

國立臺灣大學電機資訊學院電機工程學系



碩士論文

Department of Electrical Engineering  
College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

使用考量時間之相關性內容於鏈結共享之發佈/訂閱機制  
Publish/Subscribe Mechanism Using Time-Concerned Content  
Correlation for Link Sharing

王楚渝

Chu-Yu Wang

指導教授：郭斯彥 博士

Advisor: Sy-Yen Kuo, Ph.D.

中華民國 103 年 7 月

July, 2014

## 誌謝



首先特別感謝我的指導教授 郭斯彥教授，讓我在這個實驗室環境裡學習，除了有豐富的學習資源之外，也吸收了很多實務操作的經驗。也感謝紀廷運學長在帶領實驗室團隊時對我們的照顧，以及論文方向上的指導。感謝實驗室的同學和學長姊與學弟們：繹銘同學、晉廷學長、育瑄學姊、宗永學長、怡廷學長、胤澤學弟、理元學弟、詠文學弟、定遠學弟、紹航學弟等互助與合作完成計畫，達成目標，共度實驗室們大家相聚的時光。最後，感謝我的家人支持著我，讓我無所顧慮的在研究所課業上努力，由衷感謝他們無悔的付出，讓我有個溫暖的家。

## 中文摘要



近年來物聯網發展十分興盛, 在分散式系統中經常使用發佈/訂閱機制來散播資訊。不同的節點若對同一主題訂閱, 則能彼此交換資訊, 形成一層主題覆蓋層。當兩節點有訂閱許多相同的主題時, 其內容相關性就會增加, 不同的主題覆蓋層在傳遞資訊時就能在這兩個節點之間共享鏈結, 減少整體的鏈結數以降低網路傳輸之成本。然而每個節點對於一主題的訂閱在某些情況下是有時效性的, 甚至有特定的訂閱週期與排程。在尋找相鄰節點建立鏈結時, 過去單純的考量耗電量其實並不足以滿足需求, 因此我們加上考量內容的時間因素形成不同的網路拓樸, 提出了一套全新的資料訂閱與蒐集機制, 並進行了相關模擬, 結果能有效降低因為訂閱資訊的過時而必須重新建立網路拓樸的負擔。

關鍵字—發佈/訂閱, 物聯網, 鏈結共享

# ABSTRACT



In recent years, the development of Internet of Things grows rapidly, and we often use publish/subscribe mechanism to disseminate information in distributed system. Different nodes can exchange data when they subscribe to a same topic, and then form a topic overlay. When two nodes have many common subscribed topics, their content-correlation will increase, and different topic overlays may share links between both nodes to transmit data, which leads to decrease the amount of total links in order to make lower cost on network transmission. However, every node subscribes to a topic may be time-dependent in some scenarios, especially, has specific subscription period or schedule. When looking for neighbor nodes to build links, simply taking power-consuming into concern is no longer fit the requirement. Therefore we take time into concern to build a different network topology, and propose a new data publish/subscribe mechanism with some simulation. The result shows that it can lower the effort when the need to build new network graph due to the subscription time is expired.

**Keywords**—Publish and subscribe, Internet of Things, link sharing

# CONTENTS



口試委員會審定書 .....	#
誌謝 .....	i
中文摘要 .....	ii
ABSTRACT .....	iii
CONTENTS .....	iv
LIST OF FIGURES .....	vi
LIST OF TABLES .....	vii
<b>Chapter 1 Introduction .....</b>	<b>1</b>
1.1 Context .....	1
1.2 Contributions .....	1
1.3 Roadmap .....	1
<b>Chapter 2 Related Work .....</b>	<b>3</b>
2.1 Publish/Subscribe .....	3
2.2 Interest Correlation for Link Sharing .....	3
2.3 Content Life Cycle .....	4
<b>Chapter 3 Proposed System .....</b>	<b>6</b>
3.1 System Model and Assumptions .....	6
3.2 Design Rationale .....	7
3.3 Link Management .....	9
3.4 Time-concerned content .....	10
3.5 Algorithm .....	10
<b>Chapter 4 Simulation .....</b>	<b>12</b>

**Chapter 5 Conclusion.....18**

References .....19



# LIST OF FIGURES



Fig. 1 Topic-based publish/subscribe .....	2
Fig. 2 sample of link sharing (before optimization) .....	4
Fig. 3 sample of link sharing (after optimization) .....	4
Fig. 4 content life cycle .....	5
Fig. 5 Power law .....	6
Fig. 6 clustering coefficient .....	8
Fig. 7 random join-and-leave time slot.....	10
Fig. 8 subscription approach.....	12
Fig. 9 Power law simulation (# of subscriptions for topics).....	13
Fig. 10 Power law simulation (# of nodes with that # of subscriptions).....	13
Fig. 11 View Size evaluation for 1000 nodes and 100 topics.....	15
Fig. 12 View Size evaluation.....	16
Fig. 13 time-concerned comparison .....	17

# LIST OF TABLES



Table. 1 View Size in several conditions ..... 14



# Chapter 1 Introduction



## 1.1 Context

The Internet grows rapidly as the users and data increase in the network. We usually retrieve variety of data from kinds of websites or network service. Because of the demand of data from users to content providers, we need efficient dissemination systems to deliver data widely. Such as social networks or online markets need this kind of mechanism to provide their service to their users and customers.

