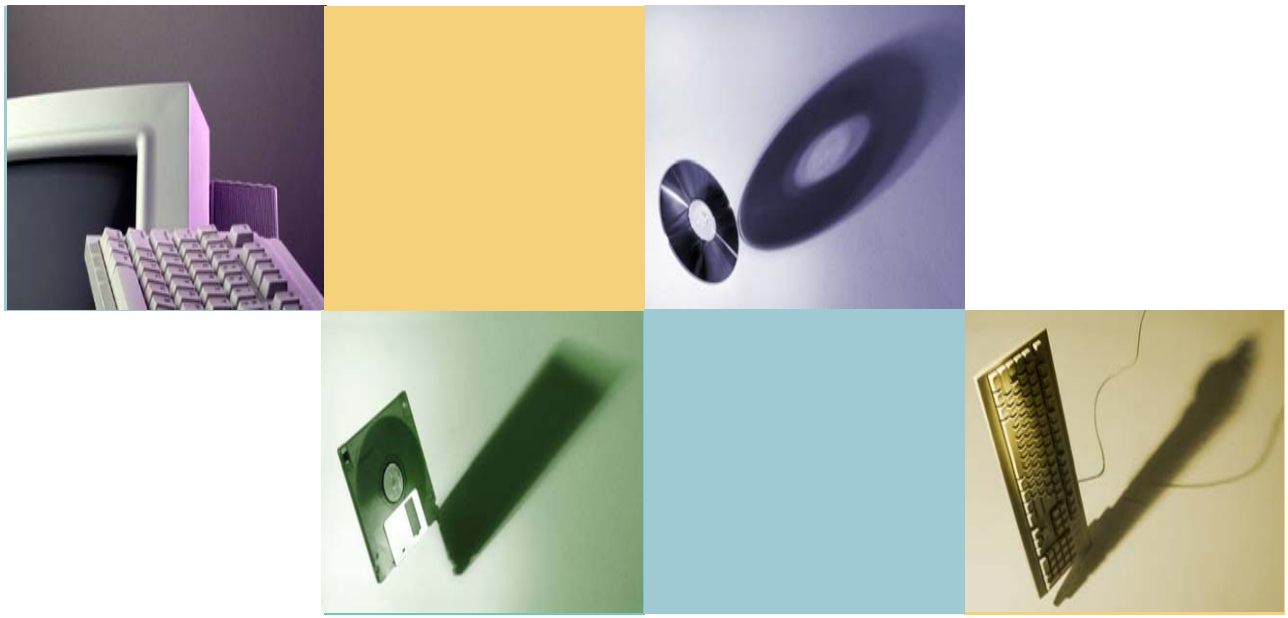
Closed P2P System for PVR-Based File Sharing



From IEEE Transactions on Consumer Electronics, **AUGUST 2005**

Presented by Ching-Lan Wang

2005/11/11

Outline

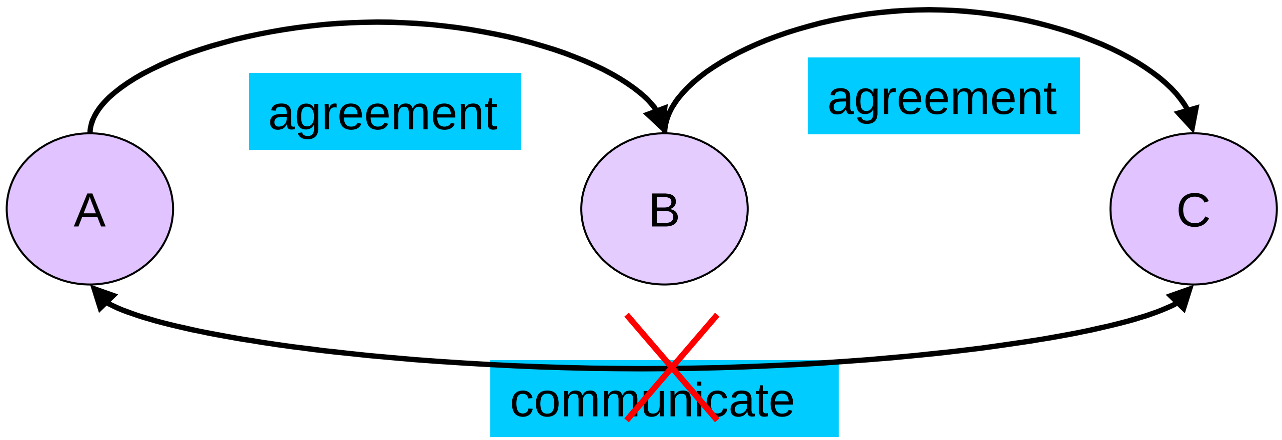
- Introduction
  - PVR (personal video recorder)
  - Closed P2P system
- Storage Extension Problem
- Data Placement Algorithm
  - Owner-based
  - Group-based
- Performance
- Conclusion

Introduction

- PVR (personal video recorder)
  - PVR (personal video recorder) is a device that stores media data on persistent storage like hard disks, whereas a VCR does it on magnetic tapes.
  - PVR can record multiple media streams, or record a stream and play back another stream concurrently.
- PVR-based file sharing
  - Closed P2P system is suitable for PVR-based file sharing.

