

# Climber: Incentive-based Resilient Peer-to-Peer Streaming Services

Taekyoung "Ted" Kwon

[tkkwon@snu.ac.kr](mailto:tkkwon@snu.ac.kr)

Seoul National University

# Outline

- Live Streaming Services
- Climber
- Discussions
- Conclusions

# Live Streaming Services



**LIVE EARTH FILM PROJECT**  
IN SUPPORT OF THE CAUSE

**CONCERTS**

- UNITED STATES
- UNITED KINGDOM
- BRAZIL
- CHINA
- AUSTRALIA
- SOUTH AFRICA
- JAPAN
- GERMANY
- MORE

Award-winning filmmakers have created these short films, commissioned by Live Earth, to showcase the diversity and creativity of the international response to the climate crisis. The Live Earth Film Project premiered at the 2007 Tribeca Film Festival and is sponsored by Absolut.

**Film Title: "Sunny Day"**  
Director: Sophie Muller  
Film Synopsis: A beautiful stylistic film encouraging the people of LA to ditch the tumble dryer and encourage use of the eco-friendly washing line instead.  
[Watch now!](#)

**Film Title: "Mermaid"**  
Director: Matt Tomaszewski  
Film Synopsis: This short film explores the ocean world as it appears to a young girl.  
[Watch now!](#)

## • MSN broadcast

- <http://liveearth.msn.com>

- Live Earth concert in July 2007 provides 15 million streams to 237,000 simultaneous viewers peak time 3/19

# Live Streaming Services

- Cost
  - Building a server farm costs too much
  - CDN is not inexpensive at all
  - P2P is the mantra
    - Encourage peers to contribute to the system
- Resilience
  - As peers dynamically join/leave, the quality of experience degrades
  - A system should work in the presence of high churning rate
- Delay
  - Live streaming is not like file downloading
  - Packets should be delivered in time

# Climber

- An Incentive-based Resilient P2P System for Live Streaming Services
- A hybrid of a tree and a mesh
  - Tree - main structure to forward streaming
  - Mesh - enhance resilience by redundant edges
- Design goals
  - Incentive
  - Resilience
  - Self improvement
  - Adapting to network condition

# Assumptions and Definitions

## • Assumptions

- Connections are symmetric
- Maximum uplink bandwidth is decided by the user

## • Definitions

- $O_i^{\max}$  : node  $i$ 's maximum possible outdegree  
(  $1 \leq O_i^{\max} \leq \text{uplink capacity/flow rate}$  )
- $O_i$  : node  $i$ 's current number of outdegree  
(  $0 \leq O_i \leq O_i^{\max}$  )

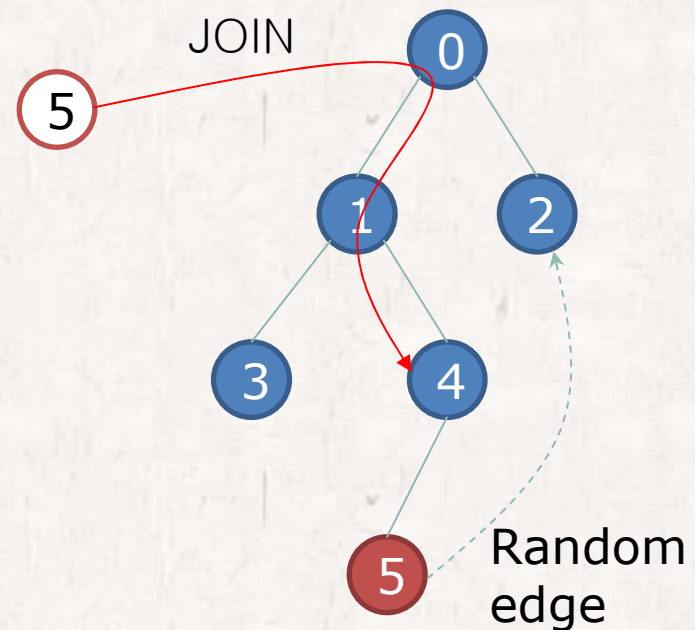
# Join/Leave procedure

## Join

- It sends a JOIN message to the root node
- Existing nodes take the new peer as a child node by the probability  $1 - O_i / O_i^{\max}$ 
  - If not OK, relayed to one of its child
- It establishes random edge(s)

