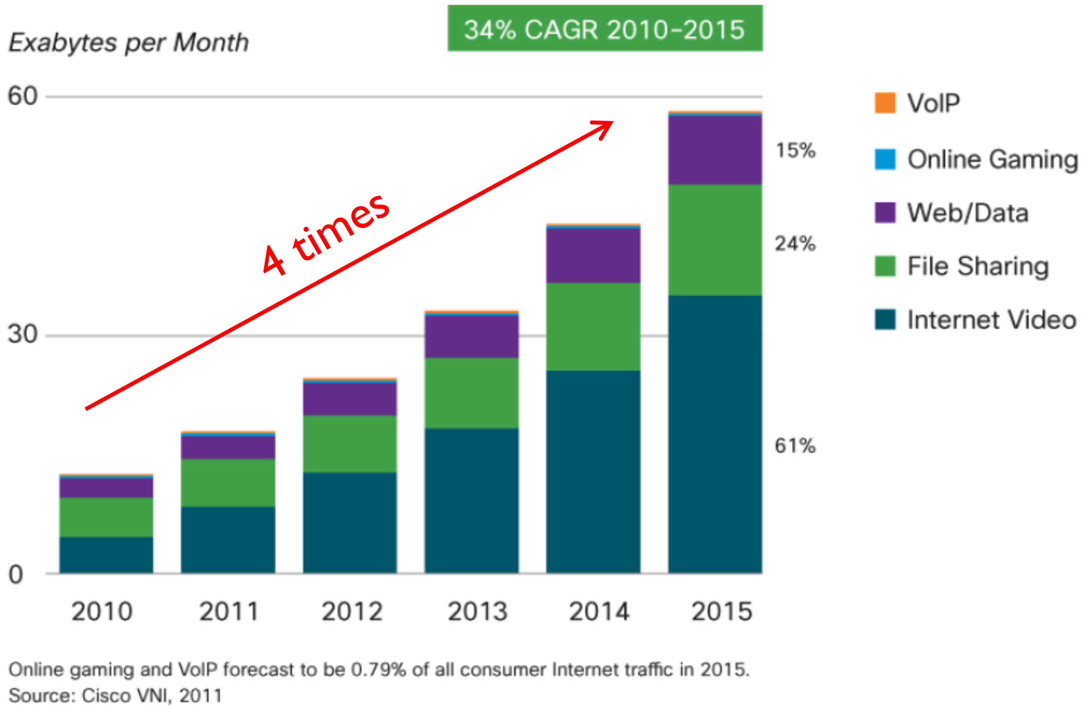
The Streaming Capacity of Sparsely-Connected P2P Systems with Distributed Control

Xiaojun Lin, Associate Professor
School of ECE, Purdue University

Joint work with Can Zhao (now at Qualcomm)
and Prof. Chuan Wu (HKU)

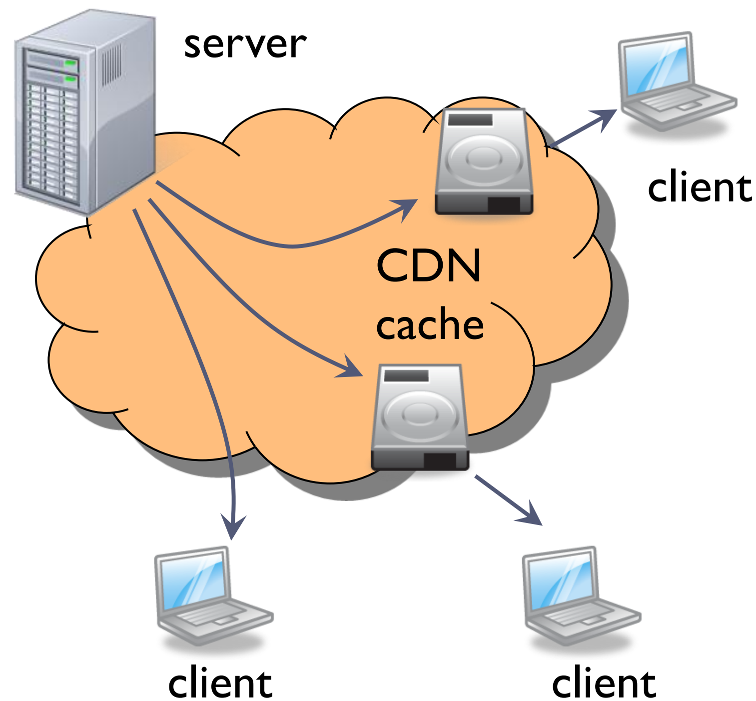
Significant Growth of Internet Video Traffic



- Internet video service providers: Youtube, Netflix, and many other.
- Consumer IP traffic will grow at a compound annual growth rate (CAGR) of 34%.
- By 2012, Internet video will account for over 50% of consumer Internet traffic.
- The sum of all video traffic (including TV, file sharing etc) will reach 90% of total IP traffic in 2015.

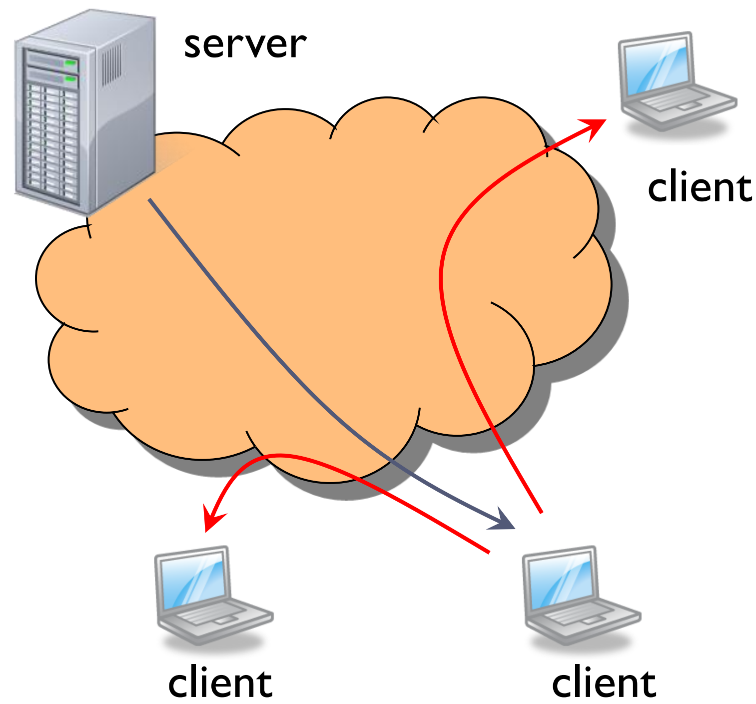
Video Streaming in the Internet: Current Status

- Video streaming directly from servers or CDNs (Content Distribution Networks) is costly
- In the US, this has been the dominant mode for video streaming
  - Credit Suisse estimated Youtube bandwidth cost in 2009 : $360M per year
  - Google likely paid significantly less due to peering with other ISP
- Licensing of video content could cost comparably or even more.
  - Netflix's lost deal with Starz: $300M for possibly a 5-year license



P2P Video Streaming: Current Status

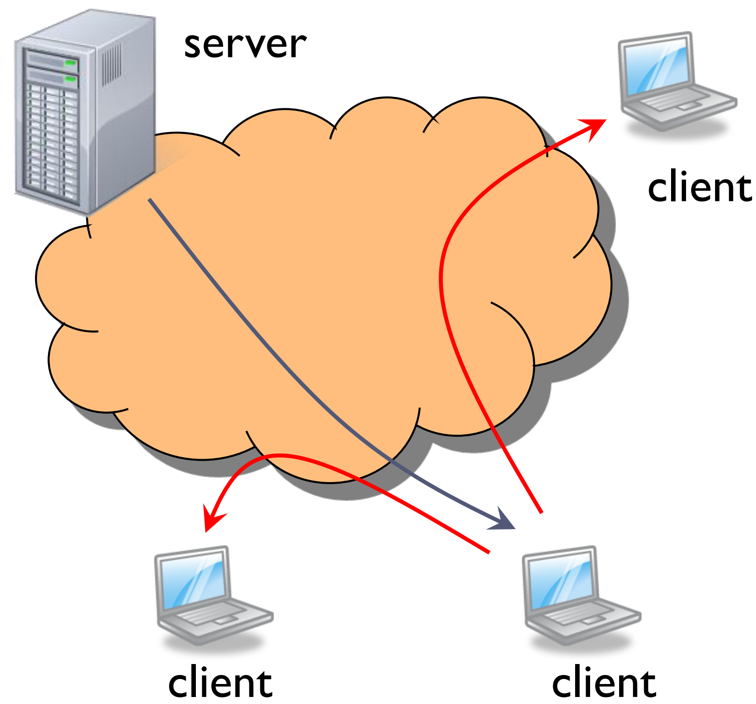
- Peer-to-peer (P2P) video streaming can potentially be much more scalable
  - Each client also contributes its upload capacity.
- In Asia, P2P streaming has become commercially successful
  - PPLive, UUSee, PPStream, etc.
- However, content has primarily been free or pirated
- High-value content appears to also move towards the server mode



Why hasn't P2P caught on yet for *high-value* content?