Closed P2P System

- A closed P2P system relies on an agreement between nodes for sharing files, and only the nodes that agree to share files can access the shared files.
- For example:

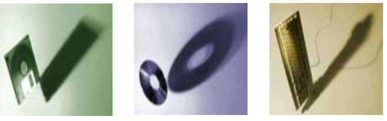


Closed P2P System for PVR-based File Sharing

- PVR is a private device.

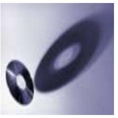
- To prevent anonymous users from accessing a node may raise a bottleneck problem, the agreement of closed P2P system is the permission to access the data of a node.

- Storages can be shared between nodes that established an agreement, and a user can use the storage of the *neighbor nodes* (nodes that have the agreement with the node).



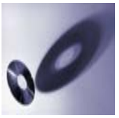
Storage extension problem

- A lack of data placement control may result in redundant file placement and reduced effective total storage, and such a space problem should not be ignored when the sizes of shared objects are large.
- The data placement scheme for such P2P system should minimize redundancy in storing shared files.

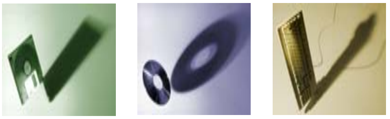




Closed P2P Model

- ***Node***
  - It means a peer in a P2P system.
  - Each node has some amount of storage, and donates a part of it to share data with other nodes.
  - It has a unique id, n_i, and N is a set of nodes.
- ***Link***
  - If two nodes have an agreement, there is a link between them. Storage between agreed nodes can be shared by both nodes. L is a set of links.


- ***Virtual storage group (VSG)***:
  - A node can use its own storage and the storage of all the linked nodes.
  - Virtual storage group *VSGi* of a node n_i is a group of nodes that share storage with the node n_i.



Virtual storage group

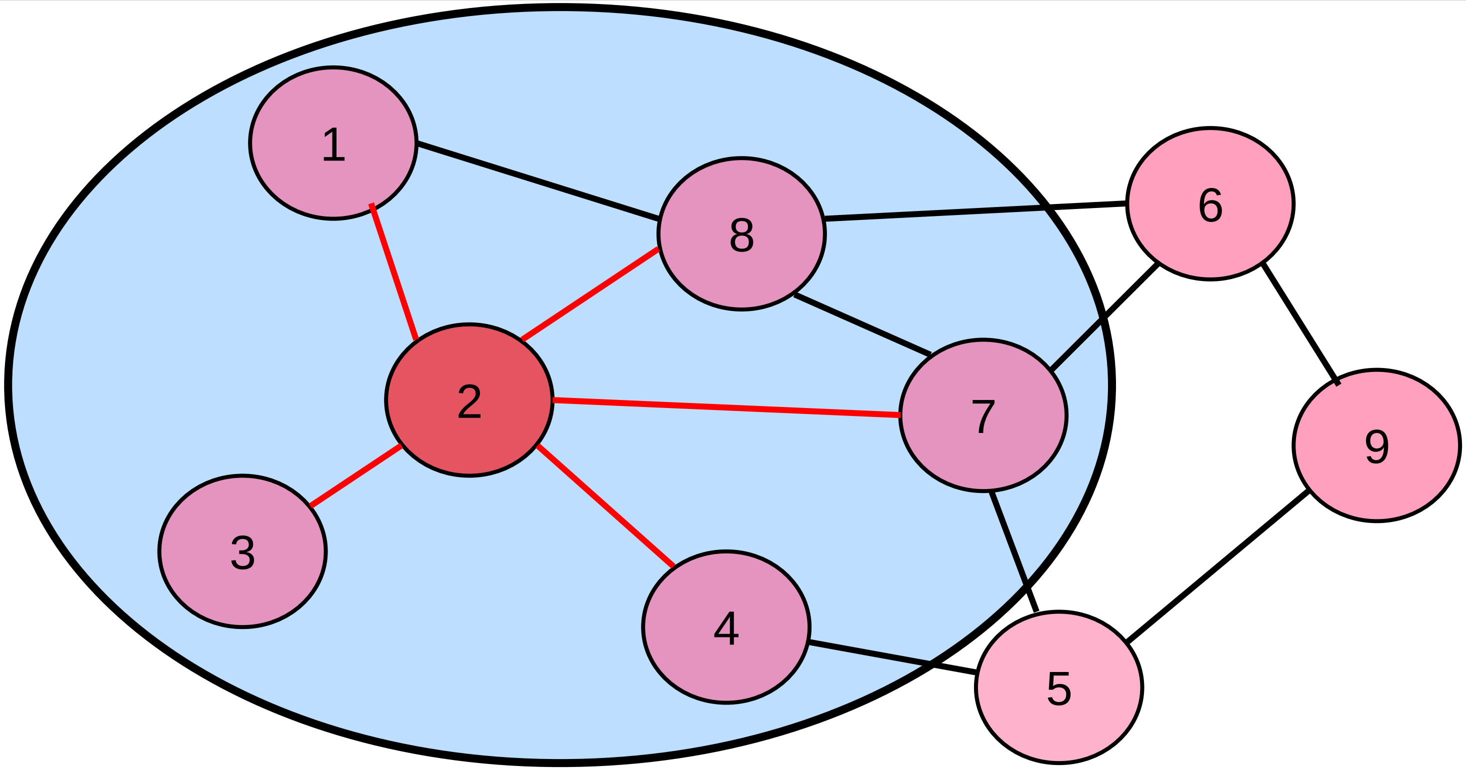
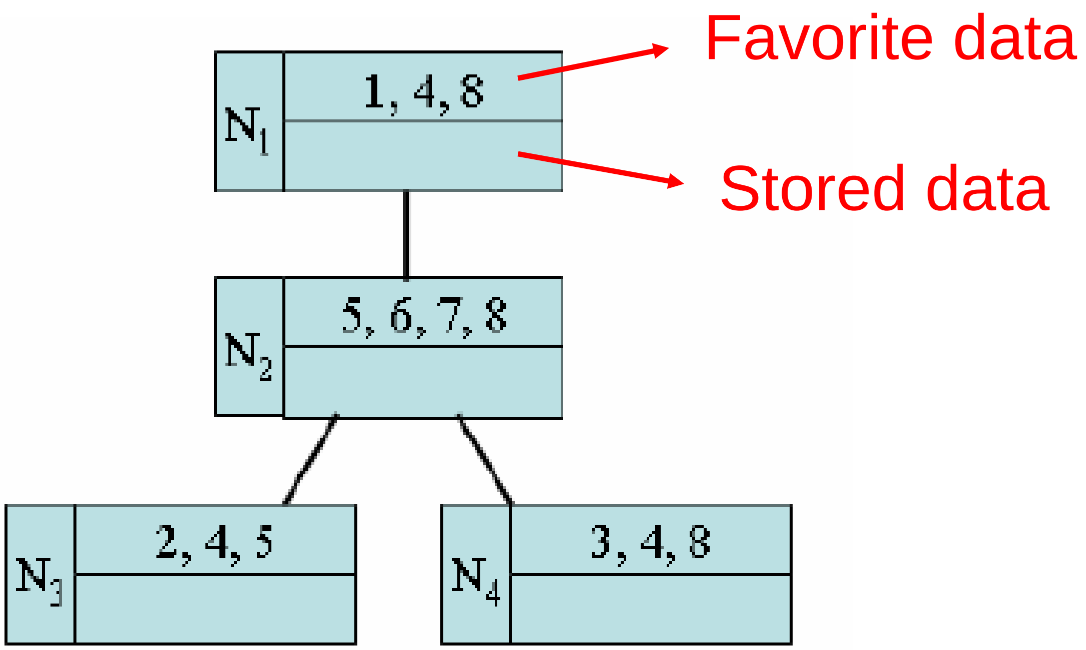


