The Impact of Storage Capacity on End-to-end Delay in Time Varying Networks

George Iosifidis and
Iordanis Koutsopoulos
University of Thessaly, Greece

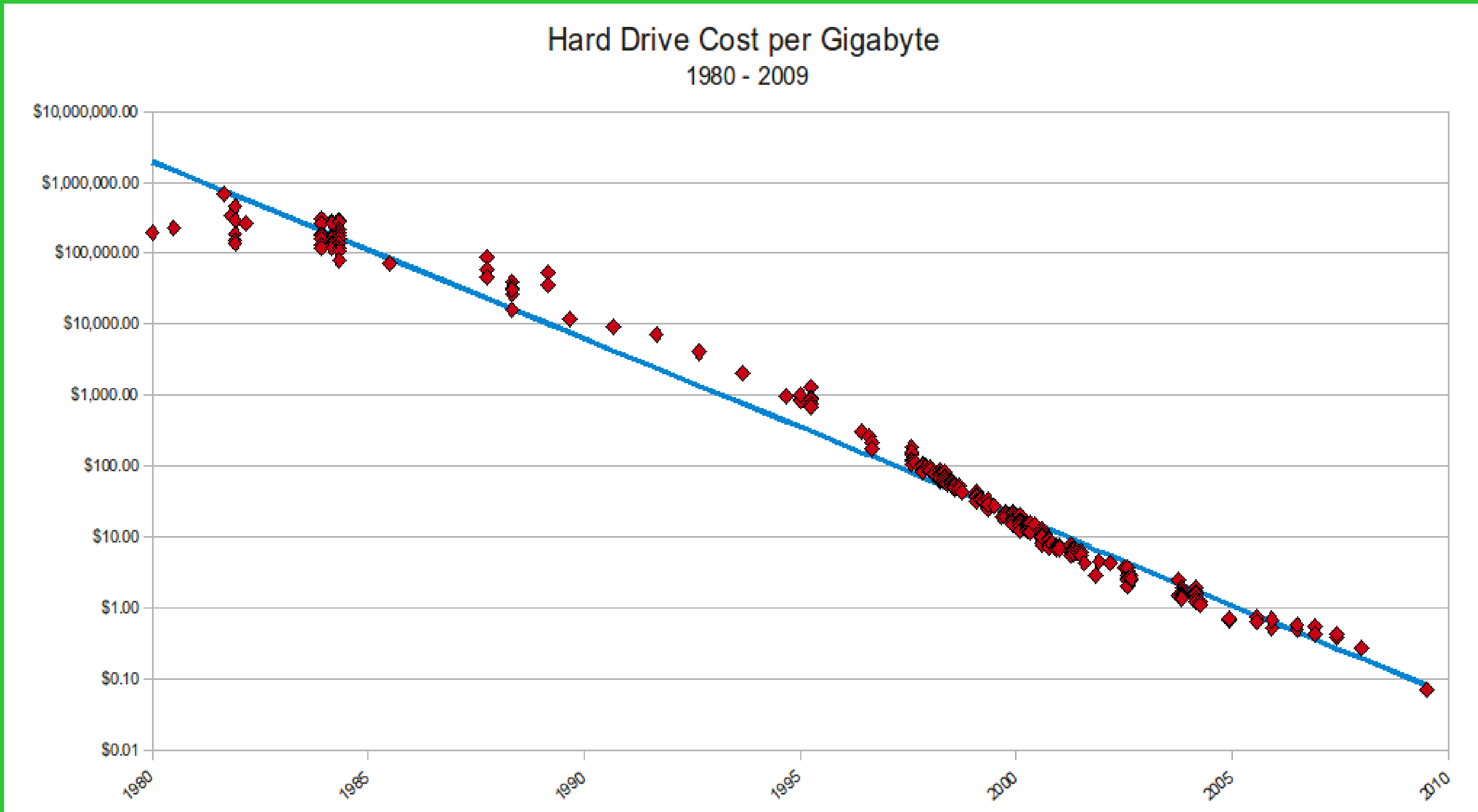
Georgios Smaragdakis
T – Labs/TU Berlin, Germany

Introduction: Motivation

- **<u>Observation #1:</u>** Storage is a cheap and at large scale available resource: <10 cents/gigabyte

Observation #1: Cost of Storage

The cost of the Hard Drive per Gigabyte decreases and now it is less than 10 USD cents.



Introduction: Motivation

- **Observation #1:** Storage is a cheap and at large scale available resource, which means that:
  - Small, portable devices can have significant storage capability
  - Central communication nodes can store enormous amount of data

- **Observation #2:** In networks, very often, links capacity varies with time

Introduction: Motivation

- **<u>Observation #1:</u>** Storage is a cheap and at large scale available resource, which means that:
  - Small, portable devices can have significant storage capability
  - Central communication nodes can store enormous amount of data

- **<u>Observation #2:</u>** In networks, very often, links capacity varies with time:
  - Link temporal failures, wireless channel impairments, etc
  - Price of capacity changes , e.g. links capacity is expensive in the rush hours .

Introduction: Motivation

- **Observation #1:** Storage is a cheap and at large scale available resource.

- **Observation #2:** In networks, very often, links capacity varies with time.

- Given the above observations we ask:

  – Can we use storage in order to improve the end-to-end delay in time varying (dynamic) networks?

Introduction: Motivation

- **Observation #1:** Storage is a cheap and at large scale available resource.

- **Observation #2:** In networks, very often, links capacity varies with time

- Given the above observations we ask:

  – Can we use storage in order to improve the end-to-end delay in time varying (dynamic) networks?

Introduction: Motivation

- **Observation #1:** Storage is a cheap and at large scale available resource.

- **Observation #2:** In networks, very often, links capacity varies with time

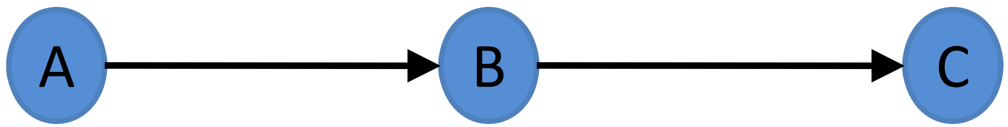
- Given the above observations we ask:

  – Can we use storage in order to improve the end-to-end delay in time varying (dynamic) networks?

  or, equivalently, to increase the amount of conveyed data within a given time interval?

Introduction: A Simple Example

- Consider a 3-nodes linear (tandem) network:

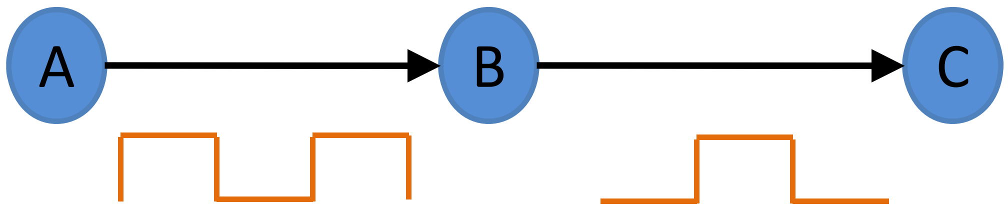


- Slotted Time: n={1, 2, 3, ….., T}, Link Traversal time = 1 slot
- C_{AB}= (D, D, 1, 1, D, D, 1, 1,….), C_{BC}= (D, 1, 1, D, D, 1, 1, D, D, ….)

How many slots are required for the transfer of D packets?

Introduction: A Simple Example

- Consider a 3-nodes linear (tandem) network:

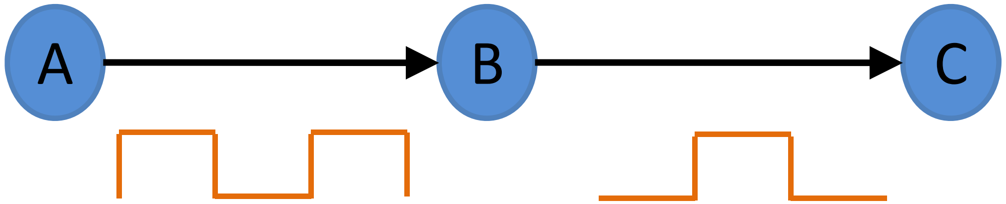


- Slotted Time: n={1, 2, 3, ….., T}, Link Traversal time = 1 slot
- C_{AB}= (D, D, 1, 1, D, D, 1, 1,….), C_{BC}= (D, 1, 1, D, D, 1, 1, D, D, ….)

How many slots are required for the transfer of D packets?

Introduction: A Simple Example

- Consider a 3-nodes linear (tandem) network:

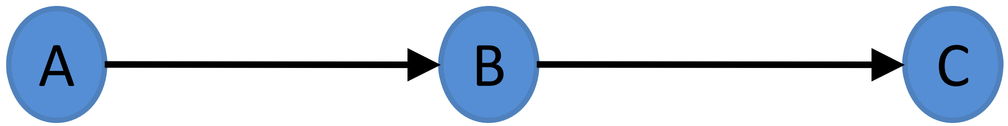


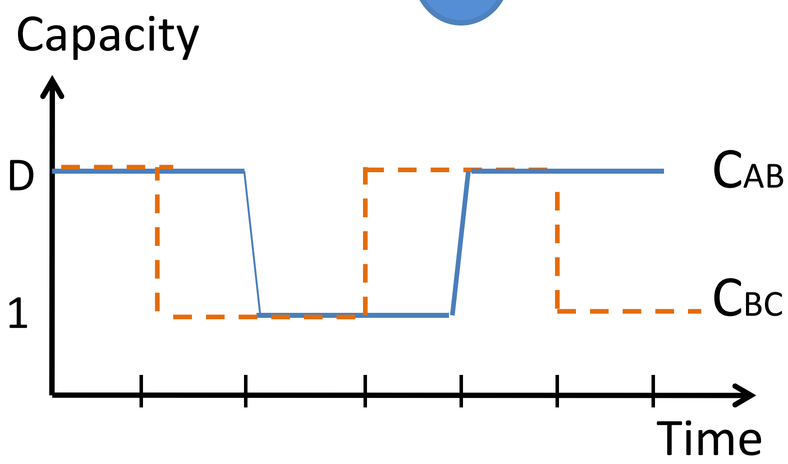
- Slotted Time: n={1, 2, 3, ….., T}, Link Traversal time = 1 slot
- C_{AB}= (D, D, 1, 1, D, D, 1, 1,….), C_{BC}= (D, 1, 1, D, D, 1, 1, D, D, ….)

