國立交通大學

資訊管理研究所

博 士 論 文

多路徑負載平衡演算法:特徵、效能分析及改良

Multihoming Load Balancing Algorithms:
Characteristics, Performance Analysis and Enhancements

研 究 生：彭祖乙

指導教授：楊　千　教授

中 華 民 國 九十四 年 十二 月

多路徑負載平衡演算法:特徵、效能分析及改良

Multihoming load balancing algorithms:

characteristics, performance analysis and enhancements

研 究 生：彭祖乙　　Student：Tsu-I Peng

指導教授：楊　千　　Advisor：Chyan Yang

國 立 交 通 大 學

資 訊 管 理 研 究 所

博 士 論 文

A Dissertation

Submitted to Institute of Information Management

College of Management

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of Doctor

in

Information Management

December 2005

Hsinchu, Taiwan, Republic of China

中華民國 九十四 年 十二 月

多路徑負載平衡演算法:特徵、效能分析及改良

學生:彭祖乙　　指導教授:楊　千

國立交通大學資訊管理研究所

摘要

Multihoming 已被運用在許多的大型的機構和企業環境用以提升網路運作的穩定性,它除了可使企業網路併用多條聯外線路減少線路失敗率(link failure rate)並可以用許多條比專線成比低廉且不需重新佈線的最後一哩線路(last miles)如 ADSL， Cable Modem， Power link ，和 Wimax 等來取代一條專線，用以減低線路成本。

在企業多條聯外路徑的 Multihoming 環境其量測及選擇路徑的方式已經有 BGP， RON 等相關研究加以探討，但是使用負載平衡方式來探討的研究還在發展。BGP 和 RON 需要連接的 ISP 特別支援以交換大量路由訊息，使用線路負載平衡方式則是一種不需要 ISP 提供支援的方式，企業使用一個單一(standalone)的線路負載平衡網路設備來連接多條 ISP 線路,這個網路設備將負責量測線路的交通情況及分配流量。目前有許多商業設備提供這樣的功能，在這些設備裡面，負載平衡的演算法扮演了網路交通量測及分配的重要腳色。

負載平衡演算法有許多種類和形式，本篇論文提出了一個分析架構，以四個特性參數來分類及比較不同演算法，並使用實境模擬(emulation)的方式產生不同的 traffic load 來比較這些特性參數和演算法在頻寬聚合及處理網路擁塞情形下的效能。模擬的結果顯示了不同演算法的特性、適用的網路環境、及相對的弱點。

本研究的第二部分，針對負載平衡演算法用於端對端(end-to-end)的量測及流量分配遇到的三個問題: 即時性、量測成本過高及頻寬使用率，提出了一個解決方法 WSDM。端對端的量測用於多路徑的網路架構主要在尋找最佳的線路。越

即時的量測越能反映網路的情況進而避開有問題的線路，但是也可能產生過多的量測封包，所以「即時」和「精簡網路資源的使用」成爲了兩難。此外，傳統在點對點量測及選擇線路的方法上都是使用某個量測時段中最佳的路徑來分配線路，這會造成其他線路頻寬使用的閒置。在這篇研究所提 WSDM 的方法，經過 emulation 的驗證，其使用權重方式能有效率的使用頻寬，並可用少量的資源達成即時點對點的量測。

Multihoming load balancing algorithms: Characteristics, performance analysis and enhancements

Student: Tsu I Peng Advisor: Chyan Yang

Institute of Information Management

National Chiao Tung University

Abstract

Multihoming has been applied in many large enterprises and organizations. In order to increase the reliability or reduce the cost for a multihoming network, many accessing technologies are used in the so-called “last mile”, such as ADSL, cable modem, power link, and WiMAX. The measuring and path-selecting operations of multihoming networks that employ Border Gateway Protocol (BGP) and Resilient Overlay Network (RON) have been discussed in many studies, but few refer to using load-balanced mechanism. BGP and RON must exchange routing information among the inter-connected ISPs rapidly to report a link failure situation, while a load-balanced mechanism can be performed on standalone equipment without support from ISPs. Load-balanced algorithms play important rules in standalone equipment to measure traffic conditions and select the proper path.

This study proposes an analytical model which uses four parameters to reveal the measuring and path-selecting behaviors of various load-balanced algorithms in multihoming networks. These load-balanced algorithms are compared under traffic aggregation and congestion conditions in an emulation environment. The emulation results display the characteristics, the network condition suitable for use, and the weakness of each algorithm.

The second part of this research proposes an end-to-end measuring algorithm to resolve problems of applying load-balanced algorithms in end-to-end transmissions in

a multihoming network. End-to-end measurement is performed in a multihoming network to locate the optimum path for a particular destination. Although this leads to more accurate network evaluations and fewer transmissions to failed links, more measurements occupy more multihoming equipment resources by using extra packets for end-to-end measurement, incurring heavy network traffic. Therefore, a trade-off exists between timely measurement and resource usage. Aside from this trade-off, bandwidth utilization is another issue in which the conventional end-to-end measurement approach only uses one optimal path within measuring interval idling links. This study also proposes a per-connection timely end-to-end measurement approach, called Weighted Self-Detected Measurement (WSDM), which consumes few resources. Our results further demonstrate that the proposed approach can effectively utilize bandwidth and keep clear of the outage path in an emulation environment.

致謝

如果仔細想想要感謝的人和事,可能篇幅將遠遠超過這一篇論文。在這裡僅能就攻讀學位期間的人事物重點式的感謝:

首先感謝楊千教授對我的指導,身教重於言教,楊老師在處理事務的速度和態度給了我莫大的學習,讓我操練了解什麼是重要的事,及給我很多切入要害的觀點。楊老師成爲了一個學習的典範,他在每個領域的全心付出及生活的紀律是我這一生要學習的重要功課。在這幾年求學和工作及小孩相繼出生中,確實讓我體會許多。

在資管所求學是我人生之大幸,(我在整理各項資料,發現了當時的錄取通知,都還記得那時興奮的心情)有機會和諸位教授學習,這裡每一位教授都爲這個美好的學習環境貢獻了相當的心力。在這裡感謝黎漢林,游伯龍,陳安斌,羅濟群,劉敦仁,蔡銘箴,陳瑞順及轉任(黃景彰教授)和新任的諸位教授。也感謝淑惠諸多的協助。由於我對資料庫,網路及決策支援特別有興趣,非常感謝授課教授們在這方面滿足我的求知需求。

論文寫作期間,傅振華學長給我諸多的指導及建議,著實讓我成長不少,十分謝謝他撥出許多的時間。同學楊耿杰也給我諸多協助,在此感謝。而 Paper 的 anonymous reviewer,對我來說,像是天使一般,謝謝 IEEE ICON 及 JIT reviewers 他們快速的 review 及指導。

另外感謝我工作期間的上司,在我以生手進入職場時,發現對實務懂得有限,幸賴交大學長黃文哲一步一步的指導我,給我打下了網路及核心開發的基礎。在第二個工作期間的老闆林義順,大力支持我"負載平衡"實務性的研究,除了讓這個研究能跨國深入許多客戶,也讓我學習許多管理領域產品的思維,在擔當 project leader 期間,所有的同仁也是我一輩子感激的對象。

在專心寫論文的期間,父母在經濟的支持及不斷的打氣,讓我能持續下去,父親已經 86 歲了,給了我許多人生的智慧。

妻子文玲,及兩個小孩國芯和國義則是每天陪伴我的對象,謝謝他們對我的鼓勵和包容,我也將努力帶給他們一個愛的環境。

最後感謝上帝,因爲我實在非常有限,常常需要支取愛、信心、智慧及勇氣。

Contents

Chapter 1 Introduction 1

1.1 Multihoming load-balancing algorithms 1

1.2 Enhanced multihoming load-balancing algorithm for end-to-end transmission 2

Chapter 2 Literature Review 3

2.1 BGP 3

2.2 RON 5

2.3 Load balance and end-to-end measuring 7

Chapter 3 Behaviors and performance analysis of load-balancing algorithms 10

3.1 Common operation parameters in LLB algorithms 10

3.1.1 Path-selecting period Pτ 11

3.1.2 Measuring period 11

3.1.3 Measuring distance D 13

3.1.4 Dispatching schemes 14

3.1.5 Measuring type 16

3.1.6 Generalized Balance Algorithm Characteristic (BAC) function 17

3.2 Characteristics of load-balanced algorithms 17

3.3 Performance indicator 20

3.4 Performance analysis 21

3.4.1 Aggregation analysis of load-balancing parameters 23

3.4.1.1 Comparison on path-selecting period 24

3.4.1.2 Comparison of dispatching scheme and measuring period 26

3.4.1.3 Comparison of measuring type and measuring distance 28

3.4.1.4 Summary of experiment results 29

3.4.2 Congestion analysis of load-balancing algorithms 29

3.4.2.1 Comparison of local congestion (last mile congestion) 30

3.4.2.2 Comparison of remote congestion 35

3.4.2.3 Discussion of experiment results 37

Chapter 4 Enhanced load-balancing algorithms for end-to-end traffic condition 38

4.1 Issues 38

4.1.1 Timely measurement 38

4.1.2 Dispatching scheme 41

4.2 Weighted Self-Detected Measurement (WSDM) 41

4.2.1 Connection cache and NAT mechanism 42

4.2.2 Algorithm 42

4.3 Comparison of operations 45
4.4 Emulation results 46
4.4.1 Congestion 46
4.4.2 Failover 48

Chapter 5 Conclusion 51

List of Tables

Table 1 **A taxonomy of load-balanced algorithm** 20
Table 2 **T1 and 512k ADSL monthly fees for the three major ISPs in Taiwan** 24
Table 3 **Computation of EBU of two workloads** 26
Table 4 **Comparison of operations** 45
Table 5 **Comparison of resource usages** 45
Table 6 **Performance comparison of WSDM and PCM** 50