The publish/subscribe system is a normally used model to disseminate events. The feature of decoupling the communication between publishers and subscribers in space, time, and synchronization make it be easily used in variety of application domain [1].

The topic-based publish/subscribe system categorizes items by topics, shown as Fig. 1. Topic is just like a channel that publishes events through it, and the subscribers receive this information by subscribing this topic first. In real world, applications like social networks use it to accomplish the event dissemination.

## 1.2 Contributions

This paper presents a proposed system call tStaN, a novel approach to topic-based publish-subscribe that aligns multiple independent overlays to promote link sharing among them, with respect to the time slot when the topic is not expired. This work shows that even in the frequently join and leave of nodes in random manner, this system can still achieve link sharing and reduce the physical link cost.

## 1.3 Roadmap

The remainder of this paper is organized as follows: Related work is discussed in Section 2, and we summarize these main idea and propose our system in Section 3. In

Section 4 we validate it by simulation. Finally, the conclusion of this paper is in Section

5.

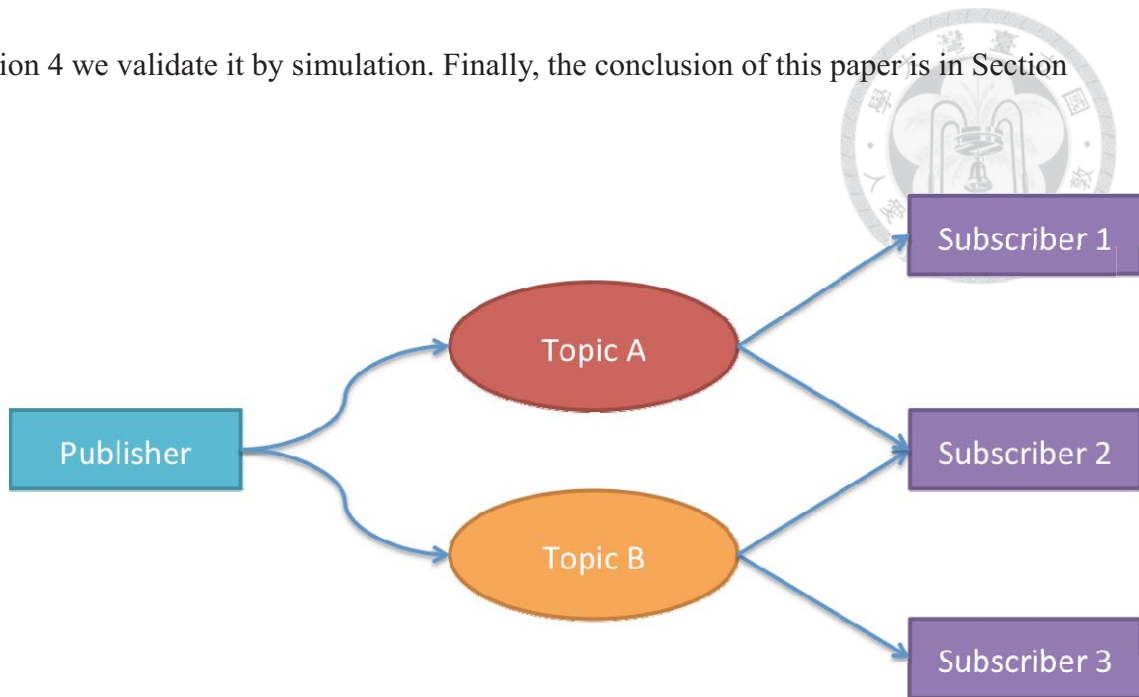


Fig. 1 Topic-based publish/subscribe

## Chapter 2 Related Work

### 2.1 Publish/Subscribe

Publish/Subscribe interaction scheme has been widely used, there are kinds of such system: namely, topic-based, content-based, and type-based. It decouples publishers and subscribers in time, space, and synchronization and make the system much more scalable [1]. This kind of system is widely used in many applications like social websites, sensor networks, event notification devices, etc. A lot of works has been done on managing overlays of topics by decentralized topic-based routing, but the impact on reliability due to clustering remains to be a problem.

### 2.2 Interest Correlation for Link Sharing

StaN [2], a protocol that takes advantage of the correlation of interests among nodes in a topic-based publish-subscribe system with the goal of decreasing the number of physical links established. The main idea shows in Fig. 2 and Fig. 3. Links can share between nodes and lower the network cost. However, it doesn't talk about time and assume that the loss is free, so we propose our modification of this, which take time into concern in later chapter.