How many slots are required for the transfer of D packets?

Introduction: A Simple Example

- ***How many slots are required for the transfer of D packets?***





In each slot, node A is able to push only as many packets as node B is capable to forward in the next time slot.

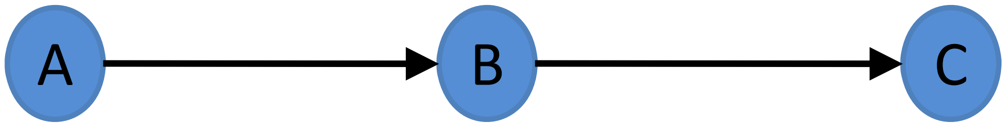
→The "end-to-end" capacity is limited by the lowest link's capacity:

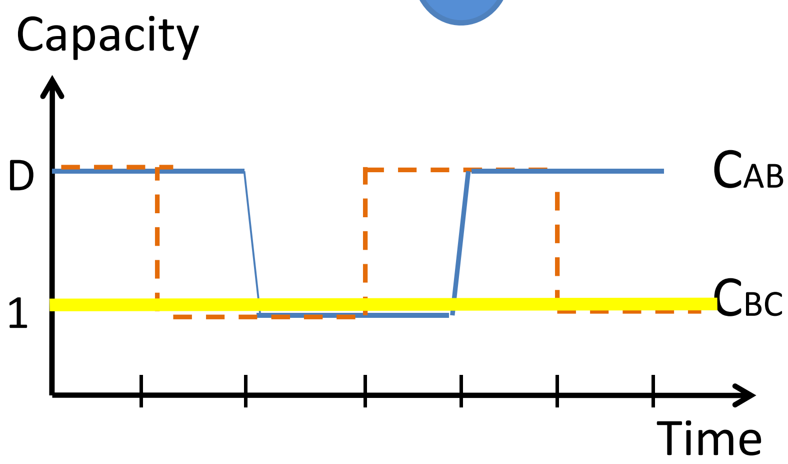
$$C_{AC}(n) = \min\{C_{AB}(n), C_{BC}(n+1)\}$$

Answer : D+1 slots

Introduction: A Simple Example

- ***How many slots are required for the transfer of D packets?***





In each slot, node A is able to push only as many packets as node B is capable to forward in the next time slot.

→The "end-to-end" capacity is limited by the lowest link's capacity:

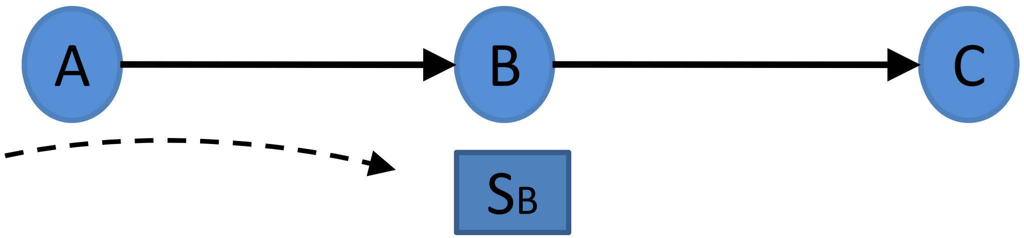
$$C_{AC}(n) = \min\{C_{AB}(n), C_{BC}(n+1)\}$$

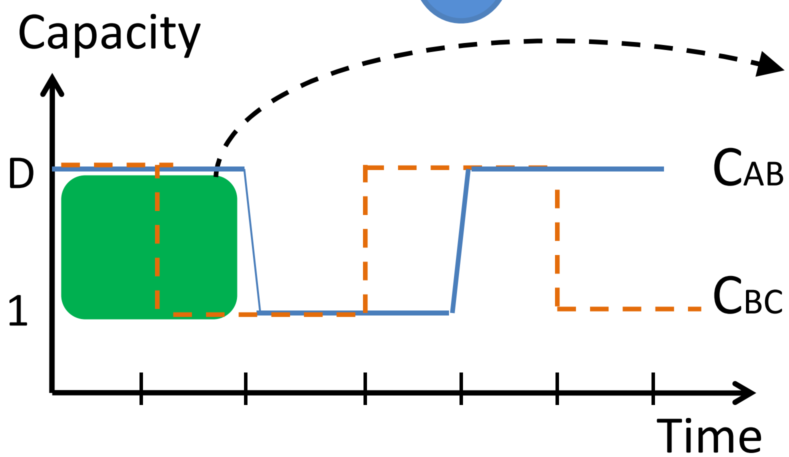
Answer : D+1 slots

Introduction: A Simple Example

- ***How many slots are required for the transfer of D packets?***

→ Assume that node B has storage capability of S_B > D packets





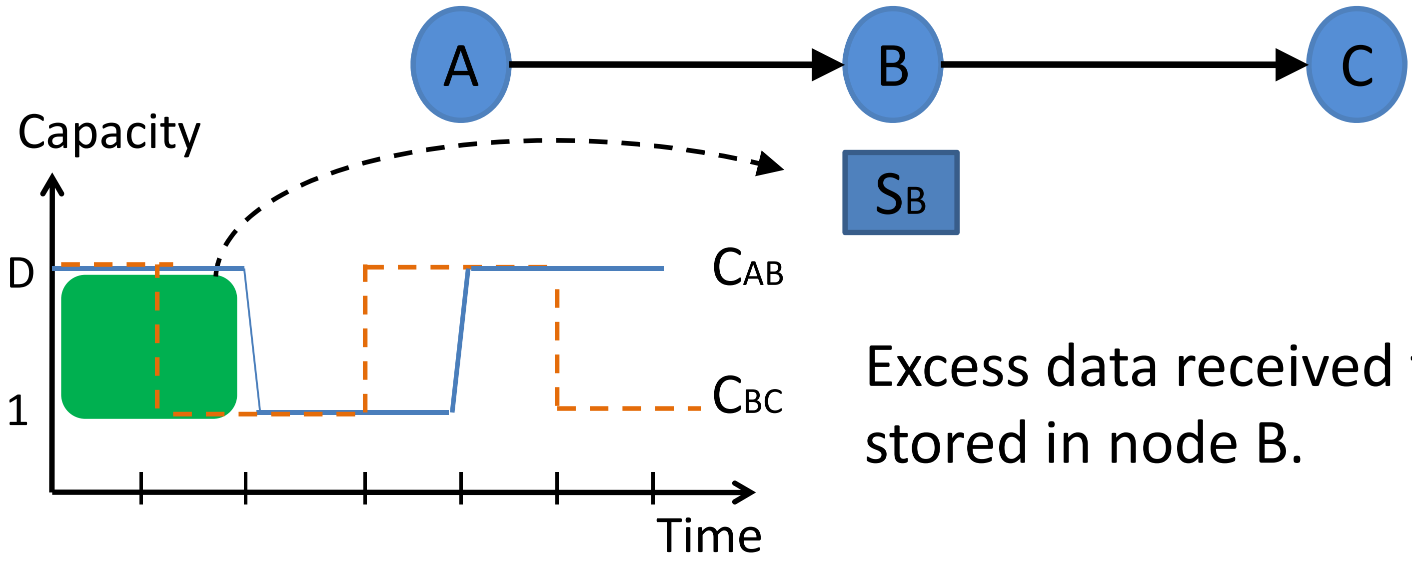
Excess data received from node A is stored in node B.

Answer : 2 slots

Introduction: A Simple Example

- ***How many slots are required for the transfer of D packets?***

→ Assume that node B has storage capability of S_B > D units



Excess data received from node A is stored in node B.

Answer : 2 slots

Storage decreases the required transfer time

Related Work

- **Delay Tolerant Networks**: to alleviate intermittent connectivity problems.
  - S. Jain et. Al, "Routing in a Delay Tolerant Network", ACM SIGCOMM, 2004
- **Cost Minimization for Bulk Data Transfer**: to minimize the monetary transfer cost.
  - N. Laoutaris, G. Smaragdakis, et. Al. " Delay Tolerant Bulk Data Transfers on the Internet", ACM SIGMETRICS, 2009.
- **Theoretical Models** where storage is consider as a routing option.
  - A. Orda, et. al., "Minimum Delay Routing in Stochastic Networks", IEEE/ACM ToN, 1993.

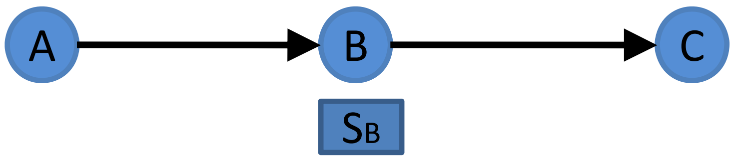
Contributions

- These related works:
  - Do not focus on the impact of storage on the network performance
  - The performance metric is not the delay (delay tolerant nets).
  - Solutions are not distributed.
- In this paper:
  - We study the storage management problem in linear networks
  - We extend the study in general network graphs
  - We define and solve the joint storage management and routing problem

Storage Management for Linear Networks

Impact of Storage Capacity in Linear Networks

- **Q:** What is the optimal storage policy and how much we can benefit from storage?



  - **Policy:** Push as many packets as possible and store the rest.
  - End-to-end capacity in each slot, without storage:

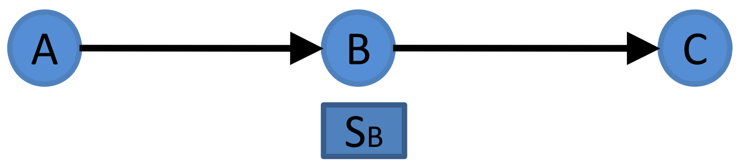
$$C_{AC}(n) = \min\{C_{AB}(n), C_{BC}(n+1)\}$$

  - End-to-end capacity in each slot, with storage:

$$C^{S}_{AC}(n) = \min\{C_{BC}(n+1), X_B(n+1)\}$$

Impact of Storage Capacity in Linear Networks

- **Q:** What is the storage policy and how much we can benefit from storage?