## Leave

- It sends LEAVE messages to its parent and child nodes

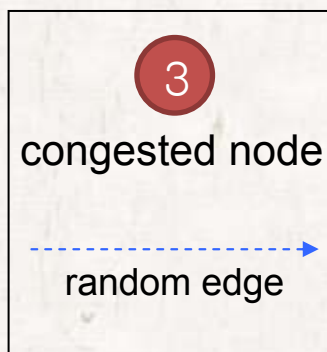
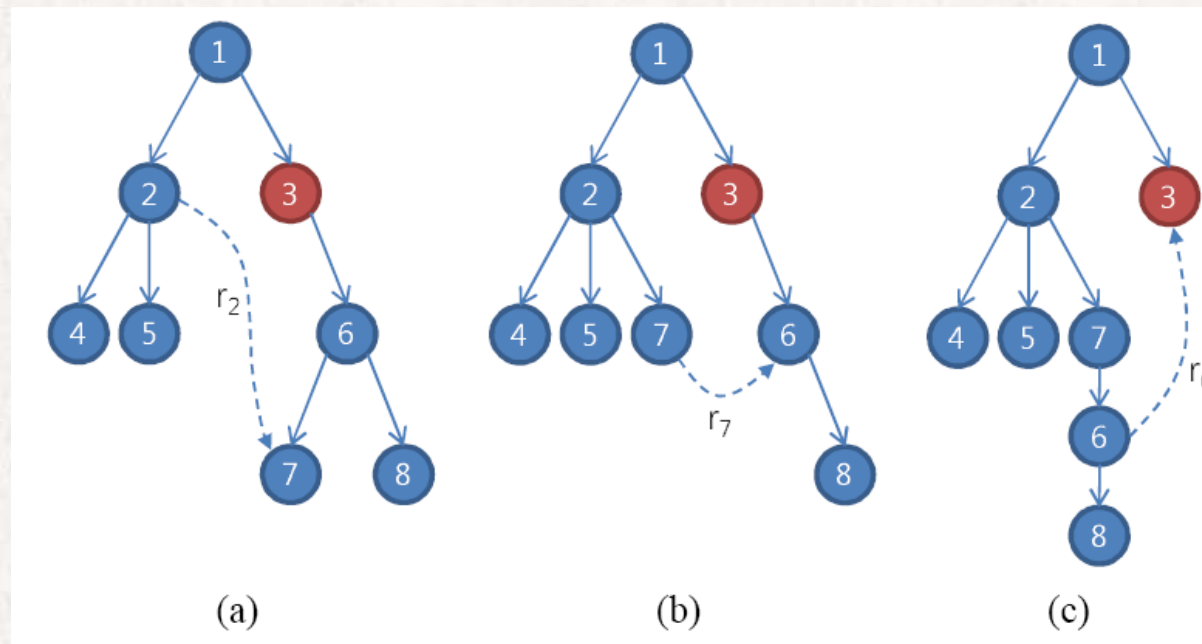


# What is a Random Edge?

- A directed edge to a random peer
  - Not to parent nor children
- For each available outdegree ( $O_i^{\max}$  - # of children), node  $i$  makes a random edge with probability  $\lambda$ 
  - $\lambda$  is piggybacked in the streaming packets from the root node
- Only the sequence number of the stream is sent along random edges
  - Streaming packets over tree edges

