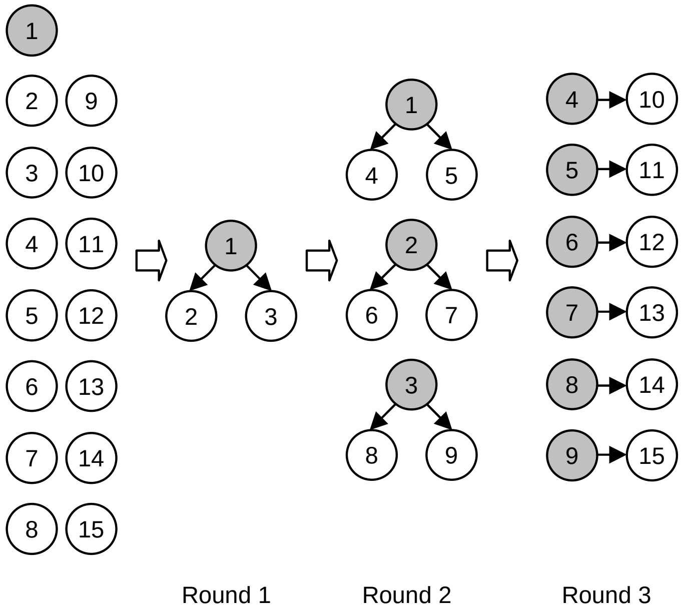
Comparison



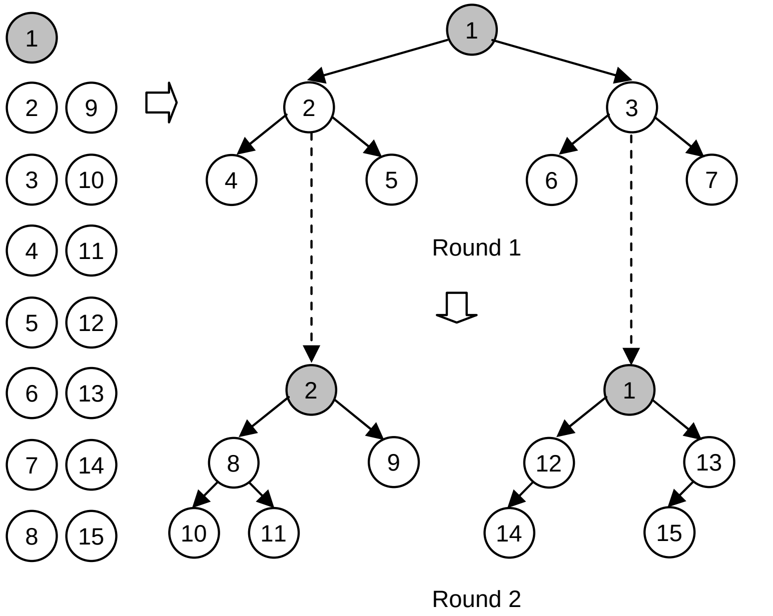
Root-Serve



Multicast



Amplification



Amplicast

Multicast Trees for File Distribution

- Amplifiable Multicasting – Amplicast

  if a requesting peer finds that

  - ***joining the tree to receive the file at the current round*** is later than
  - ***joining another multicast tree at some later round***,

  The peer would not be connected to the multicast tree at the current round

  => Amplicast may construct more than one multicast tree to distribute requested content from the source peer to a group of requesting peers