  - **Policy:** Push as many packets as possible and store the rest.
  - End-to-end capacity in each slot, <u>without</u> storage:

$$C_{AC}(n) = \min\{C_{AB}(n), C_{BC}(n+1)\}$$

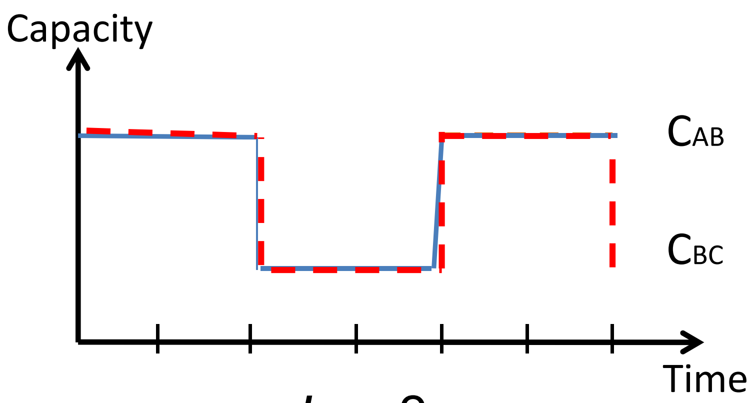
  - End-to-end capacity in each slot, <u>with</u> storage:

$$C_{AC}^{S}(n) = \min\{C_{BC}(n+1), X_{B}(n+1)\}$$

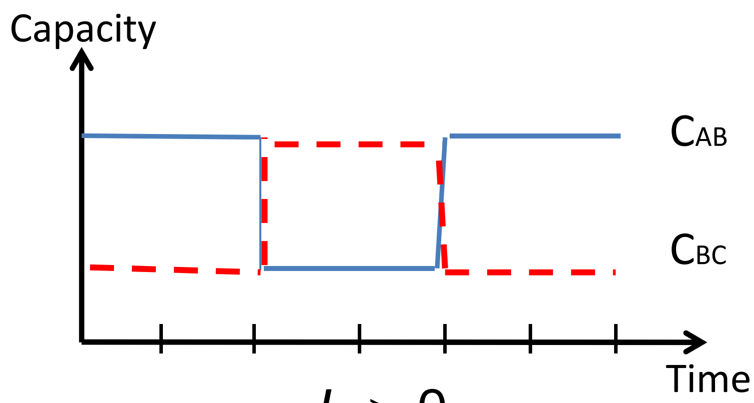
The available data at the last node before the destination

Impact of Storage Capacity in Linear Networks

- **Conclusion**: The actual delay reduction or throughput increase, depends on the relative variation pattern of the links capacities.
  - We define the *Dissimilarity Index* ***L*** to quantify this variation.



$L = 0$



$L > 0$

Impact of Storage Capacity in Linear Networks

- **Conclusion**: The actual delay reduction or throughput increase, depends on the relative variation pattern of the links capacities.
  - We define the *Dissimilarity Index* ***L*** to quantify this variation.

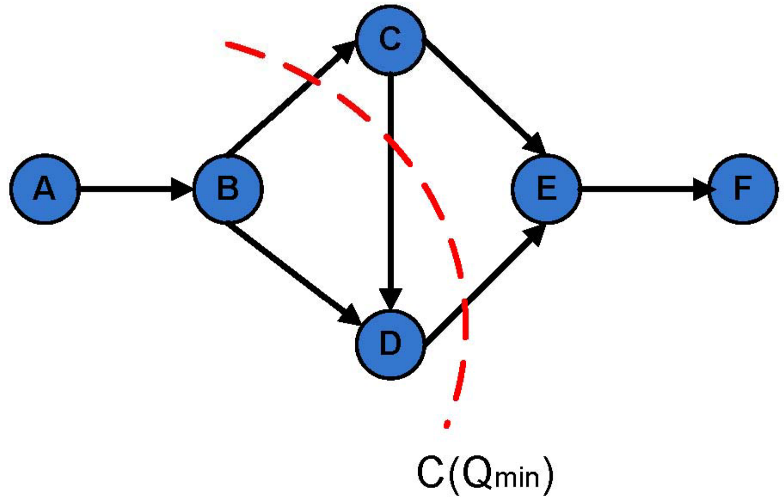
Storage is useless when **L=0.** That is, when:

- Both links capacities do not change, or they change following the same pattern.
- The links capacities change but the second link is always the bottleneck.

Storage Management Policy for General Network Graphs

Storage Management Policy for General Networks

- **Optimal Storage Management Policy:** In which nodes, when and how much to store.
- Performance upper bound of a network: Capacity of the Min-Cut $C(Q_{min})$
  - E.g. Transferred amount in T slots: $T \times C(Q_{min})$



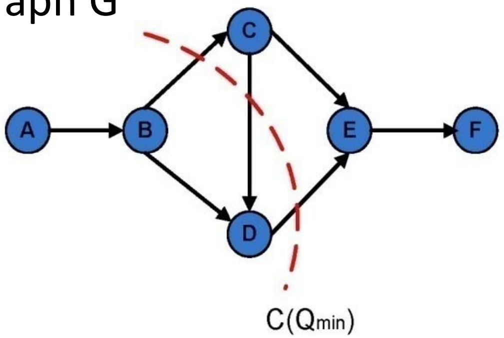
Min Cut: Q_{min}=[W, N\W]

- **Observation #3**: For many networks, it is possible to know or predict the future values of their links capacities.

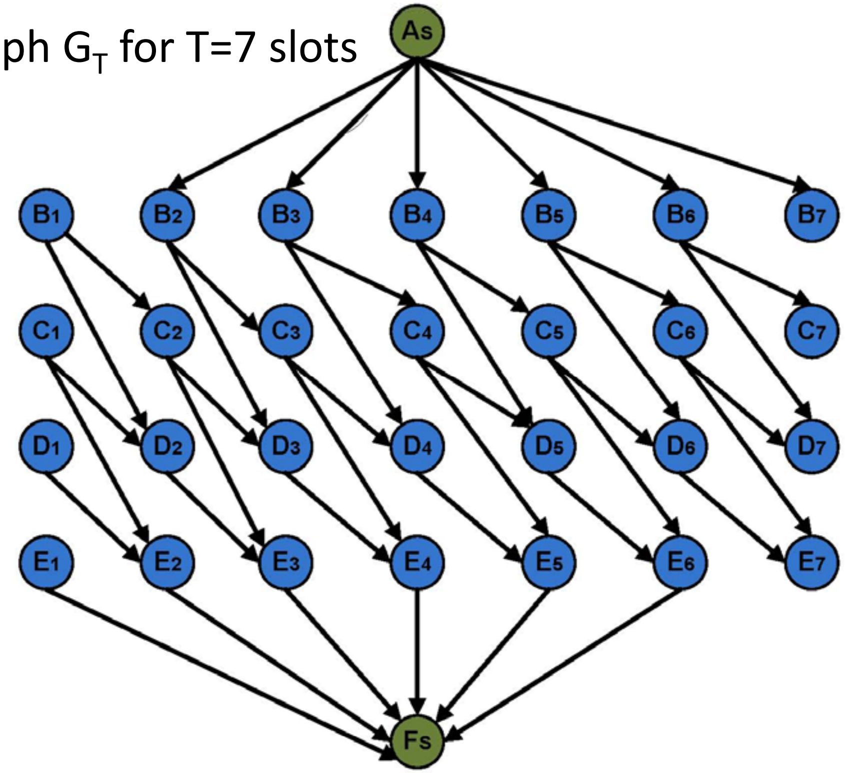
Storage Management Policy for General Networks

- **Time-Expanded Graphs**: Incorporate the notion of time in the network graph

Initial Graph G



Expanded Graph G_T for T=7 slots



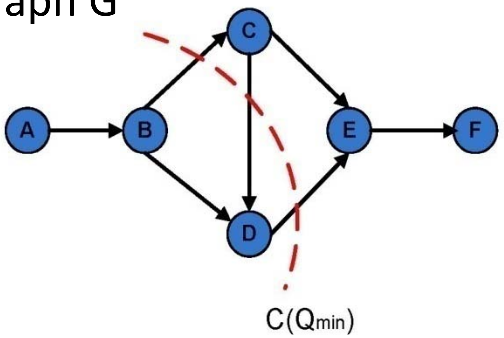
Expansion:

- For every node in G we add T nodes
- For every arc (i,j) in G we add T-1 links $(i^{(t)}, j^{(t+1)})$

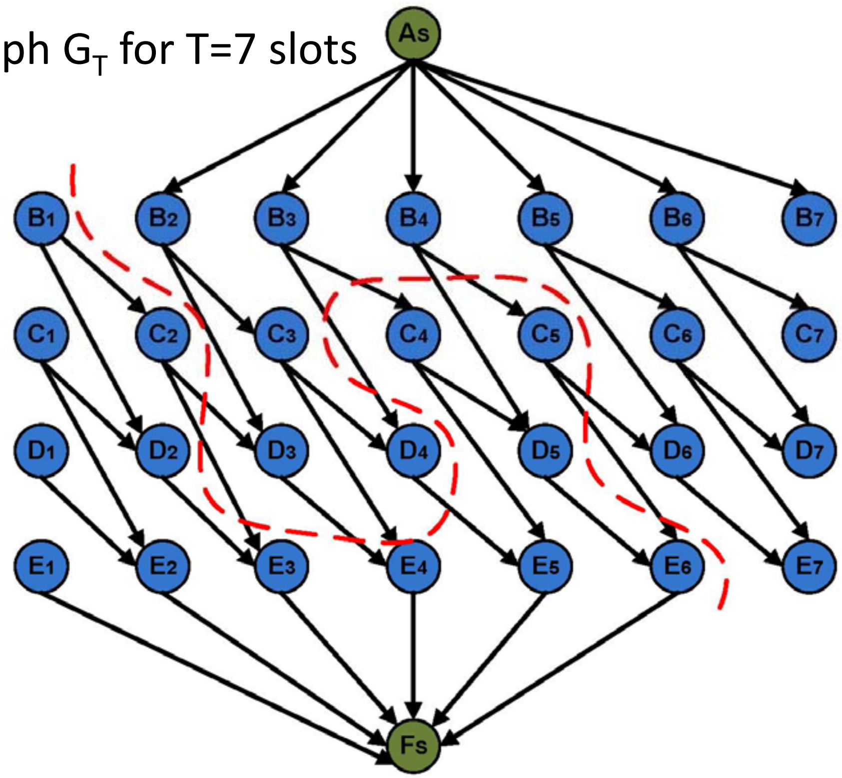
Storage Management Policy for General Networks

