



# Existence of Optimal Cooperative Regenerating Codes for Functional Repair

Kenneth Shum

(Joint work with Yuchong Hu)

July 2011

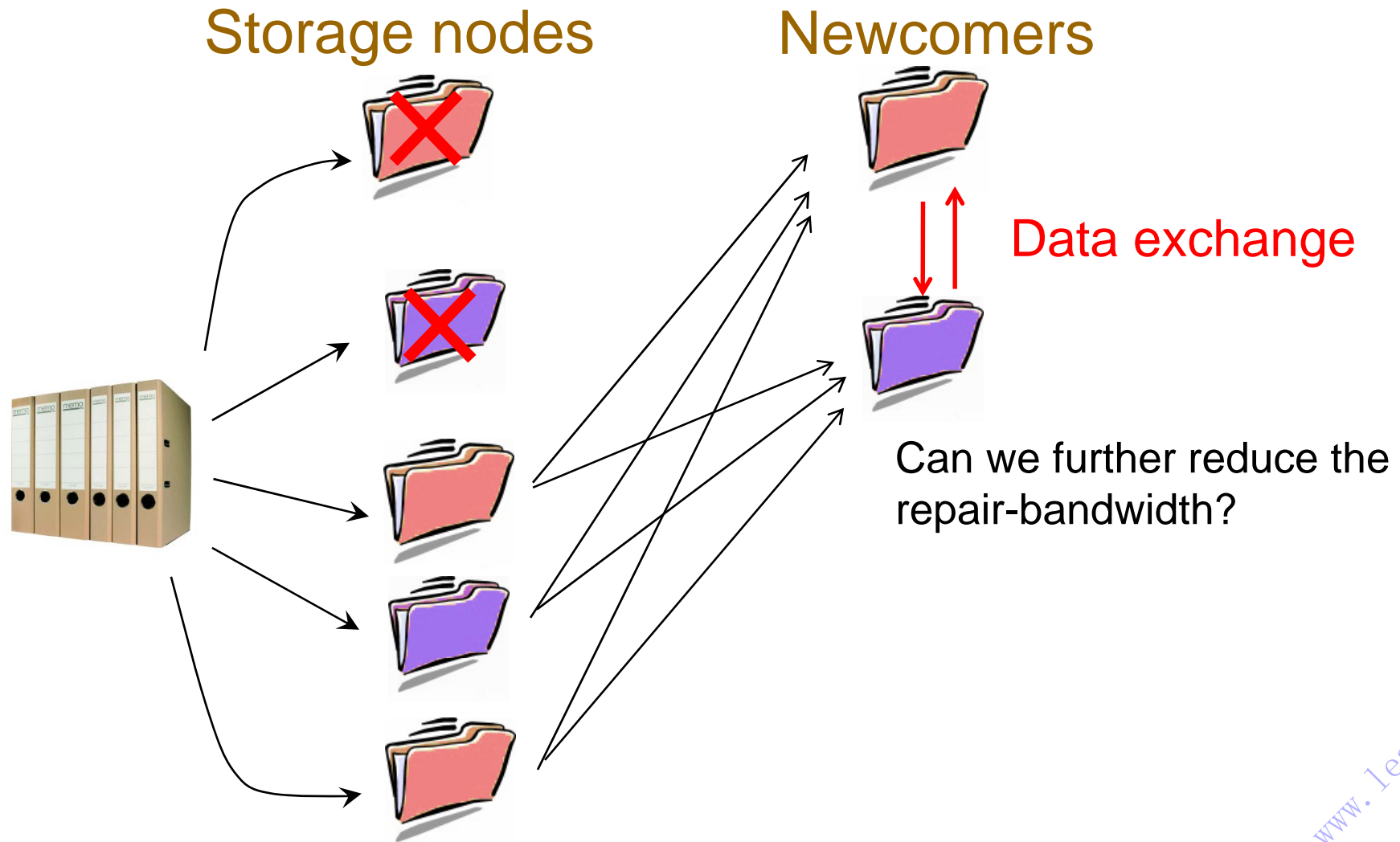
# Multiple node failures

- Large-scale storage system
  - The number of failed storage nodes may be modeled as Poisson random variable, with mean greater than one.
- Peer-to-peer storage system
  - Large churn rate

# Multiple node failures (cont'd)

- The lazy-repair policy in TotalRecall
  - The repair of a failed node is deliberately delayed until the number of failed nodes has reached a certain threshold.

# Jointly repair multiple failures

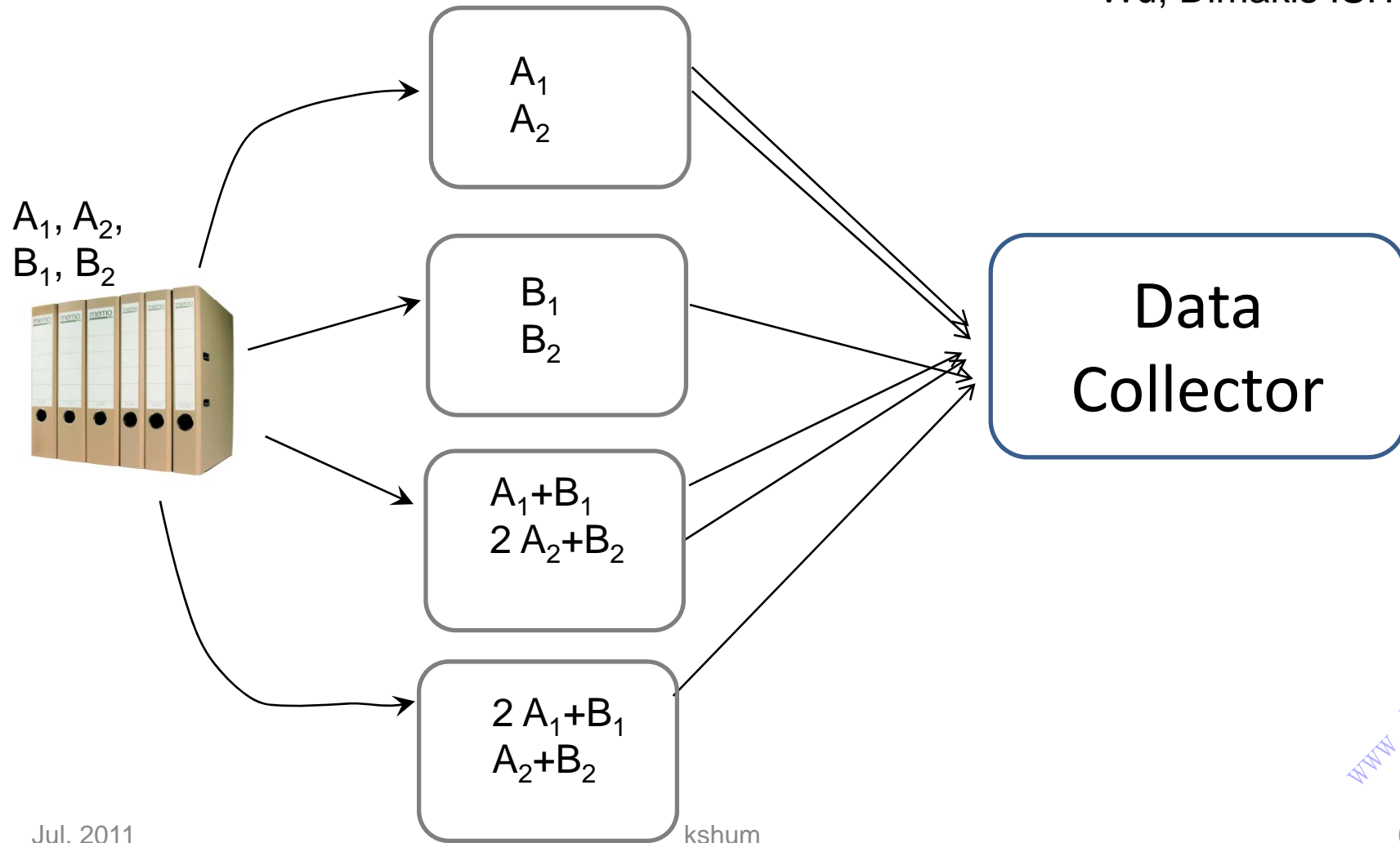


# Outline of the talk

- An explicit example for cooperative repair for exact repair
- Skectch of proof for the existence of linear regenerating codes for functional repair