List of figures

Figure 1 Multihoming load-balanced system 10
Figure 2 Measuring distance 13
Figure 3 (a) Best dispatching scheme (b) Weighted dispatching scheme 15
Figure 4 Emulation environment 22
Figure 5 (a) Comparing mean throughput of path selection periods (b) EBU of the two path selection periods 25
Figure 6 (a) Comparing mean throughput of dispatching scheme and measuring time T (b) EBU of dispatching schemes and measuring times 27
Figure 7 (a) Comparing mean throughput of measuring types and distance (b) EBU of the three measuring types 28
Figure 8 (a) Comparison of mean throughput of last mile congestion (b) EBU of last mile congestion 31
Figure 9 (a) Number of connections on each link (b) Mean absolute deviation of different users' throughput 33
Figure 10 Per session finishing time and throughput in (a) WMORBF (b) FRRTDF 34
Figure 11 (a) Mean throughput of remote congestion (b) EBU of remote congestion 36
Figure 12 Number of connections on each link at workload L(3,1) 37
Figure 13 Time slots of connection arrivals and end-to-end measurement actions 38
Figure 14 Illustration of connection cache mechanism 42
Figure 15 WSDM algorithm **44**
Figure 16 Congestion responding (a) Last-miles (b) Remote 47
Figure 17 The graphs depict the different algorithms' throughput 49

Symbol Description

PS_c	path selecting period
Mτ	measuring period
t_c^j	arrival time of the j^{th} connection
ΔT_{fm}	the time to complete a measurement over a BL
D^j_{ps}	measurement delay for j^{th} $PS^j_\tau(t_c)$
$\boldsymbol{S_{id}}$	a set of nodes in a transmission path
$H(S_{id})$	function to return the number of hops over a transmission path
D_{length}	length of measuring distance
BL	balancing Link
B_{so}	first hop of a BL
M^j_i	measuring result of the j^{th} connection over BLi
Wi	the weight of BLi
SD_i	number of dispatched sessions within a duration over BLi
BAC()	a generalized balance algorithm characteristicfunction
T_c	testing complete time
SN	number of concurrent sessions for a user
LM	a link with the bandwidth which can take the sum of many narrower links
CA_i^d	connection arrival time of the ith connection to destination d

Chapter 1 Introduction

1.1 Multihoming load-balancing algorithms

Multihoming has been applied in many large enterprises and organizations, due to its benefits such as scalability, reliability [1], and low cost [2][3]. Within a network domain, a multihoming scheme applies several links from one or several ISPs to connect to the Internet through an edge router. A multihoming scheme uses one routing and addressing scheme to dispatch network traffic to the links.

Multihoming systems use any of three routing and addressing schemes: BGP, RON and load-balance. BGP and RON must exchange routing information among the inter-connected ISPs rapidly to report link failures. The loading of exchanging routing information is heavy if every end point with multi-connected ISPs requires multihoming routing.

Link load balance schemes attempt to balance the load of traffic carried over links and to apply the NAT) [4] scheme for address assignment. A link load balance dispatches network traffic to all links with a best link selection or a weighted scheme. A link load balance scheme can measure traffic load using many methods, including last mile available bandwidth measurement and a response time measurement. Currently, many commercial products use the link load balance scheme, including Radware [5], F5 [6], and Deansoft [7]. A link load balance scheme does not need to exchange much routing and addressing information between an enterprise network and its connecting ISPs. Moreover, there are many traffic dispatching algorithms used by link load balance schemes and these algorithms can be characterized by a few generic parameters. This study compares the performance and bandwidth utilization of these parameters and algorithms by traffic aggregation and congestion conditions.

1.2 Enhanced multihoming load-balancing algorithm for end-to-end transmission

The measurement mechanisms of link load-balancing algorithms may use the number of connections, available bandwidth, or response time to measure the traffic condition over each link. Outage and congestion could occur at any point of an end-to-end transmission path. Therefore, end-to-end measurement is required to obtain the precise traffic condition in a multihoming environment.

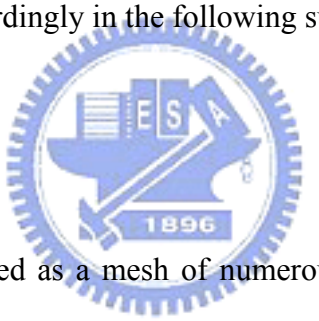
Akella [8] proposed an end-to-end measuring scheme in a multihoming network, using TCP to obtain the end-to-end response time. A timer with a fixed duration processes a TCP end-to-end measurement, making it possible to measure the response time for popular destinations of the multihoming network. However, this scheme suffers from a heavy computational load when measuring end-to-end traffic conditions efficiently for each connection arrival, as discussed in a later section.

The path-selecting scheme used by BGP, RON, and Akella's approach may utilize only one path within a measuring interval. A longer measuring interval causes other paths to become idle and lowers the bandwidth usage.

This study addresses the problems of timely end-to-end measurements in multihoming networks and proposes a scalable end-to-end measuring approach, Weighted Self-Detected Measurement (WSDM). WSDM can provide an efficient end-to-end traffic condition for each newly arriving connection without consuming too many resources to handle processes and network traffic. WSDM can also provide excellent bandwidth utilization based on its weighted dispatching scheme.

Chapter 2 Literature Review

Multihoming in a computer network means the ability to have multiple network addresses in one gateway, usually on different networks. For example, multihoming might be used to create a system in which one address is used to talk to ISP1 and the other to talk to ISP2. Large enterprises, campuses, and data centers employ multihoming as a way of ensuring continued operation during connectivity outages or traffic congestion. In particular, multihoming can be leveraged for improving wide-area network performance, lowering bandwidth costs. Multihoming systems could use any of three routing and addressing schemes, BGP, RON, and load-balance, which will be introduced accordingly in the following subsections.



2.1 BGP

Internet can be abstracted as a mesh of numerous ASs (autonomous systems) connected by inter-domain links and communicated by inter-domain routing protocol. BGP is the de facto inter-domain routing protocol that is used by BGP routers associated with different ASs to exchange reach-ability information and determine the routes for packets traversing through multiple ASs.

BGP is a path-vector protocol, as its routing information contains a sequence of AS numbers whereby the corresponding routing updates have traversed. It is an enhancement on the distance vector protocol which uses path information to prevent routing loops. BGP chooses the best route based on the shortest number of ASs on AS paths. BGP allows user to define routing policies to override the distance-based metric.

The routers participating in a BGP session are called peers. The connection between a pair of BGP routers is called a BGP session. BGP uses TCP to exchange peer routing information. After a BGP session is established, its peers will exchange the entire routing table. After the establishment, only incremental updates to the routing table are exchanged.

There are four types of BGP messages: OPEN, UPDATE, NOTICICATION, and KEEPALIVE. OPEN and KEEPALIVE are mainly for establishing and maintaining BGP sessions respectively. NOTIFICATION is sent out when an error condition is detected. UPDATE is for transferring routing reach-ability information.

BGP's routing information base consists of three parts: Adj-RIBs-In, Loc-RIB, and Adj-RIBs-Out. The Adj-RIBs-In contains unprocessed routing information that has been advertised by its peers. The Loc-RIB contains the routes that have been selected by the local BGP router's decision process. The Adj-RIBs-Out organizes the routes for advertising to specific peers through the UPDATE messages.

BGP can support multihoming, as it could use the RFC 2260 or RFC 1518[9][10] standards to connect to one or more ISPs with different network addresses. These schemes allow ISP links within an enterprise network to send and receive IP packets with different IP addresses through different ISP links. On a routing selection, BGPv4 [11] examines values for a set of attributes in the routing table and selects the best routing path. The values of these attributes could be assigned by a human operator or by the shortest path calculation.

In a multihoming environment, BGP needs to exchange an entire routing table among routers. If each enterprise network needs to exchange BGP messages with the ISPs which the enterprise network accesses, the ISPs would suffer $O(n^2)$ loading for

each BGP message exchange. Therefore, only large enterprises can afford BGP-related products and services.

Aside from the complexity, slow convergence is another well-known problem of BGP [12][13][14][15][16]. In fact, a consistent view of the network topology may take tens of minutes to reach after a routing instability situation [17][18].

A routing instability situation is described as follows.

A failure occurs on part of the network, such as a single link failure or a set of network prefixes unreachable or duplicate routes may cause routing instability. The router interconnected to the failure point would detect it and withdraw the corresponding route(s). Subsequently, the neighbor routers peering with the router will receive the new routing updates and take corresponding action like propagating the failure information. The propagation would continue with the updated information. The ripple interferes with a large portion of or even the whole Internet. At the same time, since each router receives the updates and makes its own decision based on its local information and policies, it may advertise different update results to its peers. This may lead to different views on different routers about the topology change. Finally, there may be more and more or even divergent routing updates going on, which cause a larger scale of routing instability.

2.2 RON

RON [19][20] proposed an improved scheme to solve the difficulty when BGP cannot instantly reflect network verity. It is found that RON usually bypasses 30-minute Internet failures and dramatically reduces the loss rate between two hosts.

The main goal of RON is to enable a group of nodes to communicate with each other in the face of problems with the underlying Internet paths connecting them.

RON detects problems by aggressively probing and monitoring the paths connecting its nodes. If the underlying Internet path is the best one, then that path is used and no other RON node is involved in the forwarding path. If the Internet path is not the best one, then RON will forward the packet by way of other RON nodes. RON nodes exchange information about the quality of the paths among themselves via a routing protocol and build forwarding tables based on a variety of path metrics, including latency, packet loss rate, and available throughput. Each RON node obtains the path metrics using a combination of active probing experiments and passive observations of on-going data transfers.