P2P Video Streaming: Issues

- Copyright issues?
- Lack of quality-of-service guarantees?
- Difficulty to maintain QoS in P2P systems:
  - Client upload capacity is time-varying
  - Peer "churns"
  - Large scale
  - Decentralized view and operation



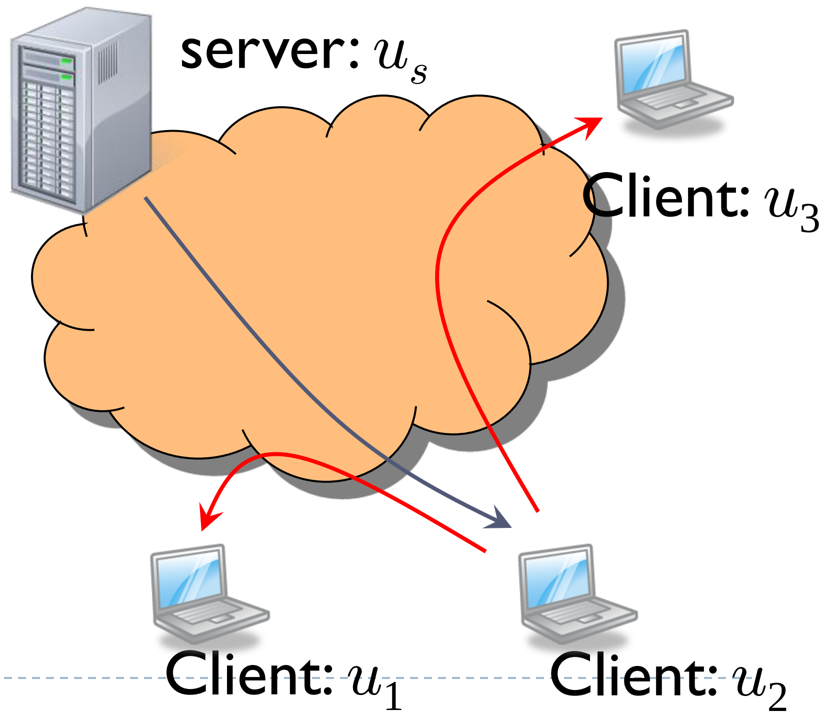
Why hasn't P2P caught on yet for *high-value* content?

Gap Between Practice and Theory

- Theoretical understanding of P2P streaming performance has significantly lag behind practice, which may have impeded further advance of P2P streaming.
- ***Our Focus:*** What is the best streaming rate C_f that a live-streaming P2P system can reliably support?
- Assuming upload capacity is the only constraint [Kumar et al '07]:

$$C_f \leq \min\{u_s, \boxed{\frac{u_s + \sum_{i=1}^{N} u_i}{N}}\}$$

- Question: Can this upper bound be attained?

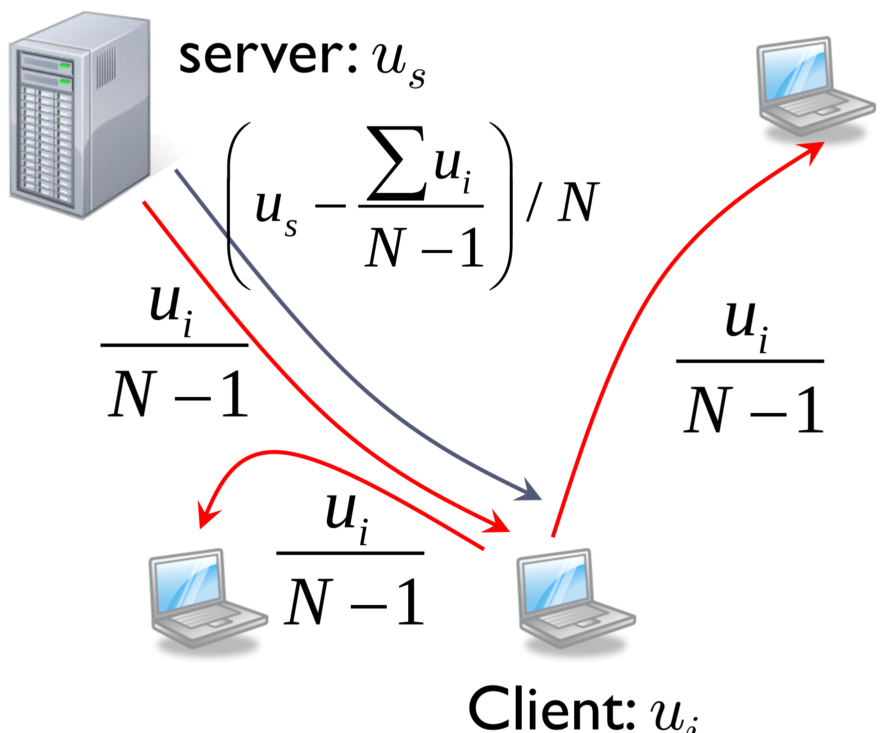


P2P Streaming Capacity

- Assume a complete graph: every peer can serve all other peers simultaneously [Mundinger et al '05, Chiu et al '06, Kumar et al '07]
- Each client gets

$$\left(u_s - \frac{\sum u_i}{N-1}\right)/N + \frac{u_i}{N-1} + \frac{\sum_{j\neq i} u_j}{N-1}$$

$$= C_f \triangleq \frac{u_s + \sum u_i}{N}$$



- How practical is this analysis?

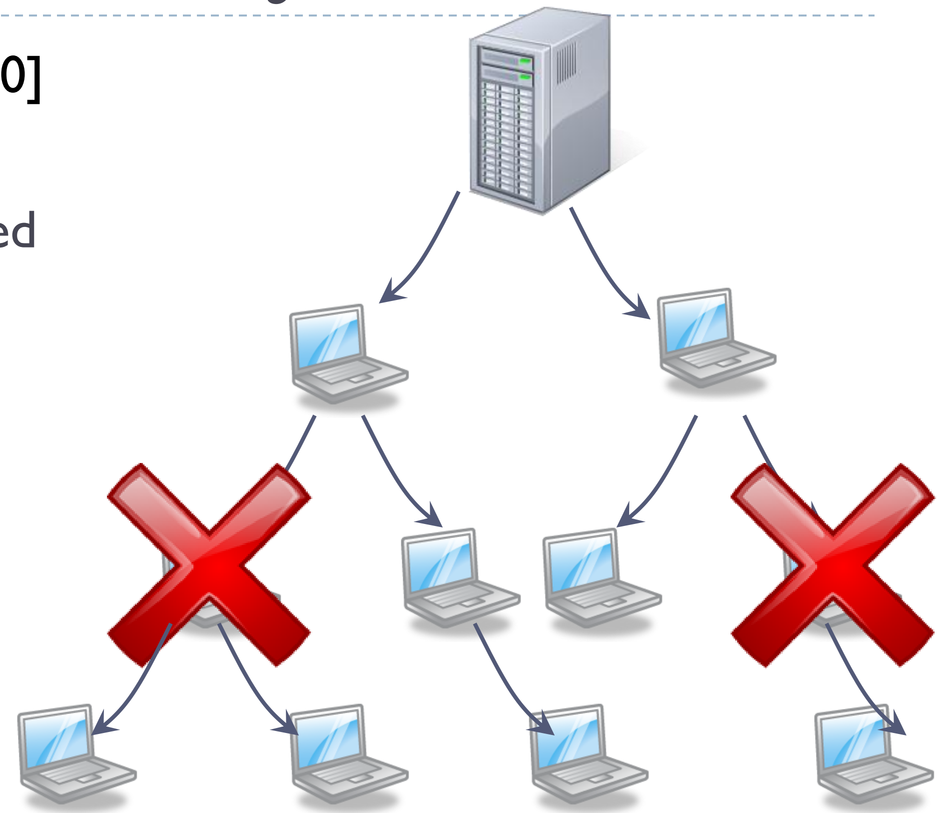
P2P Streaming Capacity

Such theoretical analysis is far from the reality in practical P2P systems!

