Tree-structured Data Regeneration in Distributed Storage Systems with Regenerating Codes

Jun Li[1], Shuang Yang[1], Xin Wang[1], Baochun Li[2]
[1]School of Computer Science, Fudan University, China
[2]Department of Electrical and Computer Engineering, University of Toronto, Canada

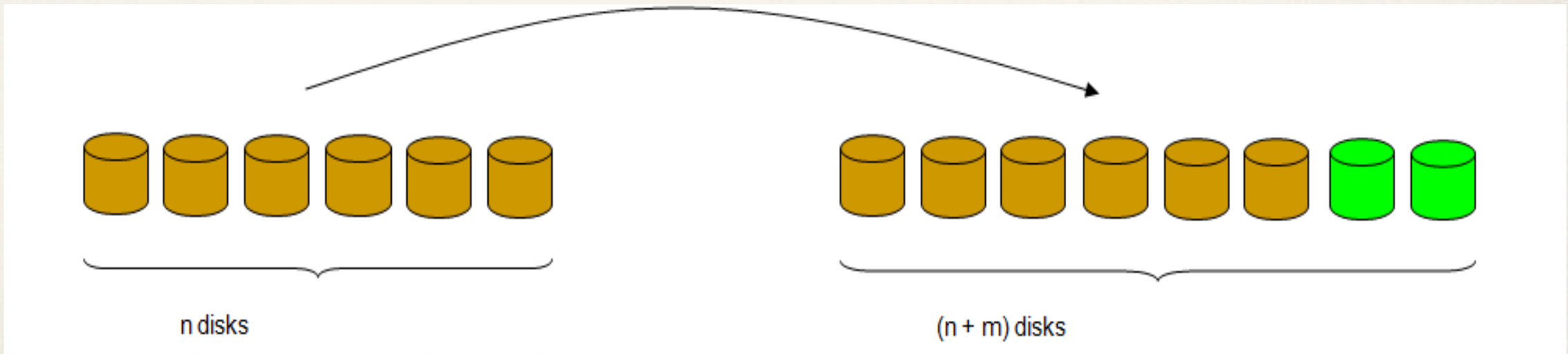
Mar. 18, 2010

Distributed Storage

- **Large-scale data storage service**
  - Data are stored in a distributed fashion, *i.e.*, in a number of storage nodes in the network
- **Disk failures are common**
  - There are so many disks in large distributed storage systems, *e.g.*, data centers
  - The permanent data loss then becomes very likely
- **We address this issue in this presentation**

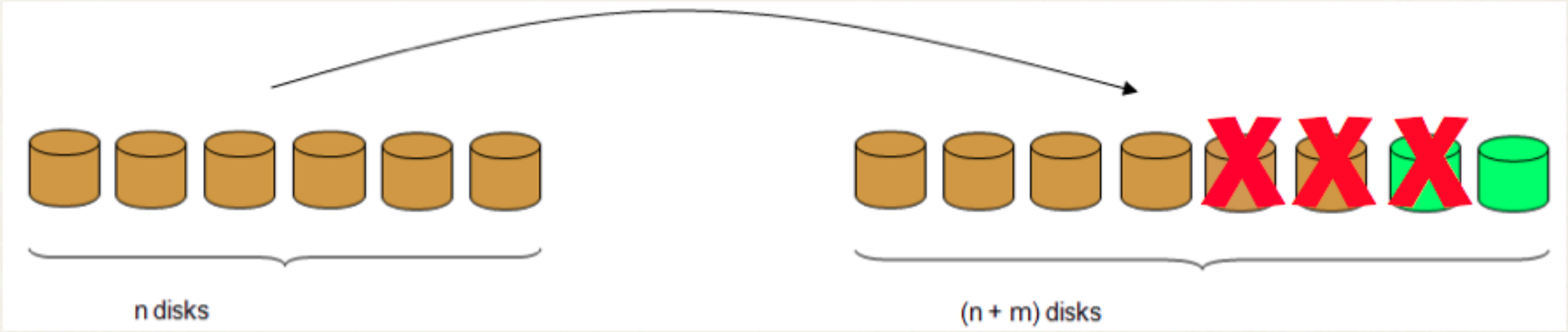
Redundancy

- **To compensate for disk failures, redundancy is introduced in the distributed storage system**
  - **Replicas**
  - **Erasure codes**



Data Regeneration

- **Redundancy is not enough.**
  - Using erasure codes, data become not available if more than *m* disk failures occur

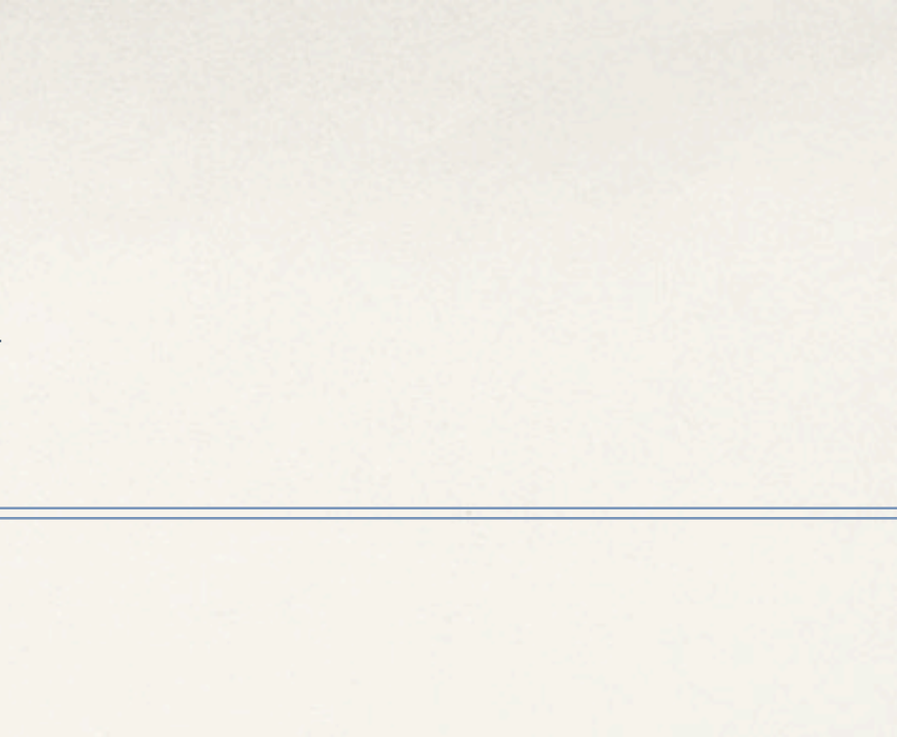


- **Newcomer and providers**
  - To repair the lost redundancy, a newcomer receives redundant data from a certain number of providers