Fig. 1. Virtual storage group

Example: $\color{red}{VSG_2}$ is $\{n_1, n_2, n_3, n_4, n_7, n_8\}$.

Data placement

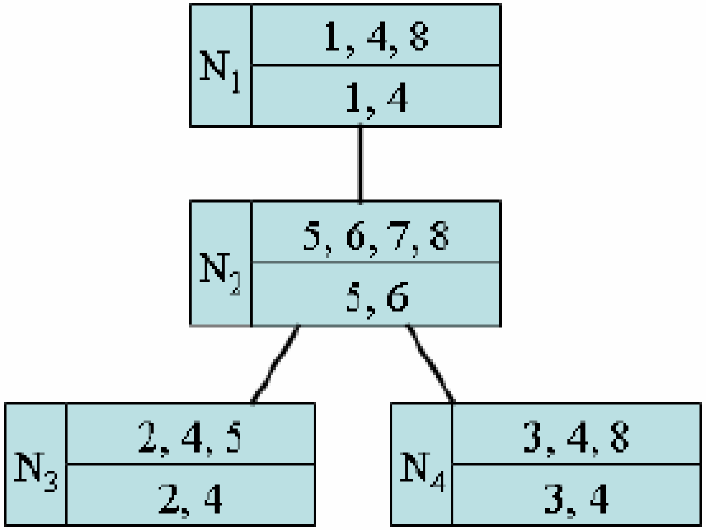
- This is a data placement scheme.
  - The sizes of all data are same.
  - The physical storage sizes of all nodes are twice the size of data.



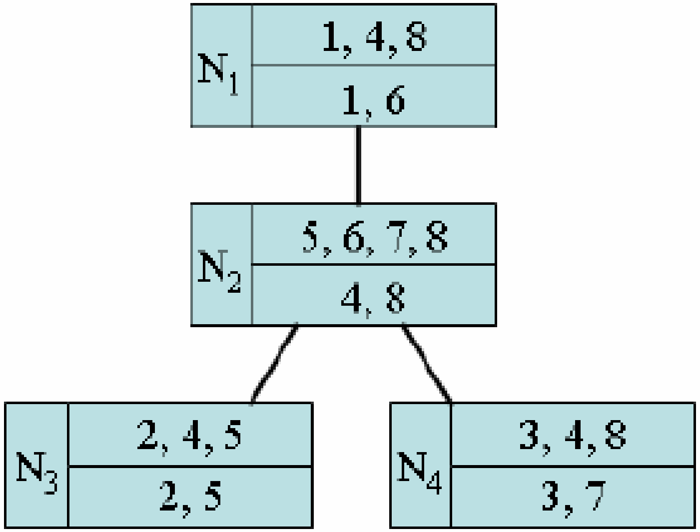
(a) An Example of data placement

Different data placement schemes

Data placement schemes affect the total virtual storage size and the available storage of each node.