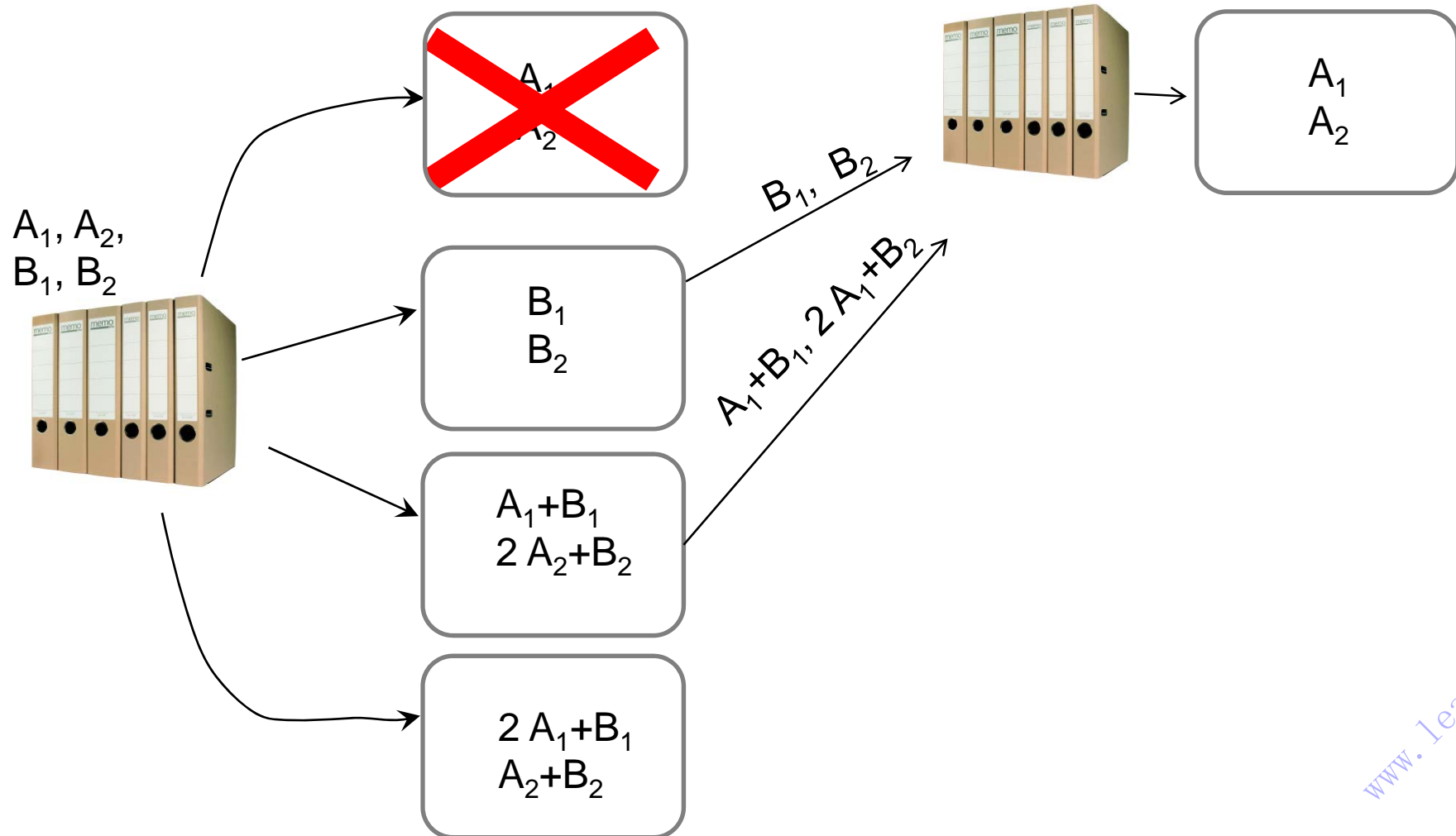
# Distributed storage (erasure coding)

Wu, Dimakis ISIT09



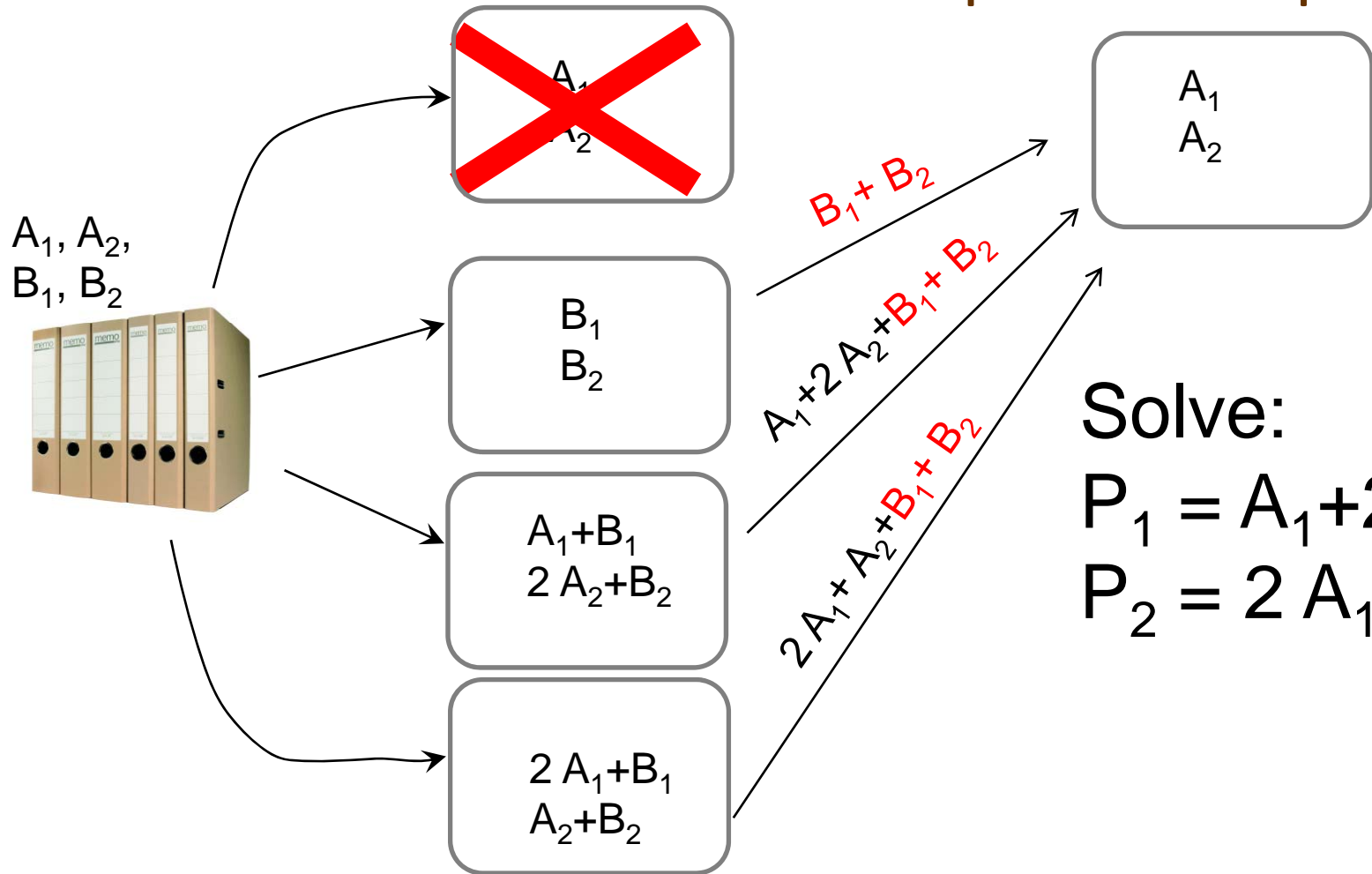
# Naive Repair

4 packets required.



# Repair with "code alignment"

3 packets required.

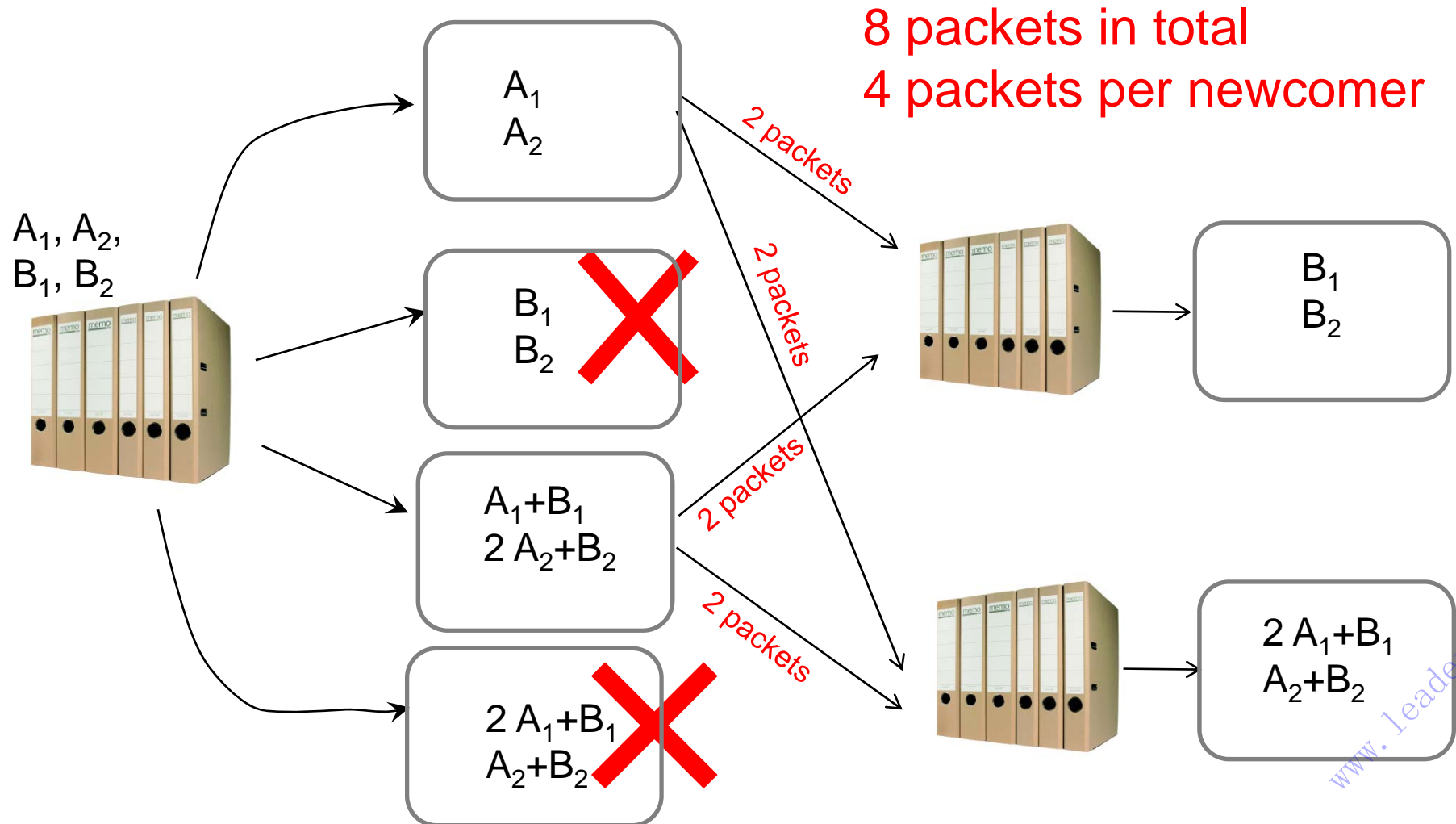


Solve:

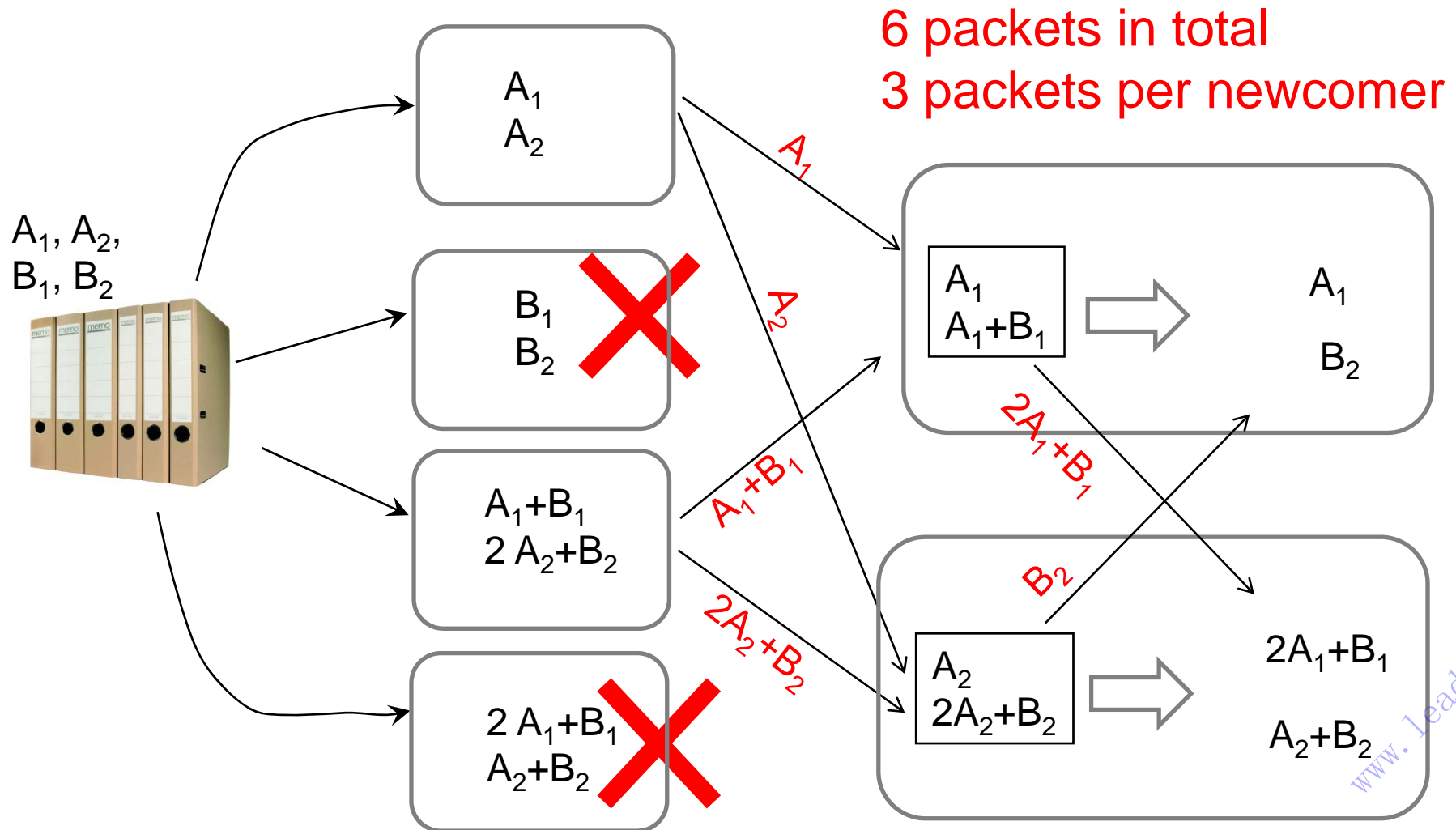
$$P_1 = A_1 + 2A_2$$

$$P_2 = 2A_1 + A_2$$

# Multiple failures, separate repair



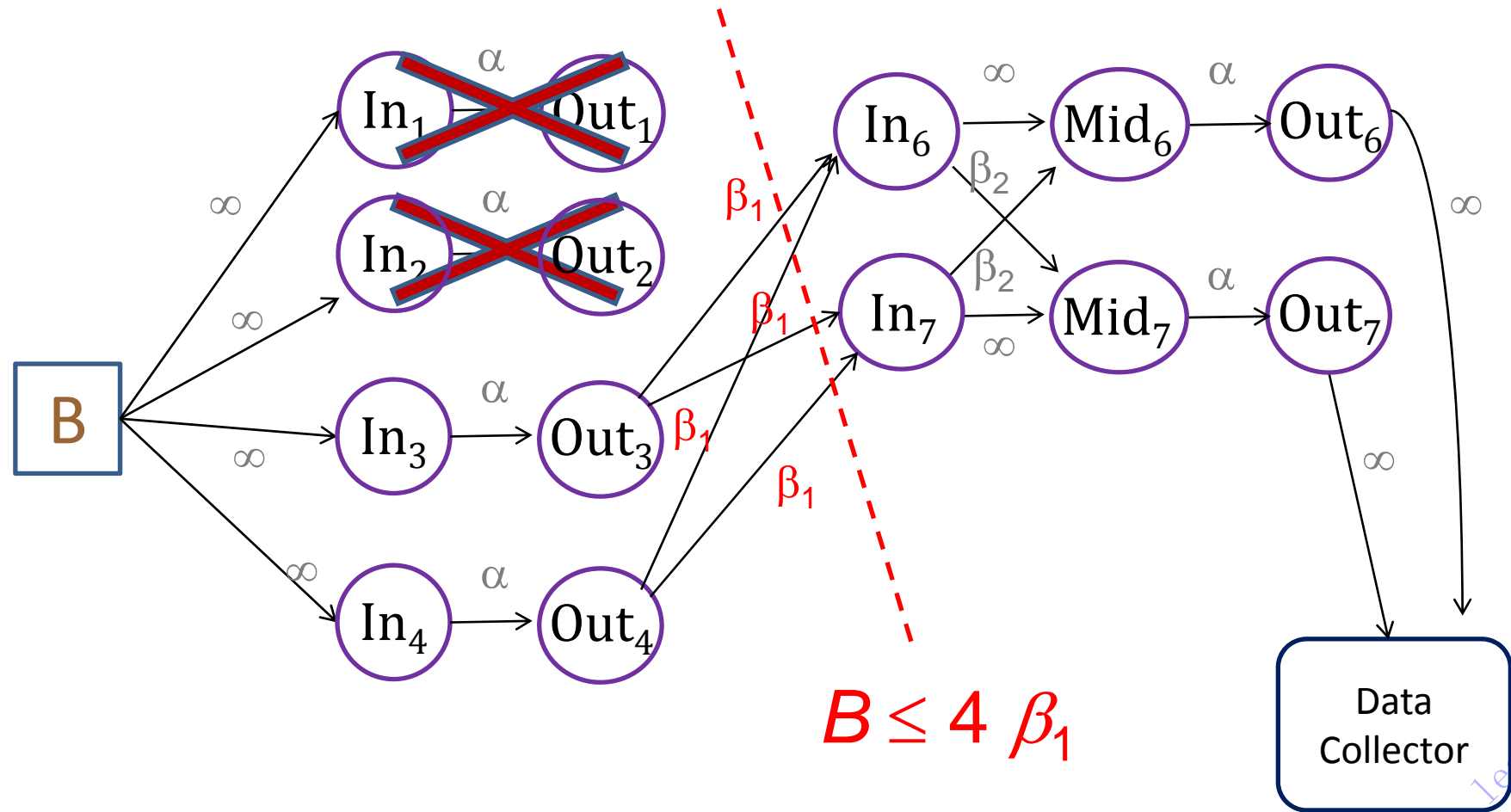
# Multiple failures, cooperative repair



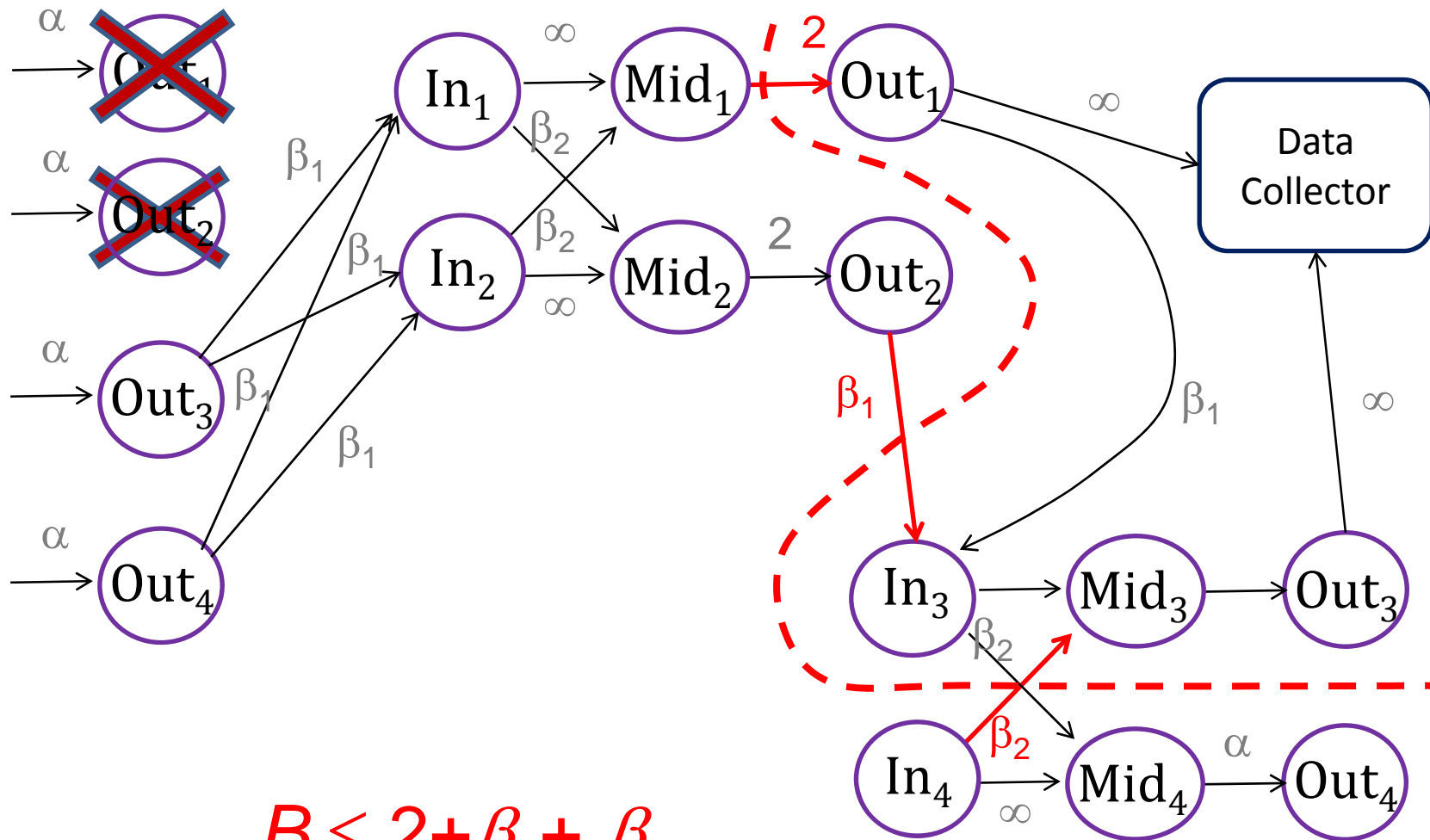
# Is this construction optimal?