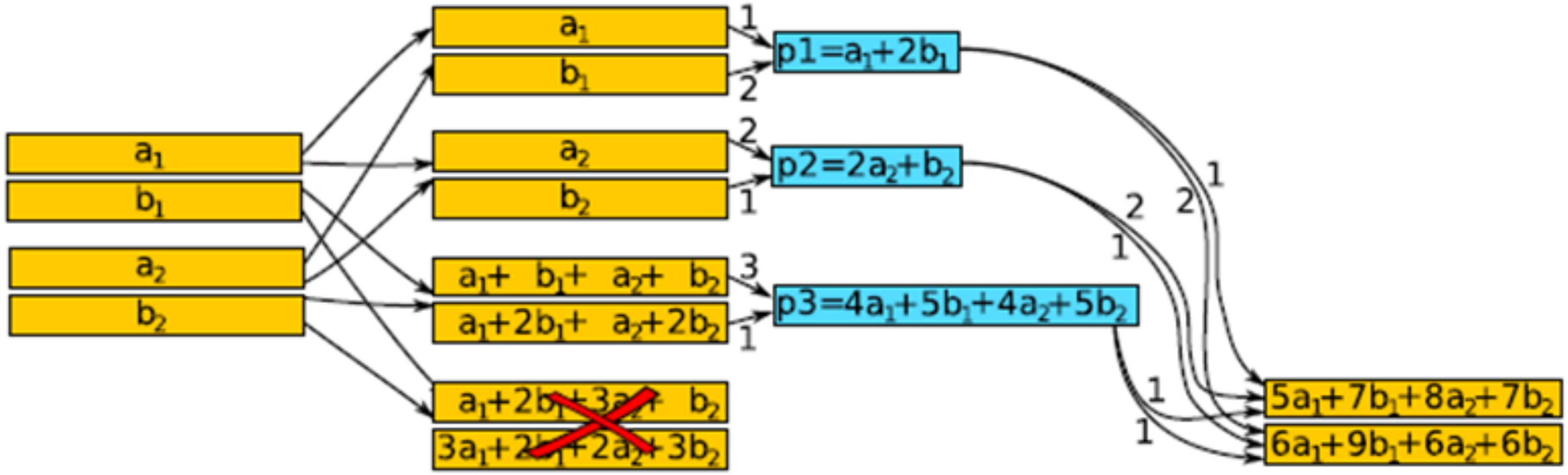
Data Regeneration

- **Redundancy is not enough.**
  - Using erasure codes, data become not available if more than *m* disk failures occur

- **Newcomer and providers**
  - To repair the lost redundancy, a newcomer receives redundant data from a certain number of providers



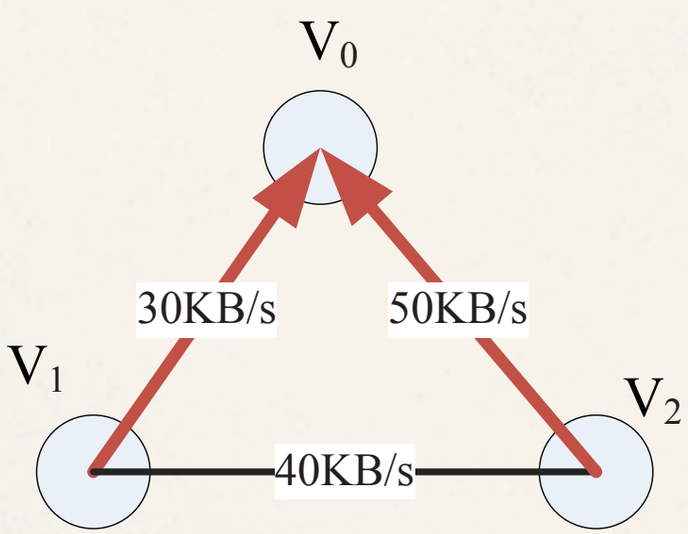
Network Coding [Dimakis07]



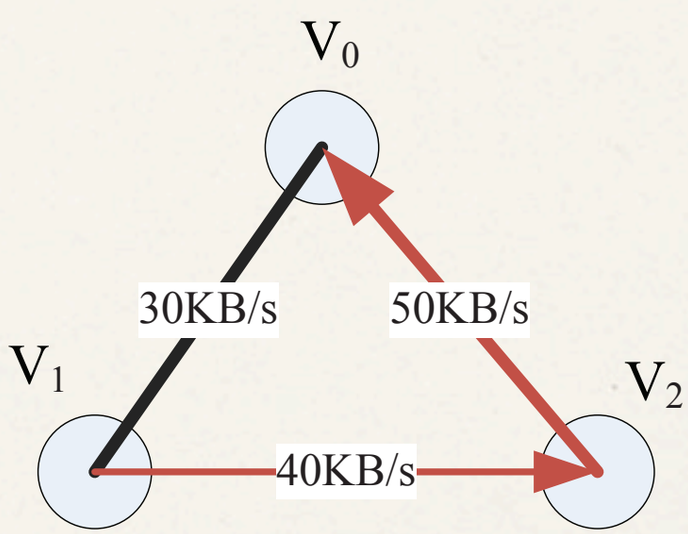
- **This paper addressed how to regenerate the lost data by transmitting data as little as possible**
- **More providers, less regeneration traffic**

Our Observation

- **An efficient regeneration should not only consider how to achieve the optimal regeneration traffic**



(a) Star-structured Regeneration Scheme



(b) Tree-structured Regeneration Scheme

- **Another factor: end-to-end available bandwidth**

Our Contributions

- **Consider how to route regeneration flows to achieve the optimal regeneration time**
  - A tree-structured regeneration scheme with linear network coding
    - [Li09] J. Li, S. Yang, X. Wang, X. Xue, B. Li, "*Tree-structured Data Regeneration with Network Coding in Distributed Storage Systems*," in Proc. of International Workshop on Quality of Service (IWQoS), 2009
- **Combine regenerating codes and the tree-structured regeneration**
- **Analyze the stability when providers may quit during the regeneration**

Our Contributions

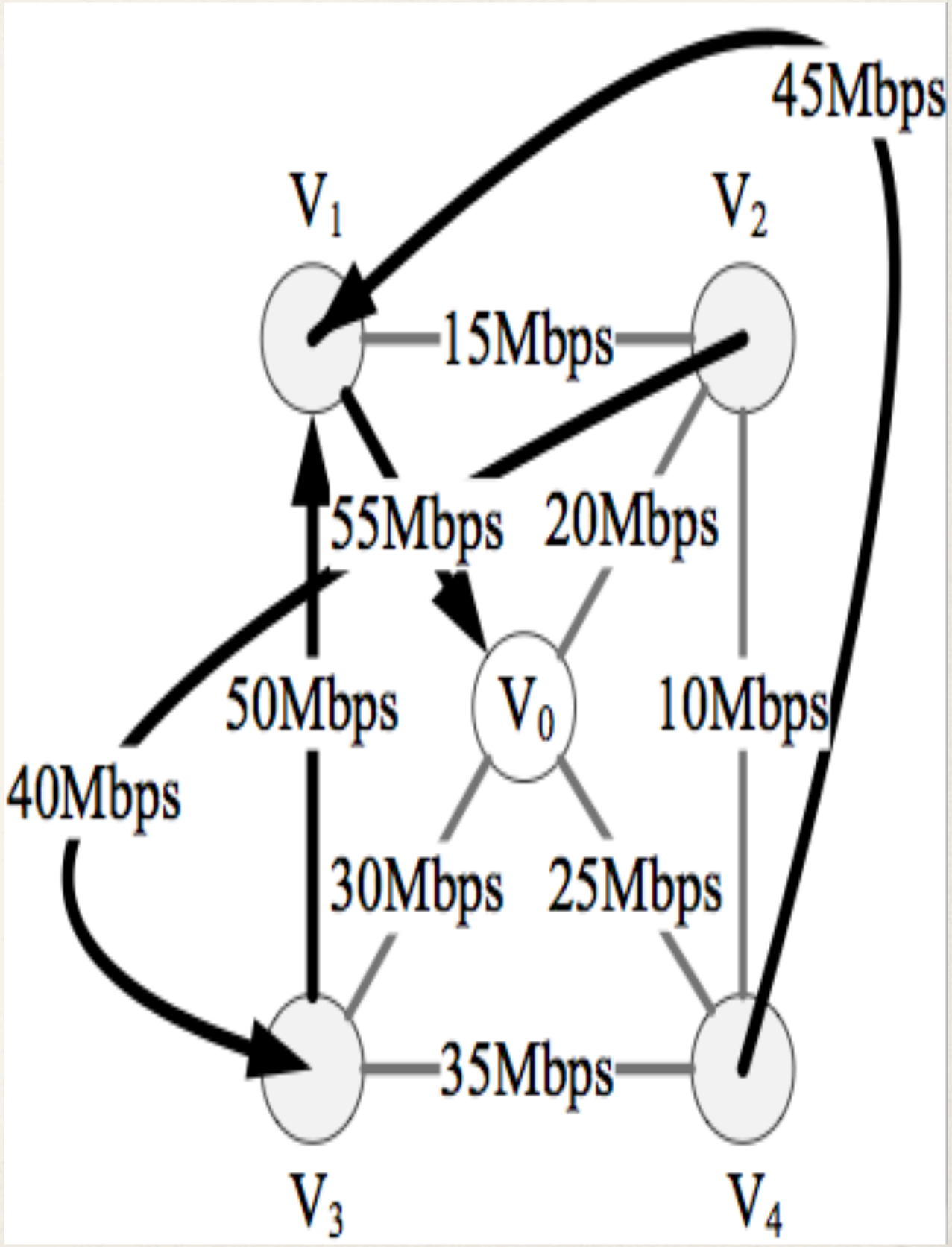
- **Consider how to route regeneration flows to achieve the optimal regeneration time**
  - A tree-structured regeneration scheme with linear network coding
    - [Li09] J. Li, S. Yang, X. Wang, X. Xue, B. Li, "*Tree-structured Data Regeneration with Network Coding in Distributed Storage Systems*," in Proc. of International Workshop on Quality of Service (IWQoS), 2009
- **Combine regenerating codes and the tree-structured regeneration**
- **Analyze the stability when providers may quit during the regeneration**