- Real P2P streaming systems are sparsely connected:
  - Each peer only knows a small subset of other peers (neighbors).
  - Infeasible for each peer to know all other peers!
- Real P2P systems are distributed:
  - No central entity can have the global/up-to-date knowledge to perform such a perfect rate allocation.

Sparsely-Connected P2P Systems

- A multi-tree topology [Liu et al. 2010]
  - Still a centralized construction.
  - More recent work uses distributed Markov approximation [Zhang and Chen, 2012]
- If a peer close to the root leaves or its upload capacity decreases, significant performance disruption will occur.



Open Question:

- Can we achieve close-to-optimal streaming capacity with sparse connectivity and decentralized control that are robust against peer churns and variations of upload capacity ?

Our Contribution

- We show that a simple distributed scheme is sufficient to achieve close-to-optimal streaming capacity with high probability for large P2P systems.
  - Each peer has a small number of downstream neighbors $M = O(\log N)$
  - Each peer can choose neighbors uniformly randomly
  - Each peer evenly divides its upload capacity among the M neighbors
- Our results reveal important insights into the dynamics of large P2P systems.
- We design improved control schemes based on these insights that further improve the system performance.
- Our work provides an important step towards understanding and controlling QoS in large and unreliable P2P streaming systems.

Overview

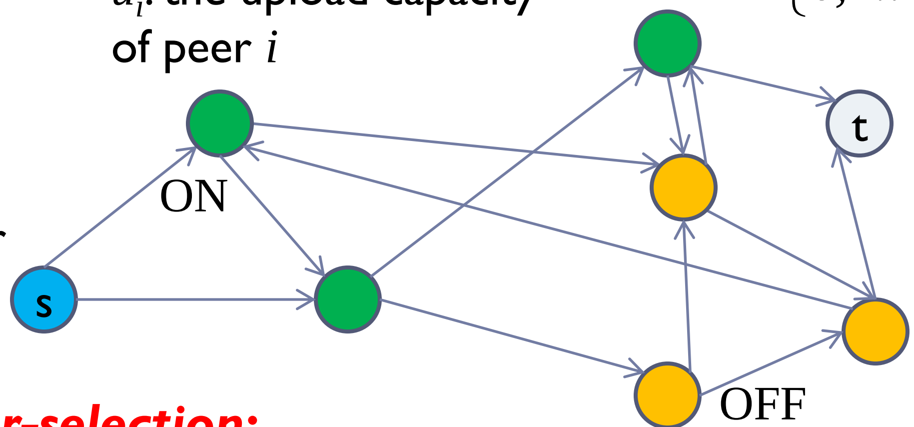
- ***System Model***
- Single-Channel: Uniform Rate Allocation
- Single-Channel: Adaptive Rate Allocation
- Multi-Channel Live Streaming
- Conclusion and Discussion

System Model: Single-Channel P2P Live Streaming with Random Peer Selection

N: number of peers
u_i: the upload capacity of peer i

$$u_i = \begin{cases} u, & \text{with prob. } p \\ 0, & \text{with prob. } 1-p \end{cases}$$

u_s: the upload capacity of server

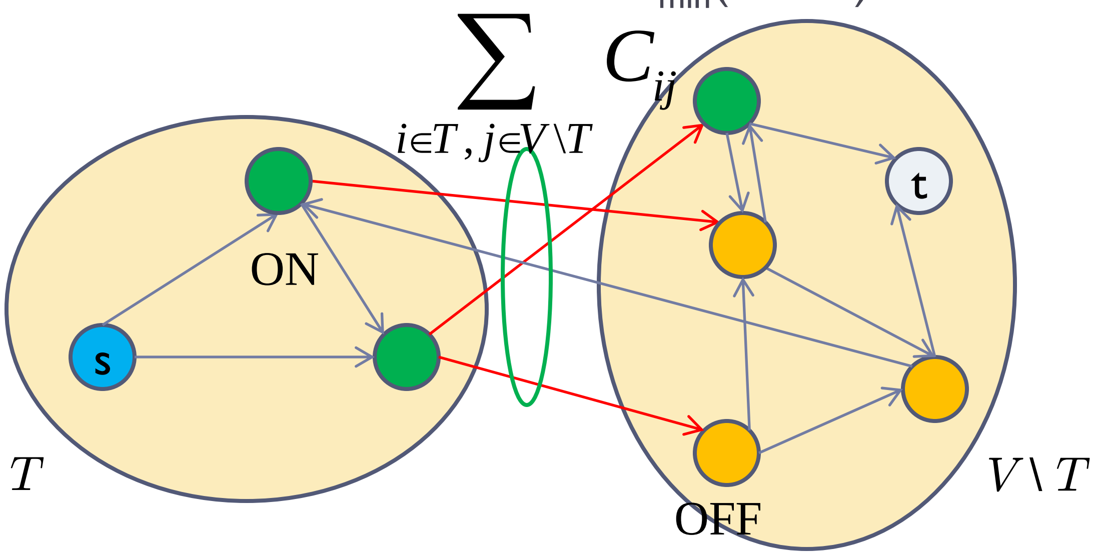


- ***Random peer-selection:***
  - Each peer randomly selects M downstream neighbors
  - Server randomly selects M ON peers as downstream neighbors
  - *Easy to implement and robust to peer churns.*
- C_{ij}: the capacity that peer i contributes to peer j.
- C_{ij}=0 if peer i is OFF or peer j is not a downstream neighbor of peer i

$$\sum_{j:\, i \to j} C_{ij} \le u_i \text{ and } \sum_{j:\, s \to j} C_{sj} \le u_s$$

System Model

- Streaming rate to destination peer t:
  - The minimum cut between s and t: $C_{\min}(s \to t)$



A directed capacitated graph

- The streaming rate of the entire system:
  - The minimum cut across all destination peers t:

$$\mathrm{C}_{\min-\min}\left(s \to V\right) = \min_{t \in V} \mathrm{C}_{\min}\left(s \to t\right)$$

  - Can be achieved in a distributed manner by network coding [Ahlswede et al 2000] or a latest-useful-chunk transmission policy [Massoulie and Twigg 2008]

Problem Statement

- C_f: The optimal streaming capacity assuming complete connectivity and centralized control

$$E[C_f] = \min\left\{u_s, \boxed{\frac{u_s + upN}{N}}\right\}$$

- ***Research Problems:***
  - How much performance penalty (compare to the optimal C_f) is incurred due to random peer-selection?
  - Are there simple and robust rate-allocation schemes that can achieve close-to-optimal capacity with minimal overhead?

Overview

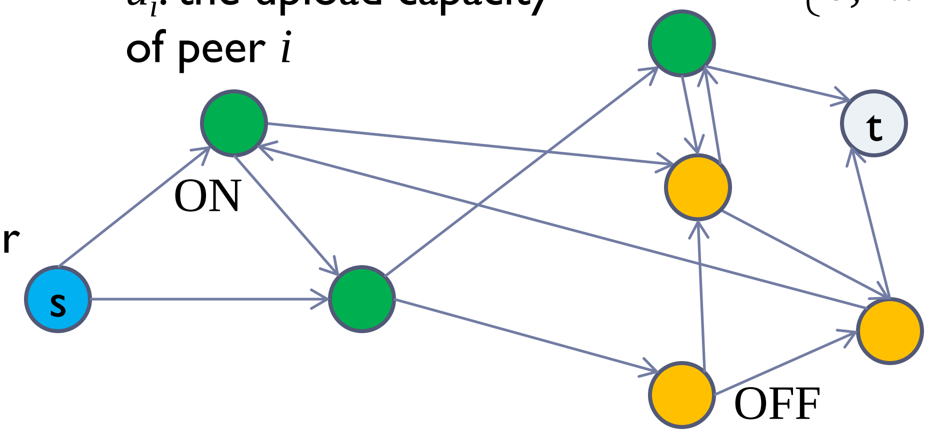
- System Model
- ***Single-Channel: Uniform Rate Allocation***
- Single-Channel: Adaptive Rate Allocation
- Multi-Channel Live Streaming
- Conclusion and Discussion

Single-Channel: Uniform Rate-Allocation

N: number of peers
u_i: the upload capacity of peer i

$$u_i = \begin{cases} u, & \text{with prob. } p \\ 0, & \text{with prob. } 1-p \end{cases}$$

u_s : the upload capacity of server



- ***Uniform rate-allocation***:
  - Each peer evenly divides its upload capacity to its M downstream neighbors
  - Same for server
- C_{ij}: the link capacity from peer i to peer j.

$$C_{ij} = \begin{cases} u/M, & \text{if peer } i \text{ is ON and } i \rightarrow j \\ 0, & \text{otherwise} \end{cases}$$

Main Result

- For any $\varepsilon \in (0,1)$ and $d > 1$, there exists $\alpha > 0$ such that if $M = \alpha \log N$, then

$$P(C_{\min-\min}(s \to V) \leq (1-\varepsilon)E[C_f]) \leq O\left(\frac{1}{N^{2d-1}}\right)$$

- Even with simple random peer-selection and uniform rate-allocation, the system can achieve close-to-optimal streaming capacity with very high probability!