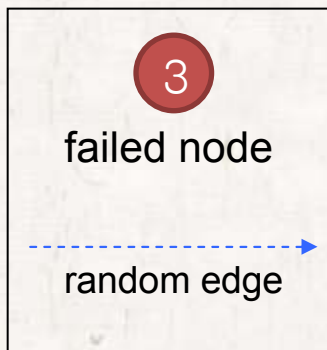
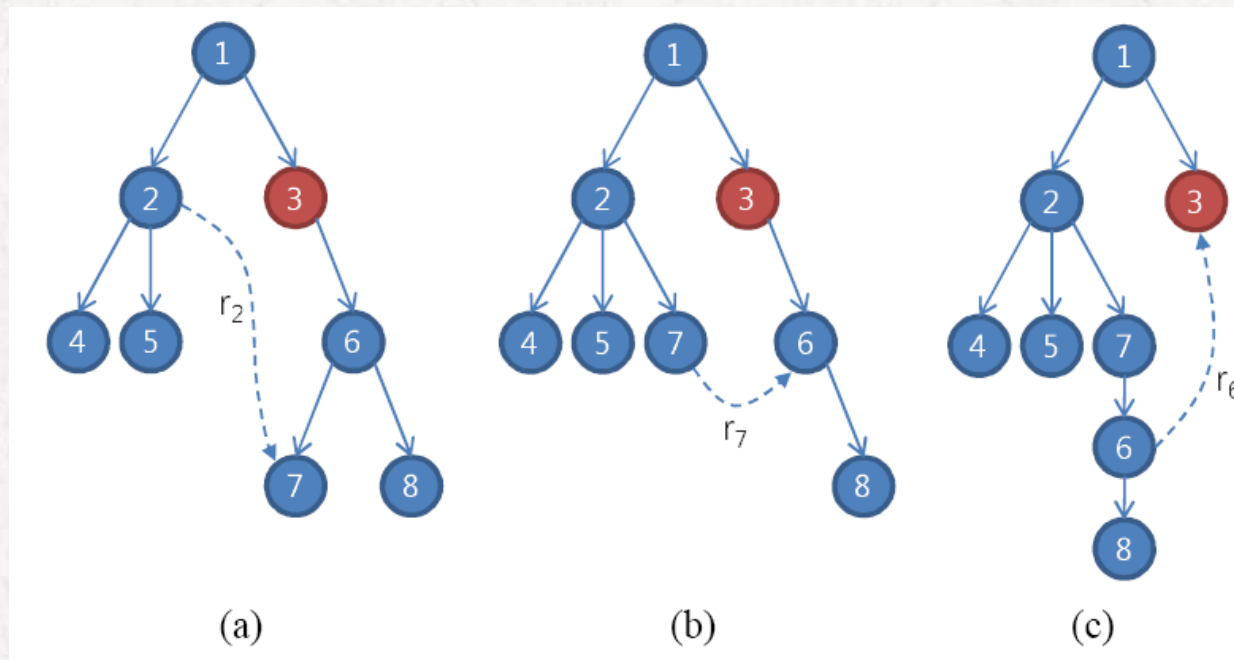
# A node climbs a tree by a Random Edge

- Node 7 climbs the tree by being a child of node 2



# Recovery Procedure

- Climber does not have additional recovery procedure
  - Climber does not distinguish node failure from node congestion
- If there is no incoming random edge, nodes are **interrupted**



# Random Edge Probability $\lambda$

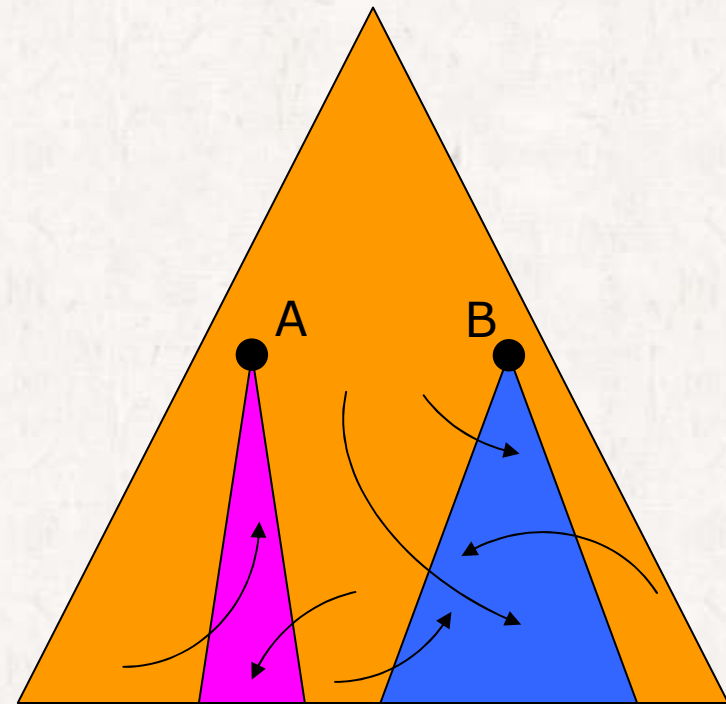
- When a node is *interrupted*, it sends a REJOIN message to the root node
  - By counting the # of REJOIN, the root keeps track of the current interruption rate  $\gamma_t$
- Root node compares  $\gamma_t$  with predefined interruption rate threshold  $\gamma$ 
  - $\gamma$  : pre-configured by a service provider
  - $\gamma_t$  : the number of received REJOIN messages over the total number of nodes at time  $t$
- If the churning rate increases ( $\gamma_t > \gamma$ )
  - The root node increases  $\lambda$  to enhance the resilience of the tree
  - Otherwise, decreases  $\lambda$  to save network bandwidth