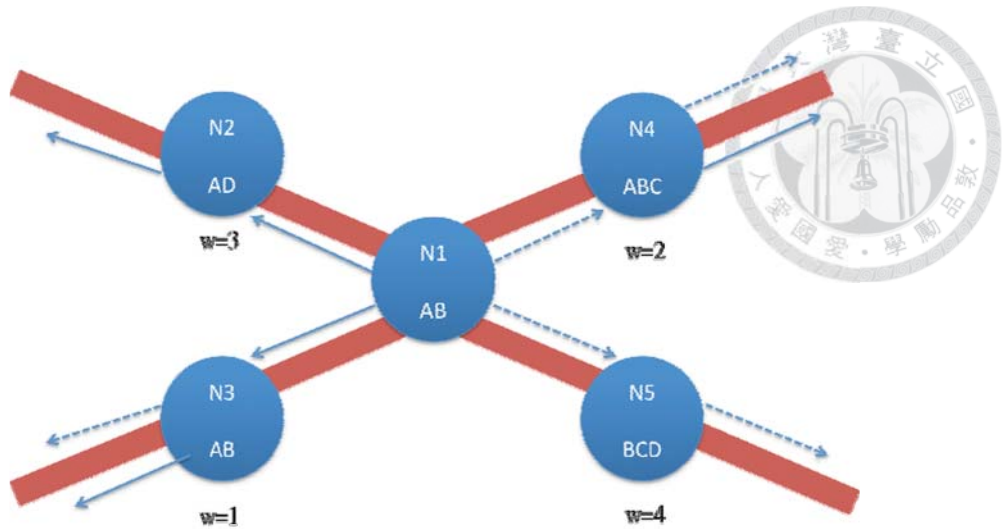


Fig. 2 sample of link sharing (before optimization)

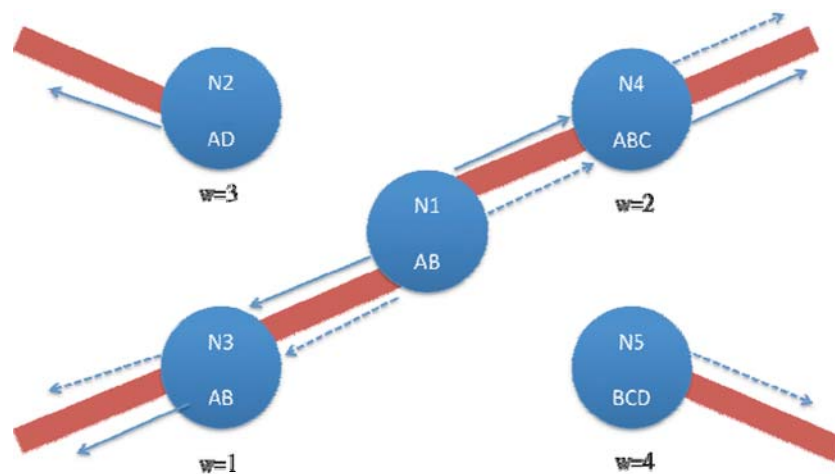


Fig. 3 sample of link sharing (after optimization)

### 2.3 Content Life Cycle

In a publish/subscribe system, the nodes' subscription of one topic may not always be the same. It may cancel it at some time. So, the node has this topic now doesn't represent it will have it later. Sometimes, the subscriptions' life cycle may affect the dissemination a lot. The node should decide if it can get data in the near future from another node when knowing that node might not have it in the future.

That is, the reliable nodes are preferred when we choose neighbor nodes [3].

As Fig. 4, in a period, the node may join and leave the subscription of the topic, and sometimes there is a neighbor node has the subscription of the topic in the same time slot. In these time slots, the link between both nodes can be established.

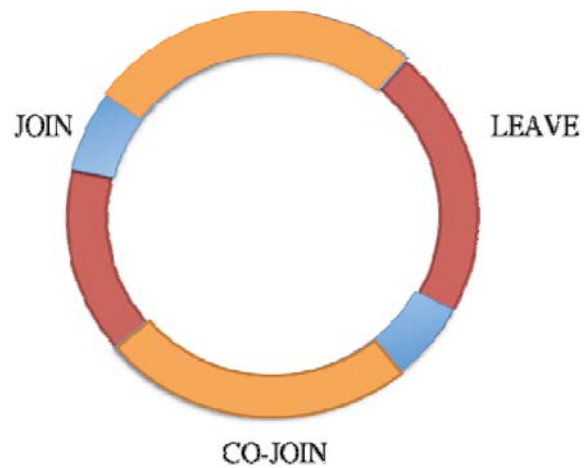


Fig. 4 content life cycle

## Chapter 3 Proposed System



### 3.1 System Model and Assumptions

In real world observations, we can conclude that there are a few topics, which have many subscribers, and a few nodes subscribe to many topics. This kind of distributions as Fig. 5 we call “power-law”, and we assume the topic popularity and subscriptions per node follow power-law distribution [4].

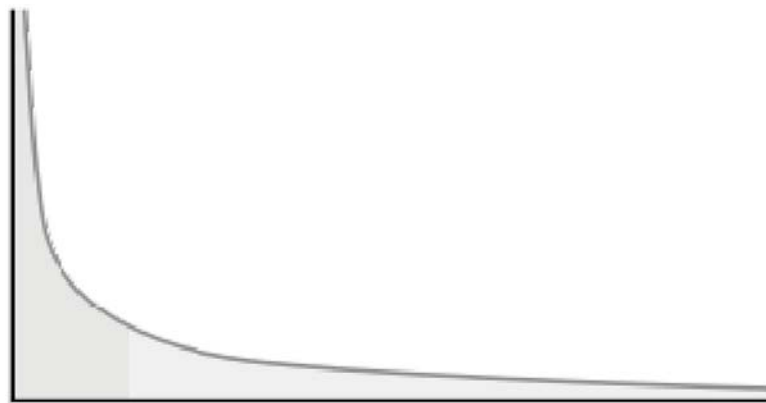


Fig. 5 Power law

We also assume that topic subscriptions are correlated, so the subscription sets may overlap. If subscriptions are uncorrelated, the system will fall into an overlay-per topic solution and disconnected components will occur in every single overlay, which will degrade the performance.