Implications

- Sparse connectivity is sufficient!
  - $M = \alpha \log N$
- Simple and decentralized control
  - Random peer selection and uniform rate-allocation
- Larger is better!
  - The larger the network size, the easier to achieve close-to-optimal capacity
- Robustness
  - Even if a peer leaves, only its upstream peer needs to re-select a downstream neighbor.
  - When a peer switches from ON to OFF, its neighbors do not need to change anything (unless it is connected to the server directly).
  - No need to reconstruct the global topology!

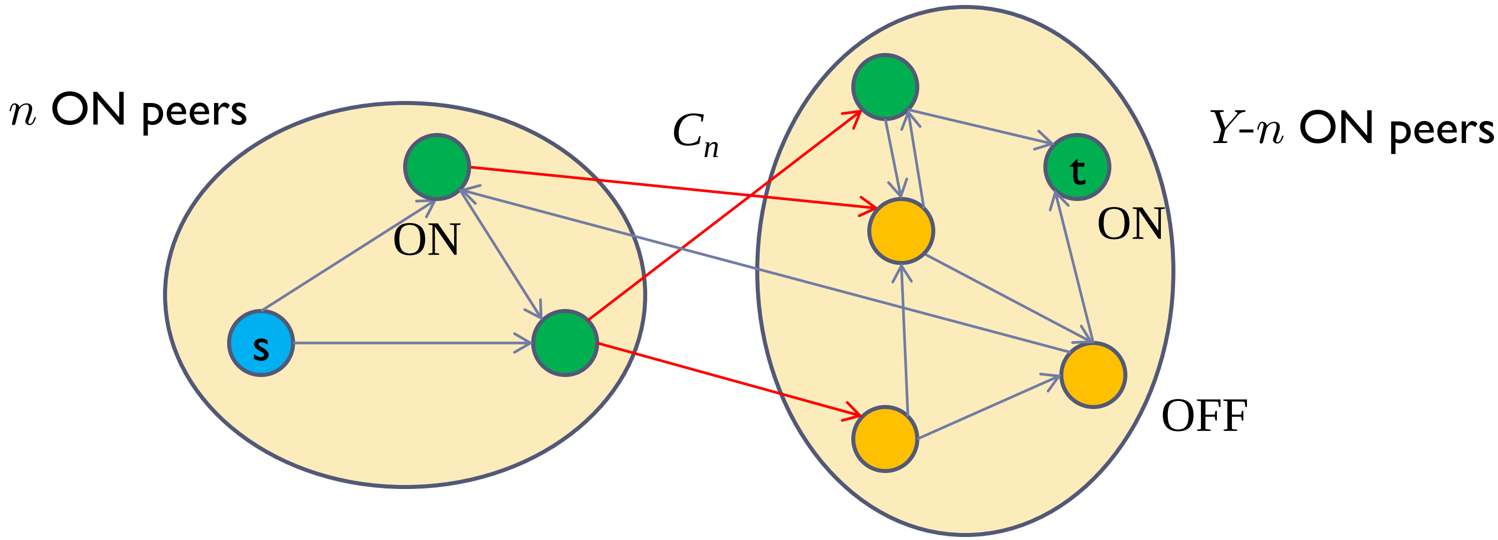
Large P2P streaming systems are in fact extremely

scalable

and

robust!

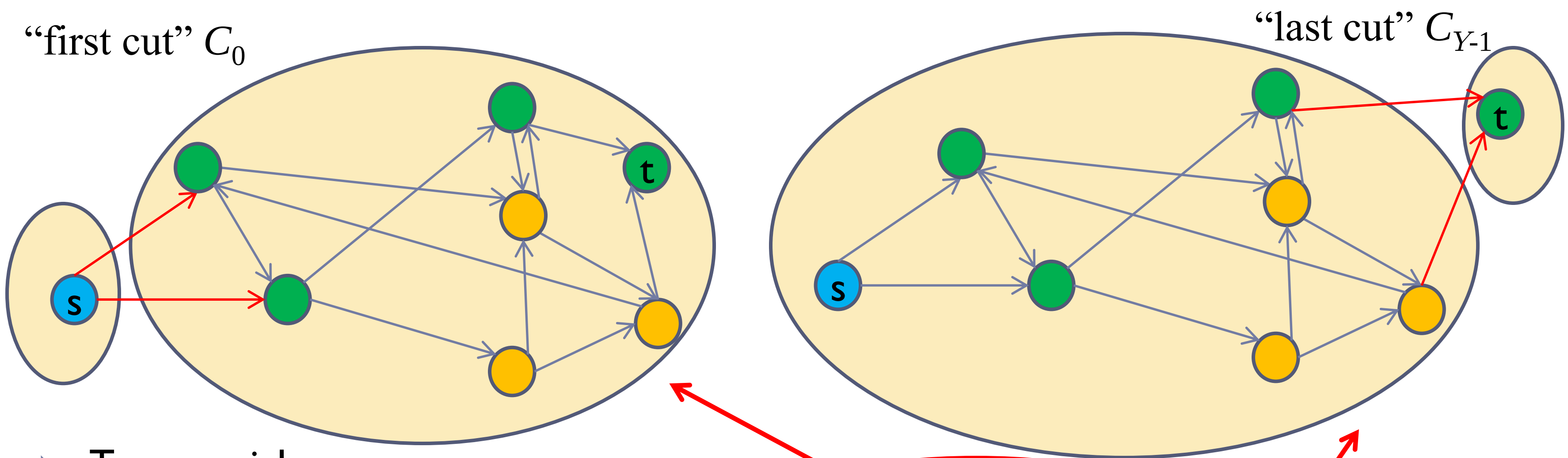
Intuition Behind the Main Result



- Fix a destination peer t. Suppose that peer t is ON.
- Let Y be the total number of ON peers: $Y \approx Np$
- C_n : the random capacity of a cut that has n ON peers on the server side

$$\mathbf{E}[C_n \mid Y] \geq \frac{u_s(Y-n)}{Y} + \frac{un(Y-n)}{N} = \begin{cases} u_s, & \text{if } n = 0 \\ \dfrac{u_s + upN}{N}, & \text{if } n = Y-1 \end{cases}$$

Intuition Behind the Main Result



- Two special cases:

$$E[C_n \mid Y = Np] \geq \begin{cases} u_s, & \text{if } n = 0 \\ \dfrac{u_s + upN}{N}, & \text{if } n = Y - 1 \end{cases}$$

- For all other n, we have

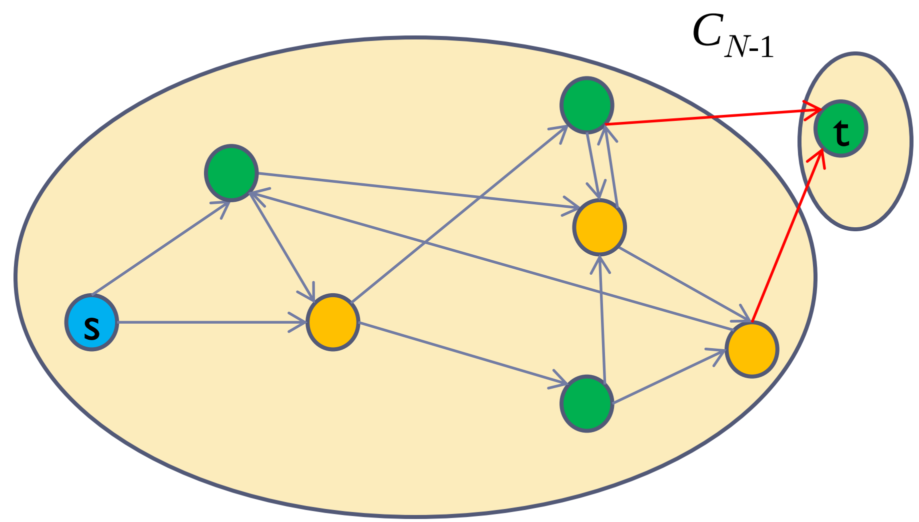
$$E[C_n] > E[C_f] \triangleq \min\{u_s, \boxed{\frac{u_s + upN}{N}}\}$$