- Path-aware Multicasting - PeerTop

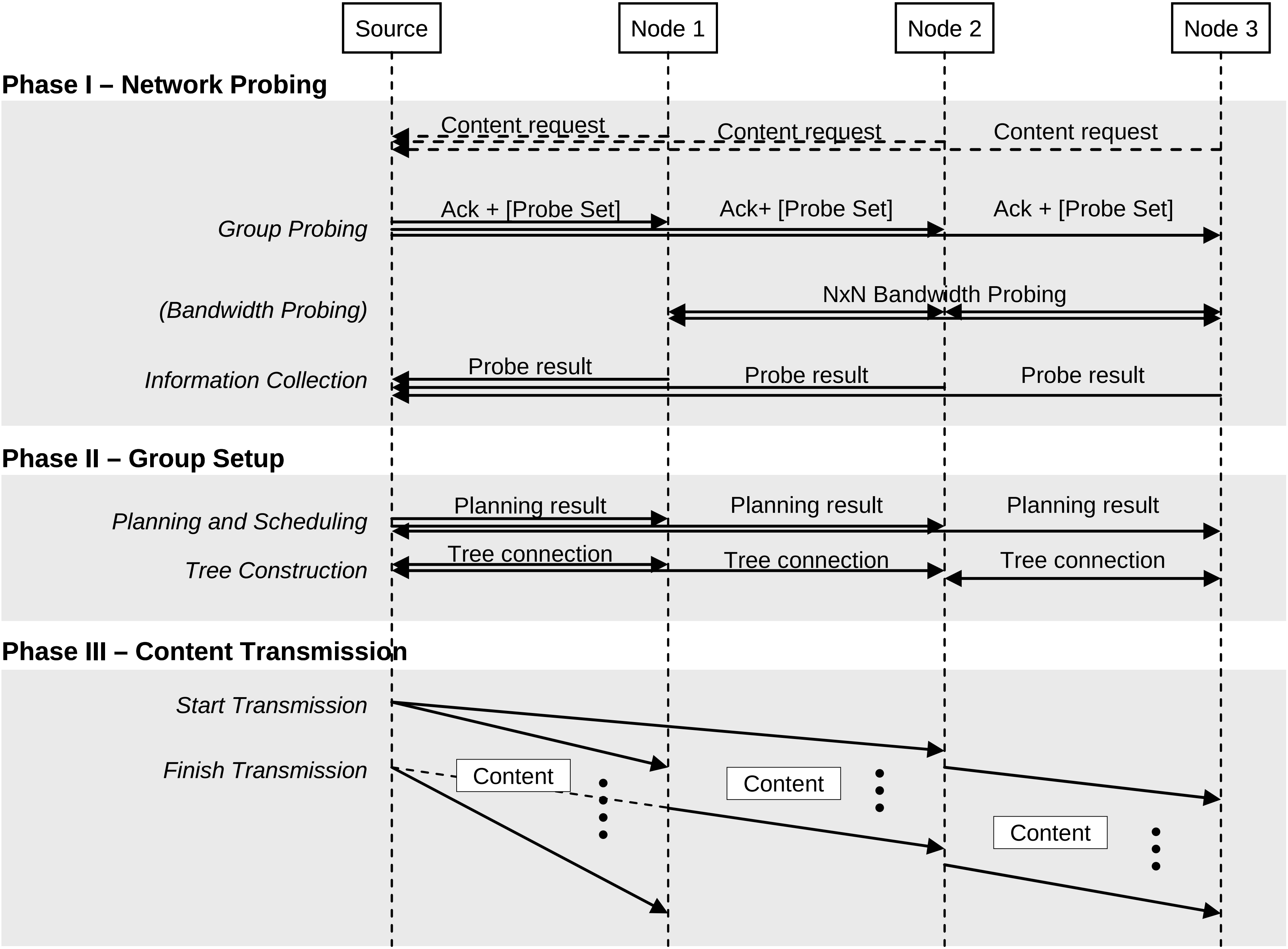
  The peers probe each other to measure real-time pair-wise network information, such as bandwidth, ping time or delay

  => cache and top-set heuristic are applied to reduce probe overhead

Basic Steps of Amplicast

1. Network Probing
   - Admitted peers measure the end-to-end download bandwidths from others and report to the source peer
2. Group Setup
   - The source peer performs the amplicast algorithm to construct amplifiable multicast trees
3. Content Transmission
   - Admitted peers begin to
     receive the file from the arranged parent peer and forward the received part to arranged child peers

Message Flow of Amplicast



Amplicast Algorithm

```
T includes S
While P is not empty
    If(all the nodes of P are leaf nodes)
        Find Pj of P and Ti of T where Ti is not occupied
          and FT(i, j) is the smallest
    Else
        Find Pj of P and Ti of T where Ti is not occupied,
          Pj is not a leaf node and FT(i, j) is the smallest
    Endif

    If(candidate peer Pj with parent Ti is found)
        Find Mk of M or Tk of T where Mk is not occupied
          and ET(k, j) is the smallest
        If ET(k,j) < FT(i,j)
            M includes Pj   // had better wait
        Else
            T includes Pj   // join the current tree
        Endif
    Else
        // Try amplification due to busy
        Find Mi of M or Ti of T, and Pj of P where Mi or Ti
          is not occupied and ET(i, j) is the smallest
        M includes Pj
    Endif
    P excludes Pj
Endwhile
Start transmission
```