# Discussions

- Cost
  - Climber encourages peers to provide more upload bandwidth by giving more incentive to a peer of more contribution
- Resilience
  - Climber efficiently recovers from node failures and adapts to network congestion condition
- Delay
  - Climber keeps improving its topology to maintain a faster path from root node

# Incentive

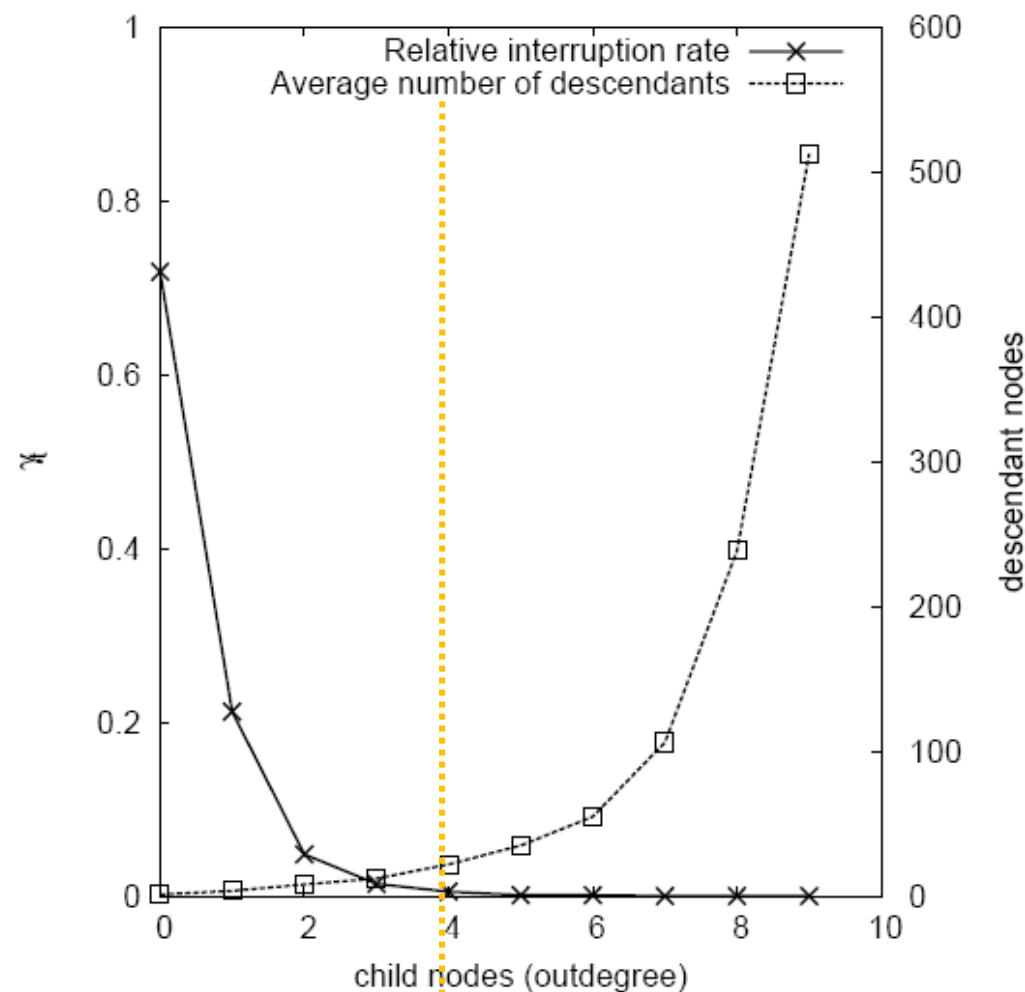
- Climber provides more resilient streaming service to the nodes of higher contribution  
-> How?
- A peer that provides **more upload bandwidth** (more children nodes) to the system will have **more descendants**, which implies **more incoming random edges** probabilistically