- Since the number of edges in a cut, **$nM(Y\text{-}n)/N$,** is large when M and N increase, the capacity of any cut C_n should be no less than $(1-\varepsilon)$ C_f with high probability

Insights for P2P Protocol Design

The most critical cut is the last cut $C_{N\text{-}1}$

- The probability that $C_{N\text{-}1}$ fails (less than (1-ε) of optimal streaming rate C_f) is much larger than the probability that any other cut fails



Two main reasons:

- The expected capacity $\mathbf{E}[C_{N\text{-}1}]$ is the smallest
- The expected number of edges is the smallest: $nM(N\text{-}n)/N = M$ when $n=N\text{-}1$

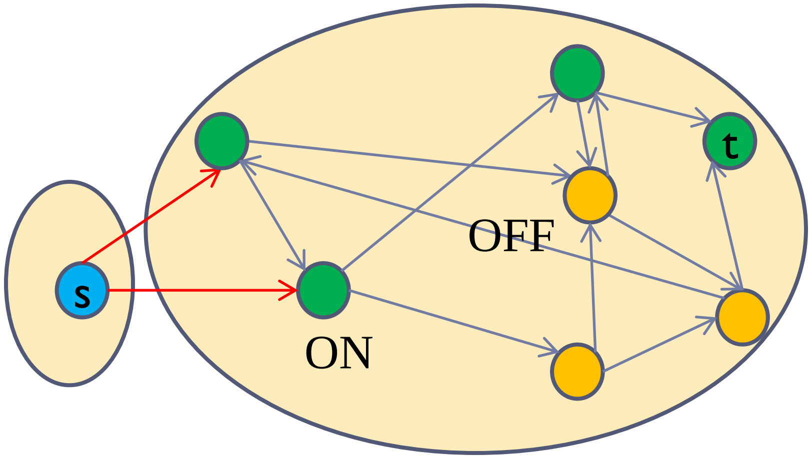
$$E[C_{N-1} \mid Y = Np] = \frac{u_s + upN}{N} = E[C_f]$$

Improved P2P control scheme should focus on improving the capacity of the last cut

Insights for P2P Protocol Design

ON/OFF status of each peer's upload capacity:

- A common wisdom is that peers close to the server should choose ON peers as downstream neighbors
- Our analysis indicates that only the server needs to be careful choosing ON peers.



Low-overhead P2P control scheme could focus on peer-selection and recovery at the server only.

Insights for P2P Protocol Design

Number of neighbors that each peer needs: $M = \alpha \log N$

- In order that

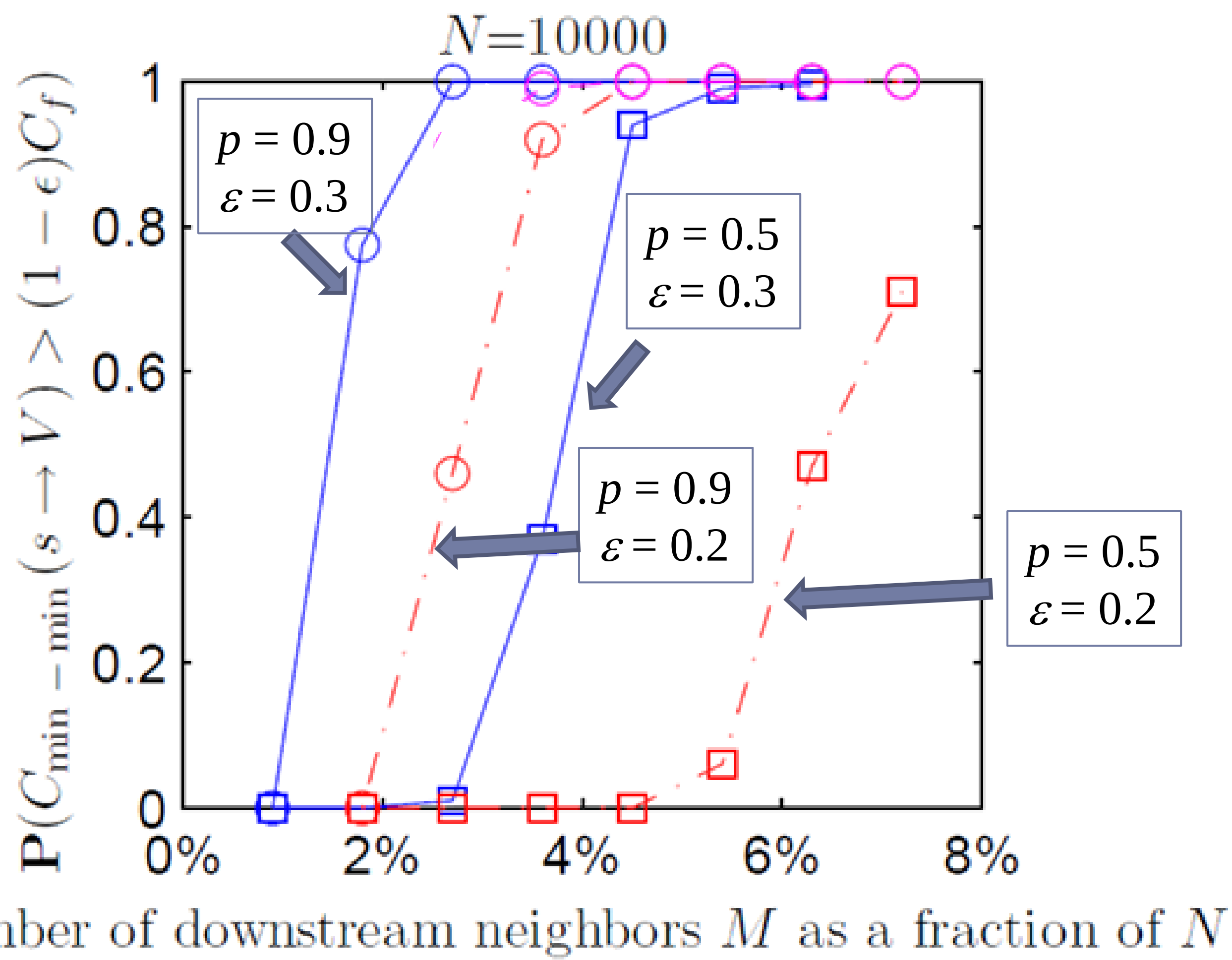
$$P(C_{\min-\min}(s \to V) \leq (1-\varepsilon)E[C_f]) \leq O\left(\frac{1}{N^{2d-1}}\right)$$

the constant α must be

$$\boxed{\alpha \geq \frac{4du_s}{pu\varepsilon^2}}$$

- Require a larger number of neighbors when
  - faster convergence rate (larger d)
  - fewer high bandwidth users (smaller p)
  - higher streaming rate requirement (smaller ε)
- This factor may be further reduced by improving the capacity of the last cut

Simulation Result – Single Channel



Overview

- System Model
- Single-Channel: Uniform Rate Allocation
- ***Single-Channel: Adaptive Rate Allocation***
- Multi-Channel Live Streaming
- Conclusion and Discussion

Adaptive Rate Allocation: Motivation

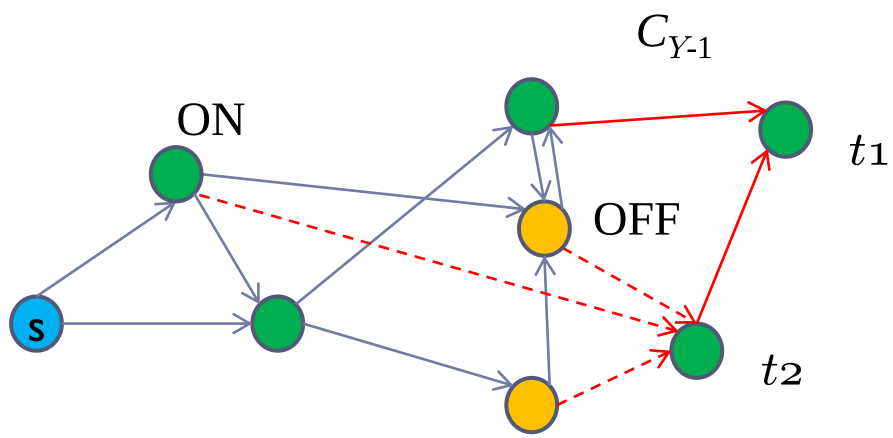
- Number of neighbors each peer needs $M = \alpha \log N$

$$\alpha \geq \frac{4du_s}{pu\varepsilon^2}$$

  - As $\epsilon \to 0$, α increases inversely proportional to ϵ^2
  - The number of neighbors of each peer can still be quite large
- Recall that the most critical cut is the last cut $C_{N\text{-}1}$
  - The capacity that each peer receives directly from its immediate upstream neighbors

- Will improving the capacity of the last cut reduce α?