- **Time-Expanded Graphs**: Incorporate the notion of time in the network graph

Initial Graph G



Expanded Graph G_T for T=7 slots

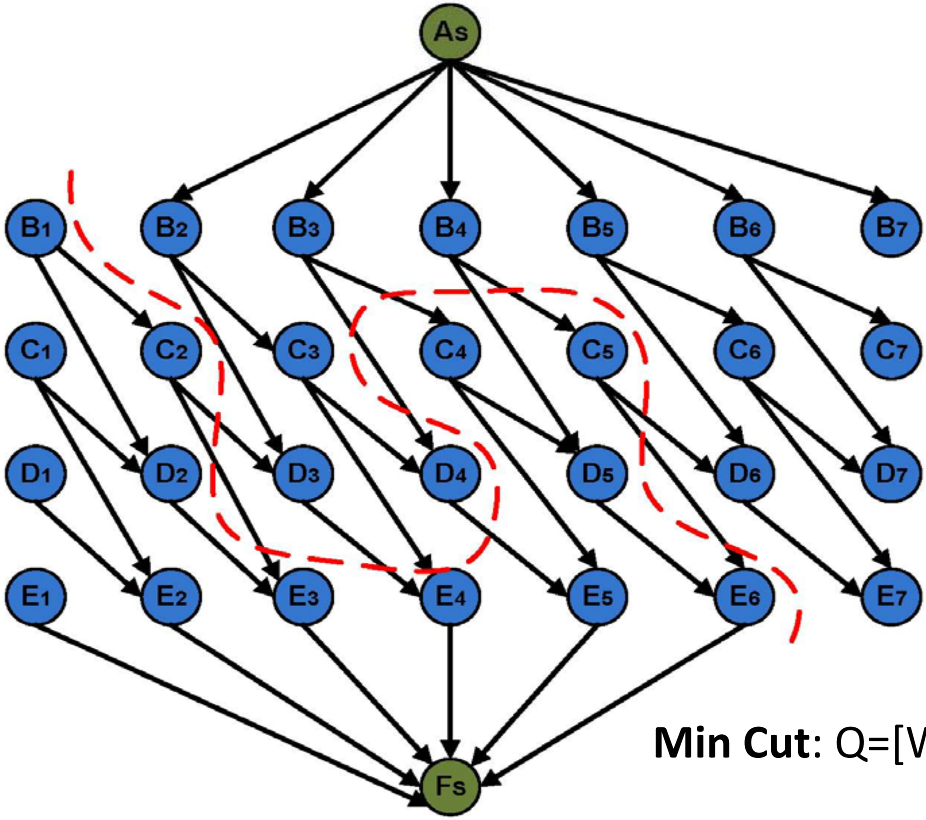


Minimum Cut of the G_T : $C(Q_T)$ is the maximum amount of data that can be conveyed from source to destination in T slots

Min Cut: $Q=[W^T, N\backslash W^T]$

Storage Management Policy for General Networks

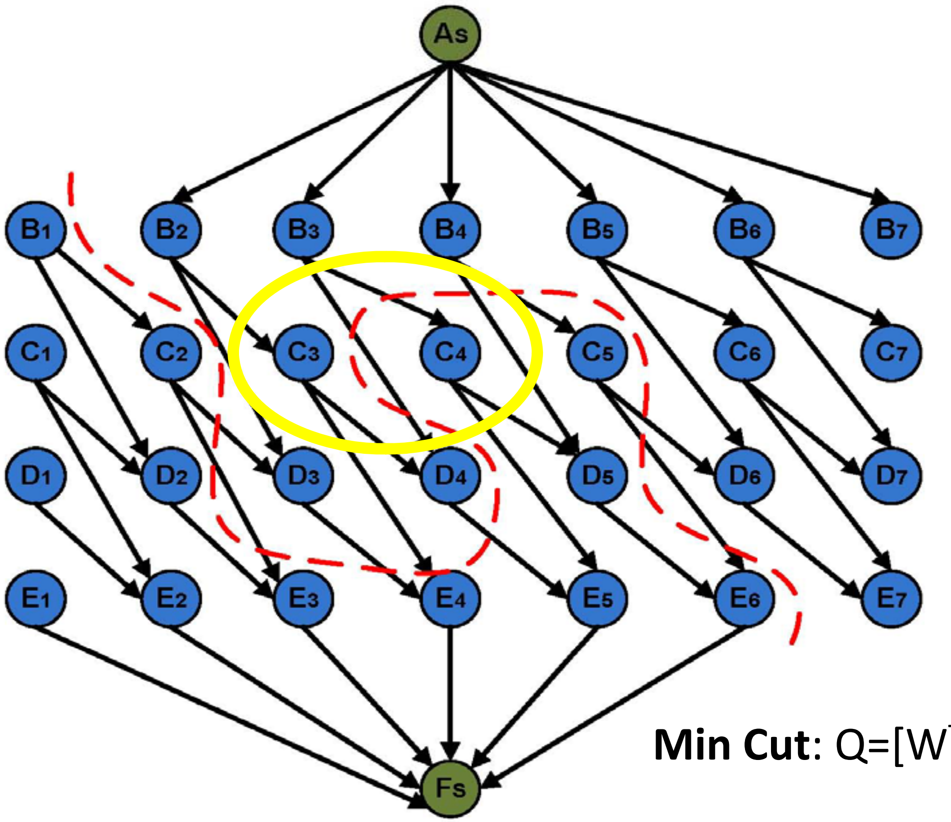
- **Goal**: Increase the min cut by adding links of special type → ***storage links***



Min Cut: $Q=[W^T, N\backslash W^T]$

Storage Management Policy for General Networks

- **Goal**: Increase the min cut by adding links of special type → ***storage links***

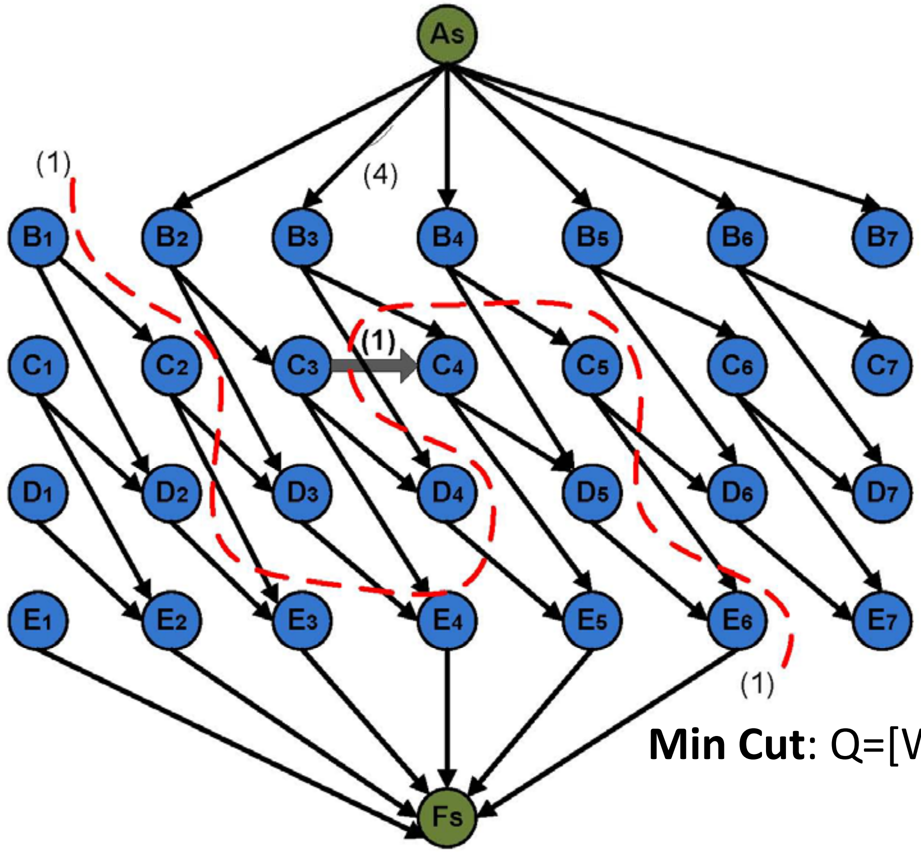


Min Cut: $Q=[W^T, N\backslash W^T]$

Observation: If the time instance (t) of a node belongs to the set W^T and the time instance (t+1) belongs to the set $N\backslash W^T$, then we can add a link to connect these 2 nodes.