In such publish/subscribe network, there are two kinds of link:

- *Logical link.*

Exist between node and its neighbors in the same overlay. Lead to the connection of event publish path.

- *Physical link.*

Exist between node and its neighbors. Represent the network transmit path in

real world.



## 3.2 Design Rationale

For a topic-based publish/subscribe system, following properties should take into concern. These properties affect reliability and effectiveness [5].

- *Completeness*

All events for a topic should publish to all nodes, which subscribe to it. No nodes will miss these events for the subscribed topic.

- *Accuracy*

Nodes should not receive the event from the topic it does not subscribe to.

- *Node scalability*

The nodes subscribe to a topic should scale well as the number of nodes grows large.

- *Topic scalability*

The topics should scale well as the number of topic grows large.

- *Connectivity*

All nodes are reachable from any other node, which can ensure completeness.

- *Clustering coefficient*

A measure of the graph to show how the nodes cluster together. It is the ratio between the number of links that exist among the neighbors of a node and the

total number of links that may exist. This measurement quantifies how close the neighbors are. For practice purpose, it is related to the dissemination cost, because highly clustered sections of the overlay will produce redundant messages. For fault tolerance, highly clustered sections of the overlay tend to easily become disconnected from each other as the graph change, which will degrade overall connectivity. The calculation of clustering coefficient is shown as Fig. 6.

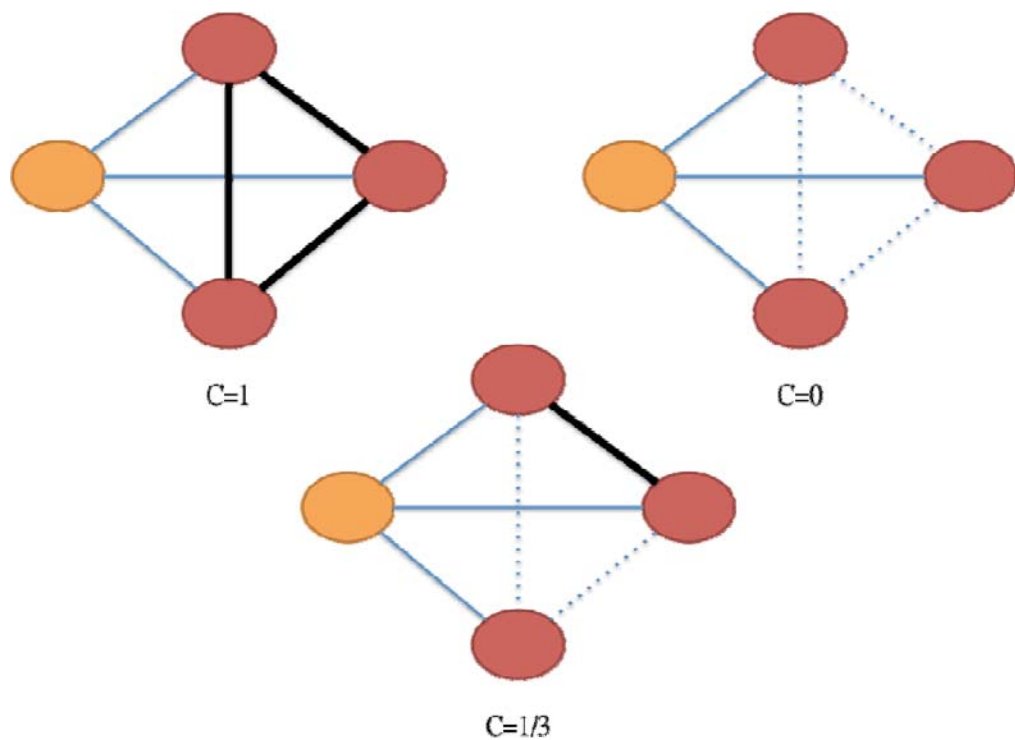


Fig. 6 clustering coefficient

- *Diameter*

It is a lower bound base on time and cost for a message to reach all nodes.



### 3.3 Link Management

For link management, we always can use global knowledge of the whole system to optimize the overlaps; however, to minimize the number of physical links is shown to be NP-complete [6]. That is, the scalability of the system cannot be satisfied.

The main idea of tStaN is simple: each node samples the subscribers of all the overlays it belongs to in a random manner periodically. In our assumption, subscriptions of a node are correlated, so these sample nodes will likely overlap in different overlays, in other words, topics. For each overlay, the node chooses a set of neighbors from the sampled set deterministically. As mentioned above, the nodes with highly probability will share across overlays; therefore, different logical link among overlays will map to single physical link and share this link. When the amount of topics grows large, the physical links of the network may grow slowly as logical links grows rapidly.

In such design, the correlation of subscriptions will lead to an undesired goal, clustering. It will impact the fitness of the overlay, because nodes will gather together with same subscriptions and increase fault rate. To face this problem, we should uniformly establish links when selecting neighbors, but not in random. By Uniformity for preserving good properties of random overlay [7] and determinism for consistency amount overlays, we can use a deterministic method to decide the nodes we select, and it will usually be the same candidates in different overlays. We can use a weight function to represent the link from node  $p$  to node  $q$ , and the value of the function is calculated by hash function [8] of the sum of the identifiers in string type. Unlike consistent hashing [9], which takes node distance into account; this value is asymmetry so it can avoid clustering.