Our Contributions

- **Consider how to route regeneration flows to achieve the optimal regeneration time**
  - A tree-structured regeneration scheme with linear network coding
    - [Li09] J. Li, S. Yang, X. Wang, X. Xue, B. Li, "*Tree-structured Data Regeneration with Network Coding in Distributed Storage Systems*," in Proc. of International Workshop on Quality of Service (IWQoS), 2009
- **Combine regenerating codes and the tree-structured regeneration**
- **Analyze the stability when providers may quit during the regeneration**

TREE: regeneration tree

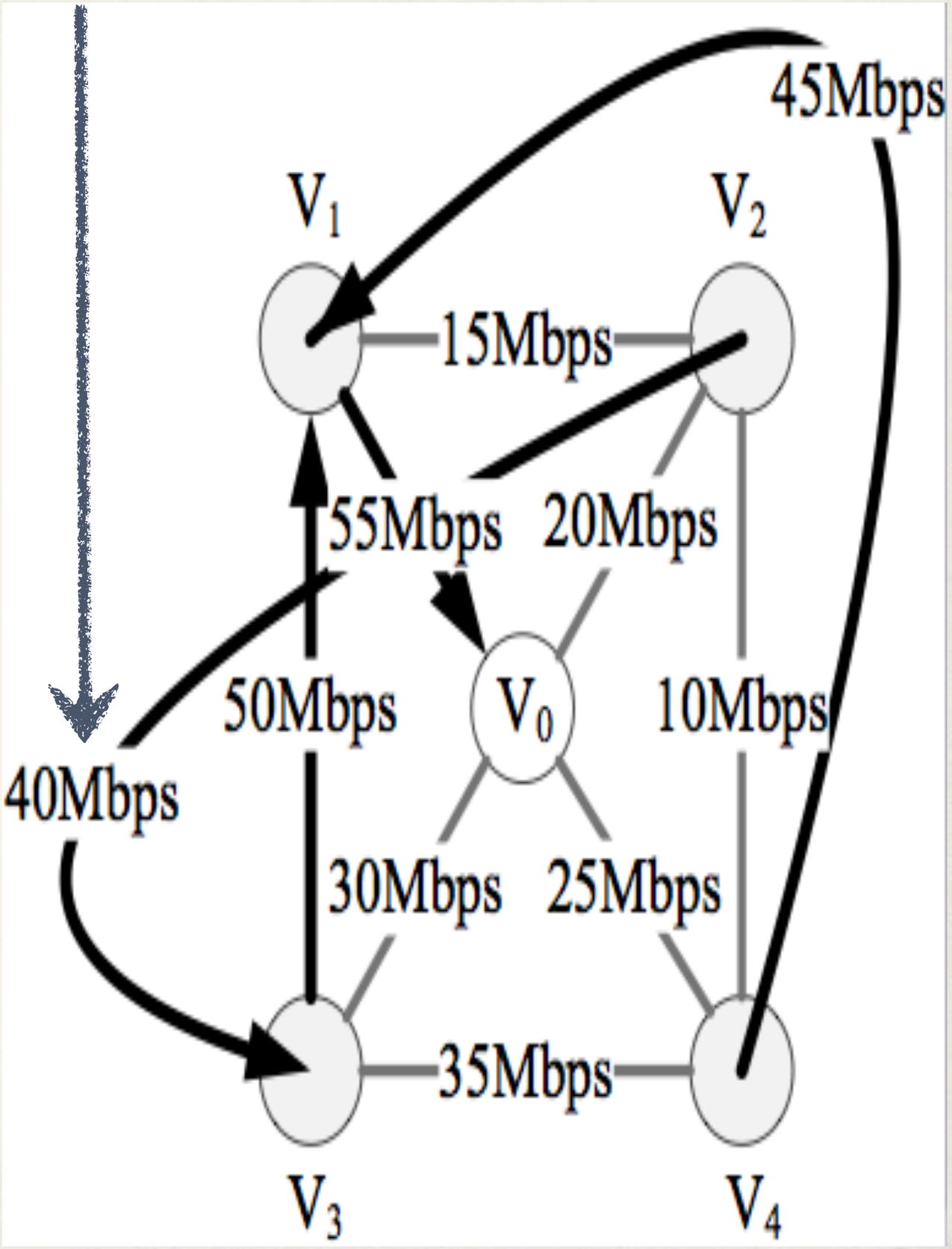
- To regenerate a block coded by network coding:
- Newcomer: V_0
- Providers: $V_1, V_2, ..., V_4$
  - not only a relay, but also an encoder
- End-to-end bandwidth measurement or estimation
- Optimal regeneration tree?
  - Prim's algorithm



TREE: regeneration tree

- To regenerate a block coded by network coding:
- Newcomer: V_0
- Providers: $V_1, V_2, ..., V_4$
  - not only a relay, but also an encoder
- End-to-end bandwidth measurement or estimation
- Optimal regeneration tree?
  - Prim's algorithm

available bandwidth



Network Model

G(d; n, k) = {V(d+1), E(d+1), ω}:

V(d+1): node set, including the newcomer and all providers

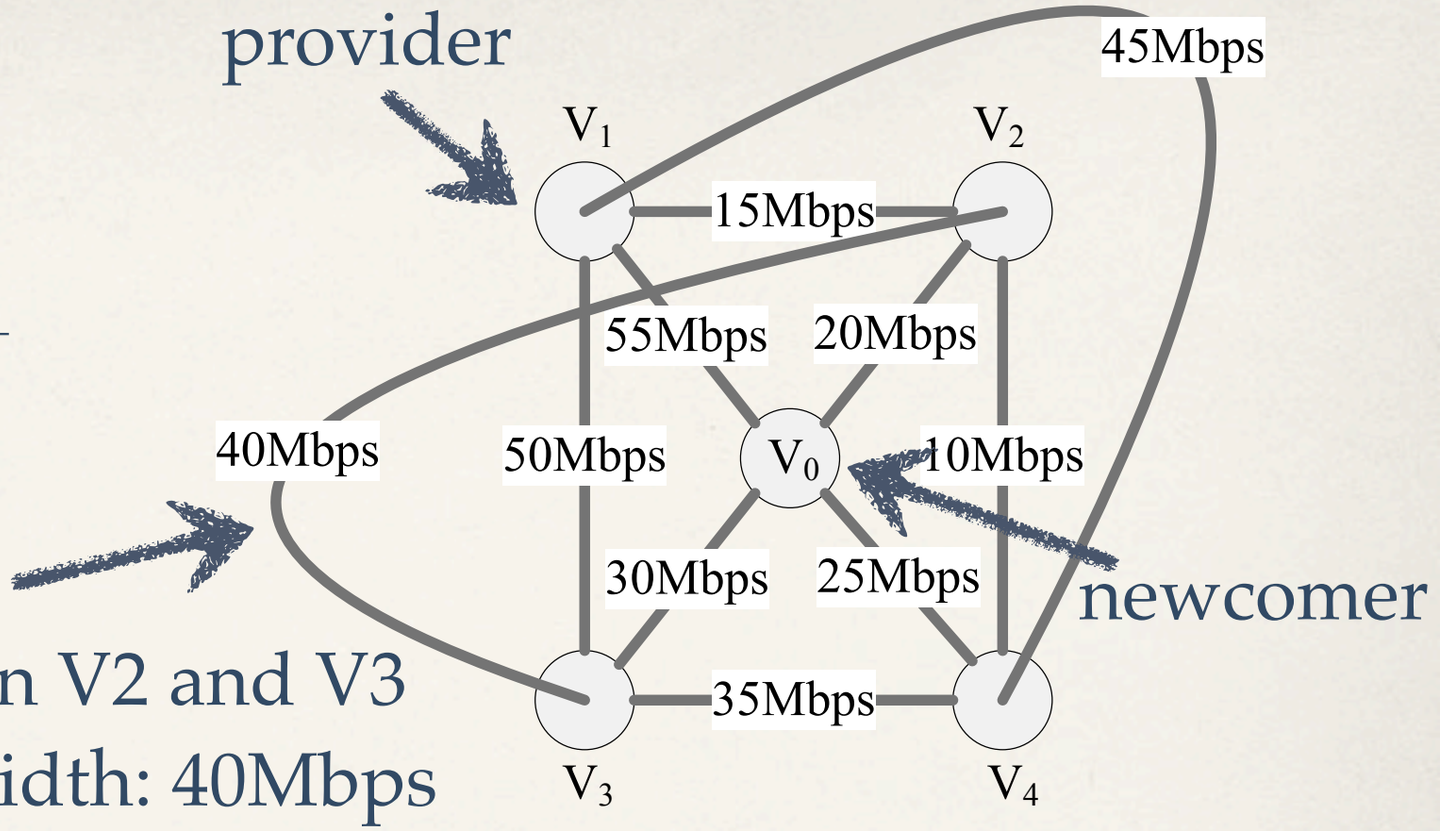
E(d+1): edge between each pair of nodes in V(d+1)

ω: available bandwidth

(r,d)-regeneration tree:

rooted by V_0, covering r providers in V(d+1)

Network Model



an edge between V2 and V3
available bandwidth: 40Mbps

$G(d; n, k) = \{V(d+1), E(d+1), \omega\}$:

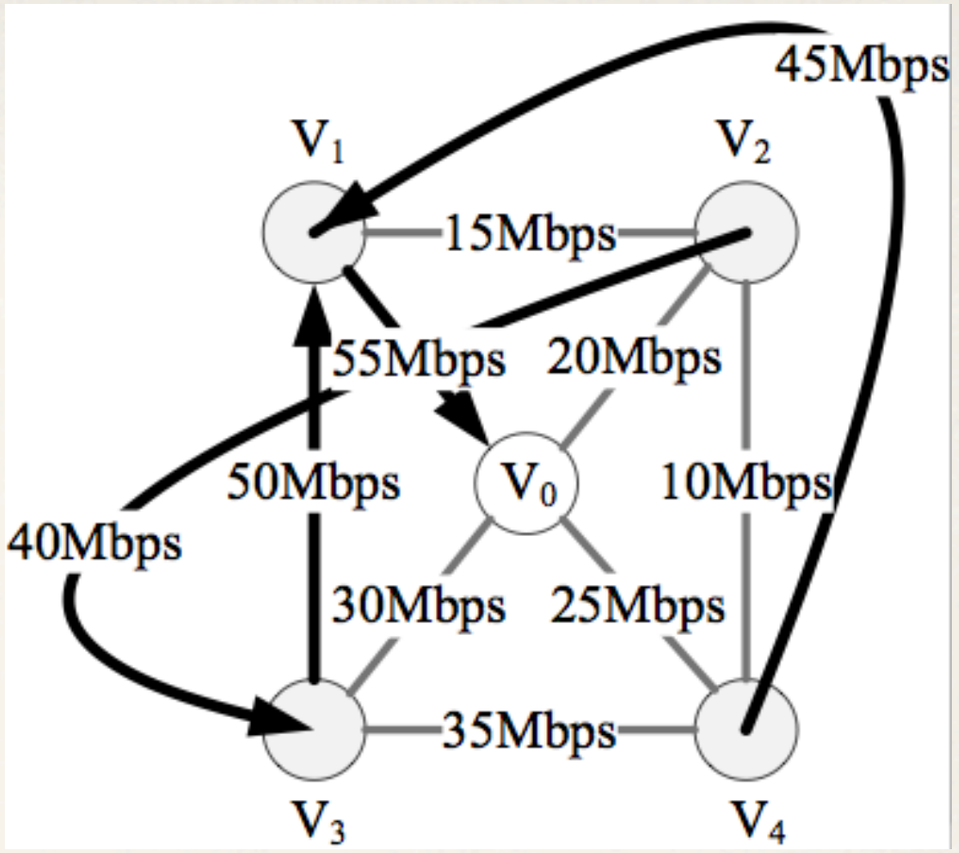
$V(d+1)$: node set, including the newcomer and all providers

$E(d+1)$: edge between each pair of nodes in $V(d+1)$

ω: available bandwidth

(r,d)-regeneration tree:
rooted by V_0, covering r providers in $V(d+1)$

Network Model



G(d; n, k) = {V(d+1), E(d+1), ω}:

V(d+1): node set, including the newcomer and all providers

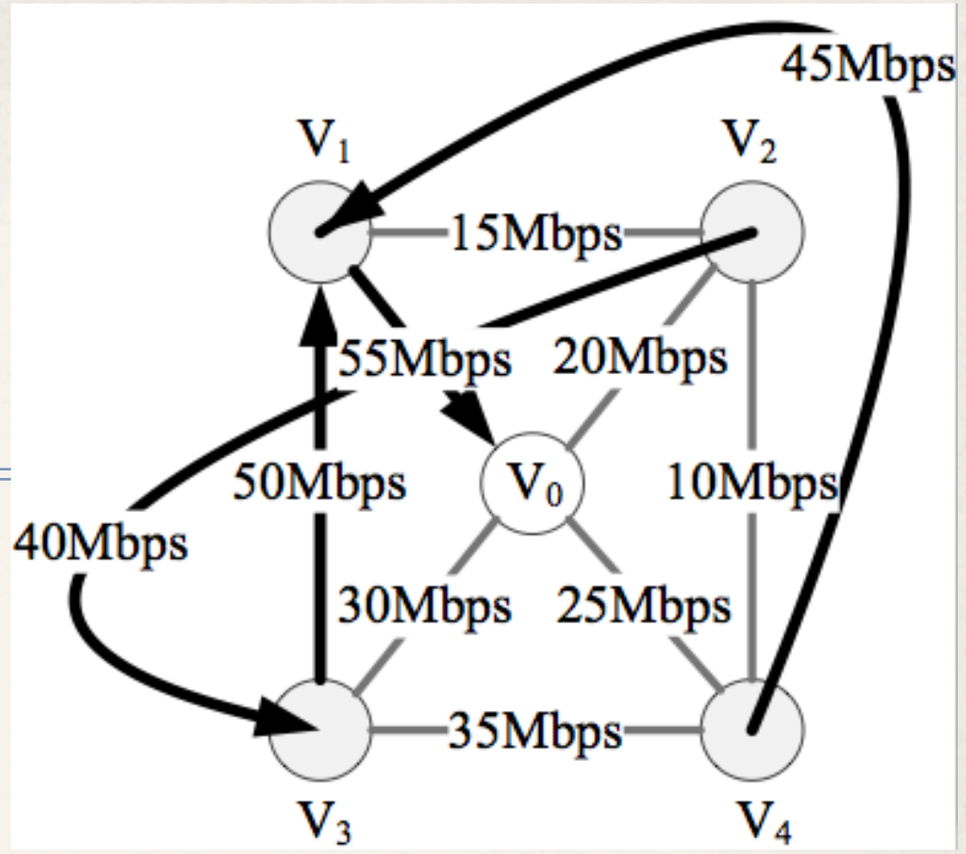
E(d+1): edge between each pair of nodes in V(d+1)