Storage Management Policy for General Networks

- **Goal**: Increase the min cut by adding links of special type → ***storage links***

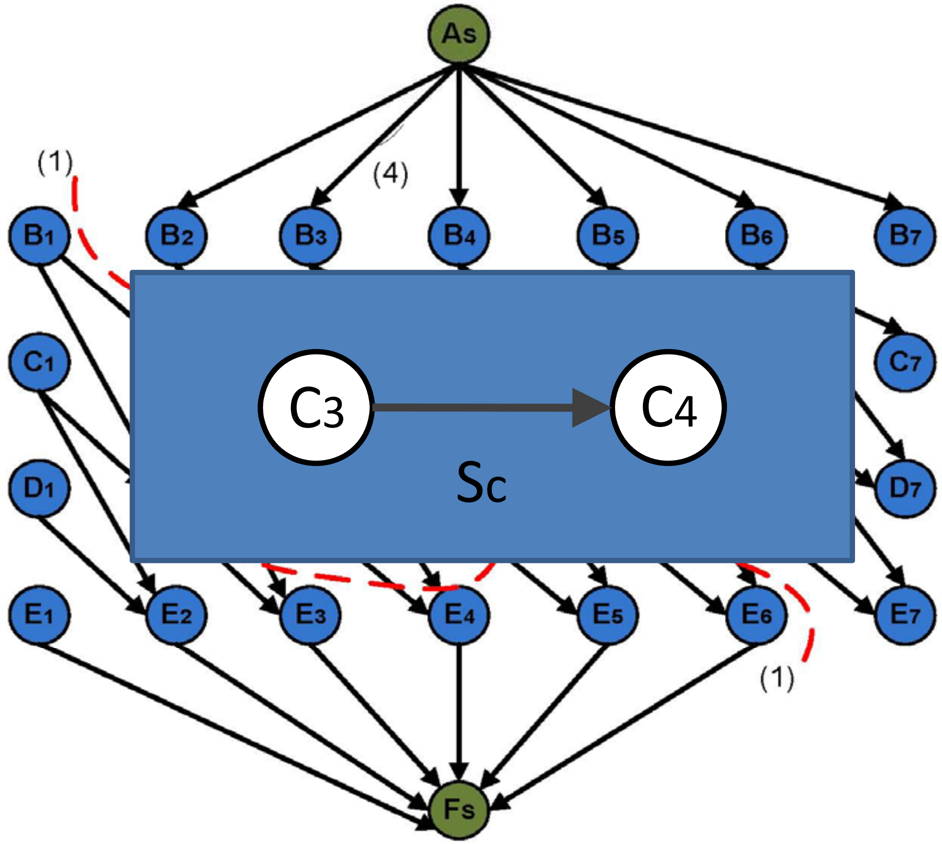


Min Cut: $Q=[W^T, N\backslash W^T]$

Observation: If the time instance (t) of a node belongs to the set W and the time instance (t+1) belongs to the set N\W, then we can add a link to connect these 2 nodes

Storage Management Policy for General Networks

- **Goal**: Increase the min cut by adding links of special type → ***storage links***

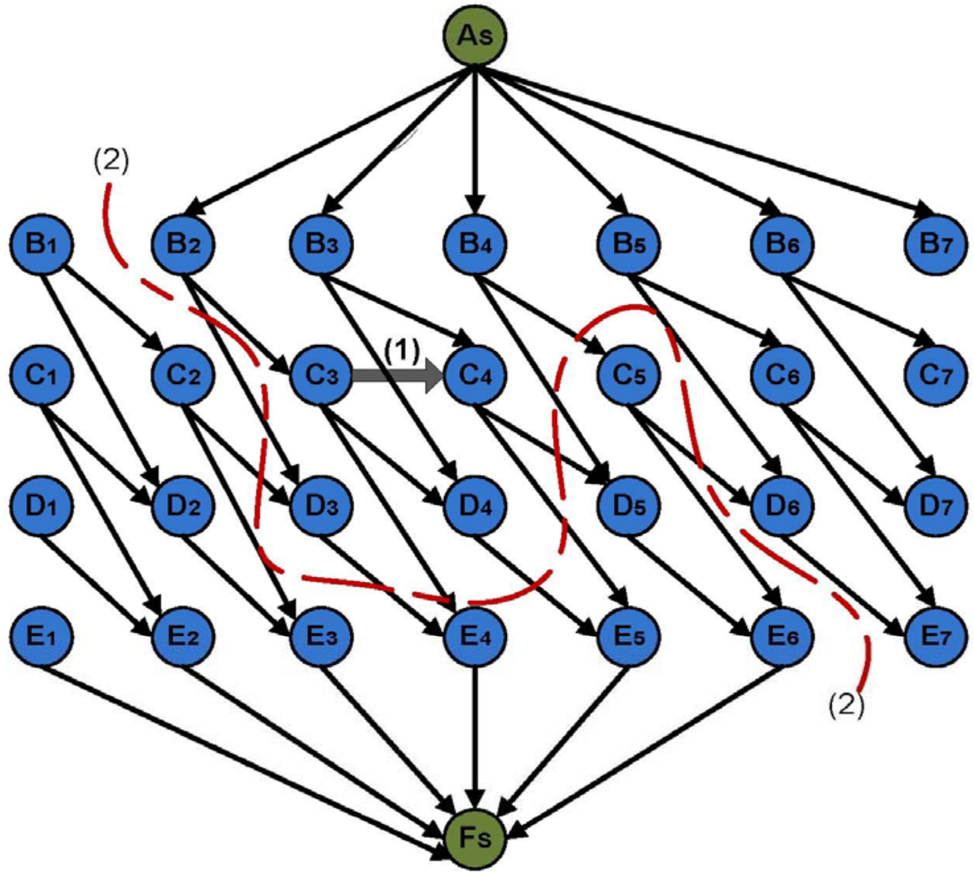


We add a storage link of Sc3 units

Observation: If the time instance (t) of a node belongs to the set W and the time instance (t+1) belongs to the set N\W, then we can add a link to connect these 2 nodes

Storage Management Policy for General Networks

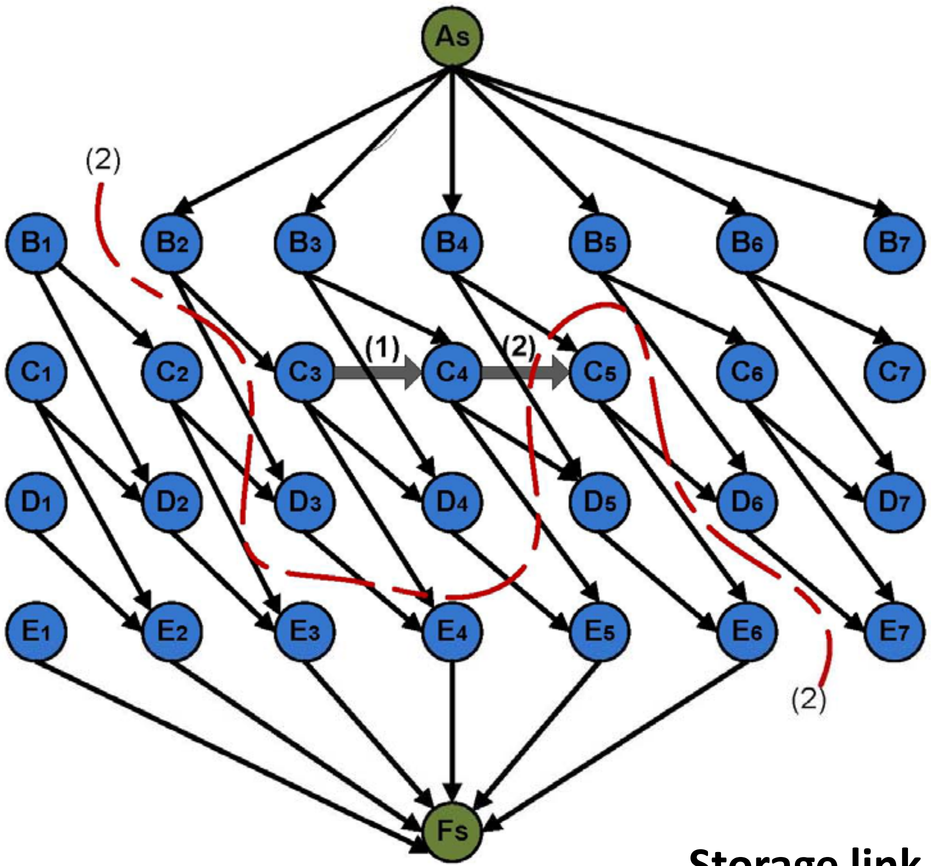
- **Algorithm 1:** Storage Management Policy – Iteratively Increasing the Min Cut



New **Min Cut**
Q_2=Q+Sc3

Storage Management Policy for General Networks

- **Algorithm 1:** Storage Management Policy – Iteratively Increasing the Min Cut