# Incentive

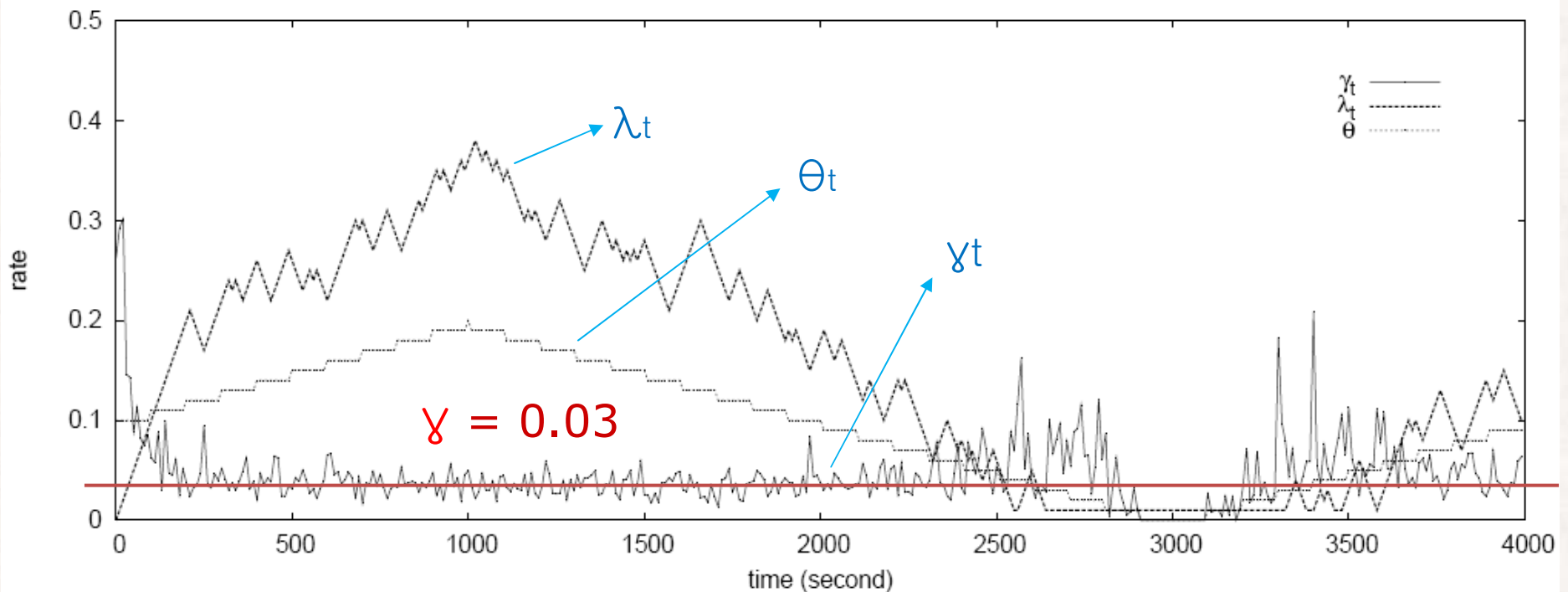
- $Inter(k)$ 
  - the number of interruptions a node experiences whose outdegree is  $k$
- Relative interruption rate of a node with outdegree  $k$

- $$\frac{Inter(k)}{\sum_{j=0}^{\max} Inter(j)}$$



# Keeping the value $\gamma$

- $\gamma = 0.03$
- $\gamma_t$  : current interruption rate at time  $t$
- $\lambda_t$ : random edge probability at time  $t$
- $\theta_t$ : network churning rate at time  $t$