- Can we repair with strictly less than 3 packets per newcomer?

# First cut



# Second cut



# A linear programming problem

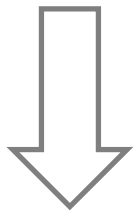
- Minimize  $2\beta_1 + \beta_2$  (repair bandwidth)

- Subject to

$$4 \leq 4\beta_1$$

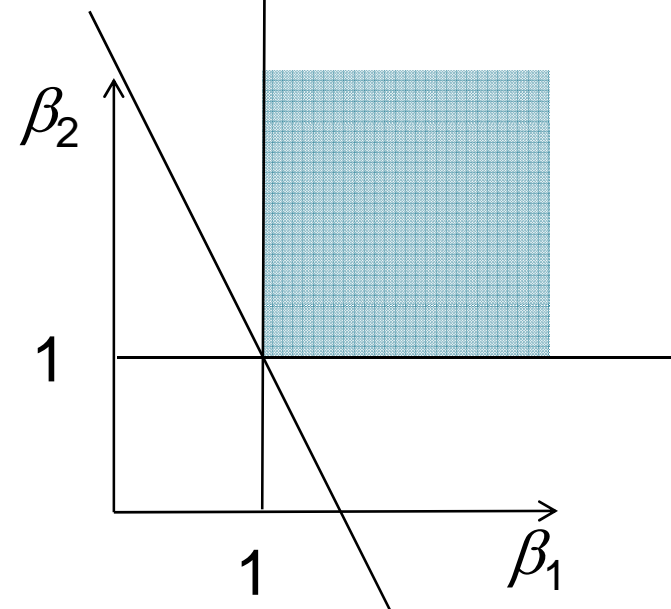
$$4 \leq 2 + \beta_1 + \beta_2$$

$$\beta_1, \beta_2 \geq 0$$



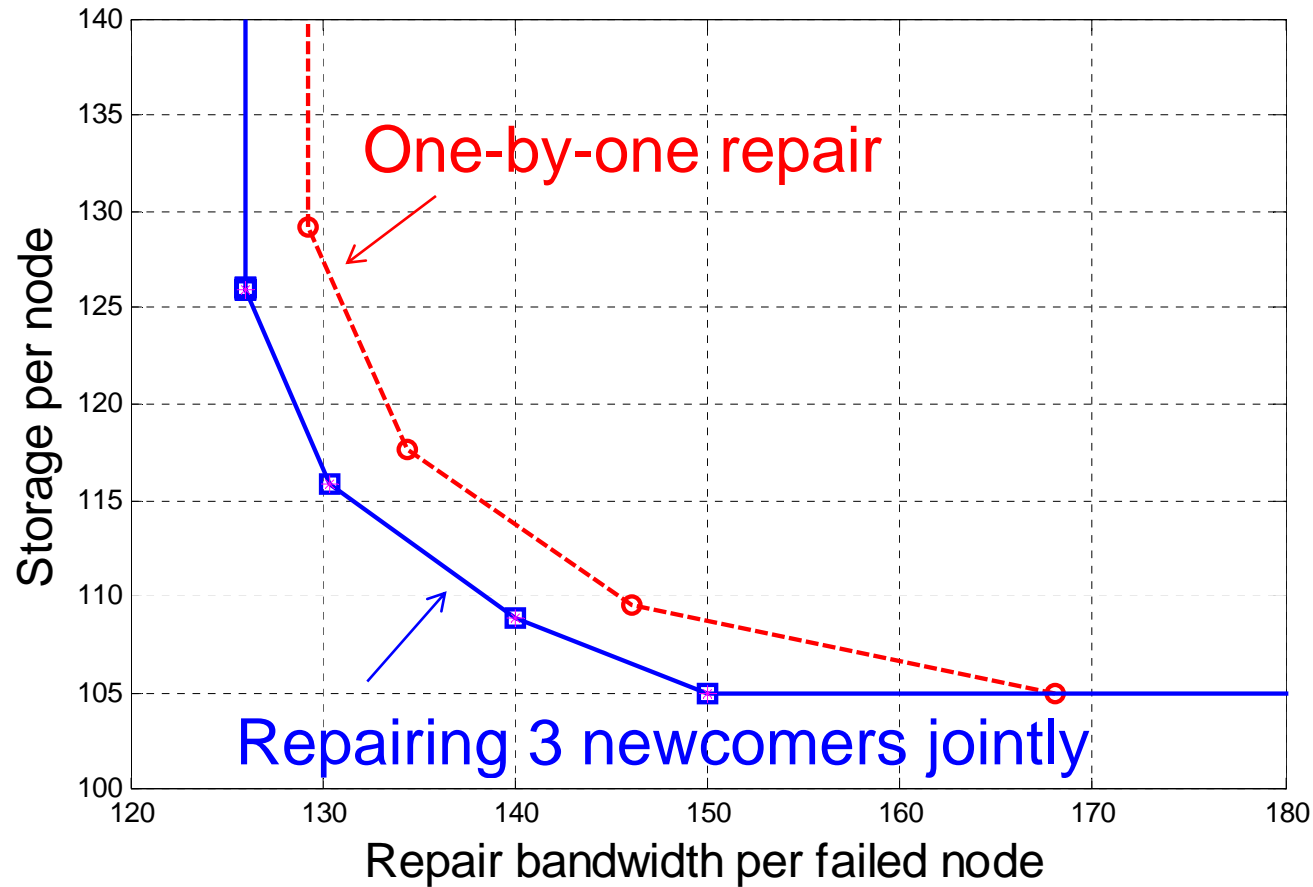
$$\beta_1 \geq 1 \quad \beta_2 \geq 1$$

$\Rightarrow$  At least 3 packets



# Storage vs Repair-bandwidth

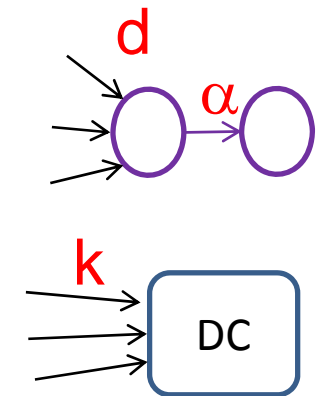
(S., ICC 2011, Kermarrec, Le Scouamec and Straub, Netcod 2011.)



File size = 420

$d = 8$

$k = 4$



[www.leaderstudio.net](http://www.leaderstudio.net)

# Outline of the talk

- An explicit example for cooperative repair
- Tradeoff between storage and repair-bandwidth
- Sketch of proof of the existence of linear regenerating codes for **functional repair**

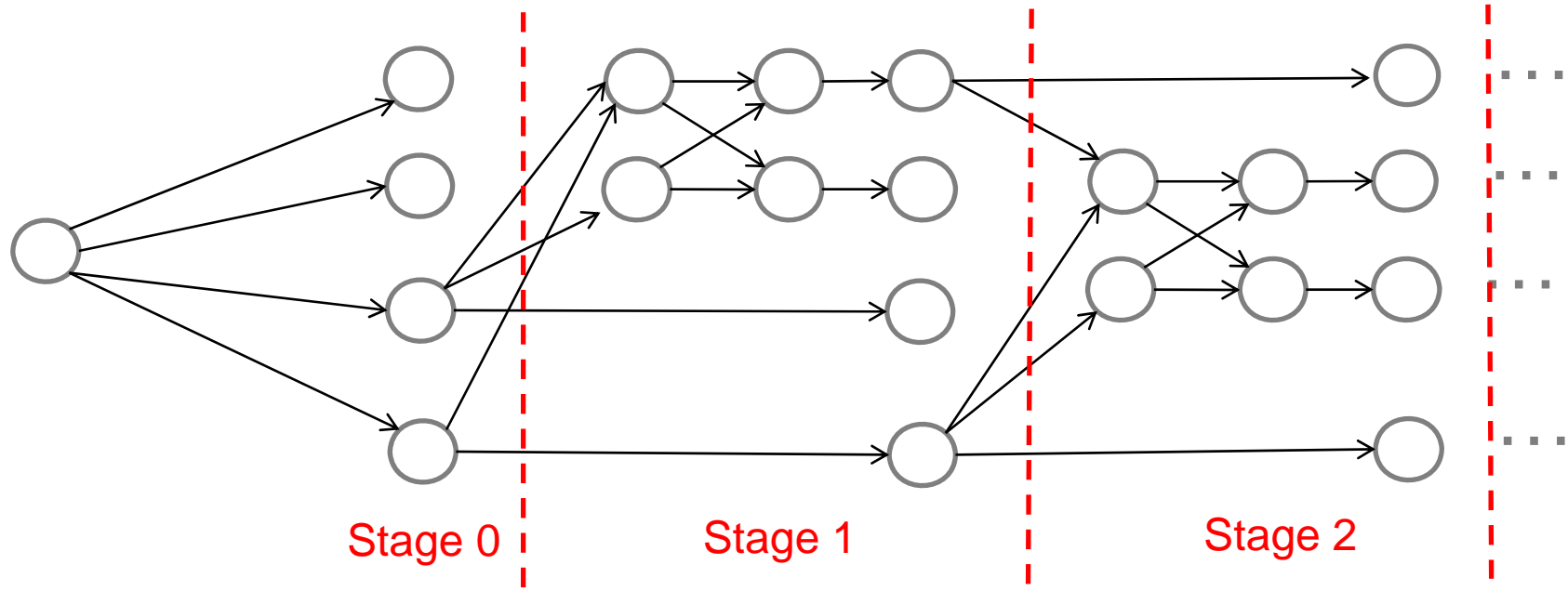
# Existence of optimal linear regenerating codes in general

- We want regenerating codes which will work after arbitrarily many repairs.
- Technical difficulty: The information flow graph is **unbounded**.
  - Most results on the construction of linear network codes for single-source multicasting require that the underlying graph is **finite**.