RON monitors its virtual links using randomized periodic probes. The active probe component maintains a copy of a peers table with a next-probe-time field per peer. When this field expires, the probe process sends a small UDP probe packet to the remote peer. Each probe packet has a random 64-bit ID. When a node receives an initial probe request from a peer, it sends a response packet to that peer and resets its probe timer for that peer. When the originating node sees this response packet, it sends a packet back to the peer, so that both sides get reach-ability and RTT information from 3 packets. The probing protocol is implemented as a RON client, which communicates with a performance database (implemented as a standalone application running on the Berkeley DB3 backend) using a simple UDP-based protocol. These network node performance measurements are placed into this performance database, and different nodes exchange a performance database via a link state routing protocol.

Given that different nodes are continually measured, RON takes only about ten seconds to run away from the outage path. However, RON has high measurement costs of about 33 kbps among the 50 nodes [19]. This measurement cost grows

exponentially, enabling hundreds of nodes undergoing mutual measurement to generate large measuring traffic flows, ultimately burdening the underlying network.

2.3 Load balance and end-to-end measuring

A link load-balanced scheme applies the server load-balanced mechanism which dispatches traffic to a server pool. RFC2391 [20] describes the algorithms and mechanisms of the server load-balanced scheme. In server load balance schemes, a load share dispatcher depends on servers' loading and access costs to dispatch different sessions to different servers and replaces the destination address in the IP header with the address of the assigned server.

RFC3291 proposes several dispatching algorithms, including Round Robin (RR), Least Loading First (LLF), Least Traffic First (LTF), Ping for Most Responsive host (PMR), Weight LLF (WLLF), and Weight LTF (WLTF).

To calculate traffic loading, LLF counts the number of sessions in each link, while LTF uses the accumulated packet length of each link. PMR estimates the response time using the round trip time of an ICMP packet. WLLF and WLTF assign weights to servers according to their service capability and route traffic accordingly. For example, if server A has a weight of 3 and server B has a weight of 1, then 75% of the sessions are dispatched to server A while only 25% of the sessions would be dispatched to server B.

When the load balance targets are not servers and the targets are ISP links, the algorithm needs to be altered. First, the loading and traffic statistics for the server load balance must be converted into ISP link statistics within an enterprise network. Second, destination addresses do not need to be switched. Instead, the original private source IP addresses are replaced with public IP address as in end-to-end measurement.

Several commercial link load balance products using LLF, LTF, and PMR-like algorithms are available, such as Radware [5], F5 [6], and Deansoft [7].

A link load balance scheme does not require much exchange routing and addressing information between an enterprise network and the ISPs, with which the enterprise network internet links are connected. Moreover, it is not necessary for a link load balance scheme to execute an end-to-end measurement. End-to-end measurement obtains the precise traffic condition in a multihoming environment in the situation when an outage and congestion could occur at any point of an end-to-end transmission path. Akella [8] proposed an end-to-end measurement scheme at a multihoming network, including two measurements, passive and active, for the end-to-end traffic condition.

Passive measurement tracks the performance to destinations by Web requests initiated by clients in the enterprise. If n ISP links can route the packet to the destination, then the measurement observes at least n requests to obtain a destination's performance data. A performance sampling of a destination on an ISP link is updated when the new sample finds a previous sample out of date by the predefined sampling lifetime.

Active measurement automatically sends TCP packets to a destination from a different link to observe the response time for every sampling interval T. A TCP measuring packet primarily utilizes a SYN packet at the connection setup phase and also enables the ACK bit. The system calculates the response time from the time of sending back the TCP RST packet from the remote site.

Reducing the measurement costs of both the passive and active measurements requires selecting limited destinations to be measured. A destination list is maintained and constrained to a specific size for end-to-end measurement.

Akella found that using the most up-to-date data and shortest measuring interval yields a well throughput.

Chapter 3 Behaviors and performance analysis of load-balancing algorithms

This study defines a routing path selection with the load balance scheme as Link Load-Balance (LLB). A diagram of a LLB system operation is shown in Figure 1, and there are multiple internet links in a LLB system. An internet link is a logical path for TCP/IP applications from one ISP link in a LLB system to all their destinations - it is named Balancing Link (BL) in this study. BL_i means the ith BL in a LLB system, and Bs_0 means the first hop over all BLs.

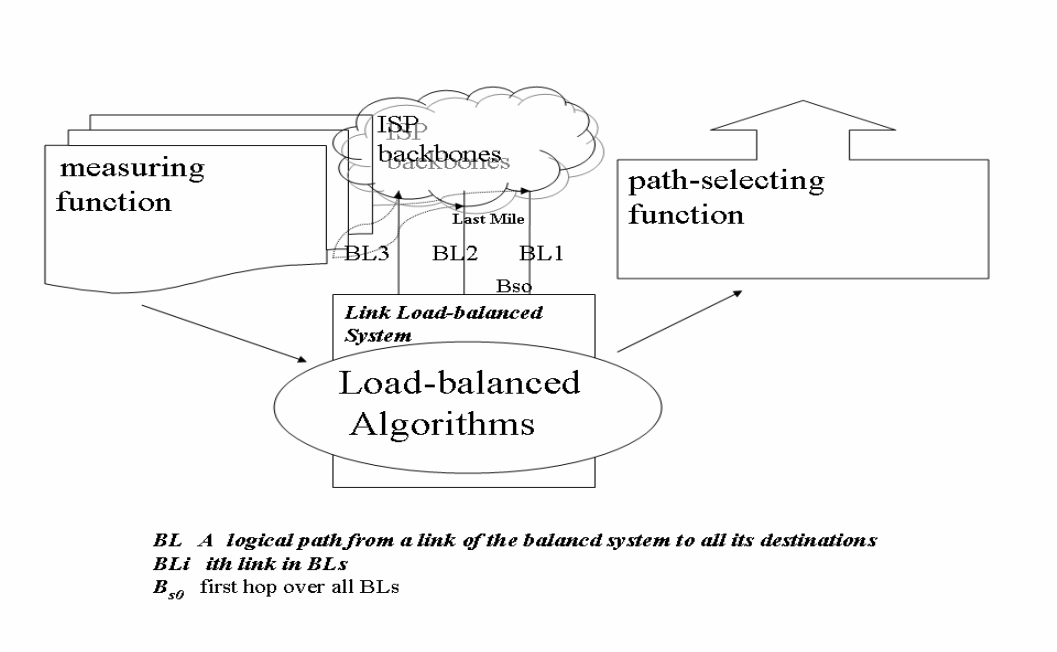


Figure 1 Multihoming load-balanced system

Several load-balancing algorithms are used by a LLB system to measure traffic load over each BL. From the measured results, a BL could be selected as the routing path for TCP/IP applications sessions.

3.1 Common operation parameters in LLB algorithms

All LLB algorithms can be characterized by five common operation parameters: path-selecting period (P_τ), measuring period ($M\tau$), measuring distance (D), dispatching-scheme, and measuring-type. We now discuss these parameters in details.

3.1.1 Path-selecting period Pτ

Every packet normally has to be assigned a routing path for its transmission. However, this path assignment can be persistent throughout a flow which may be a TCP/IP connection or the packets transmitted between a source/destination IP pair within duration.

When the persistence of a routing path is within a connection, the path-selecting procedure for the multihoming load-balanced system will only be executed at every arrival of a new TCP/IP connection. The rest of the packets belonging to this connection will have a persistent path as the first packet by some caching mechanisms. Under this connection-based persistency, the path-selecting period is a random variable of the inter-arrival time of two connections. We denote this random variable of the path-selecting period as:

PS_c ; PS_c^j is the j^{th} path- selecting period.

When the persistence of a routing path is within a range of time whereby the packets are transmitted between a source/destination address pair, the path-selecting period is the random variable of inter-arrival time two different source/destination address pairs, which we denote as PS_{ip}.

The throughputs of these two path-selecting periods are compared and discussed in Section 3.4. In the following sections, the symbol PS_c denotes the path-selecting period.

3.1.2 Measuring period

The three choices of measuring period (Mτ) are: PS_c, fixed duration, and continuous back-to-back.

When Mτ = PS_c, a measurement is triggered by a new connection. Each PS^j_c is delayed with a varying ΔT_{fm}, where ΔT_{fm} is the time to complete a measurement over a BL.

If ΔT_{fm} is larger than PS^j_c, then ΔT_{fm} would keep accumulating as a measurement delay. This measurement delay may be added to PS^{j+1}_c. Hence, the measurement delay D^j_{ps} for PS^j_c in the worst case is shown in (1).

$$D^j_{ps} = \sum_{j} MAX(0, D^{j-1}_{ps} + (t_c^{j-1} + \Delta T^j_{fm}) - t_c^j), \quad (1)$$

where t_c^j is the arrival time of the j^{th} connection.

Without considering a new connection arrival, Mτ could be assigned with a fix period T. With a fixed Mτ, there is a time gap between the j^{th} path-selecting time and its latest measurement time. The time gap is denoted as $\underline{\Delta T}_{ps\text{-}m}$ and it can be expressed as (2):

$$\Delta T^j_{ps\text{-}m} = M_\tau^j - t_c^j, \quad (2)$$

where M_τ^j is the time of the latest measuring action of t_c^j. To minimize $\Delta T^j_{ps\text{-}m}$, one can choose a back-to-back measuring that is restarted at the end of each measurement.