From Uniform to Adaptive Rate Allocation



Uniform rate-allocation:

- Each edge from an ON peer contributes u/M capacity
- However, the number of such edges to a peer is random

- ***Adaptive rate-allocation***: Balance the capacity of the last cut by carefully assigning C_{ij} (the upload rate from peer i to peer j)

$$\boxed{\sum_{j:\, i \to j} C_{ij} \le u_i, \text{ for all } i,} \quad \text{and} \quad \boxed{\sum_{i:\, i \to j} C_{ij} \ge (1-\varepsilon) C_f, \text{ for all } j}$$

Caveat: no capacity guarantee on all other cuts!

Uniform versus Adaptive Rate Allocation: Pros and Cons

- Uniform rate allocation
  - The last cut is the most difficult
  - Other cuts have larger capacity
- Adaptive rate allocation
  - Balance the capacity for the last cut
  - No capacity guarantee for other cuts

- Hybrid scheme
  - Reserve a fraction of the upload capacity of each peer for uniform rate allocation
  - Perform adaptive rate allocation with the remaining upload capacity

Hybrid Scheme - Details

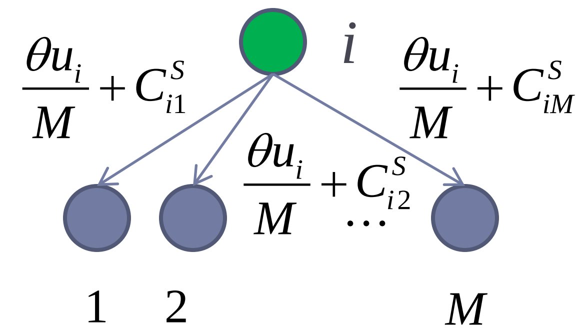
- Still perform *random peer-selection*
- Each link capacity C_{ij} consists of two parts

$$C_{ij} = C_{ij}^U + C_{ij}^S$$

- Reserve a fraction θ of the upload capacity for uniform allocation

$$C_{ij}^U = \theta u_i / M$$

  - Take care of all other cuts with high probability



- The capacity C_{ij}^S for adaptive rate allocation is given by the the solution of

$$\sum_{j:i \to j} C_{ij}^S \le (1-\theta)u_i, \text{ for all } i$$

$$\sum_{i:i \to j} C_{ij}^U + C_{ij}^S \ge (1-\varepsilon)C_f, \text{ for all } j$$

  - The solution exists with high probability $\Rightarrow$ take care of the last cut.
  - There exist fully-distributed algorithms to compute the solution.

Hybrid Scheme - Main Result

- For $0.5 < \theta < 1$, the hybrid scheme could achieve a close-to-optimal streaming rate with high probability

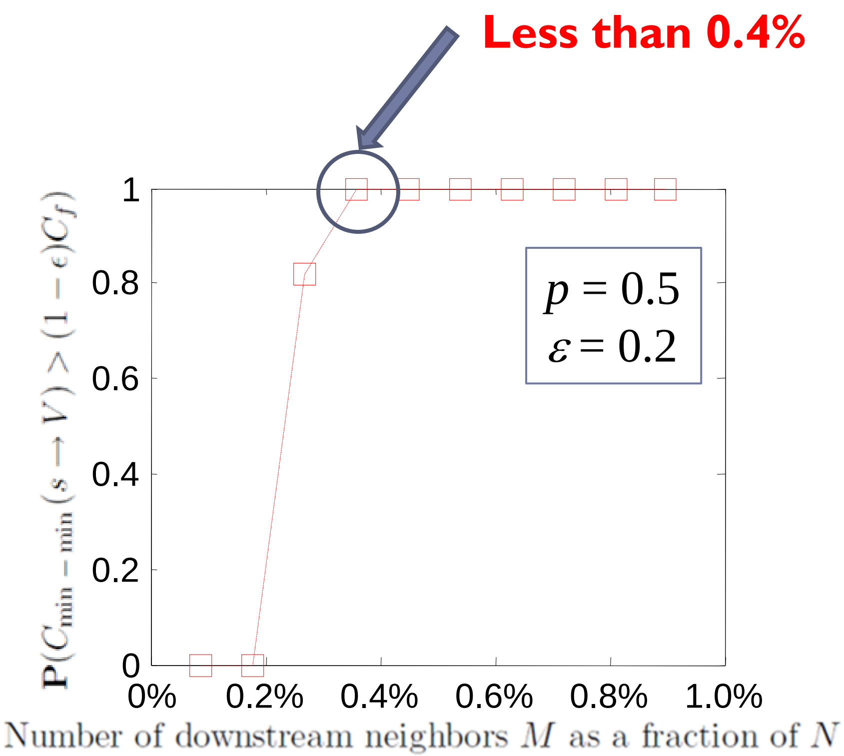
$$P(C_{\min-\min}(s \rightarrow V) \leq (1-\varepsilon)E[C_f]) \leq O\left(\frac{1}{N^{2d-1}}\right)$$

if

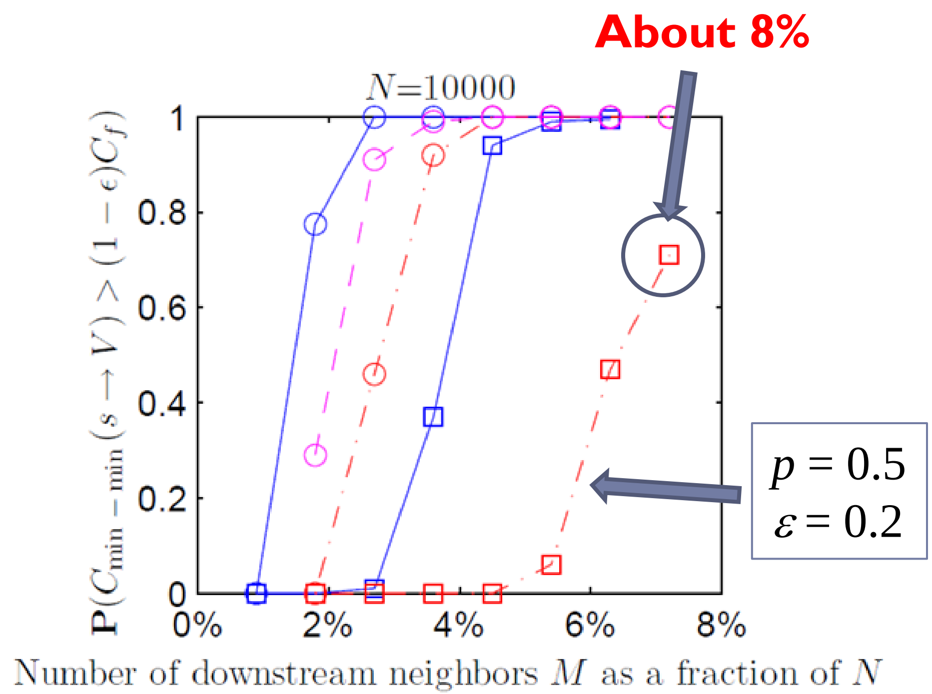
$$\alpha \geq \max\left\{\underbrace{\frac{2d}{\max\left\{\frac{\varepsilon^2}{2}, \frac{(2\theta-1)^2}{8\theta^2}\right\}p}}_{\text{For other cuts}}, \underbrace{\frac{16d}{(1-\varepsilon)p}}_{\text{For the last cuts}}\right\}\frac{u_s}{u}$$

- The dependency on small ε is virtually eliminated!
- Assume $\theta = 0.9$, $\varepsilon = 0.1$
  - Uniform rate allocation: $\alpha > 400 du_s/pu$
  - Hybrid Scheme: $\alpha > 17.8 du_s/pu$

Hybrid Scheme – Simulation Result



Hybrid (N=10000)



Uniform (N=10000)

Overview

- System Model
- Single-Channel: Uniform Rate Allocation
- Single-Channel: Adaptive Rate Allocation
- ***Multi-Channel P2P Live Streaming***
- Conclusion and Discussion