# Existence of optimal linear regenerating codes in general

- Can we work over a *fixed* finite field, for unlimited number of regenerations?
  - Yes for cooperative functional repair in general.

# Trellis structure



**m**

Message vector  
(row vector)

$$\mathbf{mT}_0$$

$T_0$  is the “transfer matrix” in stage 0

$$\mathbf{mT}_0T_1$$

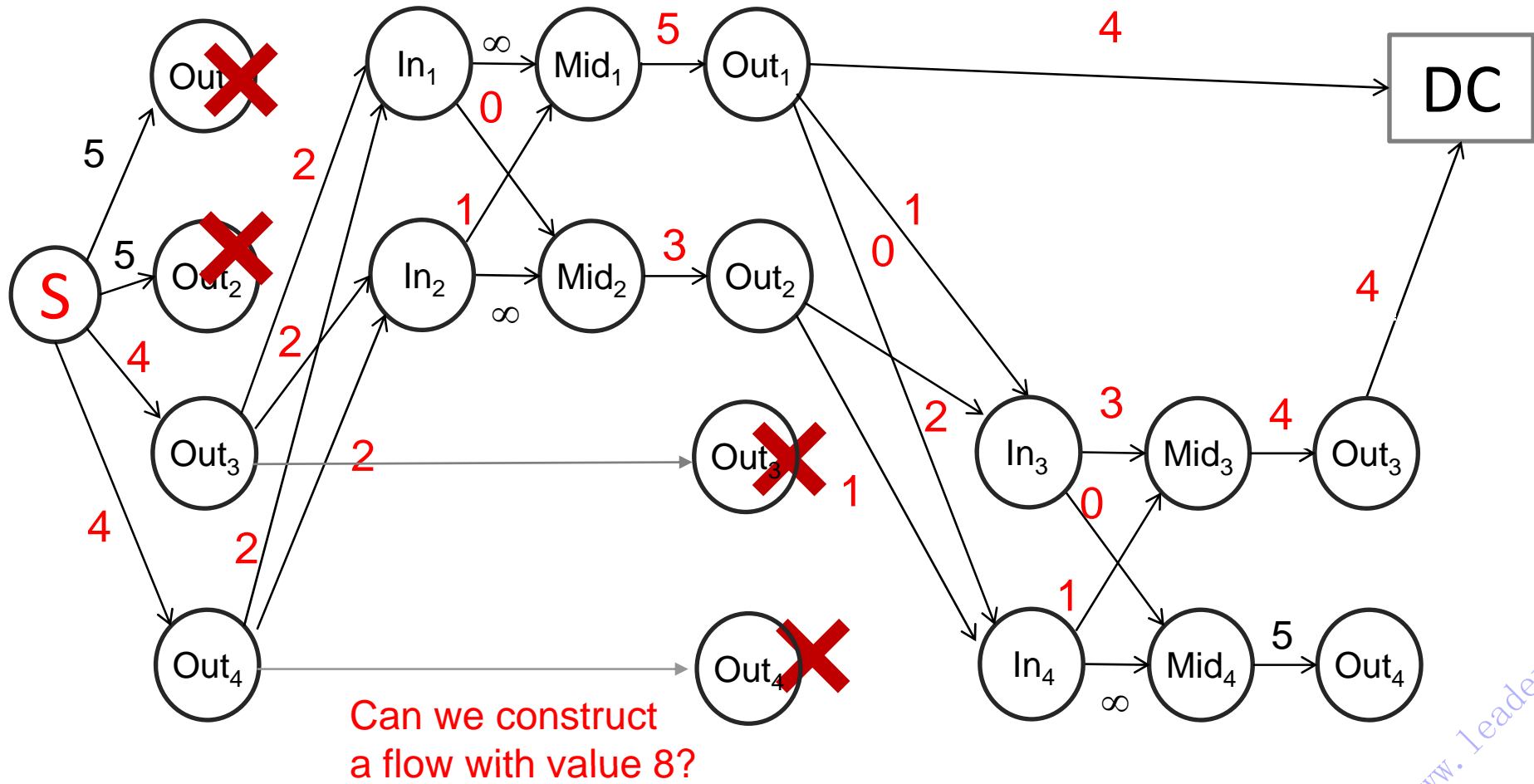
$T_1$  is the “transfer matrix” in stage 1