ω: available bandwidth

(r,d)-regeneration tree:

rooted by V_0, covering r providers in V(d+1)

Expected Available Bandwidth



Let e be the bottleneck edge of an optimal $(r, d)-$regeneration tree in $G(d; n, k)$, produced by Prim's algorithm. If e is the i^{th} maximum edge in $E(d+1)$, $\sigma_{\mathrm{TREE}}(r, G(d; n, k)) = i$.

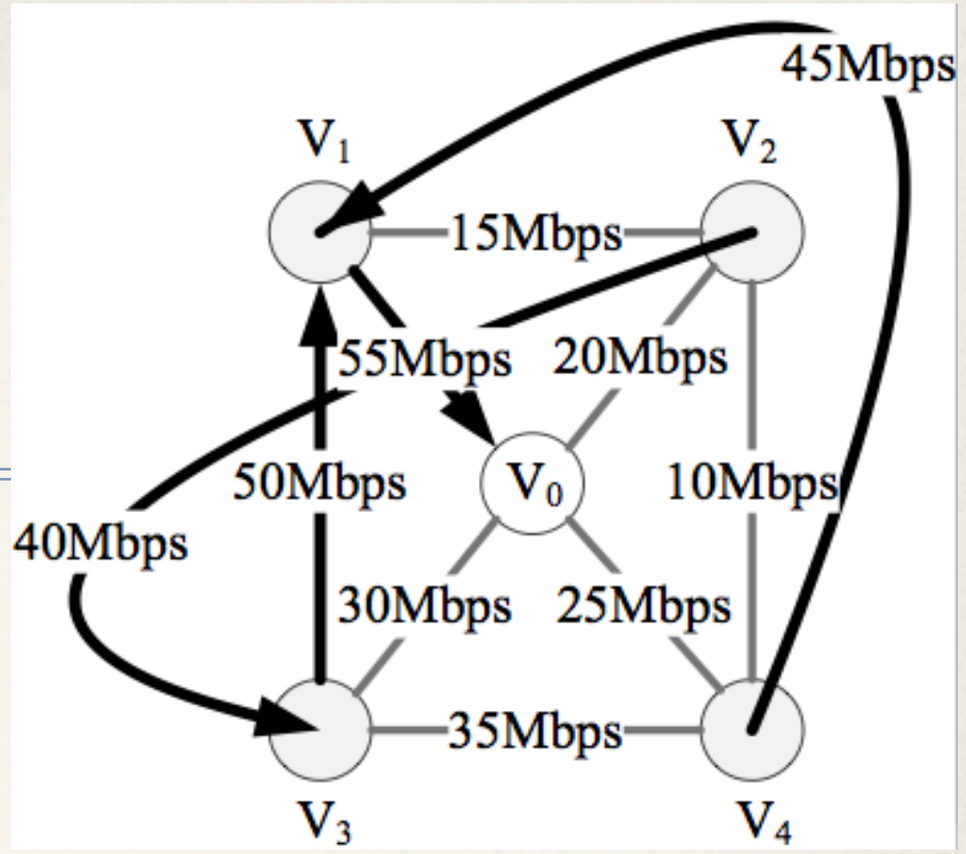
Let $p(d+1, r+1; i)$ be the probability that $\sigma_{\mathrm{TREE}}(r, G(d; n, k)) = i$.

$E_{(i;M_{d+1})}$:
expected value of i-th edge in $G(d;n,k)$

$$E_{\mathrm{TREE}}(r, G(d; n, k)) = \sum_{i=r}^{M_{d+1}-d+1} p(d+1, r+1; i) E_{(i:M_{d+1})}.$$

Expected Available Bandwidth

$\sigma_{\text{TREE}}(4, G(4; n, 3)) = 4$



Let e be the bottleneck edge of an optimal $(r, d)-$regeneration tree in $G(d; n, k)$, produced by Prim's algorithm. If e is the i^{th} maximum edge in $E(d + 1)$, $\sigma_{\text{TREE}}(r, G(d; n, k)) = i$.

Let $p(d + 1, r + 1; i)$ be the probability that $\sigma_{\text{TREE}}(r, G(d; n, k)) = i$.

$E_{(i;M_{d+1})}$:
expected value of *i*-th edge in *G(d;n,k)*

$$E_{\text{TREE}}(r, G(d; n, k)) = \sum_{i=r}^{M_{d+1}-d+1} p(d + 1, r + 1; i) E_{(i:M_{d+1})}.$$

Regenerating Codes [Wu07]

For (*n*,*k*)-MDS codes, if each node stores α bits and downloads β bits from each provider

$$(\alpha_{MSR}, \beta_{MSR}) = \left(\frac{M}{k}, \frac{M}{k(r-k+1)} \right)$$

M: file size;

r: # of providers used in the regeneration

MSR codes: Minimum-Storage Regenerating codes

MSR codes achieve the minimum β for MDS codes

RCTREE: To Combine TREE with MSR

- The newcomer needs at least *r-k+1* adjacent providers since

$$\frac{\alpha_{MSR}}{\beta_{MSR}} = r - k + 1$$

- A degree constraint at the newcomer

Algorithm 1 Find an optimal (r, d, k)–regeneration tree T in $G(d; n, k)$, $k \leq r \leq d$. Define $E_{root} = \{(V_0, V_i) | i = 1, 2, \ldots, d\}$, and $D(T) = |E_{root} \cap \{\text{edges in } T\}|$.

1: $T \leftarrow \emptyset$
2: **for** $i \leftarrow 1$ to r **do**
3: $e_i \leftarrow$ the largest edge making $T \bigcup \{e_i\}$ a rooted tree
4: $T \leftarrow T \bigcup \{e_i\}$
5: **end for**
6: **for** $i \leftarrow D(T) + 1$ to $r - k + 1$ **do**
7: $e_1 \leftarrow$ the largest edge $\in E_{root} - T$
8: $e_2 \leftarrow$ any edge $\in T - E_{root}$ making $T \bigcup \{e_1\} - \{e_2\}$ a tree rooted by V_0
9: $T \leftarrow T \bigcup \{e_1\} - \{e_2\}$
10: **end for**

Analysis (details in [Li10])

- **Available bandwidth affected by the topology constraint**

Algorithm 1 Find an optimal (r, d, k)−regeneration tree T in $G(d; n, k)$, $k \leq r \leq d$. Define $E_{root} = \{(V_0, V_i)|i = 1, 2, \ldots, d\}$, and $D(T) = |E_{root} \cap \{\text{edges in } T\}|$.