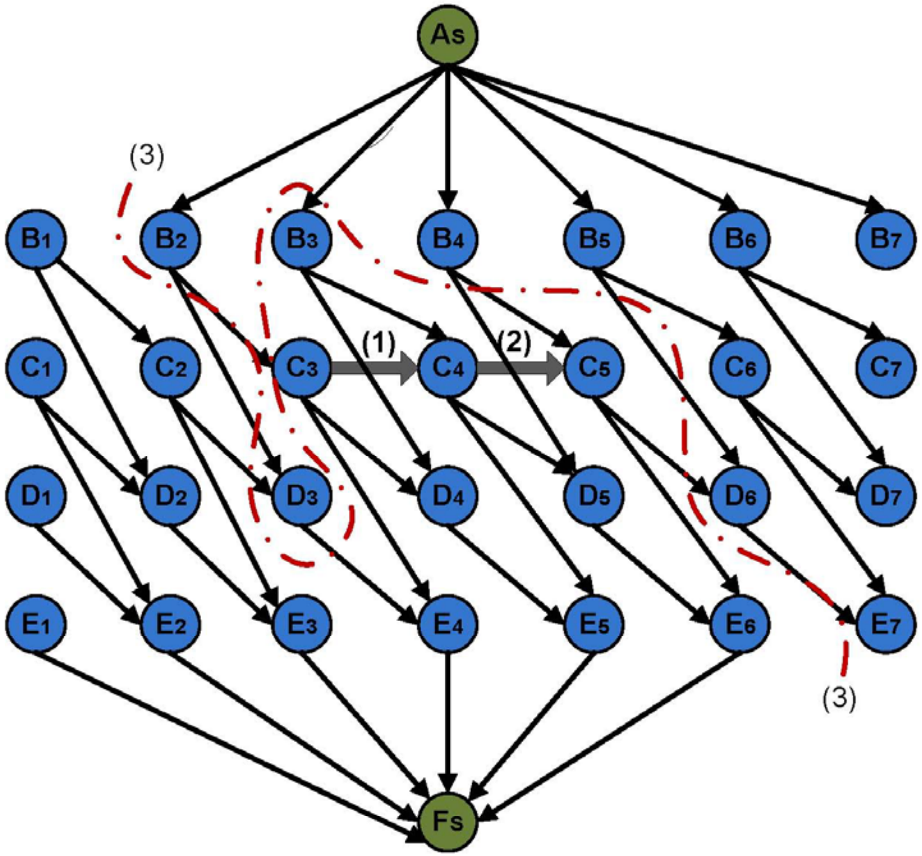


We add a storage link of S_{C4} units

Storage link C_4 - C_5

Storage Management Policy for General Networks

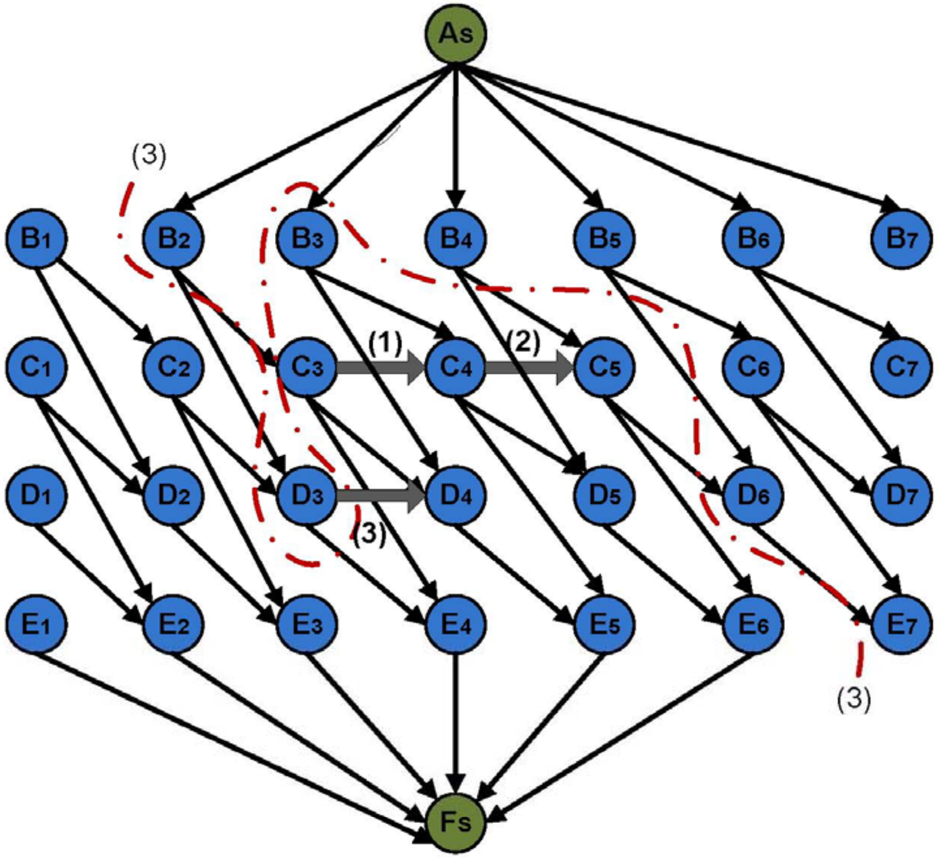
- **Algorithm 1:** Storage Management Policy – Iteratively Increasing the Min Cut



New **Min Cut**
Q_3=Q+Sc3 +Sc4

Storage Management Policy for General Networks

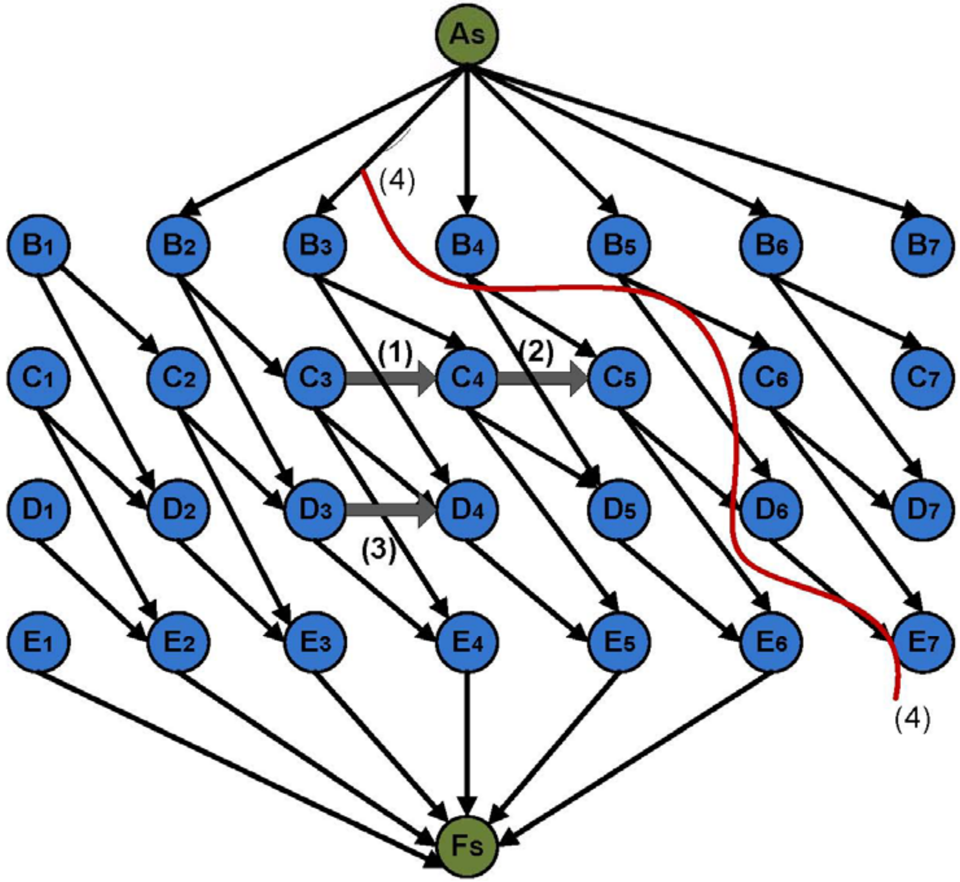
- **Algorithm 1:** Storage Management Policy – Iteratively Increasing the Min Cut



We add a storage link of S_{D3} units

Storage Management Policy for General Networks

- **Algorithm 1:** Storage Management Policy – Iteratively Increasing the Min Cut



Final Storage Enhanced Min Cut

$Q_F = Q + S_{C3} + S_{C4} + S_{D3}$

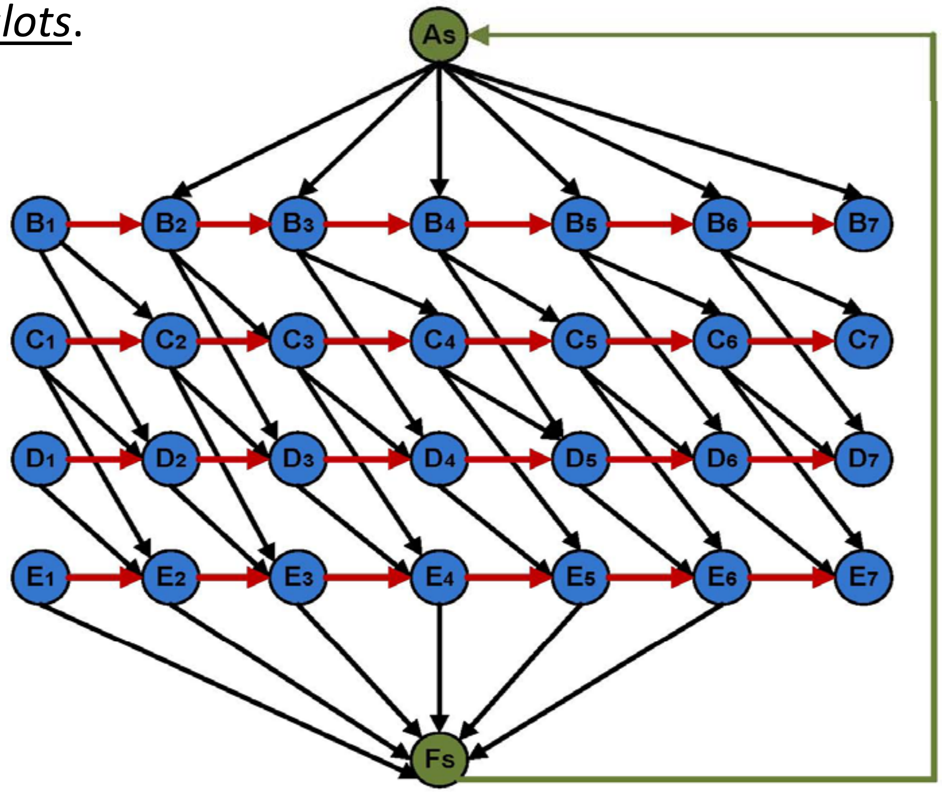
Joint Storage Management and Routing Policy

Joint Storage and Routing Policy

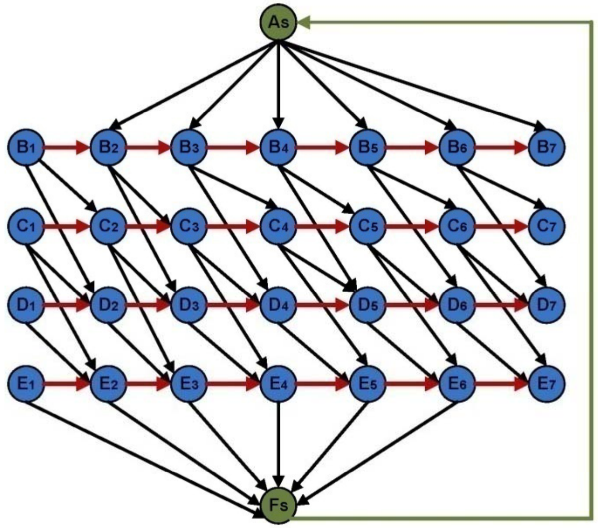
- **Joint Storage Managment and Routing Problem:**
  – Given a dynamic network G=(V,E) with a single source and a single destination, and with nodes that have time varying storage capacity, find how much data should be stored in each node and how much data should be routed over each link, in every time slot, in order to *maximize the amount of transferred data within a certain time period of T slots*.