A measurement period, Mτ, should compromise its measurement loading and delay, ΔT_{fm}. Using Mτ = PS_c in a busy station, the BL measurement is triggered as many times as the number of new arrivals. Suppose that a server has 1000 connections arriving within one second. There are 1000 path selections required and each PS_c is allowed 1ms at most. However, if ΔT_{fm} is 50 ms, then the delay of the 1000^{th} connection would be 50 seconds (D^{1000}_{ps} = 1000 × 50ms = 50 seconds) in the worst case.

Although a long measurement period reduces the measurement cost, new connections use the obsolete networking status to conduct a load balance. However, a short measurement period offers timely measurement information to balance traffic load over each BL with a high cost. Therefore, there is a trade-off between measurement periods and measurement costs.

3.1.3 Measuring distance D

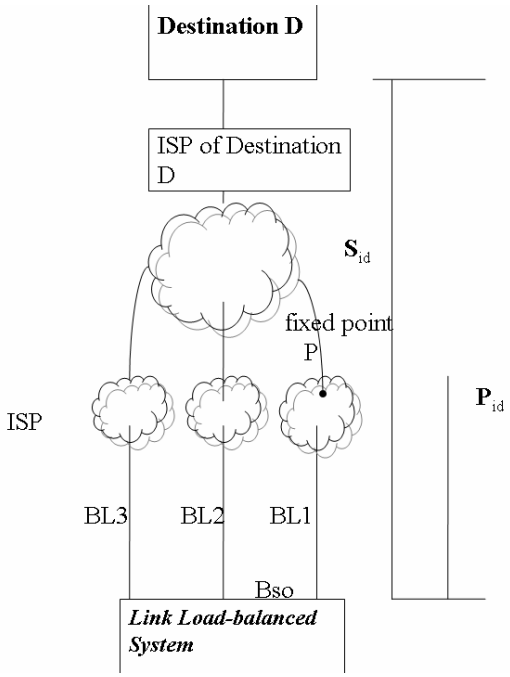


Figure 2 Measuring distance

As shown in Figure 2, a measuring distance means the distance from B_{so} of the LLB system to one specific measurement node: B_{so} itself, an access ISP, a fix node, or a destination of a session. Several symbols in measuring distance are defined as follows.

$\boldsymbol{S_{id}} = \{B_{so}, h_i1, h_i2 \ldots\ldots d\}$, where h_ij represents the jth hops from B_{so} to destination d over BLi; $H(S_{id})$ is a function to return the number of hops over a

transmission path; $H(S_{id})$ can be used as the length of measuring distance, D_{length}. Additionally, $\mathbf{P}_{id}$ is a set of hops from B_{s0} to a fix node P, $\mathbf{P}_{id} \in \mathbf{S}_{id}$.

Several cases with different visibility or measuring distance lengths are described as follows.

$D_{length} = H(B_{so}) = 0$

The traffic load information of each BL_i, i.e. the number of connections, is derived locally from the LLB system itself.

$D_{length} = H(\{B_{so}, h_i 1\}) = 1.$

It is possible for all links to share the same routing path, but differ in the last miles in the case of all BLs accessing the same ISP. In this case, a traffic load measurement is processed at the last miles, from B_{so} to an access ISP over a BL. The measurement of the last mile therefore dominates the loading situation of each BLi.

$D_{length} = Hn(\boldsymbol{P_{id}})$.

When selecting a specific middle point P over a BL as a measurement destination, a traffic load can also be measured from the B_{so} to this selected node over the BL.

$D_{length} = H_n(\boldsymbol{S_{id}})$.

The end-to-end measurement cost of a connection is high and the measurement increases the process burden of the destination host.

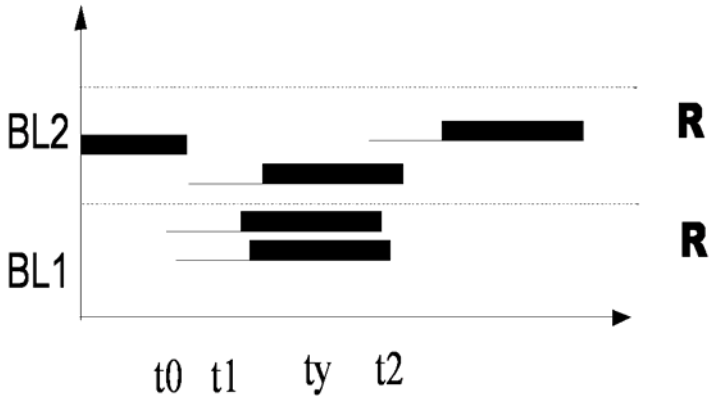
3.1.4 Dispatching schemes

After measuring traffic load over BLs, dispatch schemes are required to assign a BL to a specific session. There are two dispatching schemes used in a load balance system: "best" and "weighted".

The "best" dispatching scheme dispatches a session to the path with the best measuring result. The best dispatching method might cause a "self-congested condition" where all sessions are dispatched to the best BL in a measuring period.

Figure 3(a) shows that three sessions are dispatched to the best BL, i.e., BL_1, during the interval $[t_0, t_1]$. These three sessions cause congestion at time t_y until BL_2 becomes the best measured path in t_2. However, the available bandwidth of BL_2 is wasted during interval $[t_0, t_1]$.

(a) Best



(b) Weight

Figure 3 (a) Dispatching traffic using "Best" scheme (b) Dispatching traffic using "Weighted" scheme. The thin line indicates that the loading of a session in the beginning phase is not heavy. R is the rate provided by each BL.

Let M_i^j be the measuring result (available bandwidth, round trip delay, or number of connections) of the j^{th} connection over BLi. The "self-congested condition" is caused with the same measuring results during consecutive periods of path selections which can be expressed as (3).

$$\exists \quad M^{j}_{i} = M^{j-1}_{i}, \text{ for some } j, \tag{3}$$

where M^{j}_{i} is the measured result of BLi during the PS^{j-1}_{c} and PS^{j}_{c} periods.

To avoid the "self-congested condition", the "weighted" dispatching scheme can be used. According to the measuring results, the "weighted" dispatching scheme is based on a calculated ratio to dispatch sessions over all BLs. A path selection applies Equation (4) as follows.

$$MIN(SD_i \quad / \quad W_i) \quad for\ i = 1 \ldots n, \tag{4}$$

where SD_i means the number of dispatched sessions within a duration over BLi_and Wi is the weight of BLi.

The value W_i can be calculated from Equation (5).

$$Wi = MAX(W_{max} \times Ratio(M_i, Best(\mathbf{M_{path}})), W_{min}). \tag{5}$$

In Equation (5), W_{max} is the assigned maximum weight of an ISP path; M_i is the measuring result from $path_i$; $\mathbf{M_{path}}$ is the set of all the measured results of all ISP paths; and W_{min} is the assigned minimum weight for an ISP path. The terms W_{max} and W_{min} are used to adjust a dispatching number range. The ratio function returns the ratio of Mi and Best ($\mathbf{M_{path}}$). The Best ($\mathbf{M_{path}}$) is either a maximum (for available bandwidth) or a minimum (for round trip time).

Figure 3(b) is a diagram of the "weighted" dispatching scheme with $W_1 = 2$ and $W_2=1$. The weights are dynamically adjusted in each measurement period Mτ or based on physical link bandwidth to set a static value.

3.1.5 Measuring type

There are three measuring types, counting connection number (CSN), subtracting traffic (ST), and measuring response time (MRT), in a load balance algorithm. The CSN counts the number of sessions in a load balance system and

expects to dispatch sessions over BLs in order to share bandwidth fairly. The measuring distance used by CSN is $D_{length} = 0$. The ST uses the last mile ($D_{length} = 1$) as its measuring distance. Since its measuring distance is from B_{s0} to the first node and the physical link bandwidth is known, the bandwidth of the last mile over a BL can be easily measured by subtracting the used bandwidth from the physical link bandwidth. The MRT uses a probe packet to measure the round trip time from B_{s0} to a node as in the case of measuring distance $H(\boldsymbol{P}_{id})$ or $H(\boldsymbol{S}_{id})$ described in section 3.1.3.

The LLB algorithms discussed in RFC 2391 are RR, LCF, LTF, and PMR. These algorithms are summarized in section 2 and can be categorized into three measuring type as (1) CSN, such as RR and LCF; (2) ST, such as LTF; and (3) MRT, such as PMR.

3.1.6 Generalized Balance Algorithm Characteristic (BAC) function

A generalized balance algorithm characteristic (BAC) function can be used to describe the characteristics of LLB algorithms. The BAC function of a connection-based system has four parameters, ***Mτ***, ***D***, **Dispatching-scheme**, **and Measuring-*Type***, which are discussed in Section 3.1. For discussion purposes, an LLB algorithm is represented as BAC (***M*$_\tau$**, ***D***, ***Dispatching-scheme***, ***and Measuring-Type***). This function will be used in the next section to show the characteristic of each LLB algorithm. The measuring function as depicted in figure 1 of an algorithm uses a **Measuring-Type** to measure the traffic load at distance **D** for every ***Mτ***. The path-selecting function as depicted in figure 1 of an algorithm assigns **Dispatching-scheme** to a link.

3.2 Characteristics of load-balanced algorithms

The LLB algorithm using Least Loading First (LLF) is called Least Connection First (LCF) [5][6]. RR and LCF can be expressed by BAC (PS_c, 0, Best, counting session number), where PS_c is the measuring period for counting the number of connections. The measuring distance is $H(B_{s0})= 0$, and the dispatching scheme is 'Best'. LCF differs from RR in that LCF measures bandwidth by the number of incomplete connections over a BL, since these incomplete connections still consume bandwidth over the BL.