1: $T \leftarrow \emptyset$
2: **for** $i \leftarrow 1$ to r **do**
3: $e_i \leftarrow$ the largest edge making $T \bigcup \{e_i\}$ a rooted tree
4: $T \leftarrow T \bigcup \{e_i\}$
5: **end for**
6: **for** $i \leftarrow D(T) + 1$ to $r - k + 1$ **do**
7: $e_1 \leftarrow$ the largest edge $\in E_{root} - T$
8: $e_2 \leftarrow$ any edge $\in T - E_{root}$ making $T \bigcup \{e_1\} - \{e_2\}$ a tree rooted by V_0
9: $T \leftarrow T \bigcup \{e_1\} - \{e_2\}$
10: **end for**

- **Line 1 ~5**

- **Line 6 ~ 10**

p(d,r,c;i): the probability that
$\sigma_{RCTREE}(r, G(d; n, k))$ = **i** and **c** edges attached to the newcomer in the tree have weights more than the i-th maximal edge in E(d+1)

Analysis (details in [Li10])

- **Available bandwidth affected by the topology constraint**

Algorithm 1 Find an optimal (r, d, k)−regeneration tree T in $G(d; n, k)$, $k \leq r \leq d$. Define $E_{root} = \{(V_0, V_i)|i = 1, 2, \ldots, d\}$, and $D(T) = |E_{root} \cap \{\text{edges in } T\}|$.

1: $T \leftarrow \emptyset$
2: **for** $i \leftarrow 1$ to r **do**
3: $e_i \leftarrow$ the largest edge making $T \bigcup \{e_i\}$ a rooted tree
4: $T \leftarrow T \bigcup \{e_i\}$
5: **end for**
6: **for** $i \leftarrow D(T) + 1$ to $r - k + 1$ **do**
7: $e_1 \leftarrow$ the largest edge $\in E_{root} - T$
8: $e_2 \leftarrow$ any edge $\in T - E_{root}$ making $T \bigcup \{e_1\} - \{e_2\}$ a tree rooted by V_0
9: $T \leftarrow T \bigcup \{e_1\} - \{e_2\}$
10: **end for**

- **Line 1 ~5**

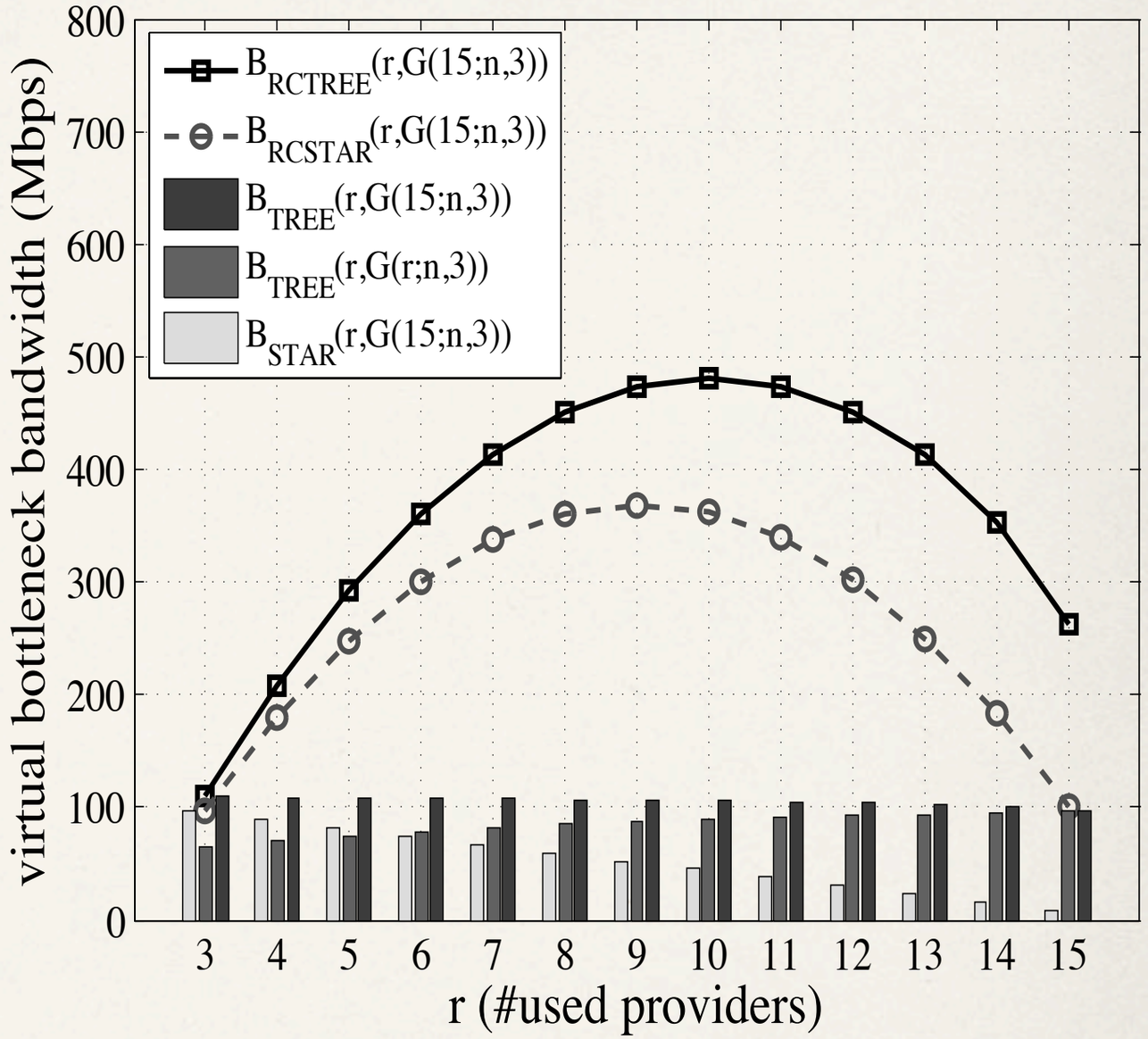
- **Line 6 ~ 10**

$p^{RCTREE(k)}(d+1, r+1, i)$:
the probability that $\sigma_{RCTREE}(r, G(d; n, k)) = i$

Pr{available bandwidth is not affected by topology constaint} +
Σ[Pr{available bandwidth is affected}×
Pr{the bottleneck edge becomes the i-th maximal edge}]

How efficient is the regeneration tree?

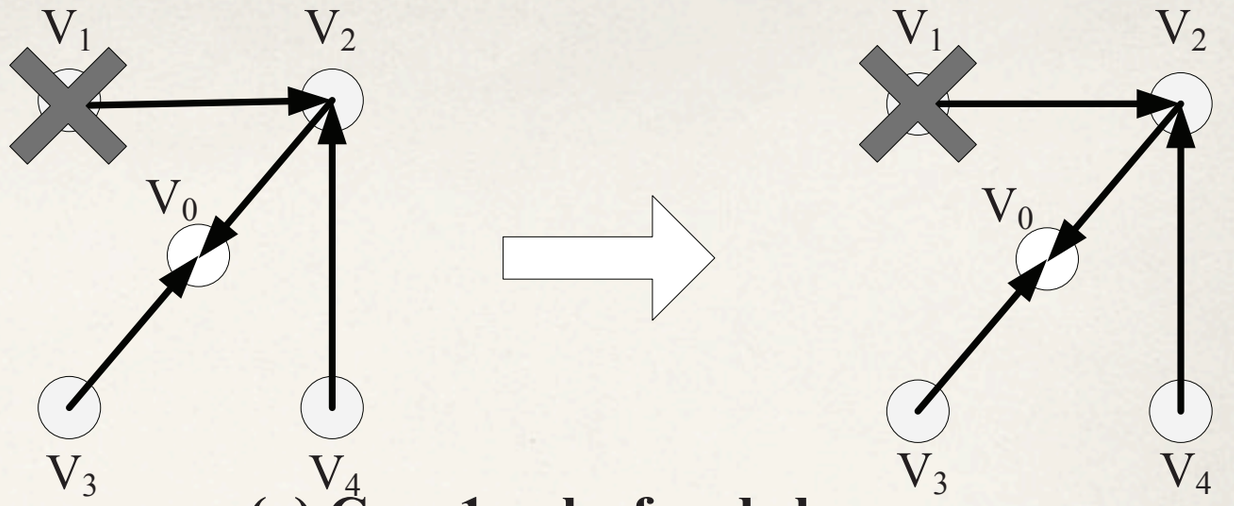
Bandwidth distribution measured in PlanetLab [Lee05]