Multi-Channel P2P networks

- Existing P2P systems typically serve a large number of channels/videos at the same time
- Traditionally, each channel is treated separately
  - Peers viewing a channel only serve other peers in the same channel
- View-Upload Decoupling (VUD [Wu et. al. 2009])
  - Peers can view one channel but serve/upload videos for peers in a different channel
  - Streaming capacity for multi-channel P2P system is improved
  - Still assume complete connectivity and centralized operation
- Our work
  - We propose a simple distributed scheme that has a similar flavor of VUD
  - Close-to-optimal streaming capacity region can still be achieved for multi-channel systems

Multi-channel - System Model

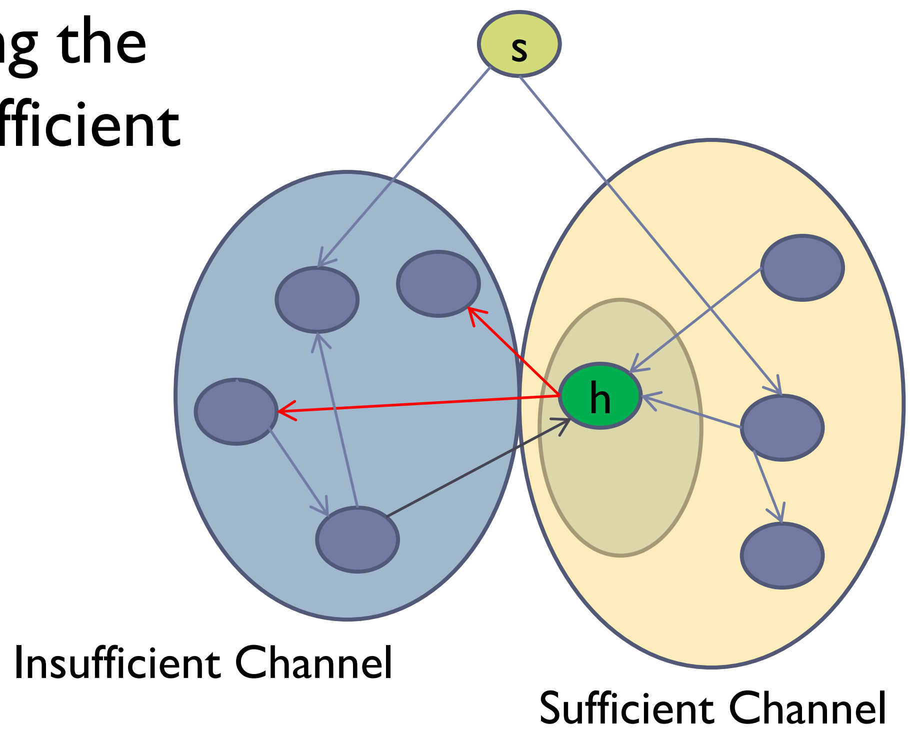
- Consider a multi-channel P2P system with J different channels
- $\mathcal{N}_j$: the set of peers that are viewing channel j
  - $N_j = |\mathcal{N}_j|$: The number of peers in channel j
- $u_{s,j}$: the capacity that server allocates to channel j
- R_j: the targeted streaming rate of channel j
- For each single channel, the optimal achievable streaming rate is

$$C_{f,j} = \min\left\{ u_{s,j}, \frac{u_{s,j} + \sum_{i \in \mathcal{N}_j} u_i}{N_j} \right\}$$

- For a given $\mathbf{R}$,
  - some channels may have a $C_{f,j} > R_j$, ➡ sufficient channels
  - some channels may have a $C_{f,j} < R_j$ ➡ insufficient channels

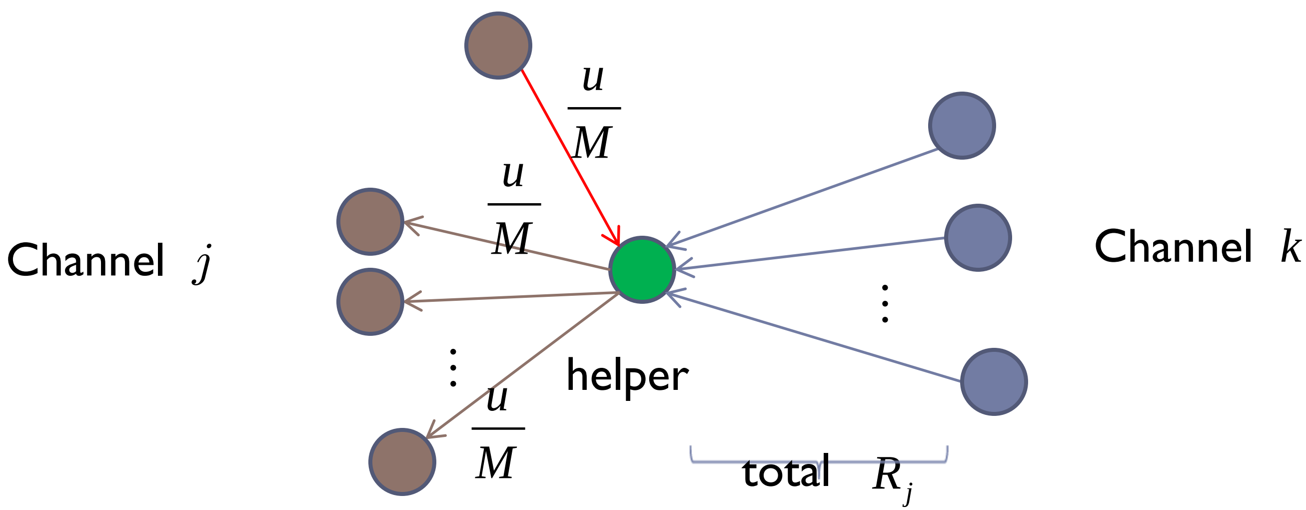
Multi-channel – View-Upload Decoupling

- VUD: Some peers from the sufficient channels become helpers to help improving the performance of the insufficient channels



Multi-channel - Helper

- A helper that is viewing channel k and helping channel j
  - Receives full streaming rate R_k of the content of channel k
  - Must be ON
  - Receives a rate u/M of the content of channel j
  - All of its downstream neighbors are peers viewing channel j



Multi-channel: Capacity Region

- R_j : the targeted streaming rate of channel j
- We can define the capacity region Λ as the set of streaming rate vectors $\mathbf{R} = [R_1, R_2, \ldots, R_J]^{\mathrm{T}}$ such that any $\mathbf{R} \in \Lambda$ is supportable by some control algorithm.
- The largest possible capacity region

$$\Lambda = \left\{ \mathbf{R} \,\middle|\, \sum_{j=1}^{J} N_j R_j \le u_s + \sum_{i \in V} u_i, \sum_{j=1}^{J} R_j \le u_s \right\}$$

Total capacity demand

Total upload capacity

- Given $\mathbf{R} \in (1-\epsilon)\,\Lambda$, can this rate vector $\mathbf{R}$ be achieved by a simple and distributed control scheme?

Multi-channel - Algorithm

- Let H_j be the number of helpers that channel j needs
- We would like to choose H_j so that

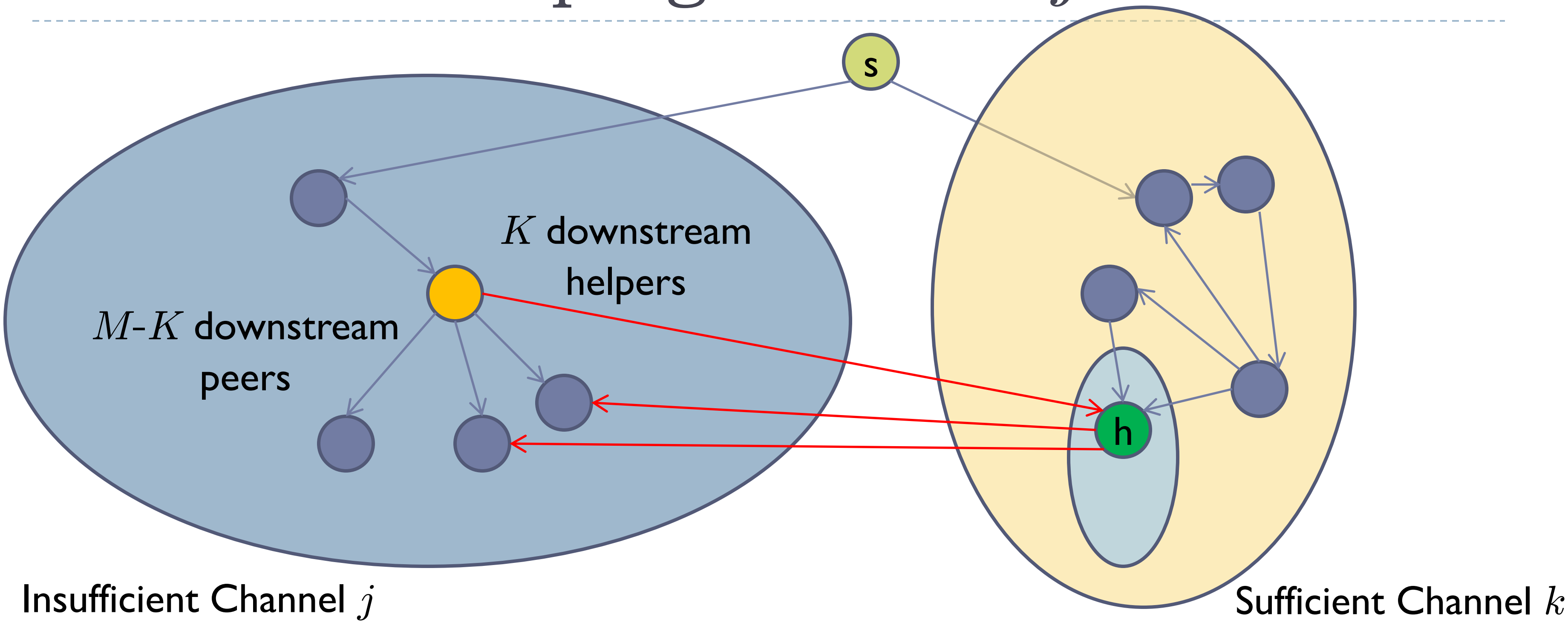
$$C_{f,j} = \min\left\{u_{s,j}, \frac{u_{s,j} + \sum_{i\in\mathcal{N}_j} u_i + \sum_{i\in\mathcal{H}_j} u_i}{N_j}\right\} = \frac{R_j}{1-\varepsilon}$$