The path-selecting function of RR and LCF can be expressed by Equation (4) to set each $W_i=1$, which indicates that the CSN-type algorithms have a built-in weight property and can be free from "self-congested condition". This built-in weight property is also realized if the measuring result of connection numbers is different for each consecutive PS_c and the condition in Equation (3) does not hold.

LTF has been revised and extended to two variant algorithms: Maximum Inbound/Outbound Remaining Bandwidth First (MIRBF, MORBF) and Weighted Maximum (Inbound/Outbound) Remaining Bandwidth First (WMIRBF/WMORBF). The modifications to LTF are based on two considerations: (1) direction, due to the speed asymmetry of physical media, and (2) availability of bandwidth, where a line of least traffic does not ensure its timely availability.

The measuring function of these algorithms calculates the remaining (available) bandwidth in one direction over a BL, which can be expressed in Equation (6):

$$AB_i(t_r\ ,\ d)\ =\ MAX(\ FB_i^d\ ,\ AB_i(t_l\ ,\ d)\ +\ FB_i^d\ \times\ (t_r\text{-}t_l)\text{-}CL(t_r,t_l))\,, \qquad (6)$$

where t_r is the recent observing time; t_l is the last observing time; d is the direction which may be outbound or inbound; FB^d_i is the physical bandwidth of BLi at the last mile in direction d; and CL is the traffic load from t_l to t_r in direction d.

The BAC of MIRBF and MORBF is given by BAC (T, 1, Best, subtracting traffic), where the measuring period is set to a fix timer T that is applied in Equation (8), and the measuring distance is given by $H(\{B_{so}, h_i 1\}) = 1$.

WMORBF and WMIRBF are revised versions of MIRBF and MORBF using the weighted dispatching scheme which can be expressed by BAC (T, 1, Weight, subtracting traffic).

PMR has been extended to three algorithms: Fastest Round Trip Time to each Destination First (FRRTDF), Fastest Round Trip Time to a Fixed Node First (FRRTFNF), and Weighted Fastest Round Rrip Time to a Fixed Node First (WFRRTFNF).

FRRTDF and FRRTNF differ by their measuring distance and measuring period. FRRTDF uses $H(\mathbf{S}_{id})$ as the measuring distance, where for a given destination FRRTDF utilizes the connection setup phase of the TCP/IP applications to produce n raced requests over n BLs. The fastest responsive path to the destination is used for the following packets in the connection. Thus, the measuring period equals PS_c. FRRTNF uses $H(\mathbf{P}_{id})$ for the measuring distance. The FRRTFNF measuring method probes a fixed node repeatedly for every BL during each measuring period, ΔT_{fm}. The BACs of FRRTDF and FRRTNF are BAC(PS_c, $H_n(\mathbf{S}_{id})$, Best, measuring response time) and BAC(ΔT_{fm}, $H_n(\mathbf{P}_{id})$, Best, measuring response time), respectively.

WFRRTFNF is a variant of FRRTNF using a weighted dispatching scheme and is expressed as BAC (ΔT_{fm}, $H_n(\mathbf{P}_{id})$, Weight, measuring response time). The FRRTDF algorithm cannot be easily altered to implement a weighted version. FRRTDF only waits for one replied packet of the fastest responsive BL to reduce the measuring time conflicting with Equation (4) which needs the measurement results of all BLs. Table 1 summarizes the BACs of all LLB algorithms.

Table 1 A taxonomy of load-balanced algorithm

Measuring period(Mτ) = { path-selecting period PSc, a fixed timer T, back-to-back ΔT_{fm} }

Algorithm	**Mτ**	**D**	**Dispatching Scheme**	**Measuring Type**
RR/LCF	PS_c	0	weight	Counting Session Number (CSN)
MORBF/ MIRBF	T	1	best	Subtracting Traffic (ST)
WMORBF/ WMIRBF	T	1	weight	Subtracting Traffic (ST)
FRRTDF	PS_c	$H_n(\mathbf{S}_{id})$	best	Measuring Response Time (MRT)
FRRTFNF	ΔT_{fm}	$H_n(\mathbf{P}_{id})$	best	Measuring Response Time (MRT)
WFRRTFNF	ΔT_{fm}	$H_n(\mathbf{P}_{id})$	weight	Measuring Response Time (MRT)

3.3 Performance indicator

To evaluate the performance of LLB algorithms, several connections aere transmitted during a testing period and evaluated from their mean throughput and bandwidth utilization. The mean throughput is derived from the average throughput. The bandwidth utilization is discussed in detail in the following paragraphs.

Appling utilization law in the queuing network gives:

$$U = X \times S, \tag{7}$$

where X is the total throughput and S is the service time for transferring a unit. When *n* connections are transferred during the period T_c in the load-balanced system with *k* links, the total bandwidth required by the tested connections is given by:

$$\sum_{i=1}^{n} bandwidth\_required(Ci), \tag{8}$$

where Ci denotes the i^{th} connection.

The total service rate provided by this multihoming load-balanced system is given by:

$$\sum_{j=1}^{k} bandwidth(Lj), \tag{9}$$

where L_j denotes the jth link. The analogy in the LLB system to (7) is thus:

$$X=\sum_{i=1}^{n} bandwidth\_required(C\,i) / T_c$$

$$S=1/\sum_{j=1}^{k} bandwidth(Lj).$$

Equation (7) can thus be expressed as:

$$U = \sum_{i=1}^{n} bandwidth\_required(C\,i) \Big/ \left(\sum_{j=1}^{k} bandwidth(Lj) \times T_c\right). \tag{10}$$

Notably, this equation measures the parameter $\sum_{i=1}^{n} \text{bandwidth\_required(C i)}$ from the application sight. The size of the file transferred, rather than the traffic count from the underlying link, is used to calculate the required bandwidth. The underlying link would have many retransmissions of traffic. The term Effective Bandwidth Utilization (EBU) is used to represent network utilization in the LLB system disregarding retransmission. EBU is easy to calculate, since each value of Equation (10) can be measured externally from the LLB.

3.4 Performance analysis

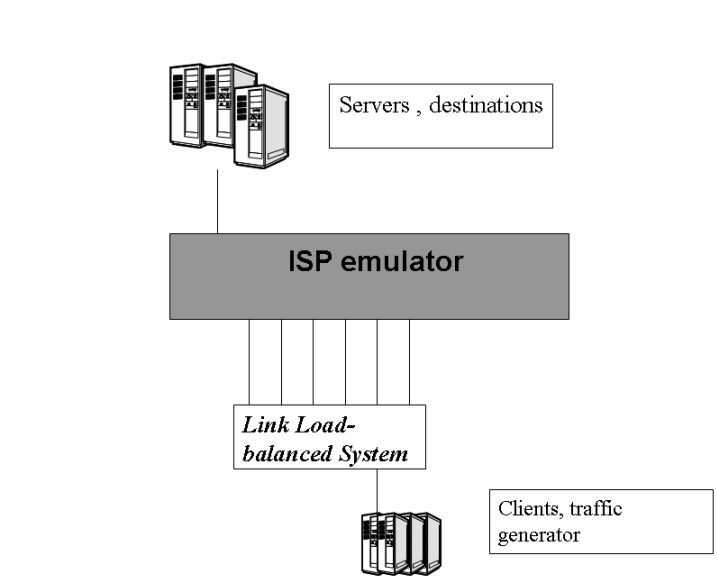


Figure 4 Emulation environment

Figure 4 is our emulation environment. The ISP emulator can create different ISP links with arbitrary download/upload speeds. The clients can generate http download requests to the servers with distinct source-IP addresses as different users.

Let UN be the number of users and SN be the number of concurrent sessions for a user. A notation L_n (UN , SN) is used to denote the combination of the links and the workload. The subscript n in Ln is the number of ISP links. We also assume that total bandwidth requires being constant at a T1 data rate of 1536k. For example, L_1 would be one line with a data rate of 1536k, whereas L_3 indicates 3 lines of 512k.

Each user in a workload (UN , SN) has a repeated downloading procedure for 10~20 rounds. In each round, there are SN http sessions to download files simultaneously.

The repeated procedure is like a user who is browsing a web page including many objects. When all the objects have been downloaded, the user browses another web page. This connection-depending traffic model has been discussed in Seldmann [21].

Many studies have discussed the burstness of TCP connection request arrivals [21][22][23][24][25]. Therefore, our workload uses different degrees of burstness to show the performance of the system. By generating different workloads of traffic, the throughput and utilization can be evaluated.

3.4.1 Aggregation analysis of load-balancing parameters

Let *LM* be a link with the bandwidth which can take the sum of many narrower link ℓi 's bandwidth. The objective function of bandwidth aggregation can be represented as (11).

$$
\begin{aligned}
&Max\left(\sum_i Throughput\left(li\right) - Throughput\left(LM\right)\right)\\
&Min\left(\prod_i FailRate\left(li\right)\right)\\
&Min\left(\sum_i Cost\left(li\right)\right)\\
&subject\ to\\
&\quad \prod_i FailRate\left(li\right) < FailRate\left(LM\right)\\
&\quad \sum_i Cost\left(li\right) < Cost\left(LM\right)\\
&\quad li\ is\ an\ existing\ choice\ in\ the\ market
\end{aligned}
\qquad (11)
$$

The goal is to use the most economical and reliable combination of existing links ℓ to realize the bandwidth demand of *LM*. Different countries might have various combinations to meet this cost and bandwidth requirements since ℓ could be 256kbps in England while 128kbps in the U.S. However, our focus is not to find the combinations, but instead we want to make sure that it is possible to gain the throughput (*LM*) from a combination of many ℓ with a lower cost.