S	the source peer
P; Pi	set of requesting peers; a peer of **P**
T; Ti	set of tree nodes; a node of **T**
M; Mi	set of nodes waiting for next rounds; a node of **M**
FT(i,j)	expected finish time for peer *j* to receive streamed content from peer *i*
ET(i,j)	expected finish time for peer *j* to wait a round and receive content from peer *i*

Design Issues of Amplicast

- Peer Selection
  - Find first the peers that can upload to others
    - that is, not behind a firewall nor freeloaders
    - freeloaders will then have lower priorities
  - Serve the above peer that keeps the finish time small
    - tend to have the largest pair-wise bandwidth to some tree node
    - a heuristic like traditional packet/stream multicast algorithms but using dynamic pair-wise link information
- Finish Time Prediction
    - The source peer selects the peer with the smallest finish time
    - A candidate peer will evaluate whether it is faster to wait to get the content from another peer that is occupied in this round
- Incentives
    - The service capability of a peer is measured by other peers and reported to the source peer
    - Freeloaders have lower priorities during peer selection

PeerTop Network Probing

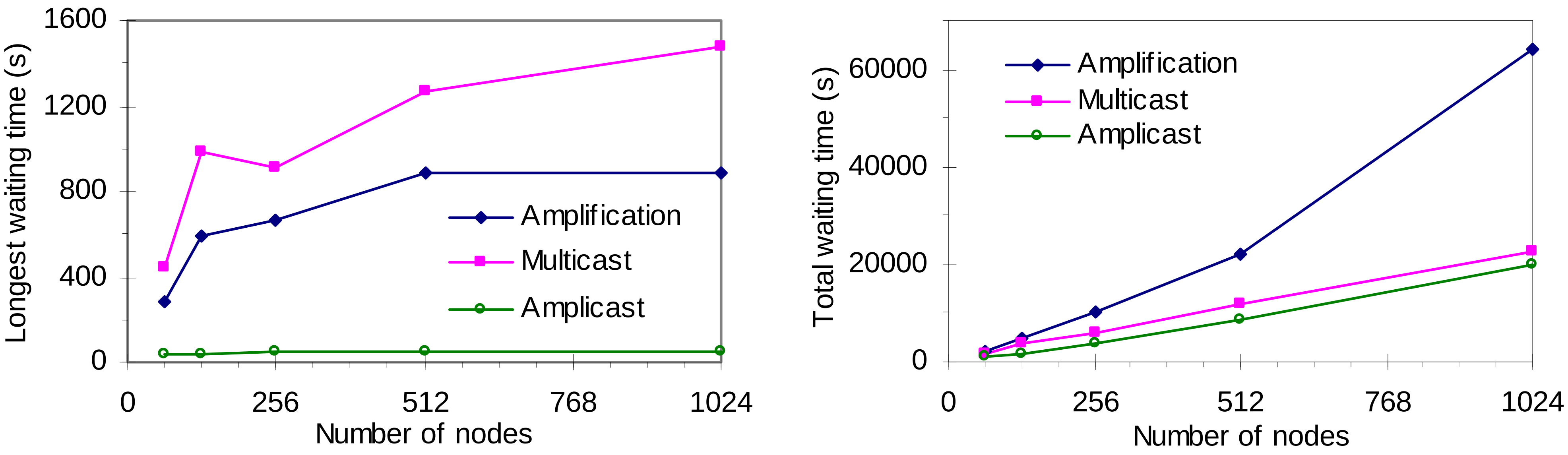
1. Utilizing the link information probed by other overlay networks such as RON, Sprobe and PDF
   - no extra overhead to implement Amplicast
2. PeerTop – lightweight probing
   - probe cache: each peer caches all the download information newly probed or collected
     - the (freeloader or firewall'ed) peers that can't upload to the peer are then detected
   - probe order (or preferred list): based on the download bandwidths from other peers
     - in case it can not probe all the nodes requested by the source
   - top node set: a portion of the probe set that supports high upload bandwidths to the peer
     - rather than exhaustedly probing all the links to the probe set