### 3.4 Time-concerned content

For every node, the subscription to a topic can vary in time; it may cancel it now and re-join it later. So, in our assumption, we randomly let these nodes to join and leave for each topic in the future as Fig. 7, and record it as a timeslot to let other nodes to know it. This change make every node should take this into concern when it select neighbors, by compare this timeslot from other nodes to it self's time slot. Therefore, this selection may guarantee the higher reliability in the future.



Fig. 7 random join-and-leave time slot

### 3.5 Algorithm

Following is the proposed algorithm, the node periodically choose a node from its neighbors randomly in a topic (called view size), check the time slot is matched and add this node into candidate and pass to other neighbors, it a given TTL to maintain the collecting cost. At last, the set is sent to original node and the new view size is regenerated. By the hash function, node will always choose the same node in different overlay, so the purpose of link sharing is achieved.



tStaN protocol(node  $p$ )

**periodically**

**foreach** topic  $t \in p.$ topics **do**

$q = \text{RANDOMNODE}(p.\text{views}[t])$

**call**  $q.\text{COLLECTWALK}(p, \{\}, \text{TTL}, t, p.\text{slot}[t])$

**function** COLLECTWALK( $src, set, ttl, topic, slot$ )

**if**  $slot$  is in  $p.\text{slot}[topic]$  **then**

$set = set \cup \{p\}$

**if**  $ttl > 0$  **then**

$q = \text{RANDOMNODE}(p.\text{views}[topic])$

**call**  $q.\text{COLLECTWALK}(src, set, \text{TTL}-1, t, slot)$

**else**

**call**  $src.\text{COLLECTREPLY}(src, topic)$

**function** COLLECTREPLY( $set, topic$ )

$ViewSize = \text{size}(p.\text{views}[topic])$

List =  $\{q \in set \cup p.\text{views}[topic]$  sorted by WEIGHT( $q$ ) $\}$

$p.\text{views}[topic] =$  first  $ViewSize$  nodes from List

**function** WEIGHT( $q$ )

**return** HASH( STRINGIFY( $p.id$ ) + STRINGIFY( $q.id$ ) )

## Chapter 4 Simulation

For simulation, first we should fit the assumption of the topic and node subscriptions that follow the power law distribution. At first, we should give each node its subscription topics. We randomly put nodes and topics on a plane, and for each topic, use power law distribution to assign a radius to it. The nodes in the circle are the nodes subscribe to this topic, shown in Fig. 8. The simulation is as Fig. 9 and Fig. 10 that shows the assumption is taken into concern. Fig. 9 shows that some topics have many subscriptions and some topics have a few subscriptions, and the distribution is not uniform. Fig. 10 shows that there are a few nodes with many subscriptions and most of the nodes have a few subscriptions, which matches the power law.