# Delay from root node

- $\lambda = 0.04$
- 80% of the nodes receive packets within 11 seconds when  $\theta = 0.16$
- To improve the delay, we should increase  $\lambda$ , which in turn enhances resilience of the tree

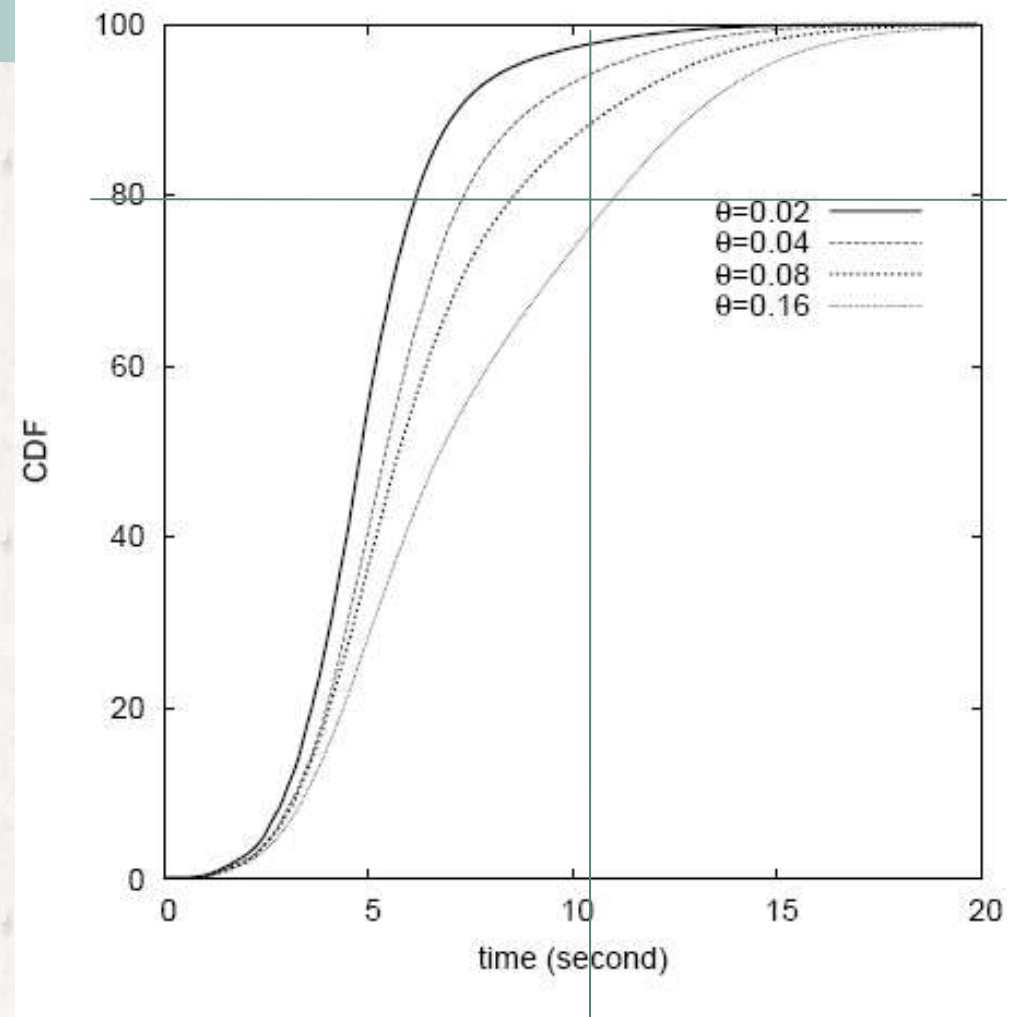
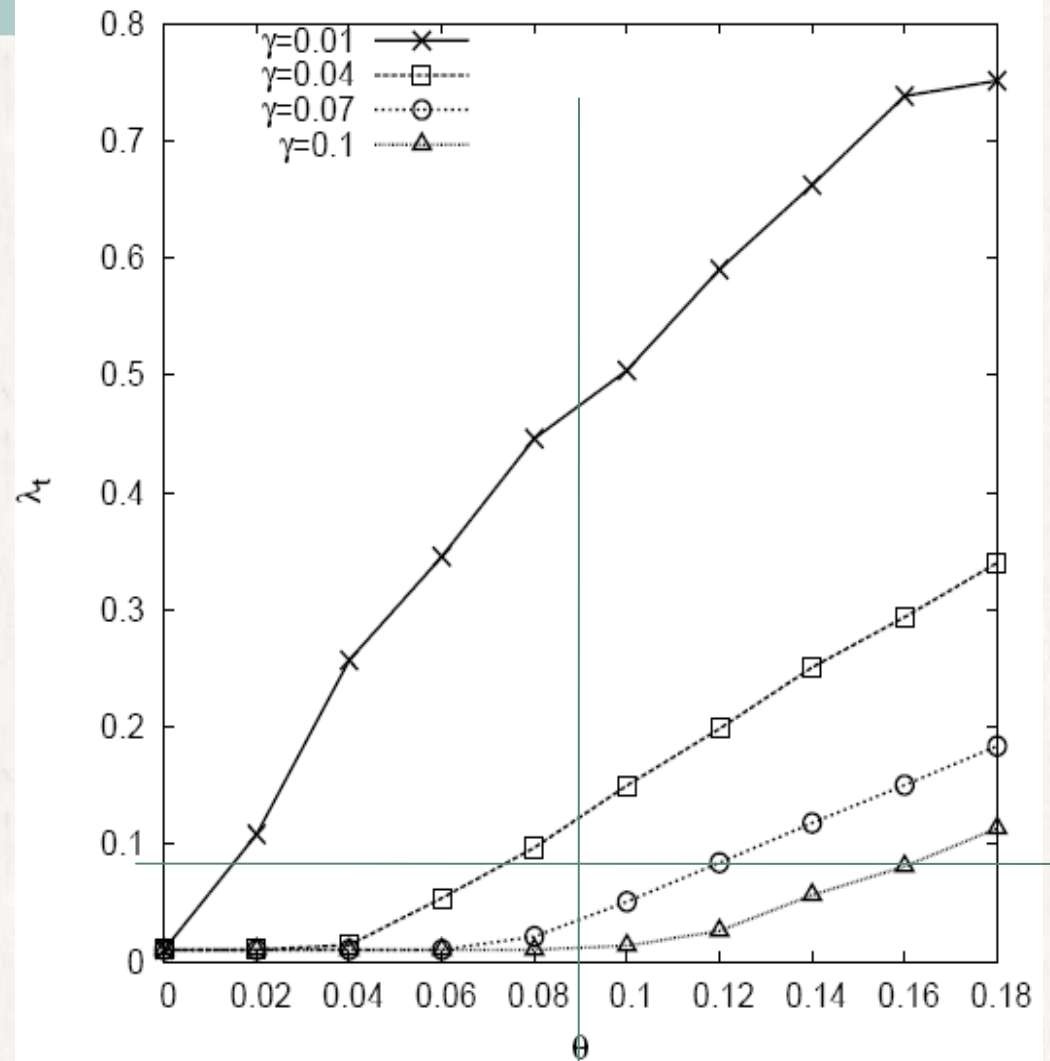