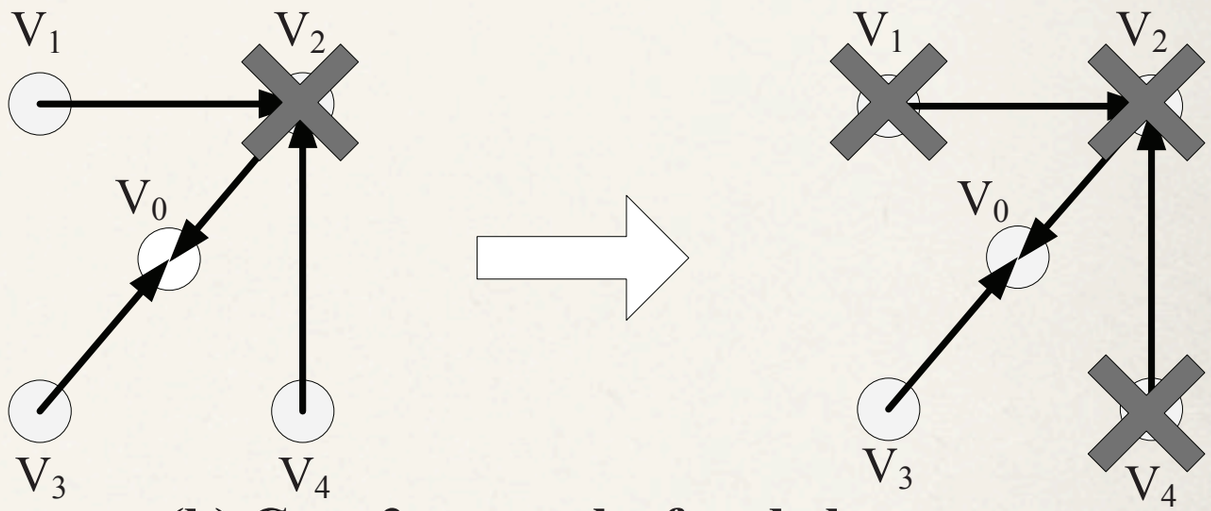
For STAR and TREE, $B_{\mathrm{STAR}}(r, G(d;n,k)) = E_{\mathrm{STAR}}(r, G(d;n,k))$ and $B_{\mathrm{TREE}}(r, G(d;n,k)) = E_{\mathrm{TREE}}(r, G(d;n,k))$. For RCSTAR and RCTREE, $B_{\mathrm{RCSTAR}}(r, G(d;n,k)) = (r-k+1)E_{\mathrm{STAR}}(r, G(d;n,k))$ and $B_{\mathrm{RCTREE}}(r, G(d;n,k)) = (r-k+1)E_{\mathrm{RCTREE}}(r, G(d;n,k))$.

Another Concern: Stability

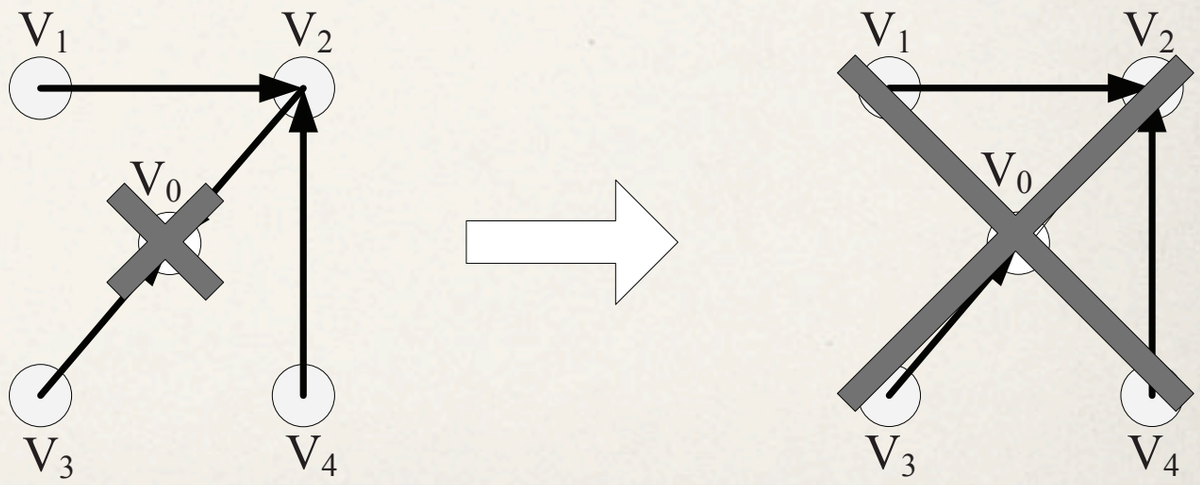
Nodes may quit during the regeneration and thus a regeneration tree crashes.



(a) Case 1: a leaf node leaves



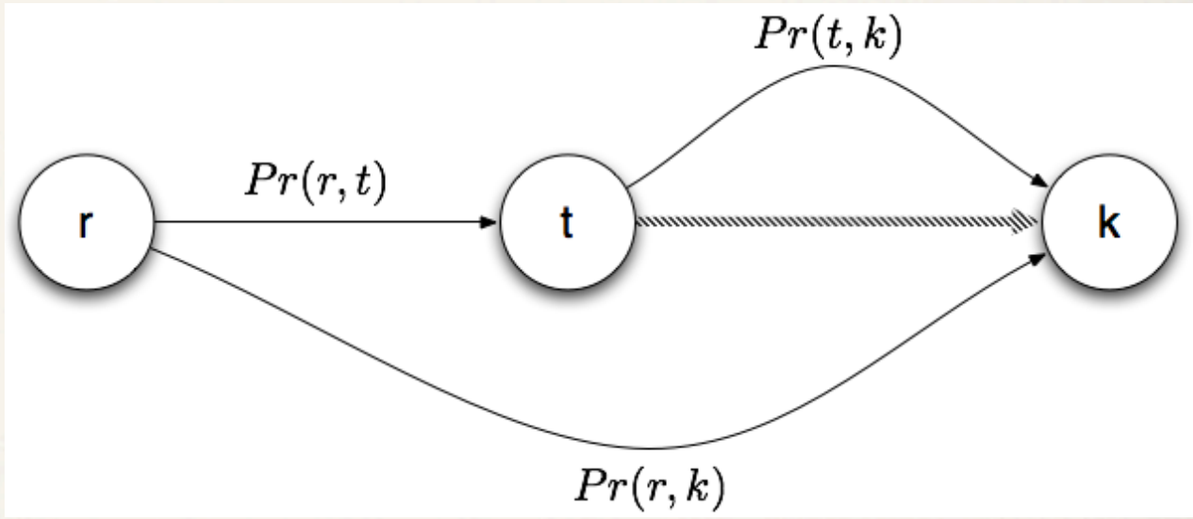
(b) Case 2: a non-leaf node leaves



(c) Case 3: the newcomer leaves

Lifetime

- The lifetime of a regeneration tree in *G(d;n,k)* is the time between when the tree is constructed and when less than *k* providers remain in the subtree