(b) Data placement 1



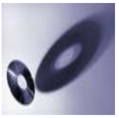
(c) Data placement 2

Definition

- ***Data (D)***:
  - A set of data.

- ***Physical storage size (P)***:
  - The size of data that can be stored in a node n_i.

- ***Favorite data (FD)***:
  - We define *FD* as a set of pairs (n_i, F_i), where F_i is a set of data that are of interest to the node n_i.


- ***Stored data (SD)***:
  - *SD* is a set of pairs (n_i, S_i) where each S_i represent a set of data that stored in the storage of node n_i.
  - Note that S_i needs not be a subset of F_i.

- ***Stored data of virtual storage group (SVSG$_i$)***:
  - A set of data stored in VSG_i.

- ***Virtual storage size (VSS)*:**
  – Only the space of the VSG storage that contains favorite data of a node is meaningful to the node.

- ***Total virtual storage size (TVSS)*:**
  – The total sum of VSS of all the nodes in a system.

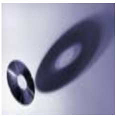
Data Placement Algorithms

- Data placement is important in a closed P2P system because smart algorithms will enlarge the total storage of the system.

- Owner-based Algorithm
- Group-based Algorithm
  - Group-based random (GR)
  - Most-popular (MP)
  - Most-popular with neighbor consideration (MPNC)
  - Maximum popular value first (MPVF)

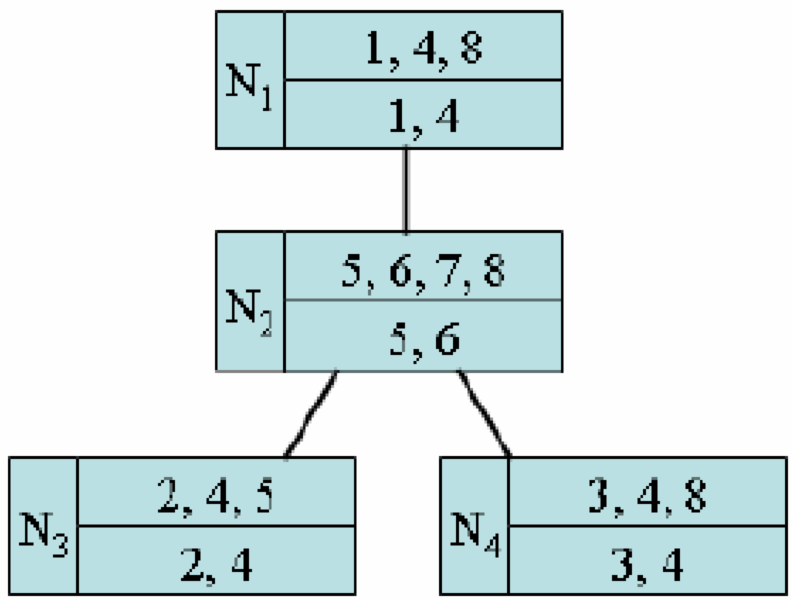
Assumptions

- Same storage size:
  - All nodes have the same storage size.
- Same data size:
  - All data sizes are equal.
  - Each node can store the same number of data.
- Sufficient network bandwidth:
  - There is no network bottleneck and a node can always perform services to other linked nodes.
- Off-line algorithm:
  - we only consider the initial data placement assuming that there is no change in node configurations or data preference.

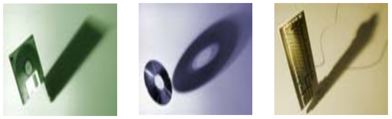




Owner-based Algorithm

- When the storage of a node is not sufficient to store all the favorite data of the node, data are randomly selected.
- Because of the greediness, this one will fail to maximize the total virtual storage size.

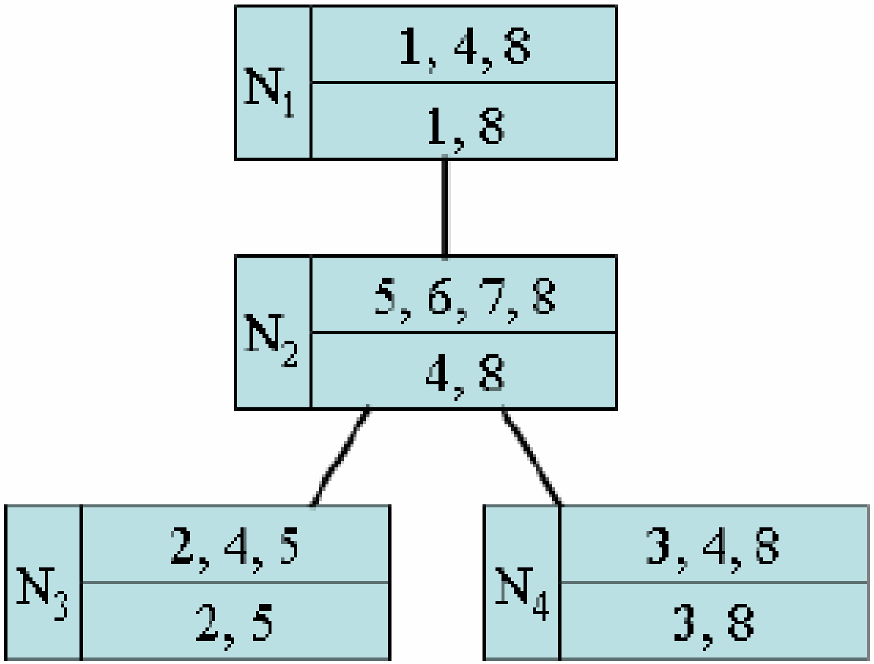


(b) Data placement 1



Group-based random (GR)

- A node randomly selects data among the favorite data of the nodes in its VSG.