$$\mathbf{mT}_0T_1T_2$$

$T_2$  is the “transfer matrix” in stage 2

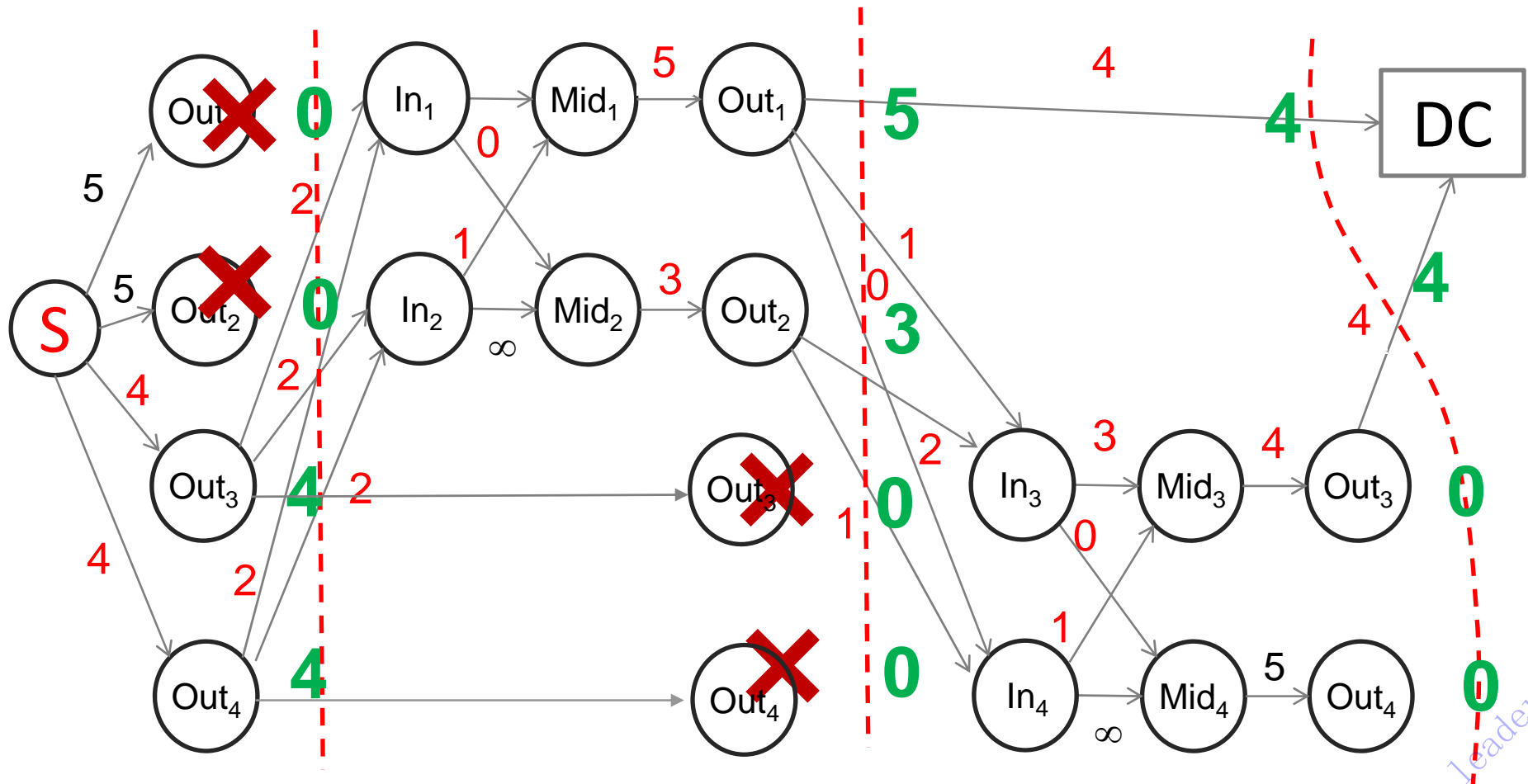
www.leaderstudio.net

# Flow in information flow graph



www.leaderstudio.net

# Cross-sectional flow patterns



www.leaderstudio.net





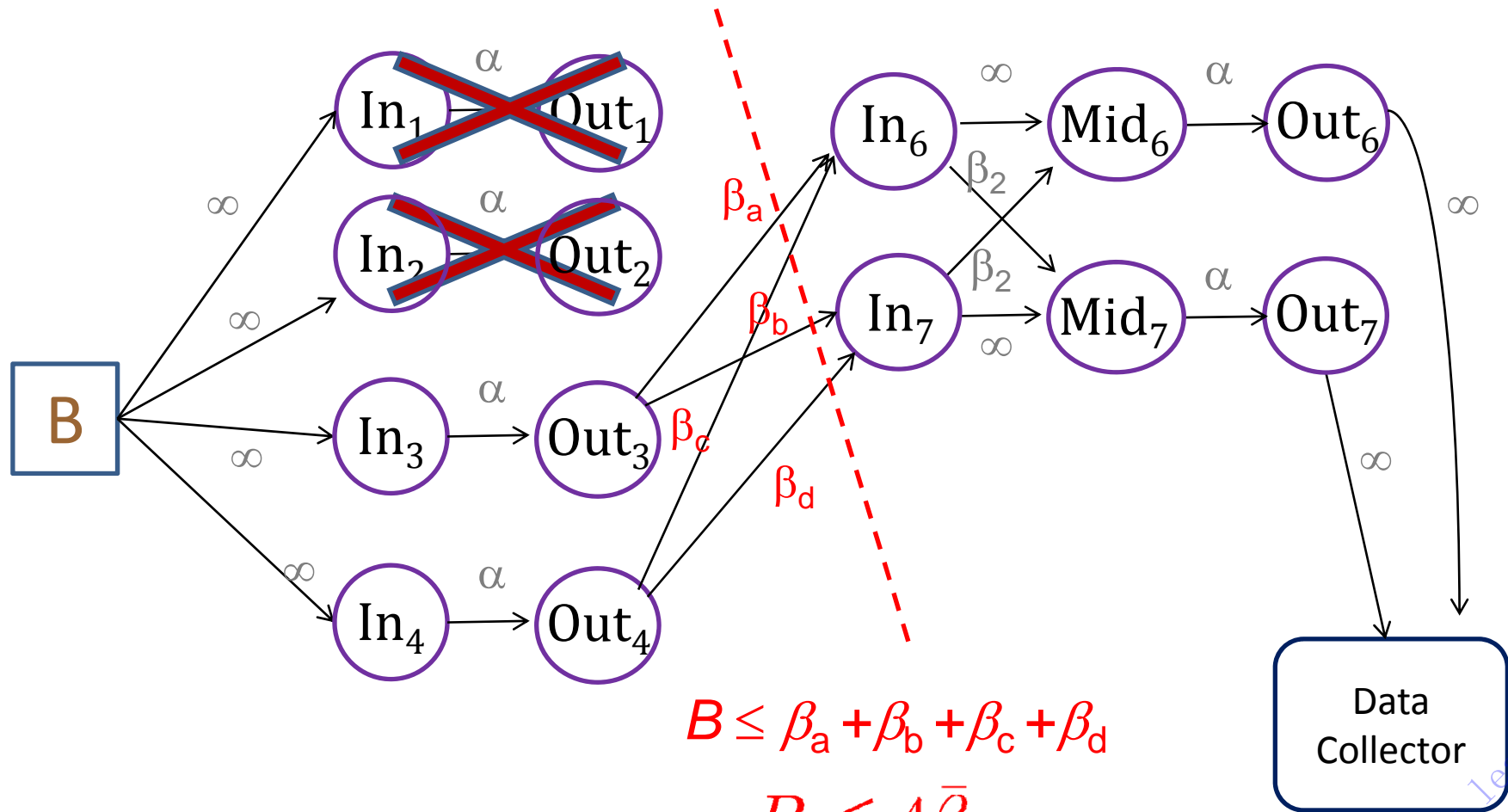
# Conclusion

- For functional repair, we can construct cooperative regenerating codes over a *fixed* finite field, which supports **unlimited** number of repair processes.
  - In the proceeding, the proof for the special case of minimum-repair-bandwidth cooperative codes is given.
- The proof technique exploits the trellis structure of the information flow graph.
  - Analysis on a segment of the graph

# References

- Y. Wu and A. G. Dimakis, *Reducing repair traffic for erasure coding-based storage via interference alignment*, ISIT, Jul, 2009.
- Y. Hu, Y. Xu, X. Wang, C. Zhan and P. Li, *Cooperative recovery of distributed storage systems from multiple losses with network coding*, J. Sel. Area Comm., vol. 28, no. 2, pp.268-275, Feb, 2010.
- K. W. Shum, *Cooperative Regenerating Codes for Distributed Storage Systems*, ICC, Jun, 2011.
- A.-M. Kermarrec and N. Le Scouarnec and G. Straub, *Repairing Multiple Failures with Coordinated and Adaptive Regenerating Codes*, Netcod, Jul, 2011.