For practical reasons, we pick one of the available combinations from the market. By surveying many ISPs we find that using one T1 leased line and three lines of 512k/512 k ADSL links is a proper choice. (See table 2)

Table 2 T1 and 512k ADSL monthly fees for the three major ISPs in Taiwan:

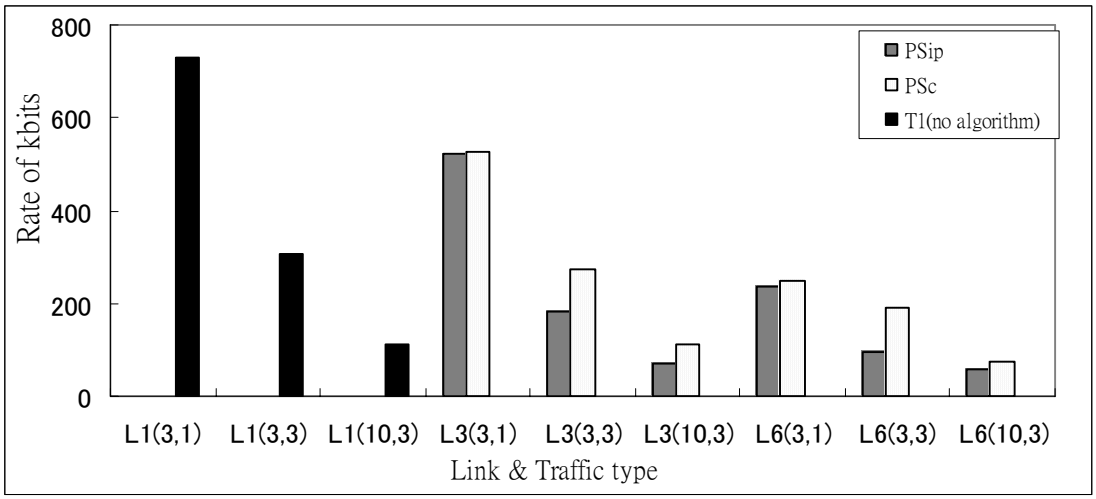
ISP Name	T1 monthly fee (NT$)	512k ADSL monthly fee (NT$)
CHT	54,600	3,700
SPARQ	145,000	3,700
TFN	144,000	3,700

Based on the discussion in section 3.1, in a connection-based system the LLB algorithm employs a **Measuring-Type** to process traffic load measurement at a distance **D** for every $M\tau$. The measurement results are evaluated in order to assign a BL by the **Dispatching-scheme.**

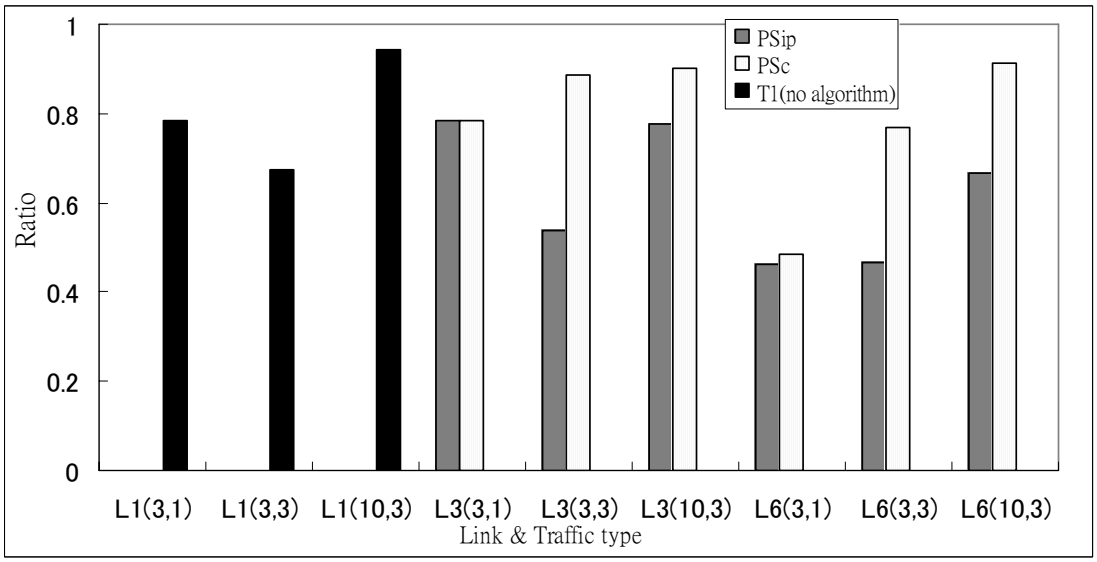
We can compare the performance and utilization of the generic parameters of the LLB algorithms. Moreover, the impact of link numbers on the bandwidth aggregation will be considered. Note that 6 lines of 256K are also included to see if the number of lines is sensitive to the experiment. The workload starts from Ln(3,1) to Ln(10,3). Ln(3,1) is about the capacity of the bandwidth provided by a T1 leased line. To focus don the bandwidth aggregation issue, this simulation environment only has traffic generated from the testing clients, and no disturbance of outward traffic is induced.

3.4.1.1 Comparison on path-selecting period

Figure 5 (a) shows that when the workload reaches L_3 (3, 3), the mean throughput of 3 aggregated links is getting closer to one T1 link. Note that in Figure 5 (a), the mean throughput of an IP-based dispatching period (PS_{ip}) in every workload is



(a) Throughput



(b) EBU

Figure 5 (a) Comparing mean throughput of path selection periods: connection-based (PS_c) and IP-based (PS_{ip}) (b) EBU of the two path selection periods

worse than the connection-based dispatching period (PSc). Moreover, due to flow control mechanism to be explained later in this section, the throughput of 3 balanced links (L3's) is always better than L6's.

In Figure 5(b) the EBU values of PS_c are all higher than those of PS_{ip}. Note that EBUs in Figure 5(b) in fact correlate to the mean throughput in Figure 5(a). An interesting situation is that the EBU value of L_1 (3,1) is higher than L_1(3,3). To compute the EBU value of L_1(3,1) and L_1(3,3) by using Equation (10), the total

service rate $\sum_{j=1}^{k} bandwidth(\ Lj)$, the total bandwidth required $\sum_{i=1}^{n} bandwidth\_\ required(C\,i)$, and the testing complete time (T_c) are listed in Table 3 respectively.

Table 3 Computation of EBU of two workloads

Workload	$\sum_{j=1}^{k}$ bandwidth(Lj)	$\sum_{i=1}^{n}$ bandwidth_required(Ci)	T_c (sec)
$L_1(3,1)$	1.536(Mbps)	150k bytes	9.8
$L_1(3,3)$	1.536(Mbps)	450k bytes	34.1

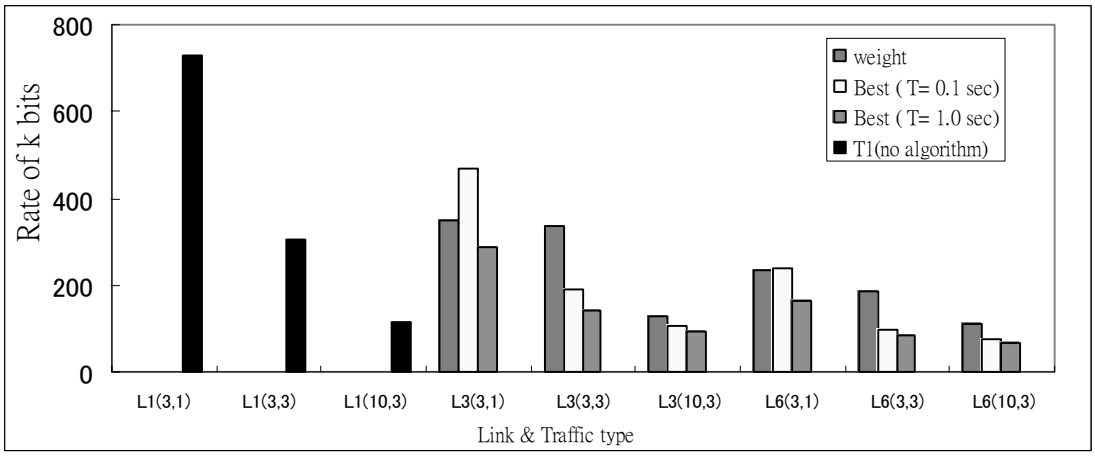
In the above EBU calculation of $L_1(3,1)$ and $L_1(3,3)$, T_c is an important factor, as it is the time needed to complete the data transfer and depends on flow control behavior. When the arrival rate of traffic is higher than the service rate of the link, there will be packet losses and retransmissions. TCP flow control will start to lower down the end-to-end transmitting speed and make T_c longer.

Like the mean throughput in Figure 5(a), Figure 5(b) also shows that the EBU of 3 balanced links is better than their counterpart of 6 links. The average measurement of T_c of L6(UN,SN) is longer than L3(UN,SN), which indicates that the flow control will have more impact on the narrower bandwidth.