- One solution:

$$H_j = \left\lfloor \frac{N_j R_j}{(1-\varepsilon)u} - \frac{u_{s,j}}{u} - pN_j \right\rfloor$$

  - $H_j > 0$ for an insufficient channel (needing helpers)
  - $H_j < 0$ for a sufficient channel (providing helpers)

Channel k Helping Channel j



- Each helper behaves like an OFF peer in channel k
- Each ON peer in channel j reserves K (downstream) slots for helpers
- Each helper finds a normal ON peer randomly from channel j as its upstream neighbor
- Each helper picks M downstream peers randomly from channel j
- Uniform rate allocation. Helpers do not connect to helpers

Multi-channel – Main Result

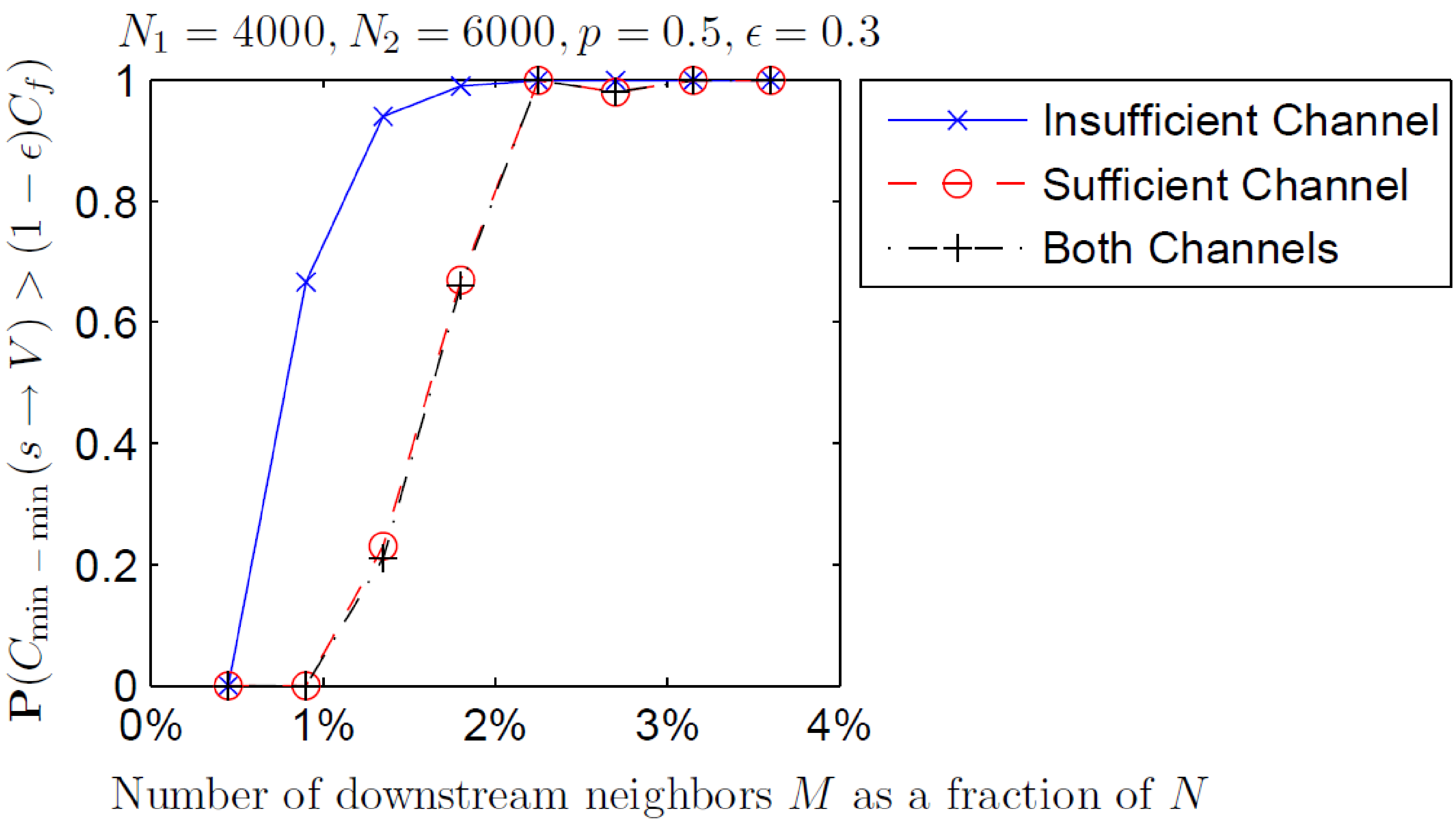
- For any $\varepsilon > 0$, $d > 1$ and $\mathbf{R} \in (1-\varepsilon)\Lambda$, there exists α such that if $M = \alpha \log N$, then for all channel j

$$P(C_{\min-\min}(s \to \mathcal{N}_j) \le R_j) \le O\left(\frac{1}{N^{2d-1}}\right)$$

  - $\mathcal{N}_j$: The set of peers that are viewing channel j
  - Λ is the largest possible capacity region.

- Our proposed scheme can achieve close-to-optimal capacity with sparse connectivity and decentralized control.
  - Each peer still needs only $O(\log N)$ neighbors
  - Helpers are chosen randomly

Simulation Result – Multi-Channel



Overview

- System Model
- Single-Channel: Uniform Rate Allocation
- Single-Channel: Adaptive Rate Allocation
- Multi-Channel Live Streaming
- ***Conclusion and Discussion***

Summary

- Close-to-optimal streaming capacity can be achieved with high probability using
  - $O(\log N)$ downstream neighbors for each peer
  - Random peer-selection
  - Uniform rate-allocation
- Our results reveal important insights into the dynamics of large P2P streaming systems
- Based on these insights, we design a hybrid scheme that further improves the system performance.
- With "helpers", a similarly simple scheme could also achieve a close-to-optimal streaming capacity region for multi-channel P2P systems

On-Going and Future Work

- P2P Video-on-Demand (VoD) Systems
  - Timing of each neighbor is important
  - Users may jump forward/backward
  - Cache placement policy is also critical
  - We show that simple and distributed control with sparse-connectivity will still suffice

On-Going and Future Work

- The packet scheduling problem
  - May use generation-based random linear network coding.
  - There is a tradeoff between rate, delay, and overhead
  - BATS code?
- Incorporating scalable video
  - Video encoding rate may be adjusted based on the optimal streaming rate
  - Layered video
- Multiple ISPs
  - Cross-ISP traffic may encounter new bottlenecks
  - Will random peer-selection and simple rate-allocation strategies still be sufficient?
- Wireless versus wireline

Thank You!

Can Zhao, Xiaojun Lin and Chuan Wu ``The Streaming Capacity of Sparsely-Connected P2P Systems with Distributed Control," in *IEEE INFOCOM*, Shanghai, China, April 2011

https://engineering.purdue.edu/~linx