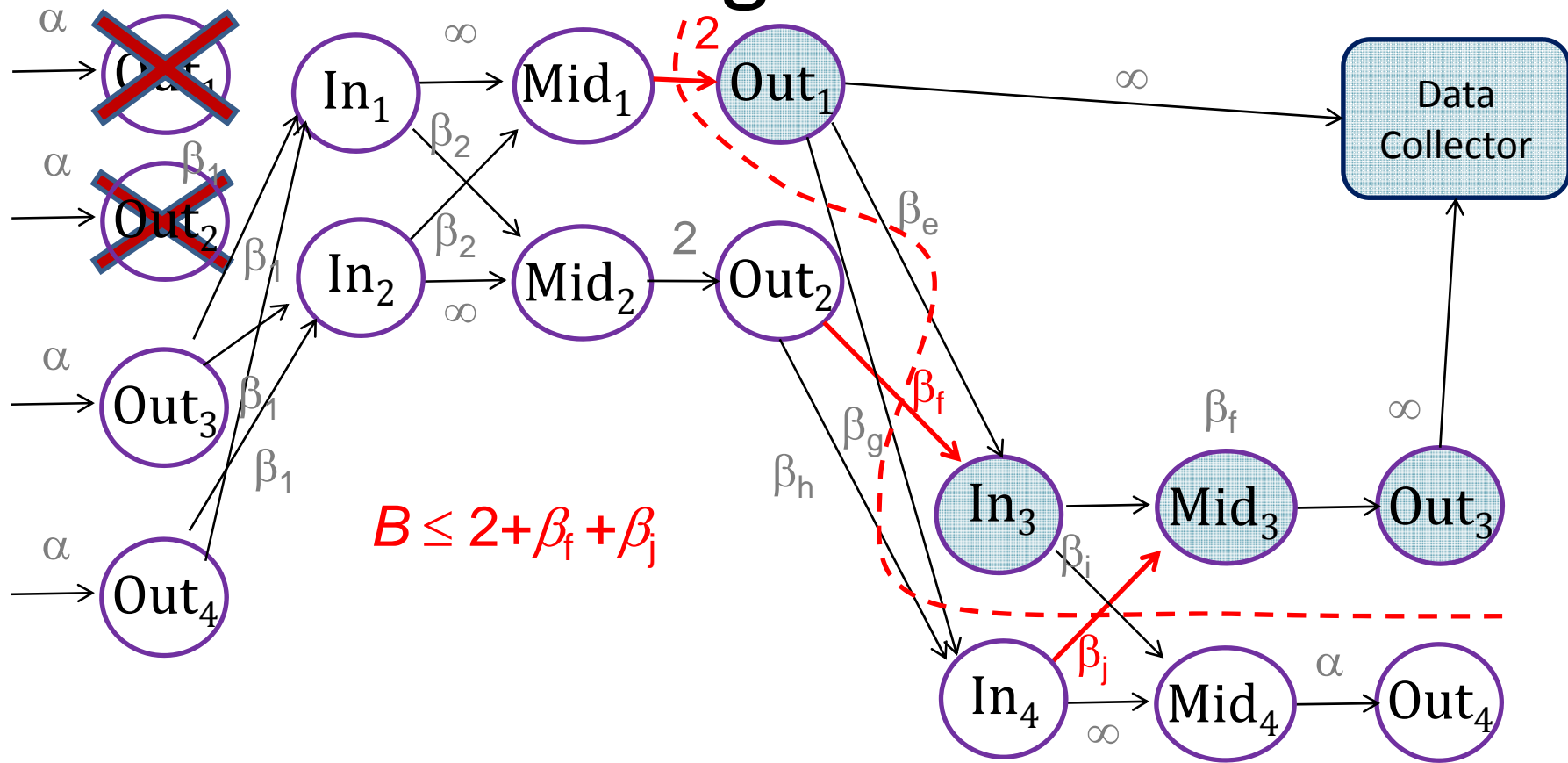
# Non-homogeneous download traffic



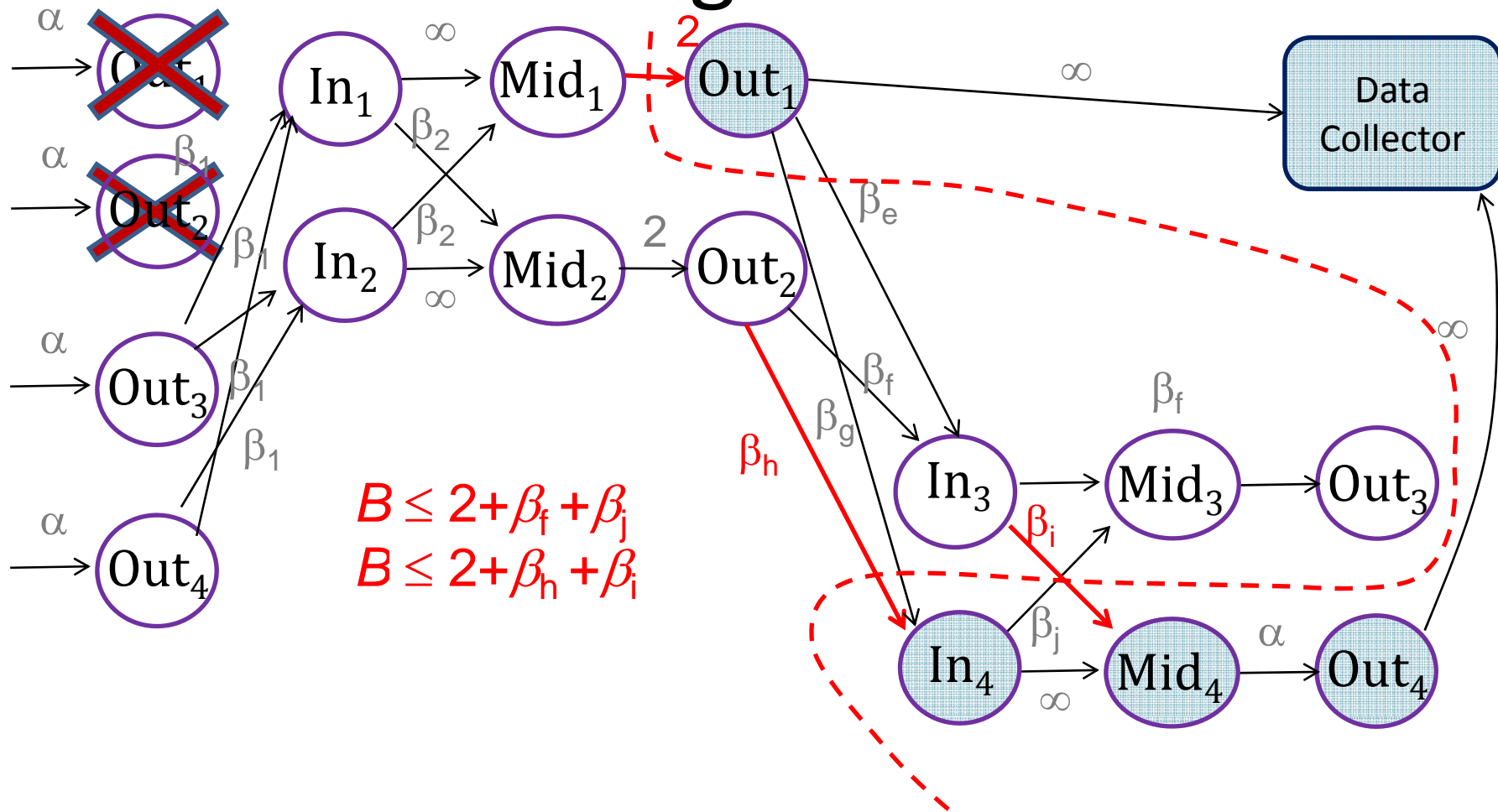
$$B \leq \beta_a + \beta_b + \beta_c + \beta_d$$

$$B \leq 4\bar{\beta}_1$$

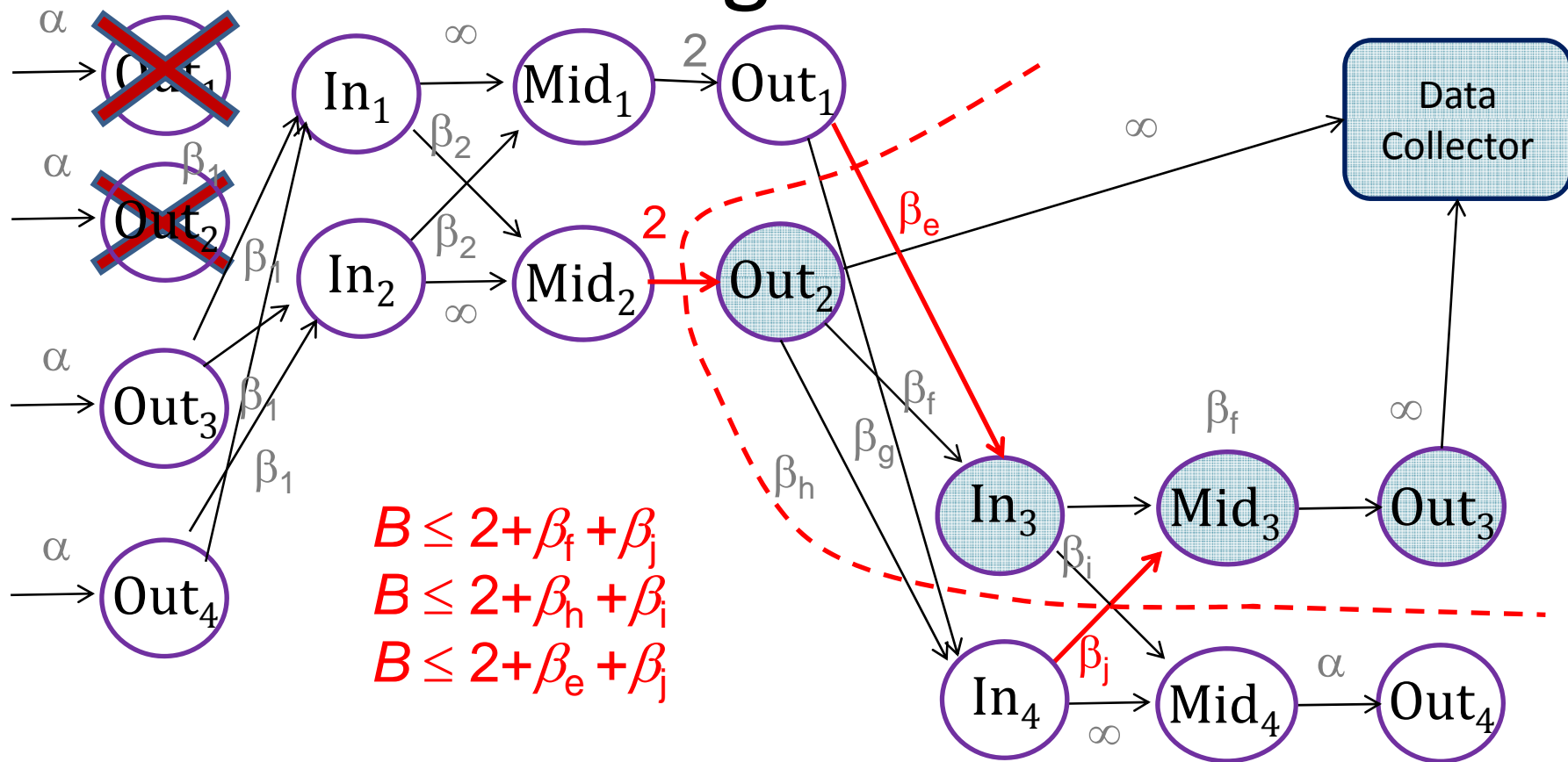
# Non-homogeneous traffic



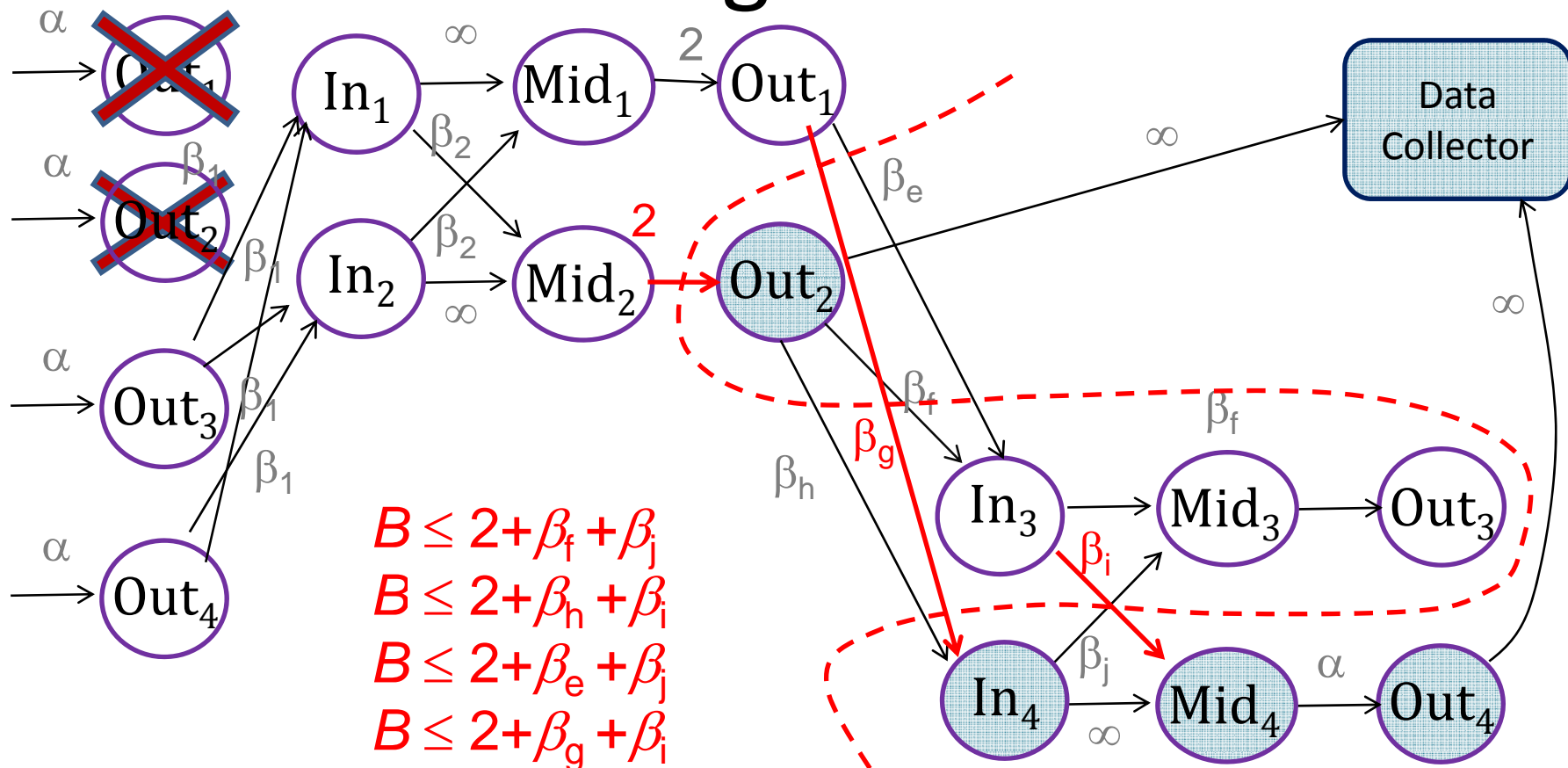
# Non-homogeneous traffic



# Non-homogeneous traffic



# Non-homogeneous traffic



$$\begin{aligned}
 B &\leq 2 + \beta_f + \beta_j \\
 B &\leq 2 + \beta_h + \beta_i \\
 B &\leq 2 + \beta_e + \beta_j \\
 B &\leq 2 + \beta_g + \beta_i
 \end{aligned}$$