Red arrows: Storage Links
Black arrows: Communication Links
Green arrow: Artificial feedback Link

- X_{ds} is the flow that reaches the destination node {s} during the T slots



Joint Storage and Routing Policy



- Joint Storage and Routing Problem
  - Max Flow Problem in the Time – Expanded Graph

$$\min\{-x_{ds}\}$$

$$s.t. \quad \sum_{j \in F_i} x_{ij} + y_{in} = \sum_{j \in B_i} x_{ji} + y_{mi}, i \in V_T$$

$$0 \le x_{ij} \le C_{ij}, i \in V_T, j \in F_i$$

$$0 \le y_{in} \le S_{in}, i \in V_T, n = i^{(t+1)}$$

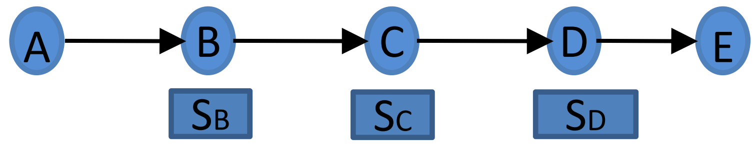
- { F_i } is the set of downstream communication neighbors of node {i}
- { B_i } is the set of upstream communication neighbors of node {i}
- { C_{ij} } is the capacity of the ***communication links***
- { S_{in} } is the capacity of the ***storage links***

- Solution method: **ε-relaxation** distributed primal – dual algorithm

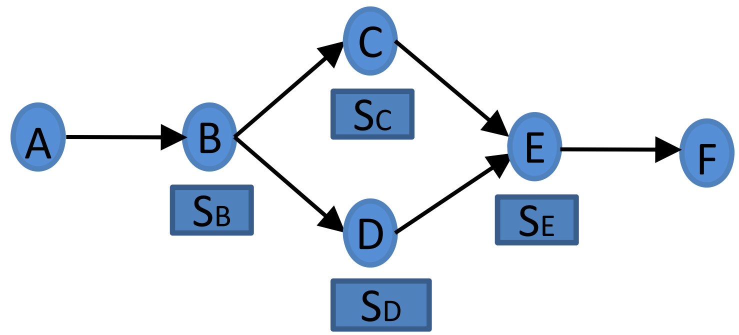
Numerical Results

Numerical Results

- Simulation Set up:
  - Linear network with 3 nodes



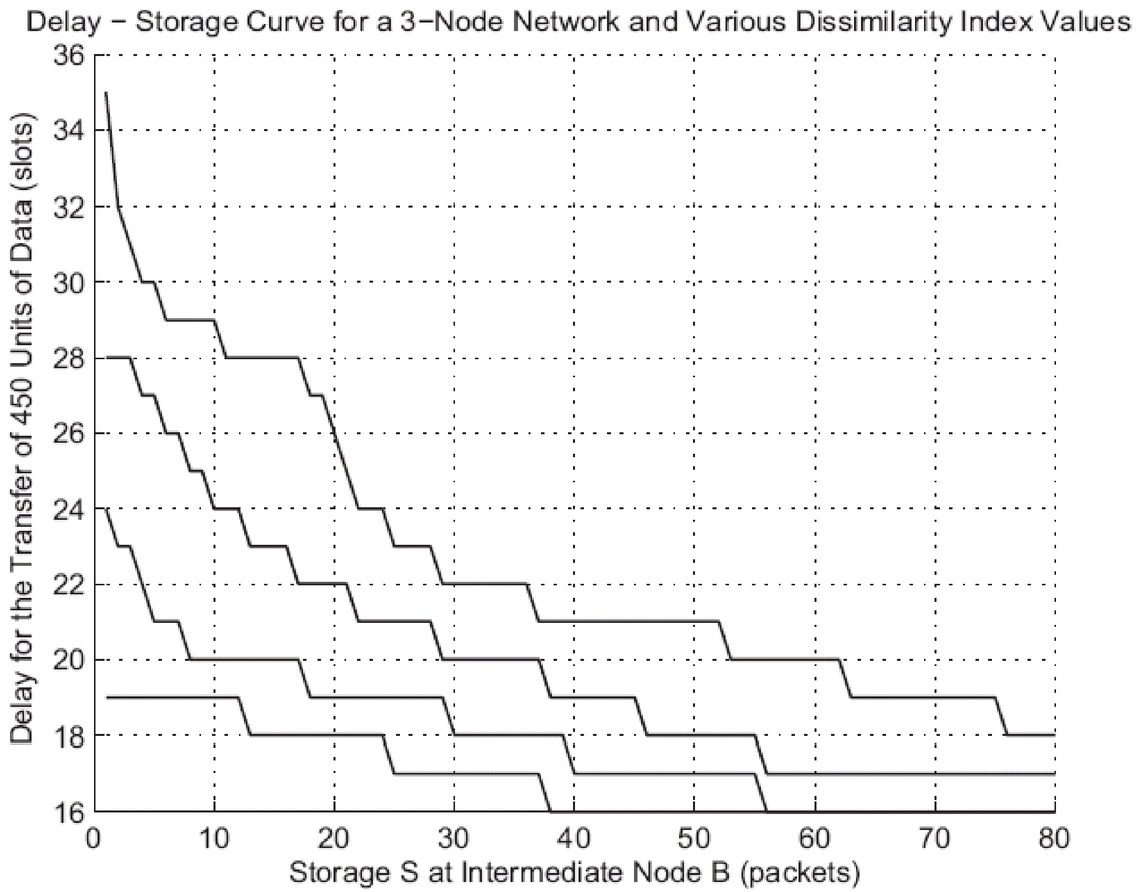
  - Linear network with 5 nodes
  - general graph



- Objective
  - Demonstrate the Impact of Storage on achieved Delay, for different values of L

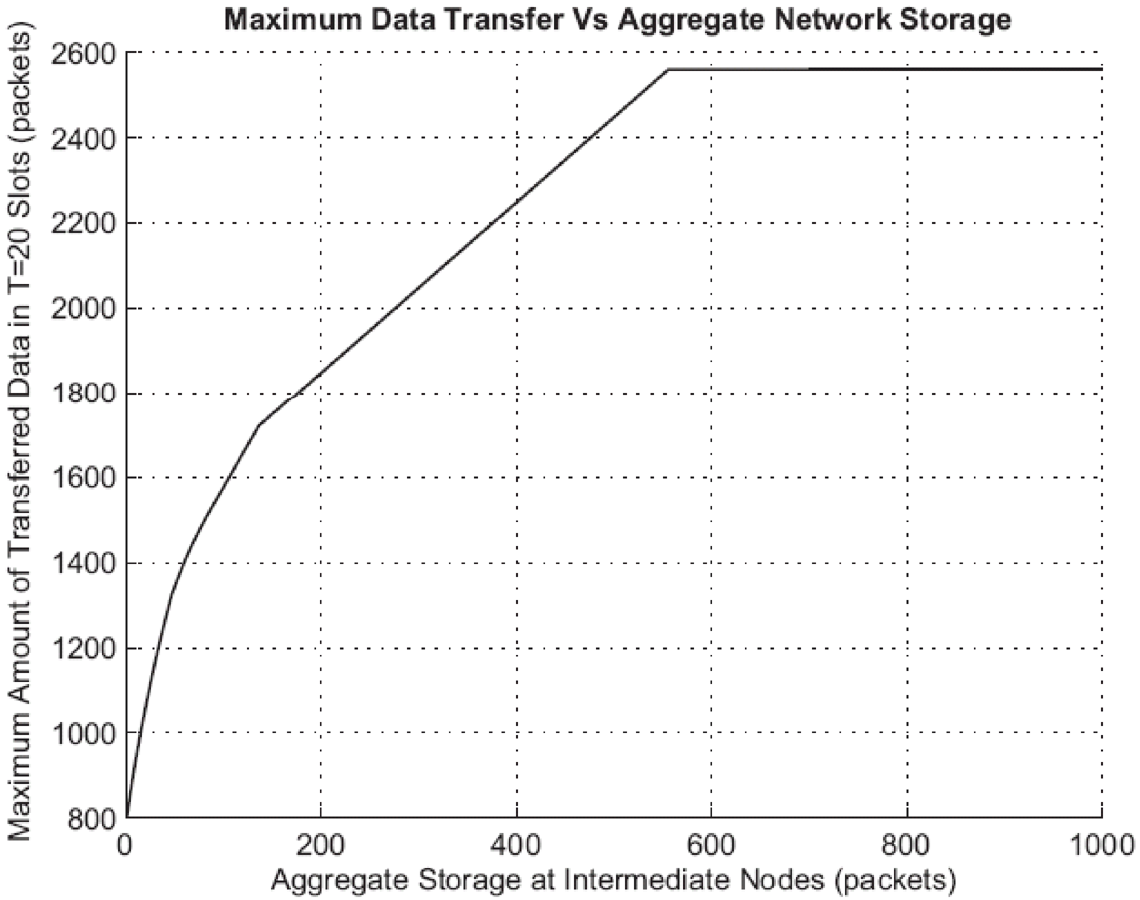
Numerical Results

- For a given amount of data, the delay improvement is analogous to the value of L



Numerical Results

- General Network: The benefit from storage has an upper limit. After a certain amount of storage we don't get any further improvement.



Conclusions

Conclusions

- Storage under certain conditions can improve network performance:
  - For fixed time interval: increase the amount of transferred data
  - For fixed amount of data: decrease the incurred transfer delay
- Storage management must be considered in conjunction with routing, i.e. JSR policies.
- What are the applications ?
  - Inter-data center communication,
  - Intra-data center networking.
  - Overlays performance enhancement.
  - Reduction of network infrastructure cost: storage instead of capacity.

Thank you!