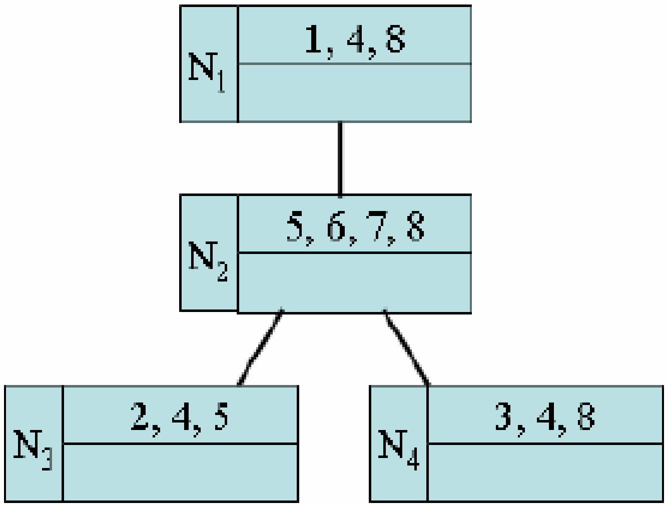


(d) Data placement 3

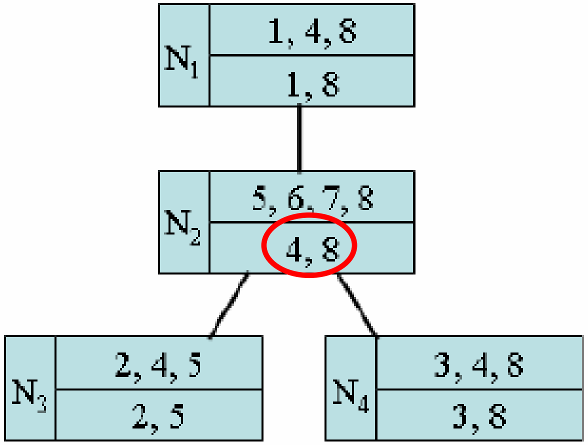
Most-popular (MP)

- This algorithm selects data among the favorite data of all the nodes in its VSG according to the popularity of the data.
- *Popular value* (V_{ij}) is the number of nodes that would be able to access data d_j if the data is stored in node n_i.
- Data to be stored in node n_i are selected according to these popular values to maximize the happiness of the nodes that are interested in the data.

Most-popular (MP) Example



(a) An Example of data placement



(d) Data placement 3

All popular values of nodes N_2 in figure(a)

Data number	1	2	3	4	5	6	7	8
Popuplar value(V_{2j})	1	1	1	3	2	1	1	3
Benefit node	n_1	n_3	n_4	n_1,n_3,n_4	n_2,n_3	n_2	n_2	n_1,n_2,n_4

Most-popular with neighbor consideration (MPNC)

- MPNC prevents such unnecessary duplication by not allowing linked nodes to have the same data.
- If two linked nodes have the same favorite data, the data are stored only in one of them.

MPNC Example

All popular values of nodes N_2 in figure(a)

Data number	1	2	3	4	5	6	7	8
Popuplar value(V_{2j})	1	1	1	3	2	1	1	3

All popular values of nodes N_1 in figure(a)

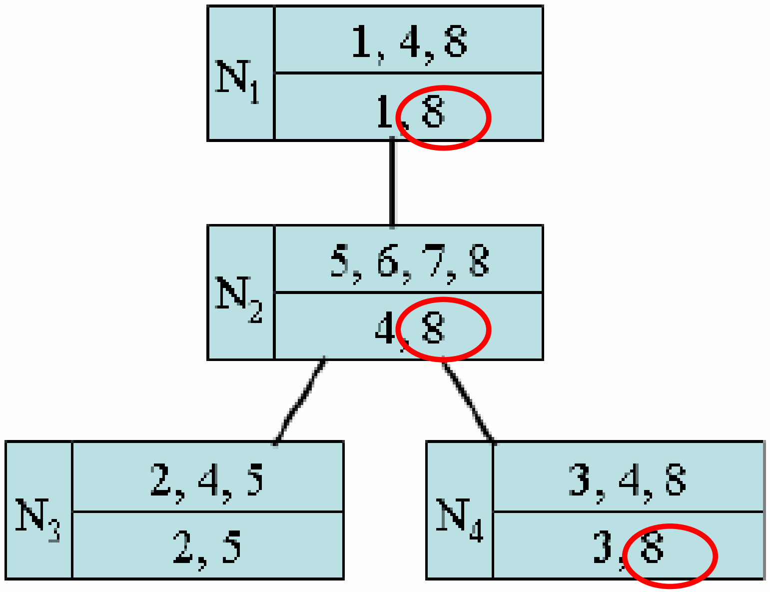
Data number	1	2	3	4	5	6	7	8
Popuplar value(V_{1j})	1	0	0	1	1	1	1	2