$$4B \leq 8 + \beta_e + \beta_f + \beta_g + \beta_h + 2\beta_i + 2\beta_j$$

$$B \leq 2 + \bar{\beta}_1 + \bar{\beta}_2$$

# The same LP problem

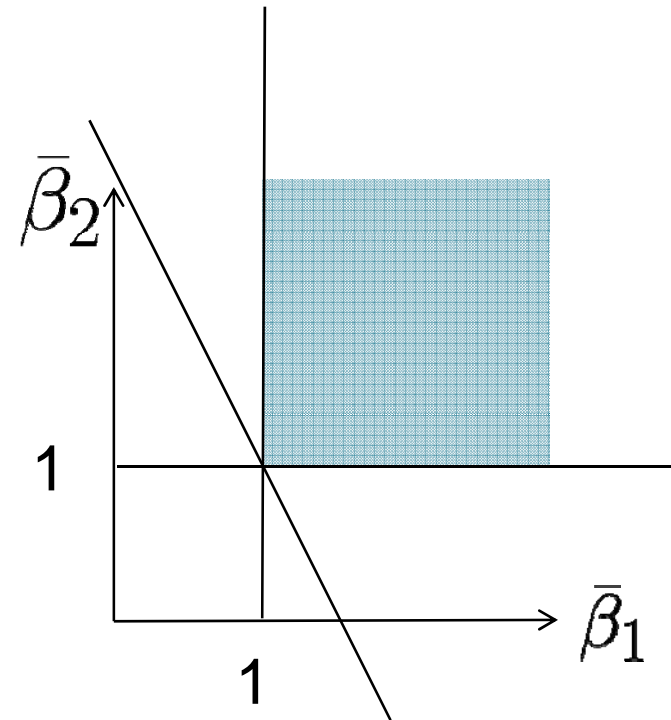
- Minimize  $2\bar{\beta}_1 + \bar{\beta}_2$

- Subject to

$$4 \leq 4\bar{\beta}_1$$

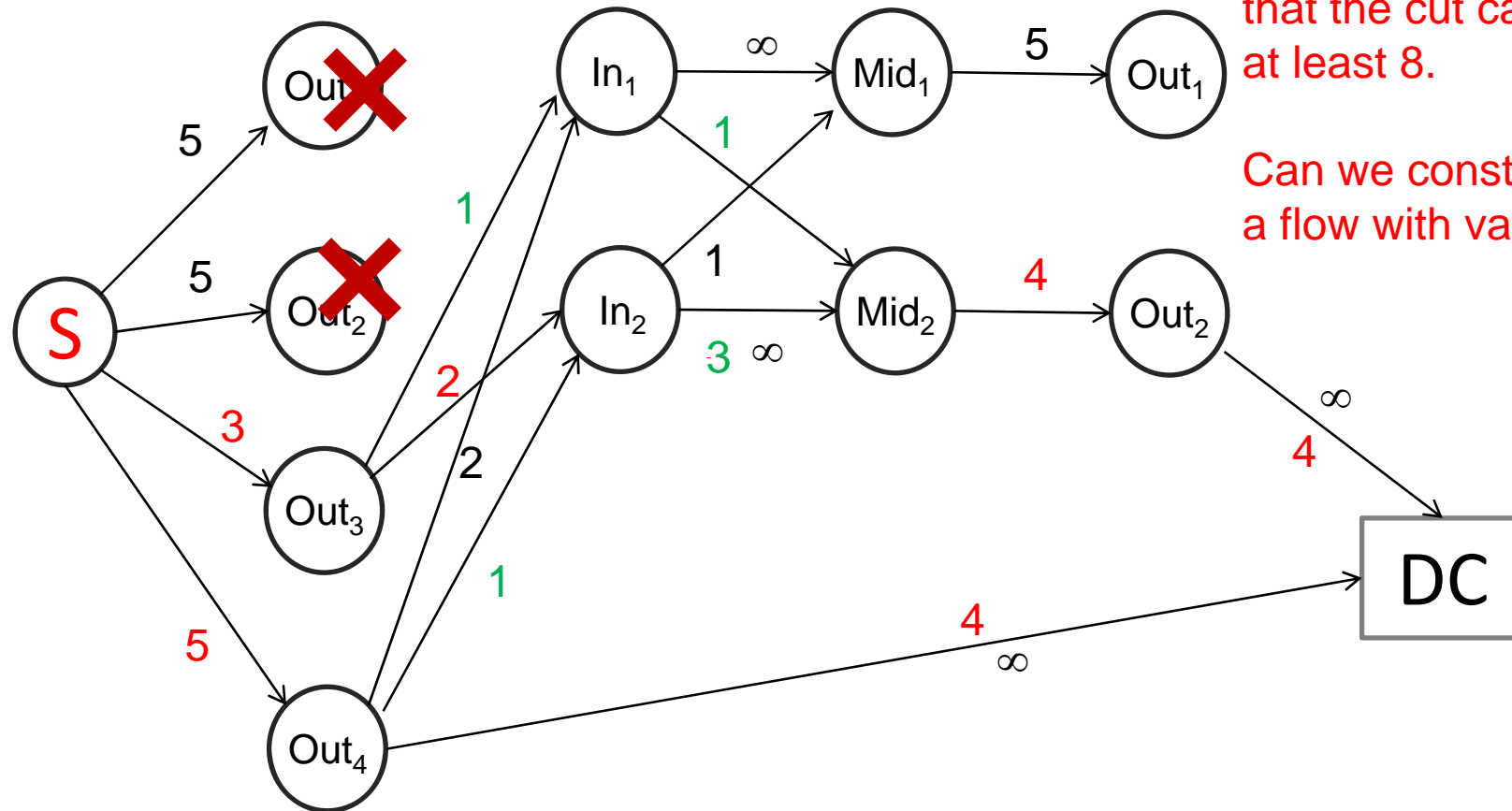
$$4 \leq 2 + \bar{\beta}_1 + \bar{\beta}_2$$

$$\bar{\beta}_1, \bar{\beta}_2 \geq 0$$



$\Rightarrow$  At least 3 packets

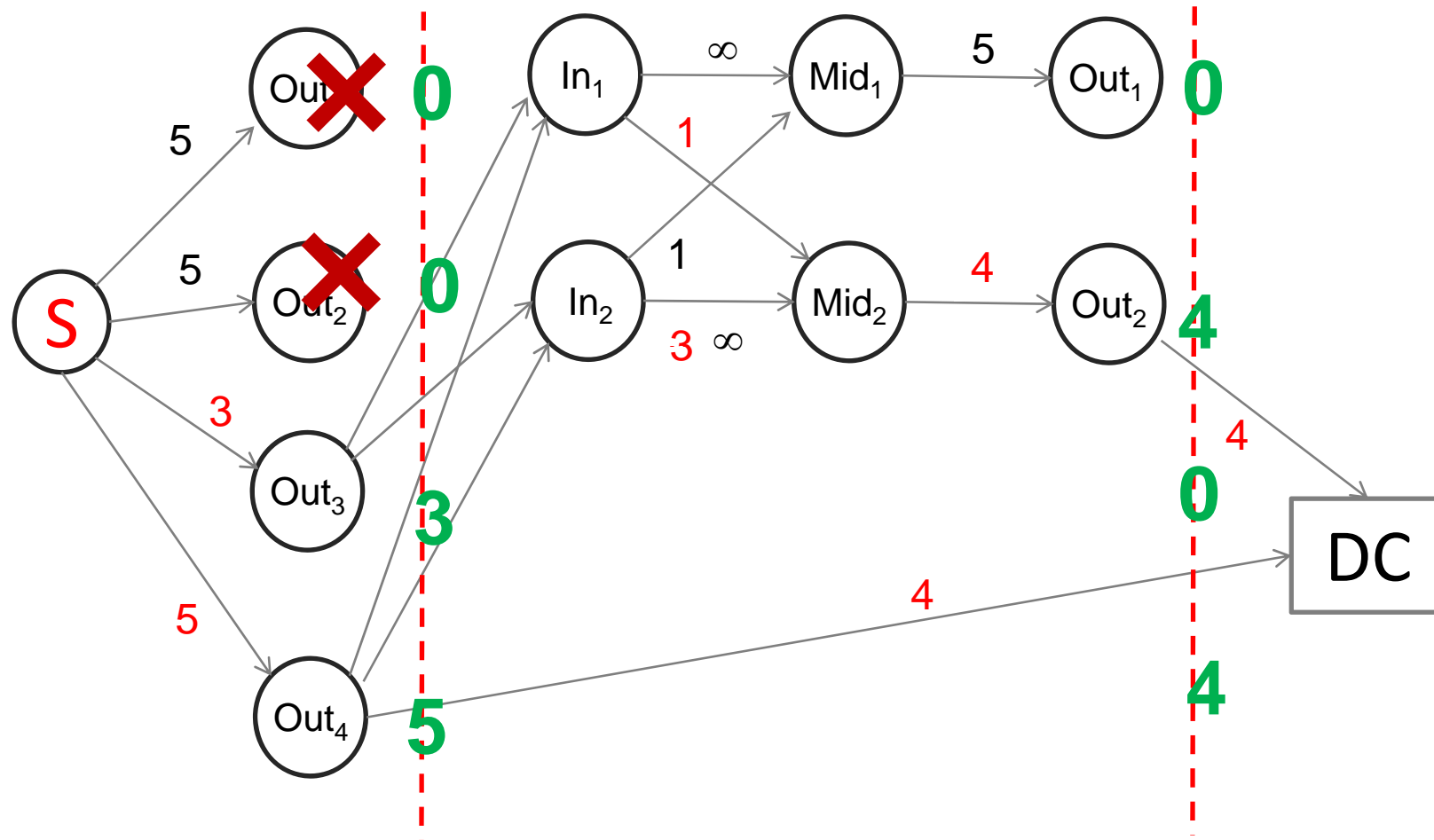
# Flow



The cut-set bound says that the cut capacity is at least 8.

Can we construct a flow with value 8?

# Cross-sectional flow



# Information flow graph