George Iosifidis: www.GeorgeIosifidis.net

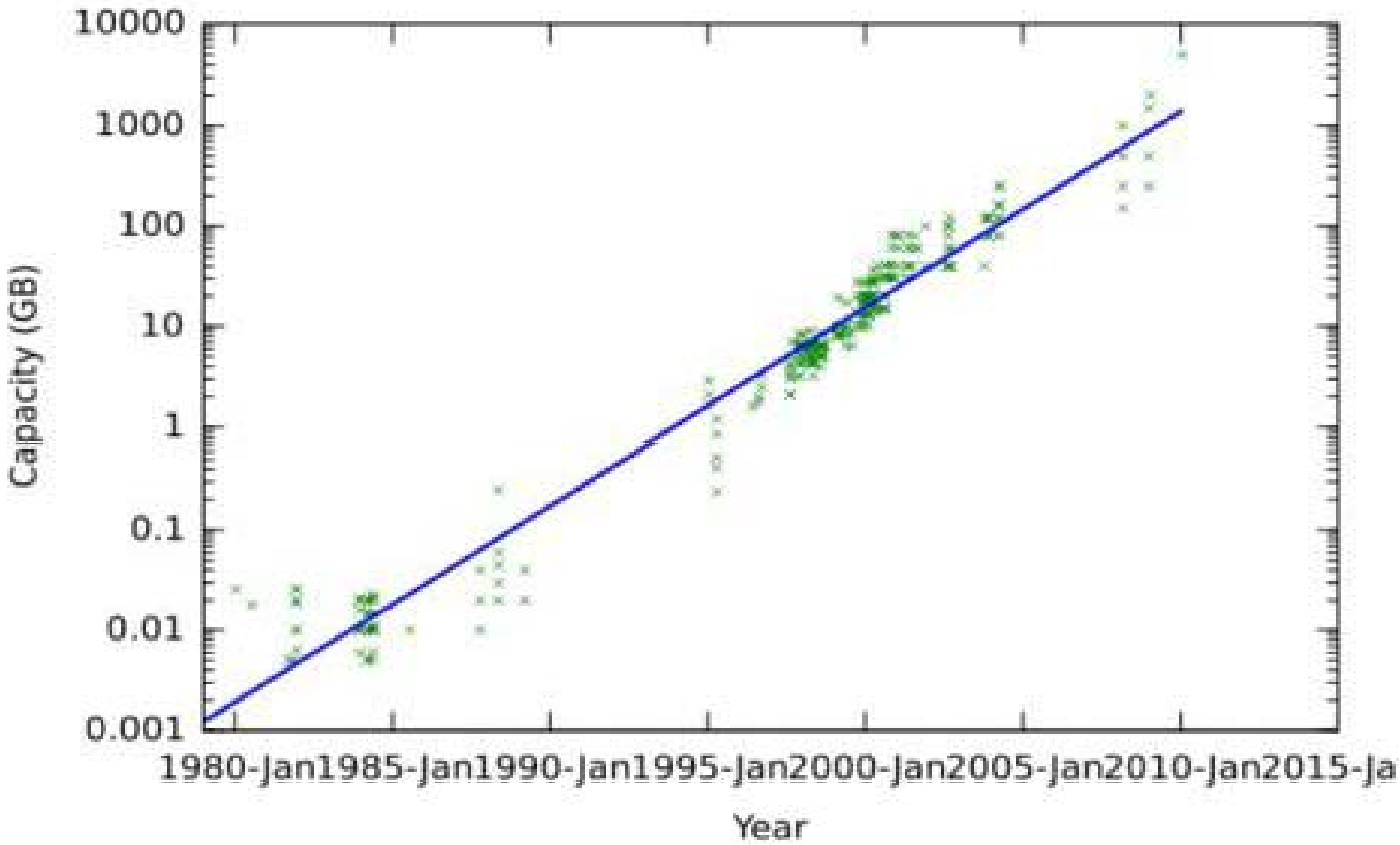
Iordanis Koutsopoulos: www.inf.uth.gr/~jordan

Georgios Smaragdakis: www.Smaragdakis.net

Back up Slides

Observation #1: Cost of Storage

The storage capacity increases with an exponential rate.



Impact of Storage Capacity in Linear Networks (3)

- **Q1:** How many slots do we need to transfer D units of data?

$$Z : \sum_{n=1}^{Z} C_{AC}^{S}(n) T_0 \geq D$$

- **Q2:** Which is the amount of data that is transferred in T time slots?

$$D = \sum_{n=1}^{T-2} C_{AC}^{S}(n) T_0$$

→ Where the end-to-end capacity in each slot n, is:

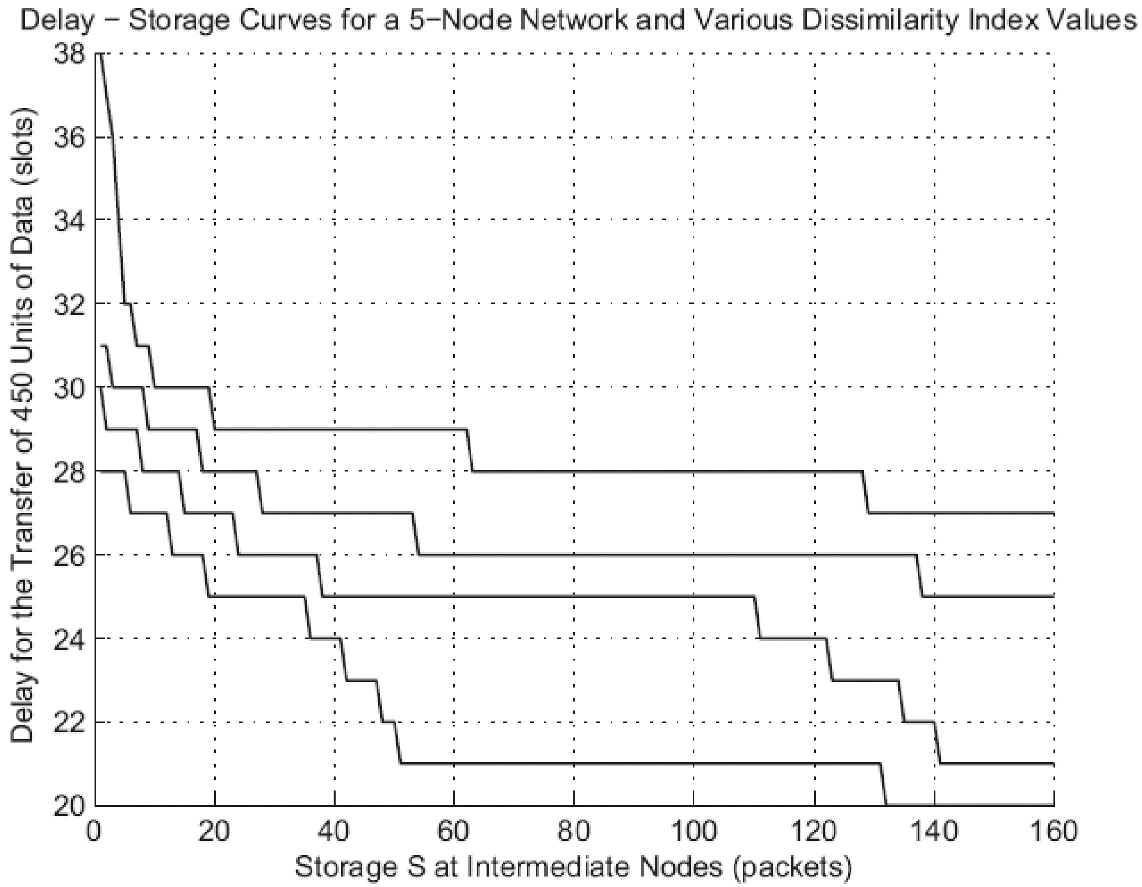
~~$C_{AC}(n) = \min\{C_{AB}(n), C_{BC}(n+1)\}$~~

$$C_{AC}^{S}(n) = \min\{C_{BC}(n+1), X_B(n+1)\}$$

The available data at the last node before the destination

Numerical Results (3)

- 5 nodes: The same conclusion. The improvement is analogous to the value of L.



Joint Storage and Routing Policy (3)

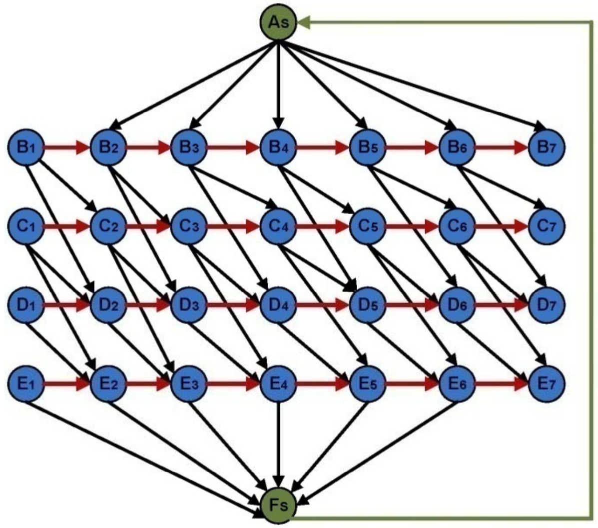
- Joint Storage and Routing Problem
  - Max Flow Problem in the Time – Expanded Graph

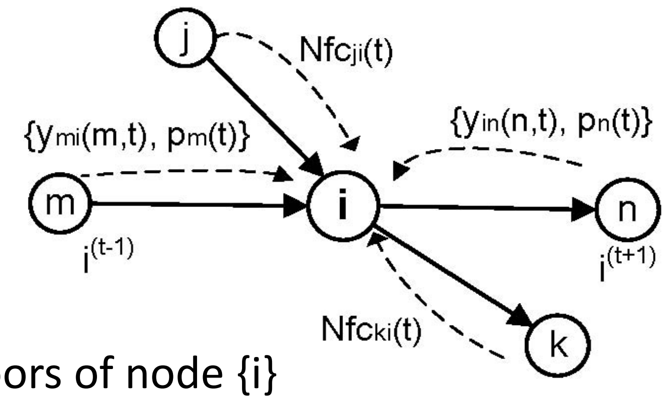
$$\min\{-x_{ds}\}$$

$$s.t. \quad \sum_{j\in F_i} x_{ij} + y_{in} = \sum_{j\in B_i} x_{ji} + y_{mi}, i \in V_T$$

$$0 \le x_{ij} \le C_{ij}, i \in V_T, j \in F_i$$

$$0 \le y_{in} \le S_{in}, i \in V_T, n = i^{(t+1)}$$





- { F_i } is the set of downstream communication neighbors of node {i}
- { B_i } is the set of upstream communication neighbors of node {i}
- { C_{ij} } is the capacity of the *communication links*
- { S_{in} } is the capacity of the *storage links*

Joint Storage and Routing Policy (6)

- Solution method: **ε-relaxation** primal – dual method
  - Primal Variables: routing and storage decisions (/link, /node)
  - Dual Variables: prices (/node)

  - 1. Nodes Decisions: data to route/ data to admit/ data to store.
  - 2. Nodes Communication: msg exchange with these vars.
  - 3. Nodes Coordination: comparison and update of the vars.