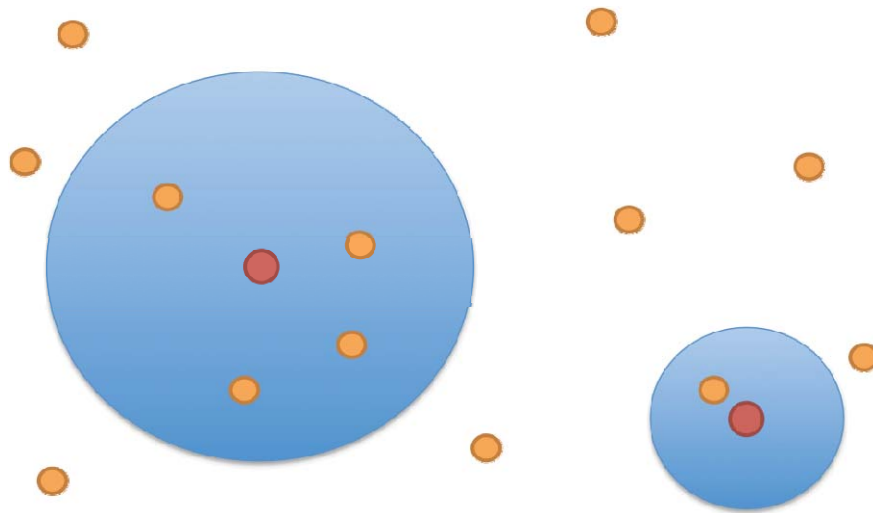


Fig. 8 subscription approach

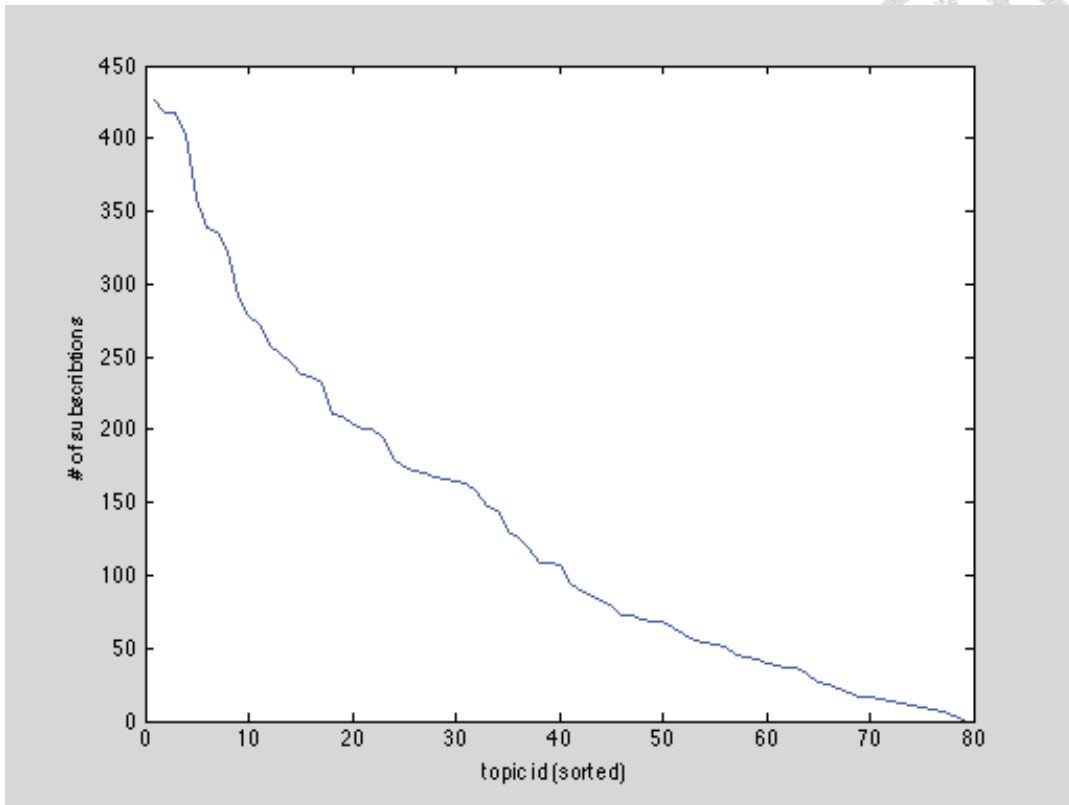


Fig. 9 Power law simulation (# of subscriptions for topics)

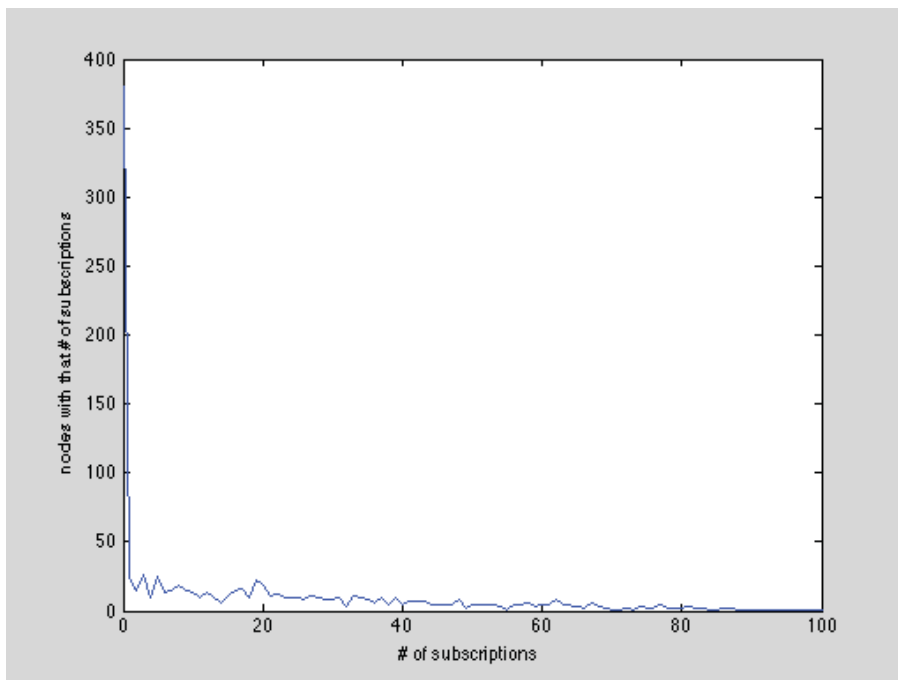
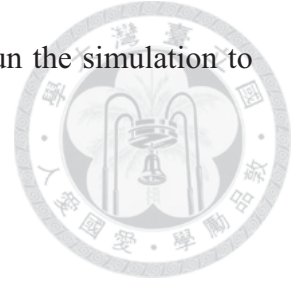


Fig. 10 Power law simulation (# of nodes with that # of subscriptions)

Table. 1, we list several conditions of nodes and topics, and run the simulation to show that the algorithm can indeed lower the physical links.



# Nodes	View Size	# Topics		
		100	200	300
1000	LVS	90	175	268
	Initial PVS	60	110	166
	Final PVS	33	50	71
2000	LVS	110	223	304
	Initial PVS	90	137	188
	Final PVS	45	62	72

Table. 1 View Size in several conditions

Fig. 11 is a condition of 1000 nodes and 100 topics. The logical view size always keep the same to ensure every node have sufficient links to neighbors in each topic. The physical view size drops down after a few cycles.