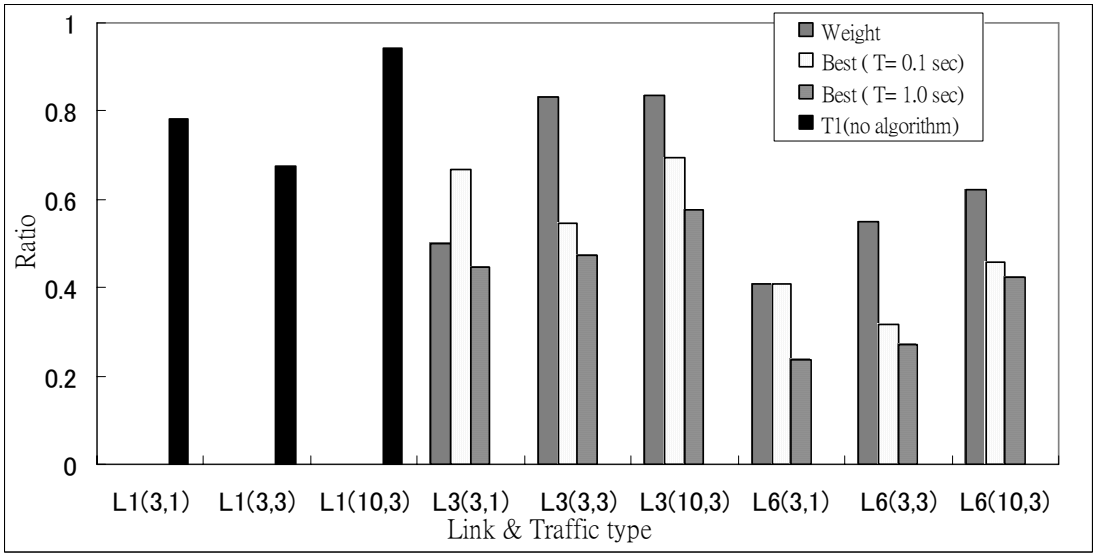
3.4.1.2 Comparison of dispatching scheme and measuring period

Figure 6(a) compares the mean throughput of the weighed dispatching scheme versus the best dispatching scheme with different measuring times, T = 1 and T= 0.1 seconds. When the workload is small, the best dispatching scheme (T= 0.1 sec) gets the best result. When the workloads are heavier, $L_n(3,3)$ and $L_n(10,3)$, the weighted

dispatching scheme outperforms the ‘best’ and its throughput approaches the T1 rate .



(a) Throughput



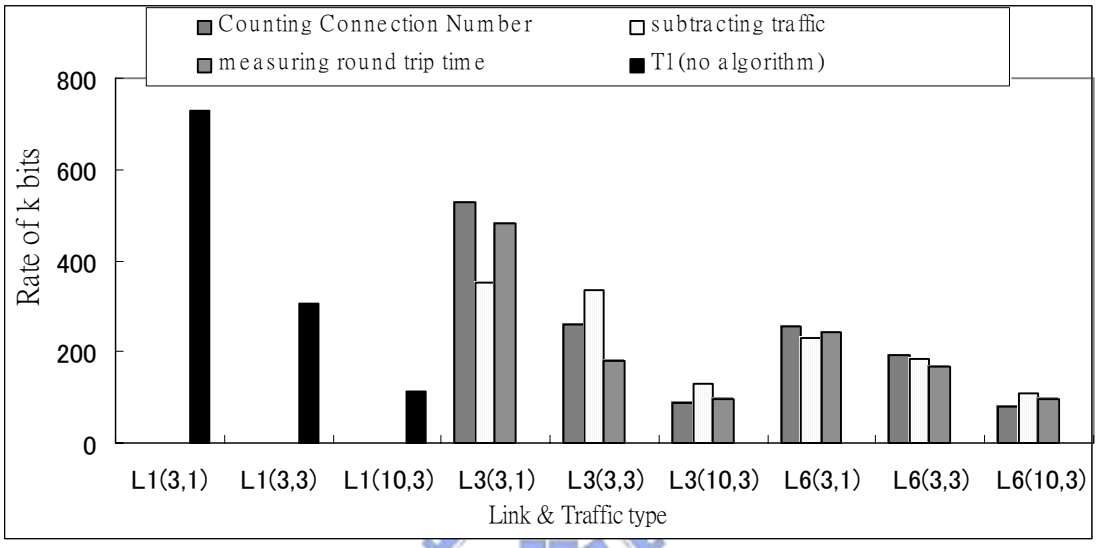
(b) EBU

Figure 6 (a) Comparing mean throughput of dispatching scheme and measuring time T (b) EBU of dispatching schemes and measuring times

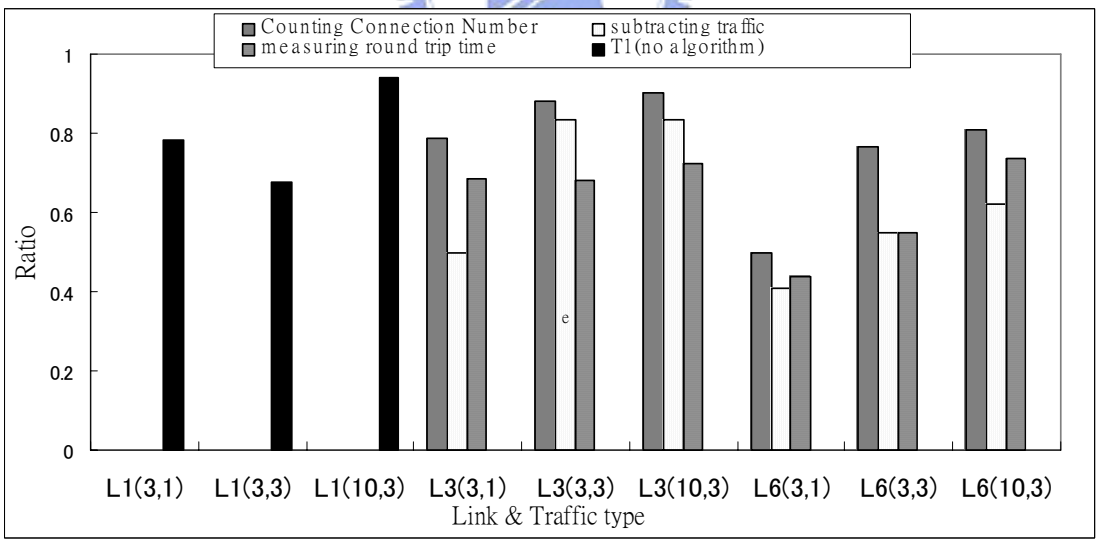
The weighted approach achieves good throughput performance only when the number of sessions is high enough so that their load can be distributed. In the comparison of measuring time T, (T= 1 sec) is always worse than (T=0.1 sec), as this is a direct consequence of a smaller $\Delta Tj_{ps\text{-}m}$ (see Equation (4)) since the measurement

error can be reduced accordingly. Figure 6(b) shows that utilization correlates to mean throughput in Figure 6(a).

3.4.1.3 Comparison of measuring type and measuring distance



(a) Throughput



(b) EBU

Figure 7 (a) Comparing mean throughput of measuring types and distance (b) EBU of the three measuring types

Figure 7(a) shows the results of various measuring types and the corresponding measuring distance as discussed in section 3.1.5. When the workload is small, the mean throughput is higher in the CSN and MRT types of algorithms. When the workload is heavier, the ST-type algorithm gets the higher throughput. However, there are no discriminations between good and bad L6 situations.

Figure 7(b) shows that the CSN type gets the highest EBU. To discuss this phenomenon, we should understand more about the operations of CSN-type algorithms. CSN-type algorithms use the counted number of sessions of each BL_i to determine the path for transmissions. The path assignment operation of the CSN-type algorithm can be expressed by Equation (6) by setting each W_i=1. This usage of the weight utility function indicates that the CSN-type algorithms have a built-in weight property. This built-in weight property enforces that the CSN-type algorithms have the weighted dispatching behavior which can gain a higher utilization of bandwidth. Thus, the result of Figure 7(b) of a higher EBU value with the CSN-type algorithm is due to the weighted dispatching property.

3.4.1.4 Summary of experiment results

To summarize up, the important factors that influence the bandwidth aggregation performance are the path-selecting period, the measuring period, and the dispatching scheme. Either the measuring type or the measuring distance does not have a significant impact on bandwidth aggregation. In the case of narrower bandwidth links to be aggregated to a pipe (like with L6's situation), the flow control would affect the total throughput and utilization.

3.4.2 Congestion analysis of load-balancing algorithms

The reaction of each algorithm to a congested link is described now. A congested BL in the multihoming load-balanced system affects the throughput and

bandwidth utilization. In this study a congested link is created by narrowing the available uploaded link bandwidth. The uploaded bandwidth consequently affects the TCP's acknowledgement of the HTTP's downloading. In this experiment one of the BLs(BL_2) in the multihoming network is made to be congested, the available bandwidth is decreased to 5k bits/sec, and the response time is increased to about 2 seconds. The influence of congestion to various LLB algorithms can hence be compared.

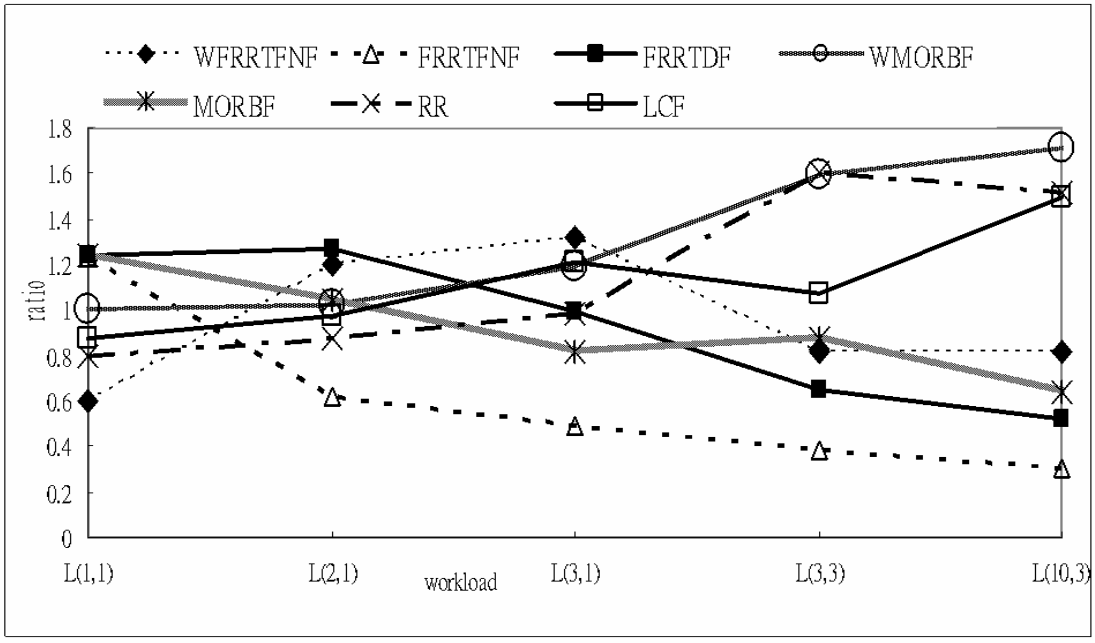
Since only the relative performance of the LB algorithms is important, the throughput values generated from all algorithms are normalized, with the average throughput set to 1.