All popular values of nodes N_4 in figure(a)

Data number	1	2	3	4	5	6	7	8
Popuplar value(V_{4j})	0	0	1	1	1	1	1	2

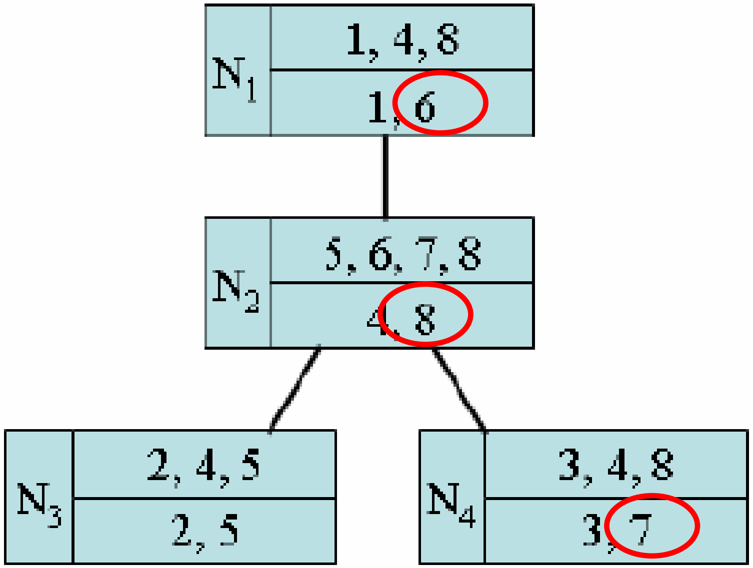
MPNC Example

(MP)



(d) Data placement 3

(MPNC)



(c) Data placement 2

Maximum popular value first (MPVF)

- The MPNC algorithm considers neighboring nodes, its optimization is still local as it does not consider data placed in nodes of more than one-hops away.
- For MPVF, we need a centralized server, and we assume that a centralized server collects all the needed information from every node, and decides which data should be placed in which node.

Global popular value table

- The centralized server prepares a global popular value table that contains the popular values of all nodes and data, and then makes a decision on data placement according to this table.

node / data	P_{1j}	P_{2j}	P_{3j}	P_{4j}
P_{i1}	1	1	0	0
P_{i2}	0	1	1	0
P_{i3}	0	1	0	1
P_{i4}	1	3	1	1
P_{i5}	1	2	2	1
P_{i6}	1	1	1	1
P_{i7}	1	1	1	1
P_{i8}	2	3	1	2

MPVF Example

data \ node	P_{1j}	P_{2j}	P_{3j}	P_{4j}
P_{i1}	1	1	0	0
P_{i2}	0	1	1	0
P_{i3}	0	1	0	1
P_{i4}	1	3	1	1
P_{i5}	1	2	2	1
P_{i6}	1	1	1	1
P_{i7}	1	1	1	1
P_{i8}	2	3	1	2