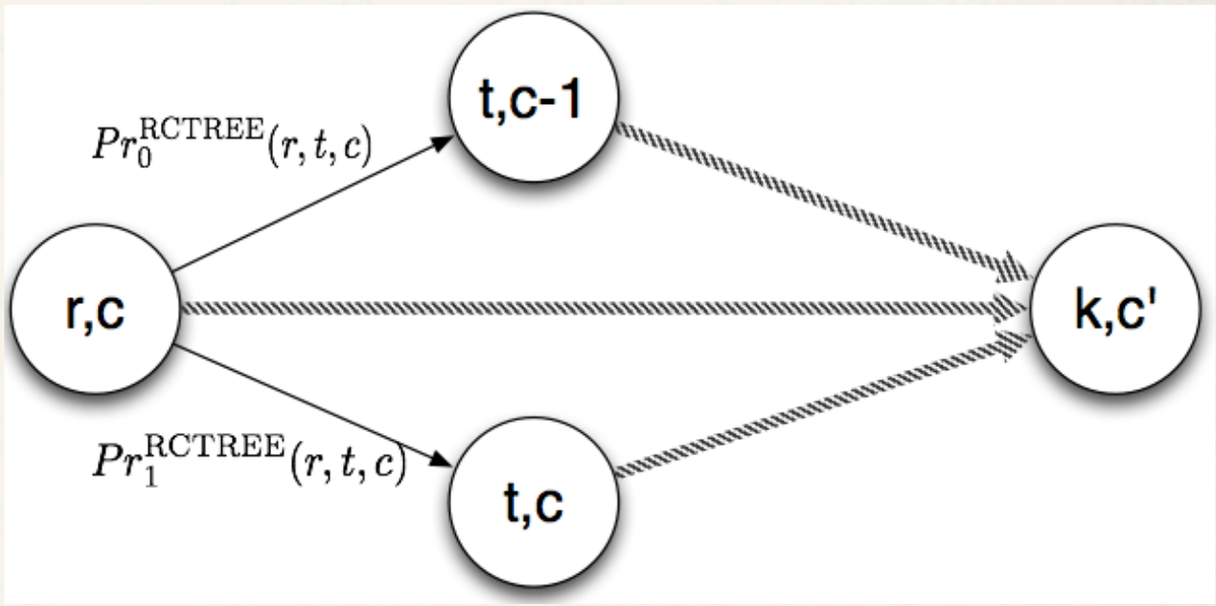


- A Markov chain:
  - *r, t, k*: the number of providers

$$L^{\mathrm{STAR}(k)}(r-1) = E(r) + \sum_{t=k+1}^{t=r-1} Pr^{\mathrm{STAR}}(r,t) L^{\mathrm{STAR}(k)}(t-1).$$

Lifetime of RCTREE

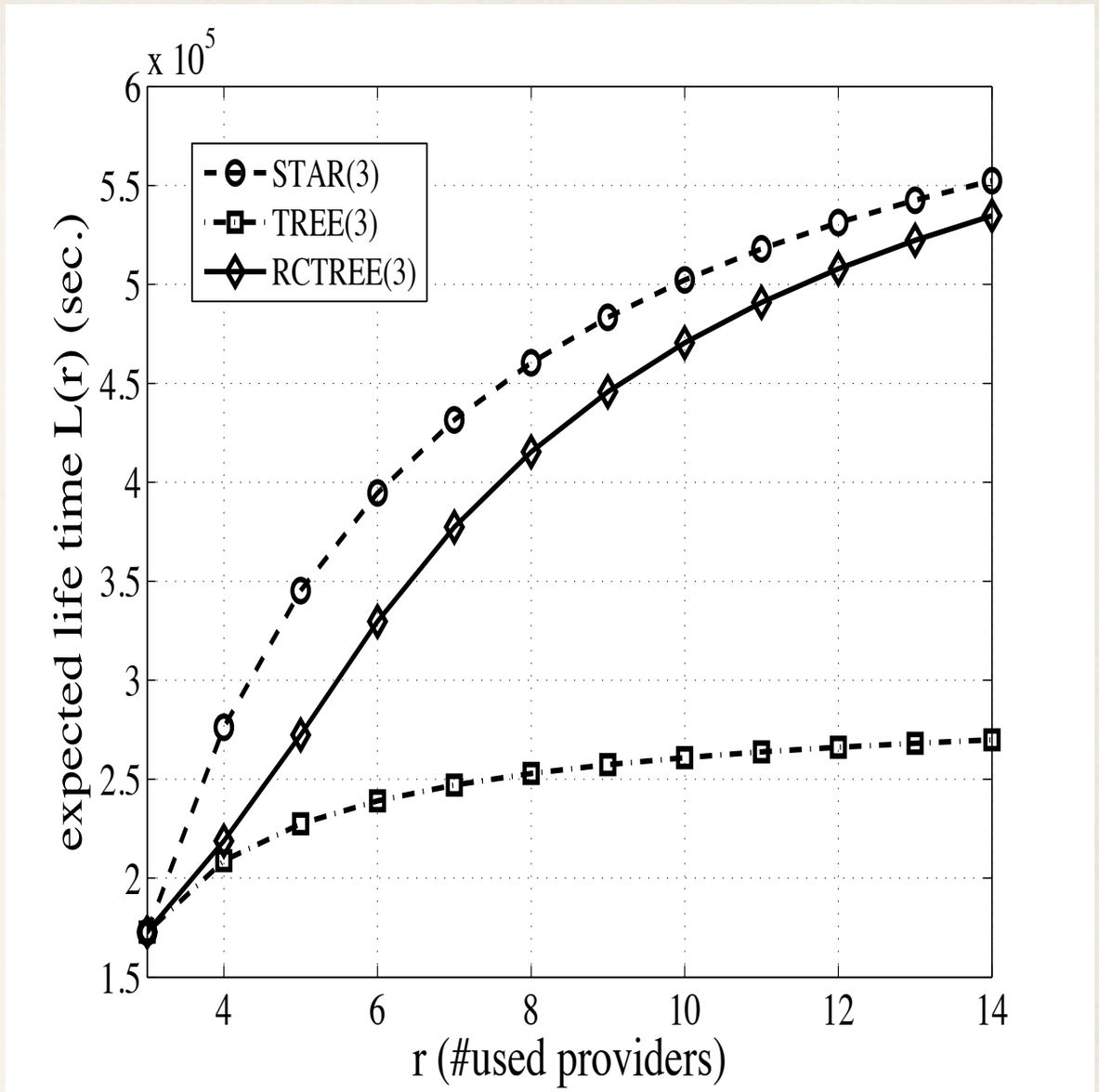
- State:
  - r: the number of providers in the tree
  - c: the degree of the newcomer
- The degree constraint always satisfied
  - The degree of the newcomer decreases by 1 at most



Evaluation

$L^{RCTREE(k)}(r,c)$:

the lifetime of the regeneration tree that there are *r* providers and the degree of the newcomer is *c*



$L^{RCTREE(k)}(r)$:

summarizing $L^{RCTREE(k)}(r,c)$ with coefficients of the frequency of the regeneration tree where the newcomer has a degree of *c*

Conclusion

- RCTREE combines the advantages of tree-structured regeneration and MSR codes
  - reducing the regeneration time: better routing & less traffic
  - providing better stability
- One step closer towards a practical repair of failed storage nodes in distributed storage systems

References

[Dimakis07] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright and K. Ramchandran “Network Coding for Distributed Storage Systems,” IEEE Conference on Computer Communications. (INFOCOM), May 2007

[Lee05] S.-J. Lee, P. Sharma, S. Banerjee, S. Basu, and R. Fonseca, “Measur- ing bandwidth between PlanetLab nodes,” Passive and Active Network Measurement, pp. 292–305, 2005

[Wu07] Y. Wu, A. G. Dimakis and K. Ramchandran, “Deterministic Regenerating Codes for Distributed Storage,” Forty-Fifth Annual Allerton Conference, September 2007

Thank you!

More information:
http://sonic.fudan.edu.cn/junli/

We appreciate the financial support of School of Computer Science, Fudan University.