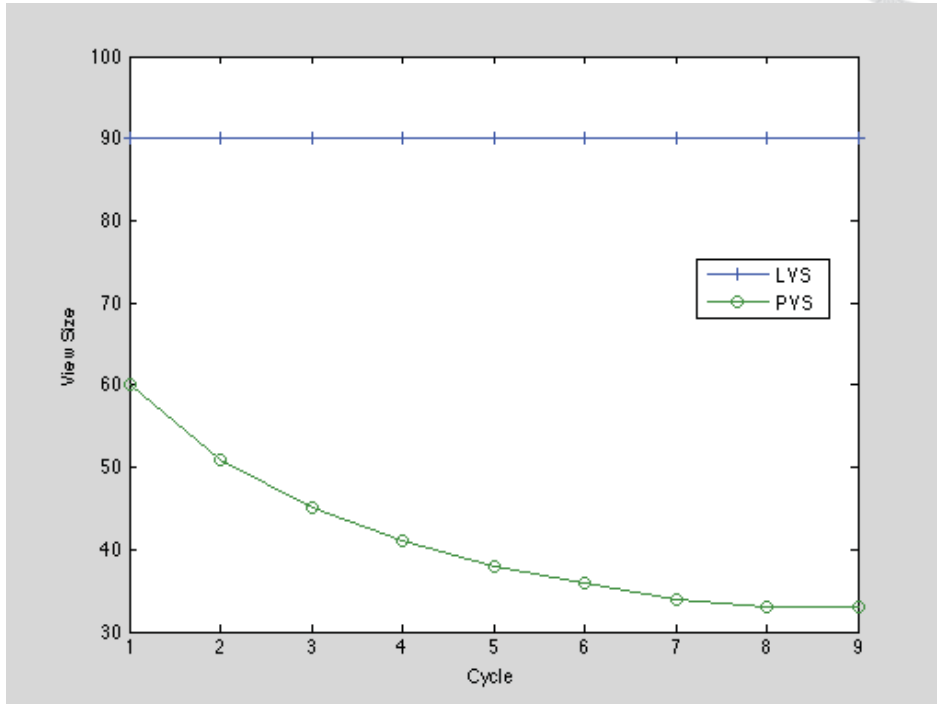


Fig. 11 View Size evaluation for 1000 nodes and 100 topics

Fig. 12 compares topics with 100 and 300, and nodes with 1000 and 2000. It shows that the algorithm can decrease the number of physical links about half.

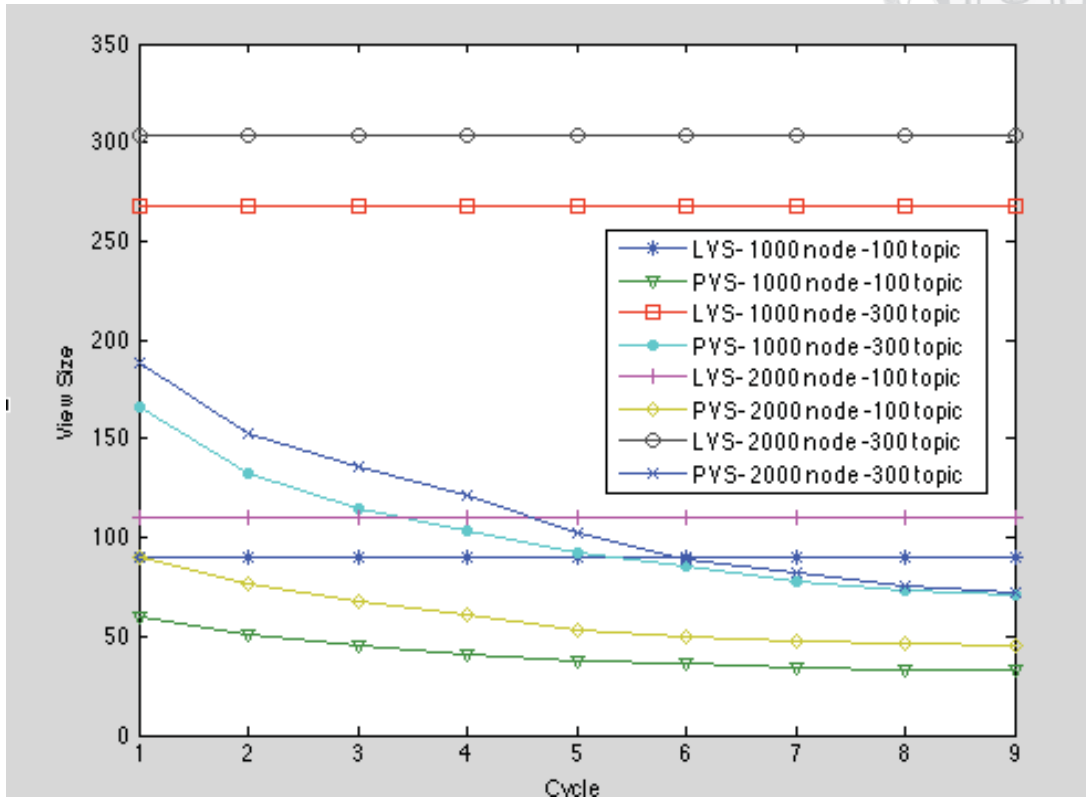


Fig. 12 View Size evaluation



Fig. 13 is a simulation comparison between algorithms with and without taking time into concern. When we use time slot matching, the ratio of link sharing is not as good as without time-concerned one, but it also can decrease physical links a lot.