P_{24} →

1	1	0	0
0	1	1	0
0	1	0	1
0	S	0	0
1	2	2	1
1	1	1	1
1	1	1	1
2	3	1	2

P_{28} →

1	1	0	0
0	1	1	0
0	1	0	1
0	S	0	0
1	2	2	1
1	1	1	1
1	1	1	1
0	S	0	0

P_{35} →

1	1	0	0
0	1	1	0
0	1	0	1
0	S	0	0
0	0	S	0
1	1	1	1
1	1	1	1
0	S	0	0

P_{11} →

S	0	0	0
0	1	1	0
0	1	0	1
0	S	0	0
0	0	S	0
1	1	1	1
1	1	1	1
0	S	0	0

P_{43} →

S	0	0	0
0	1	1	0
0	0	0	S
0	S	0	0
0	0	S	0
1	1	1	1
1	1	1	1
0	S	0	0

P_{16} →

S	0	0	0
0	1	1	0
0	0	0	S
0	S	0	0
0	0	S	0
S	0	0	0
1	1	1	1
0	S	0	0

P_{32} →

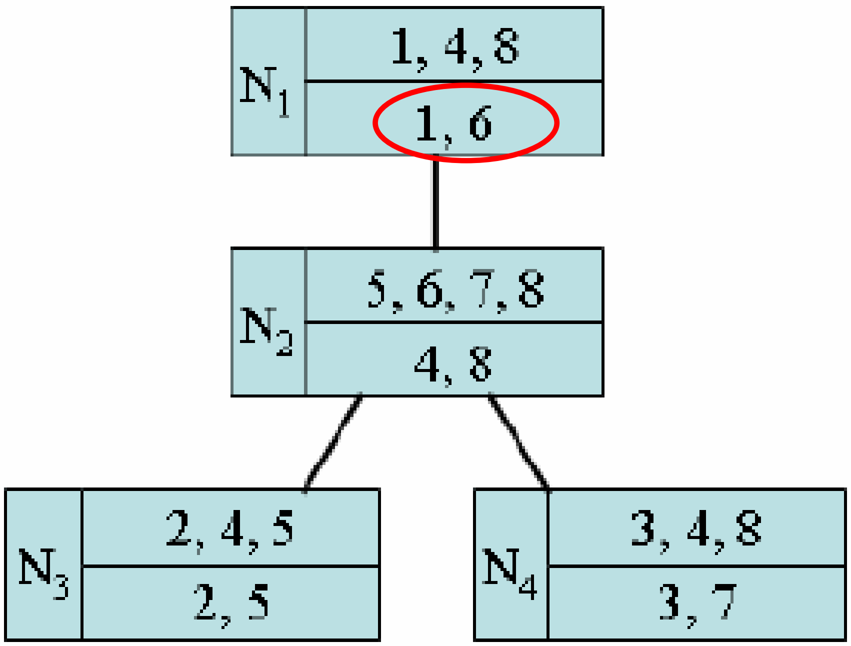
S	0	0	0
0	0	S	0
0	0	0	S
0	S	0	0
0	0	S	0
S	0	0	0
1	1	1	1
0	S	0	0

P_{47} →

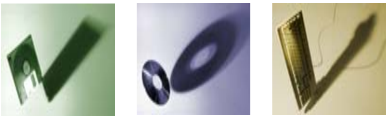
S	0	0	0
0	0	S	0
0	0	0	S
0	S	0	0
0	0	S	0
S	0	0	0
0	0	0	S
0	S	0	0

MPVF Example

data \ node	P_{1j}	P_{2j}	P_{3j}	P_{4j}
P_{i1}	S	0	0	0
P_{i2}	0	0	S	0
P_{i3}	0	0	0	S
P_{i4}	0	S	0	0
P_{i5}	0	0	S	0
P_{i6}	S	0	0	0
P_{i7}	0	0	0	S
P_{i8}	0	S	0	0