Fig. 5. CDF of delay from root node

# Overhead

- $\lambda$  required to satisfy the level of QoS ( $\gamma$ ) depends on the churning rate  $\theta$ 
  - When  $\theta = 0.08$ , the system needs to make random edges with probability 0.1 to meet the constraint  $\gamma = 0.04$



# Conclusions

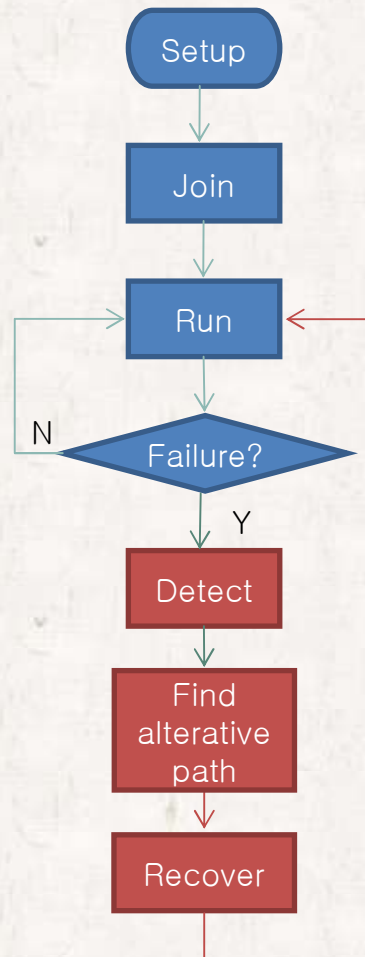
## ● Climber

- gives more incentive to a highly contributing peer
- efficiently recovers from node failures
- keeps improving the performance
- adapts to the churning rate to satisfy the QoS (interruption rate)

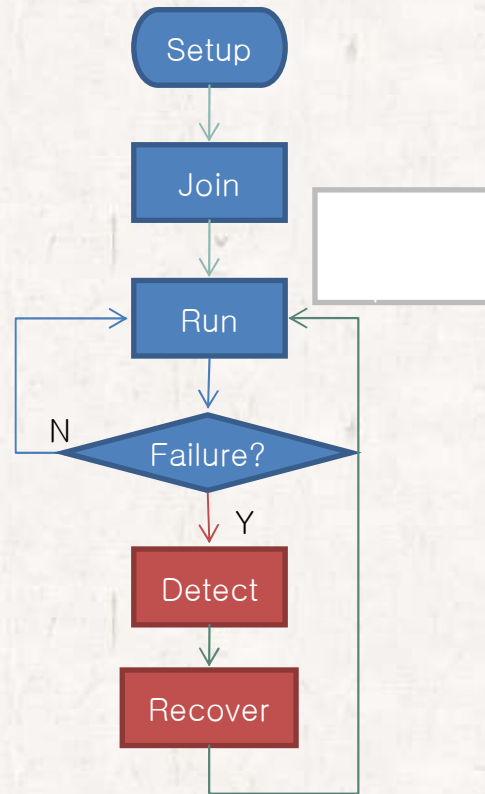
# Thank you

- Questions?

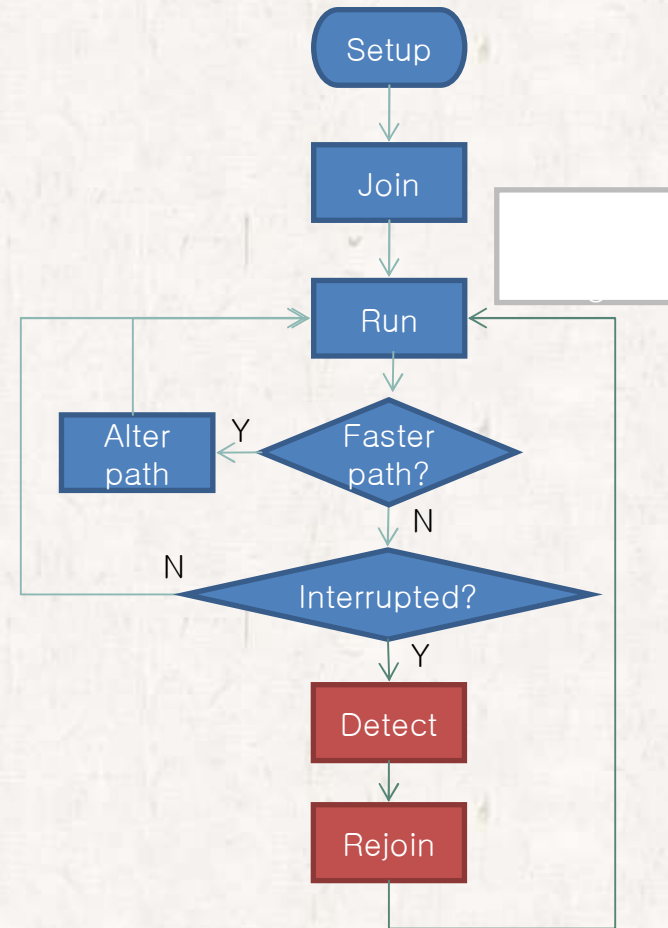
# Recovery Procedure



(a) Reactive



(b) Proactive



(c) Climber