(ref. C.M. Cheng, Y.S. Huang, H.T. Kung, and C.H. Wu, "*Low-Cost Relay Routing for Achieving High End-to-End Performance*," IEEE Globecom 2004)

Experiment Environments

- Brite Simulator
  - Waxman models (α=0.15 and β=0.2)
  - Average 100 topologies of 64, 128, 256, 512, and 1024 nodes each
  - Heavy-tailed bandwidth distribution
  - File size: 100MBytes
  - Branch factor: up to 4
  - PeerTop: 8, 16, 32, 64 and 128 top nodes for the topologies of 256 nodes
- PlanetLab Dataset
  - 212 nodes probe each other every two hours during May 24 to May 30, 2004
  - 50%: 106 nodes; 25%: 53 nodes; 12.5%: 27 nodes
- Measurements
  - waiting time (finish time) = startup time + transmission time
  - longest waiting time = how long the system takes to distribute the file to all the requesting peers
  - total waiting time = the summation of all individual waiting times

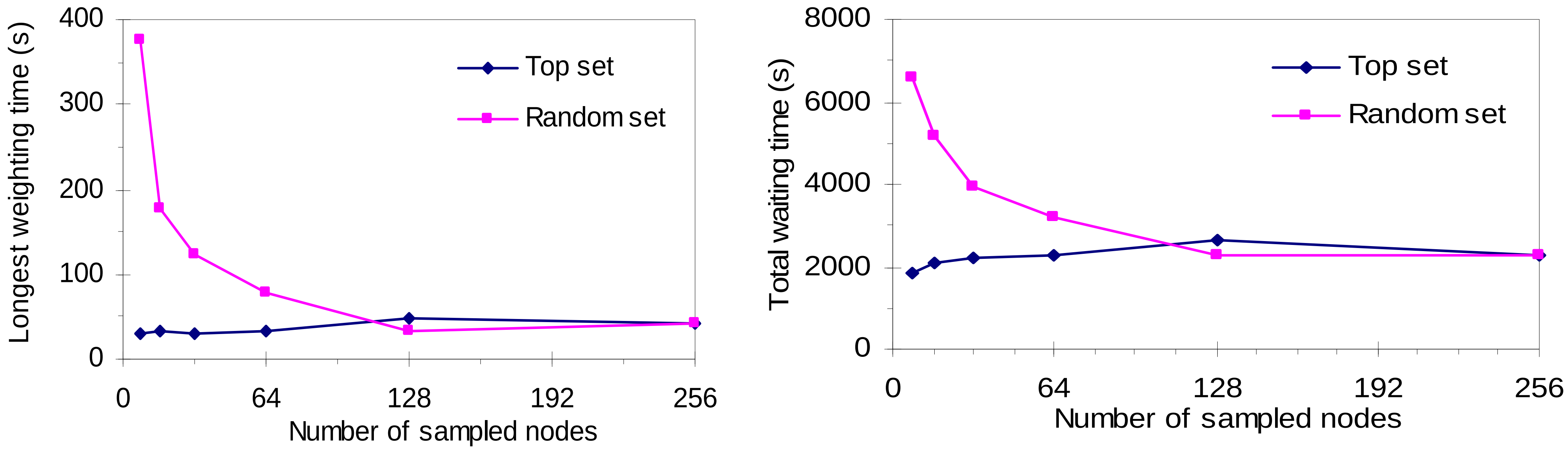
Fig. 2. Simulation of Brite model



(a) longest waiting time

(b) total waiting time

Fig. 3. Evaluation of PeerTop

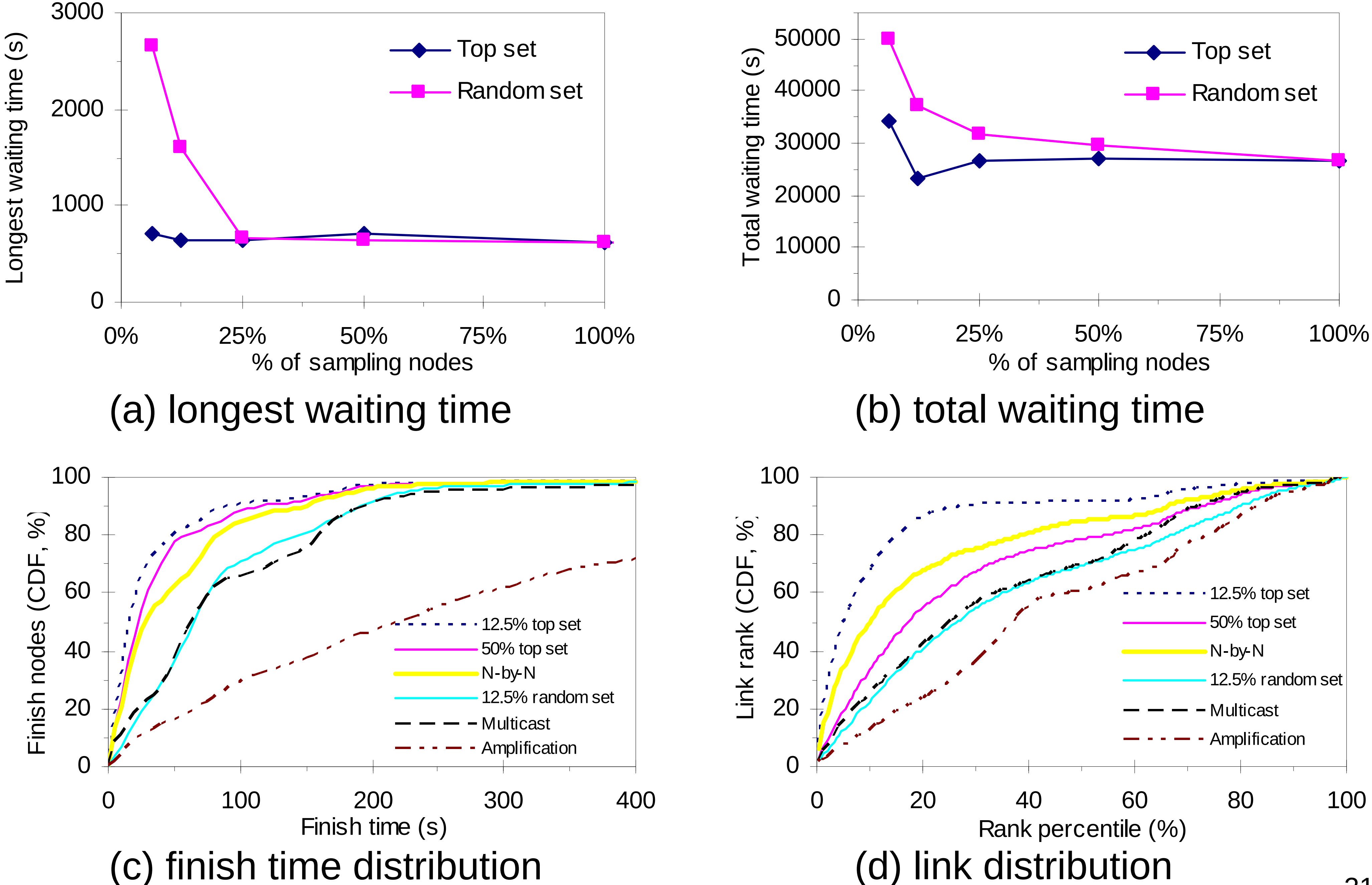


(a) longest waiting time

(b) total waiting time

# of top nodes ↑ => Probability to wait for next rounds ↑ => mis-predicted ↑

Fig. 4. Evaluation of PlanetLab dataset



(a) longest waiting time

(b) total waiting time

(c) finish time distribution

(d) link distribution

Table 1. Comparison of performance among different content distribution approaches using the PlanetLab dataset

Metrics	Amplification	Multicast	N-by-N	12.5% top set	12.5% random set
Longest waiting (s)	3,585	2,661	611	636	1,611
Total waiting (s)	142,764	41,273	26,600	23,220	37,111
Average link rank	44.0	34.4	24.0	12.0	32.9

Conclusion

- To distribute a large file, we propose
  - Amplicast: a hybrid approach of file amplification and stream multicast
    - in multicast, most peers can start to receive the file earlier, and
    - in amplification, the peers can wait to choose a better server in order to avoid receiving the file from a low bandwidth link
  - PeerTop: lightweight network probing with link cache and a heuristic of top-set sampling
- Intelligent peer selection: considering
  - Bandwidth of end-to-end paths and incentive of peers
  - Finish time prediction
- Further issues:
  - node leave
  - collusion or multiple peers within the same firewall