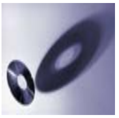


(c) Data placement 2

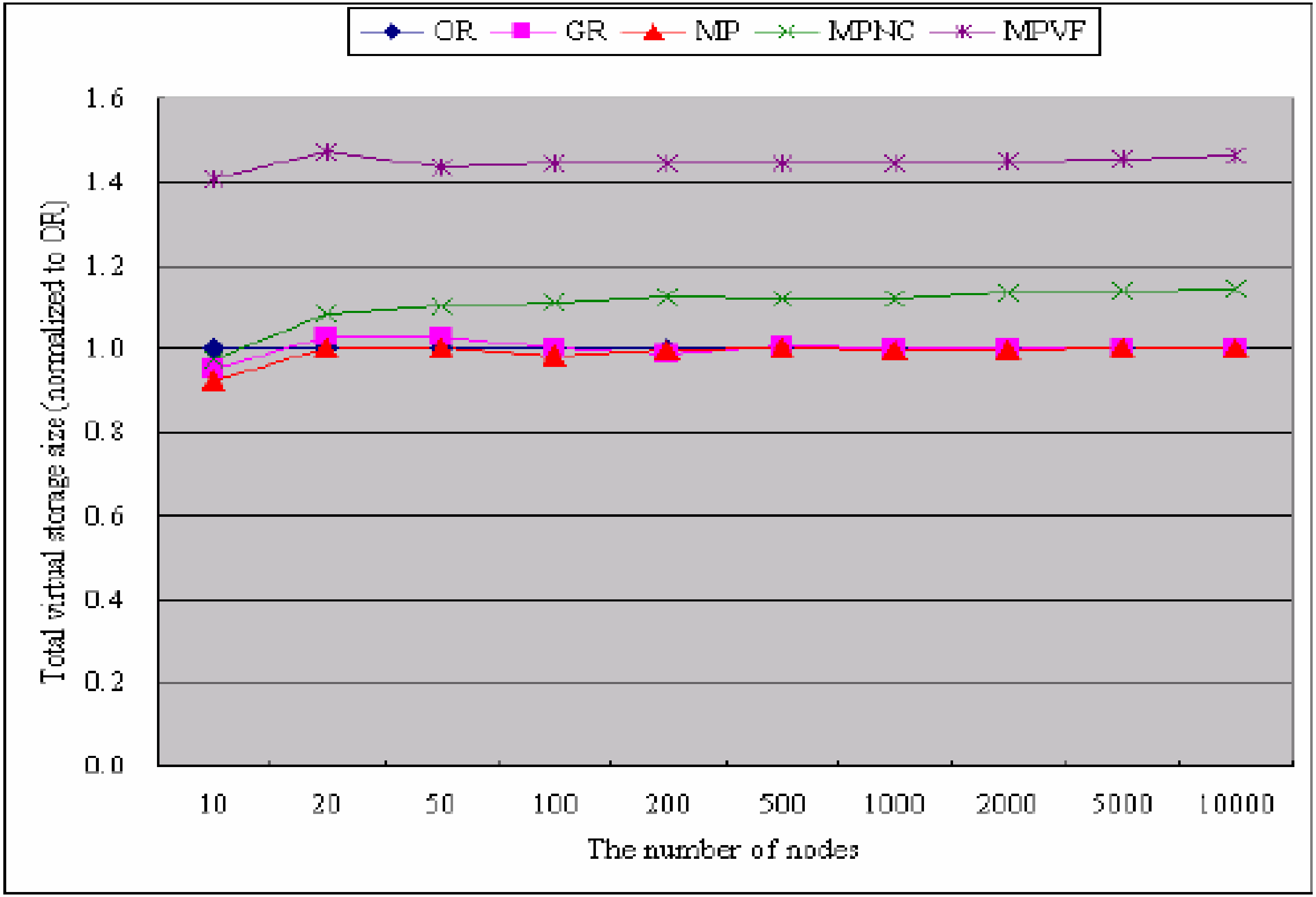


Performance

- *n* : the number of nodes.
- *l* : the average number of links for each node in the whole P2P system.
- *d* : the total number of data.
- *f* : the number of favorite data for each node.
- P : the physical storage size of a node in terms of the number of data that can be stored.

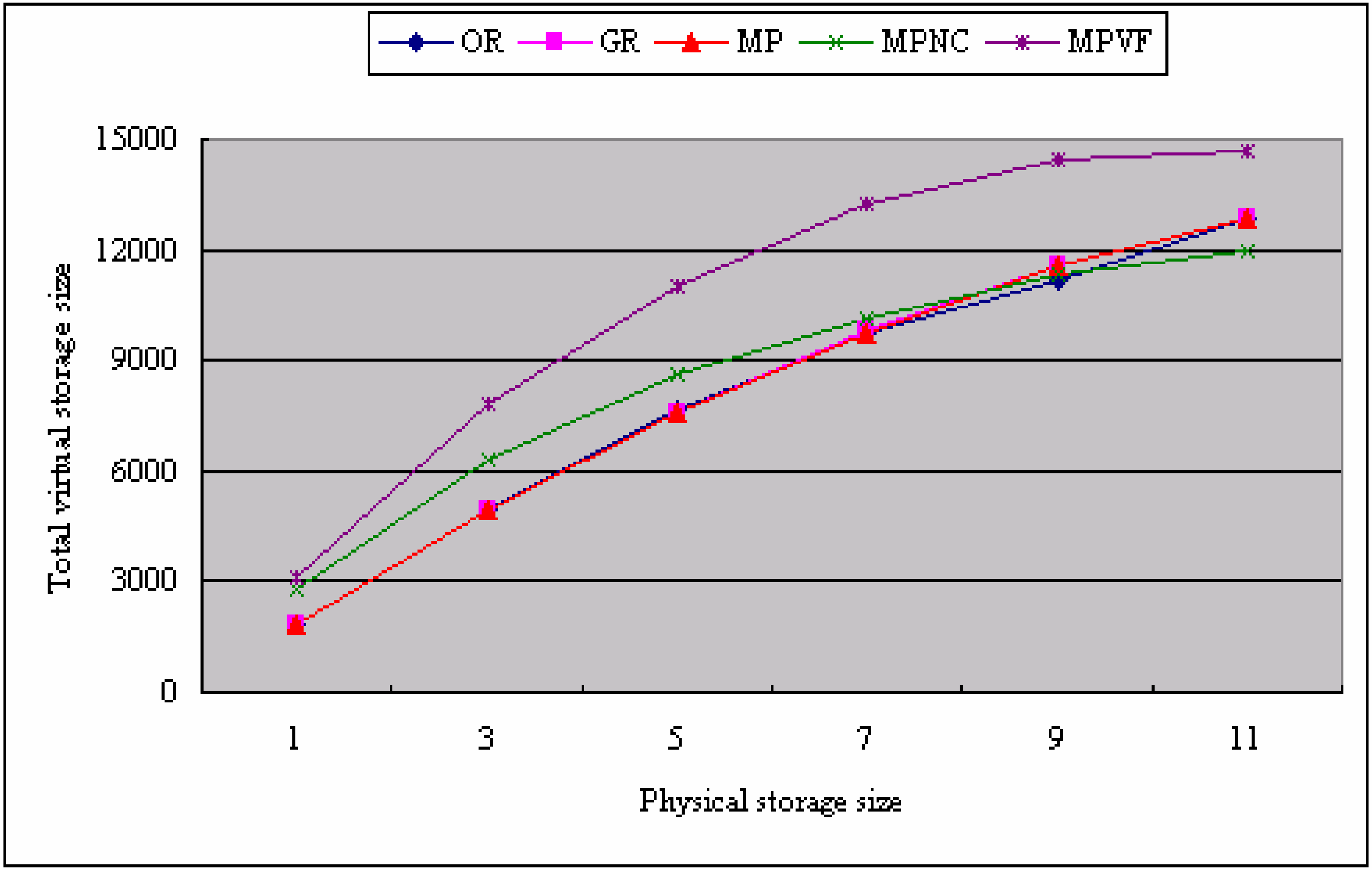




The normalized TVSS varying the number of nodes



l **= 3,** ***d*** **= 50,** ***f*** **= 15, P = 5.**

The effect of varying the physical storage size



l = 3, d = 50, f = 15.

Conclusion

- The smart data replacement would increase the virtual storage size of nodes.
- This paper suggested and evaluated five algorithms for this problem.
- The MPVF algorithm increases the total virtual storage size most because it considers interests of all the nodes in the P2P system.