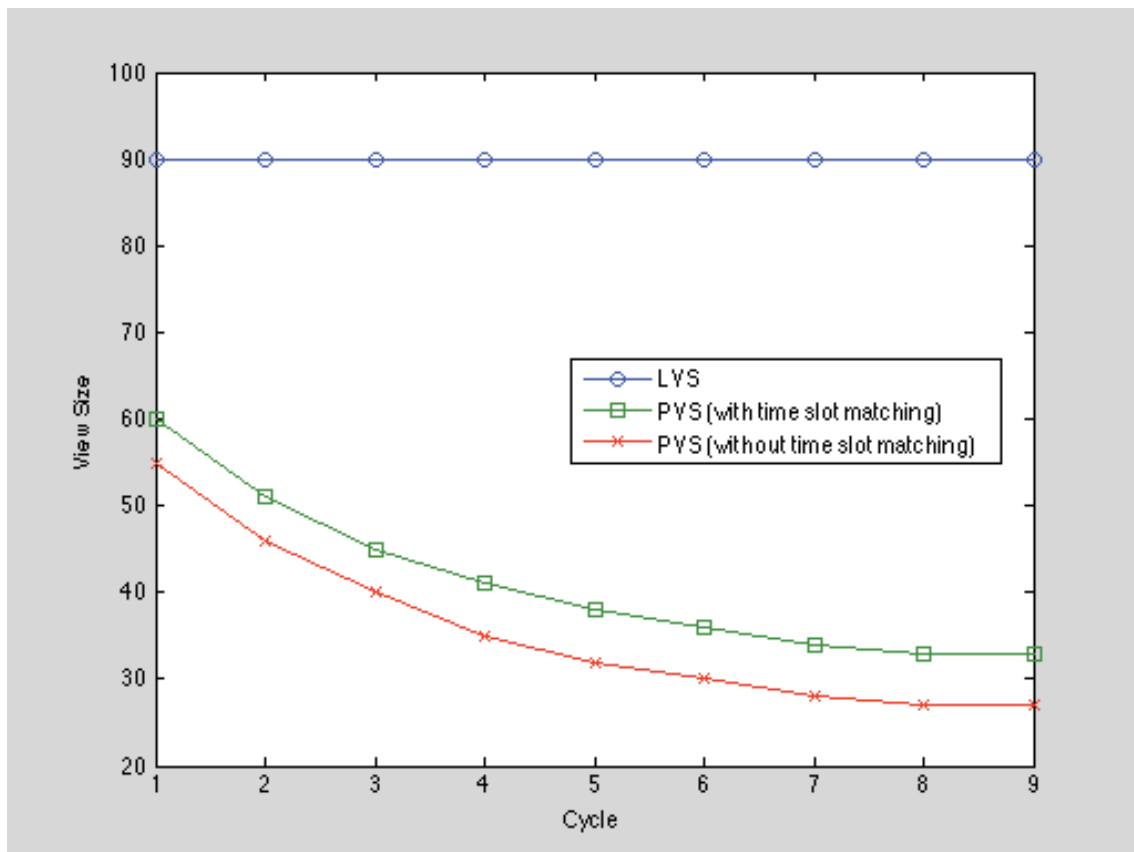


Fig. 13 time-concerned comparison

## Chapter 5 Conclusion

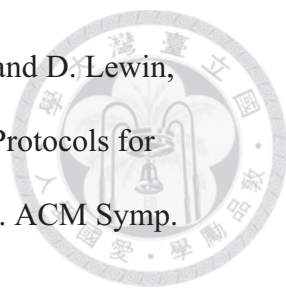
This work only discusses with link sharing and assumes that there was no fault happened. In [10] it discuss more about fault when the system is not loss free. We only take time into concern to make scenario much more real, because in publish/subscribe system, the nodes may join and leave as time going and should be take into concern. Here shows that the protocol also works in this scenario. Maybe time scheduling in subscriptions can be the future work.



## References



- [1] P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec, “The Many Faces of Publish/Subscribe,” *ACM Computing Survey*, vol. 35, no. 2, pp. 114-131, 2003.
- [2] Matos, Miguel, et al. “Scaling Up Publish/Subscribe Overlays Using Interest Correlation for Link Sharing,” *IEEE Transactions on Parallel and Distributed Systems*, vol.24, pp. 2462-2471, Dec 2013.
- [3] DAMIANI, Ernesto, et al. “A reputation-based approach for choosing reliable resources in peer-to-peer networks,” *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002.
- [4] H. Liu, V. Ramasubramanian, and E. Sirer, “Client Behavior and Feed Characteristics of RSS, a Publish-Subscribe System for Web Micronews,” *Proc. Internet Measurement Conf.*, 2005.
- [5] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, “Gossip-Based Peer Sampling,” *ACM Trans. Computer Systems*, vol. 25, no. 3, article 8, Aug. 2007.
- [6] CHOCKLER, G., MELAMED, R., TOCK, Y., AND VITENBERG, R. “Constructing scalable overlays for pub-sub with many topics,” *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing* (2007), ACM, p. 118.
- [7] P. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massoulié, “From Epidemics to Distributed Computing,” *Computer*, vol. 37, no. 5, pp. 60-67, May 2004.
- [8] M. Luby, “Pseudorandomness and Cryptographic Applications,” Princeton Univ. Press, 1994.

- 
- [9] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin, “Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web,” Proc. 29th Ann. ACM Symp. Theory Computing (STOC ’97), 1997.
- [10] M. Matos, P. Felber, R. Oliveira, J. Pereira, and E. Riviere, “Scaling Up Publish/Subscribe Overlays Using Interest Correlation for Link Sharing (Supplemental Document)”.