3.4.2.1 Comparison of local congestion (last mile congestion)

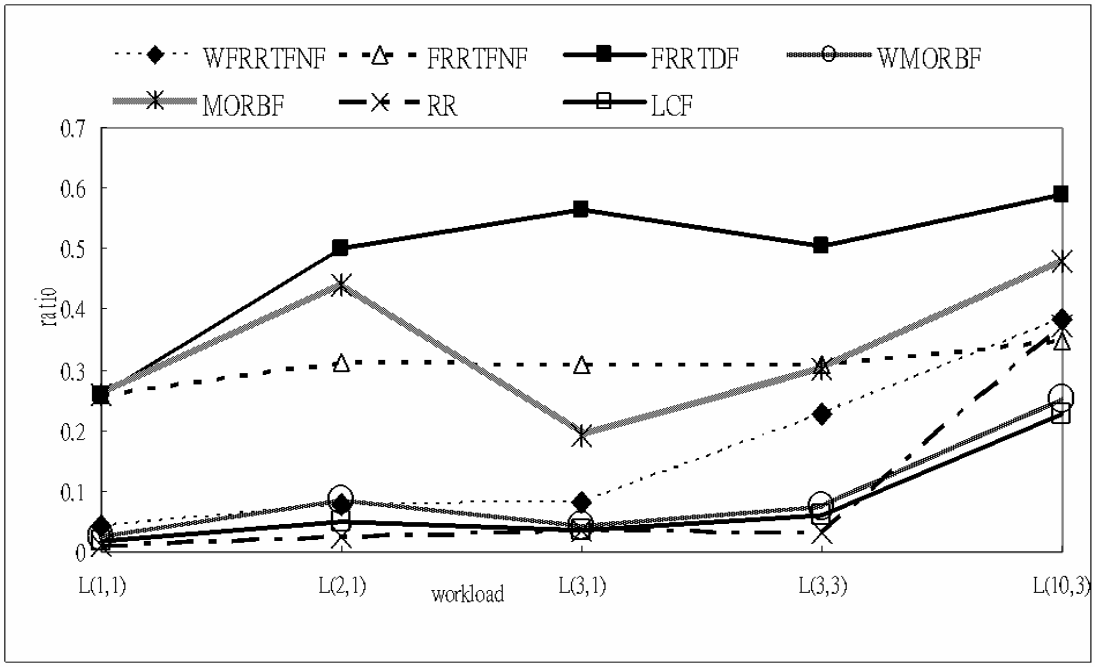
Figure 8(a) compares the mean throughput of each algorithm where the bottleneck is at the last mile. Increasing the workload leads to raised and decayed lines. With the smallest workload value L(1,1), the FRRTDF, FRRTFNF, and MORBF algorithms - which use the "best" dispatching scheme to assign the path - performed best. FRRTDF and FRRTFNF use the response time to determine the best path, and MORBF uses the calculated available bandwidth. The weighted algorithms (WMORBF and WFRRTFNF) and the CSN-type algorithms (RR and LCF) performed worse at this workload.

When the workload is increased to near the link capacity (L(3,1)), WMORBF and WFRRTFNF, the algorithms using both traffic measurement and weighted dispatching are found to have the highest throughput. The "weighted" algorithms improved when the workload increased, and the "best" algorithms deteriorated in performance.

When the workload increased to the highest value L3(10,3), the "weighted" algorithms WMORBF, RR, and LCF have the highest throughput, with WFRRTFNF's throughput slightly lower.



(a) Throughput



(b) EBU

Figure 8 (a) Comparison of mean throughput of last mile congestion (b) EBU of last mile congestion

The weight adjustment duration for WFRRTFNF is longer than that of WMORBF. WFRRTFNF had to wait for a response time for all the BLs. The algorithms using the "best" dispatching scheme, MORBF and FRRTFNF, have the worst throughput.

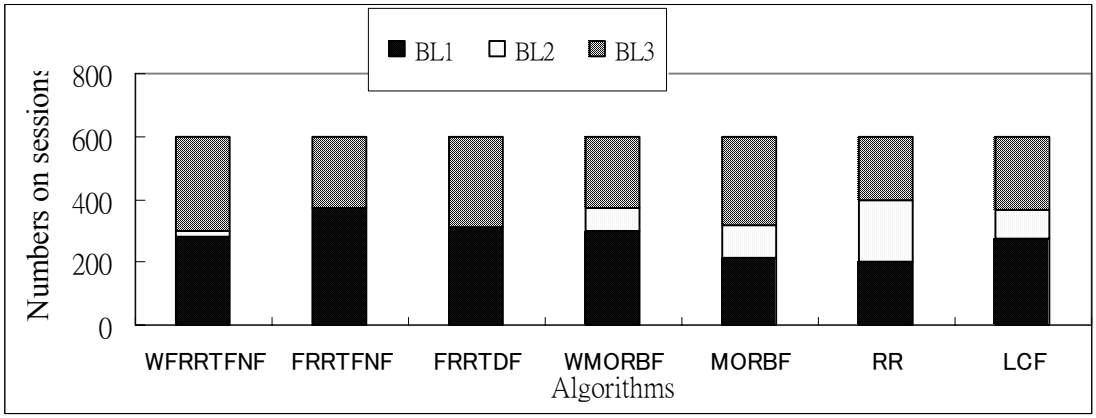
When the workload increases to near the link capacity (L(3,1)), WMORBF and WFRRTFNF, the algorithms using both traffic measurement and weighted dispatching are found to have the highest throughput. The "weighted" algorithms improve when the workload increases, and the "best" algorithms deteriorate in performance.

When the workload grows to the highest value L3(10,3), the "weighted" algorithms WMORBF, RR, and LCF have the highest throughput, with WFRRTFNF's throughput slightly lower. The weight adjustment duration for WFRRTFNF is longer than that of WMORBF. WFRRTFNF has to wait for a response time for all the BLs. The algorithms using the "best" dispatching scheme, MORBF and FRRTFNF, have the worst throughput.

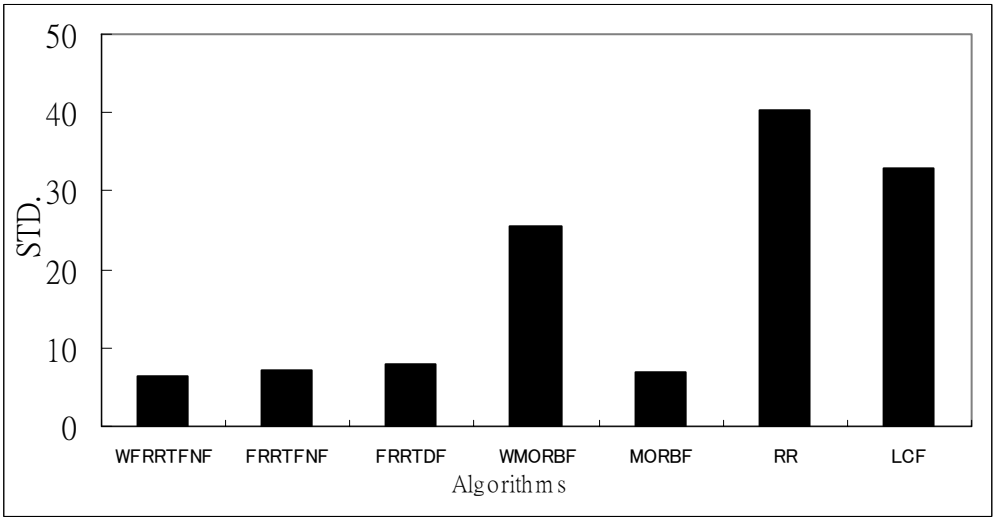
The mean throughput is higher in weighted algorithms which dispatch sessions to the congested path, because of the limited total capacity. Under heavy traffic loading, when the algorithms using the "best" dispatching scheme only dispatch traffic to the non-congested path, the non-congested path becomes overloaded. Therefore, capacity may be increased by using congested paths as in the weighted algorithms. However, delay and throughput represent a trade-off.

The sessions dispatched to congested paths are significantly delayed, inducing a higher variability. Figure 9(a) shows that the number of sessions dispatched to each BLi by MRT-type algorithms, such as FRRTFN and FRRTDF, avoid congested paths, but other "weighted" algorithms dispatch traffic to congested paths. Figure 9(b) shows

the mean absolute deviation of different users' throughput of each algorithm, revealing a larger variability correlated to Figure 9(a), if sessions are dispatched to the congested path.



(a) Number of connections

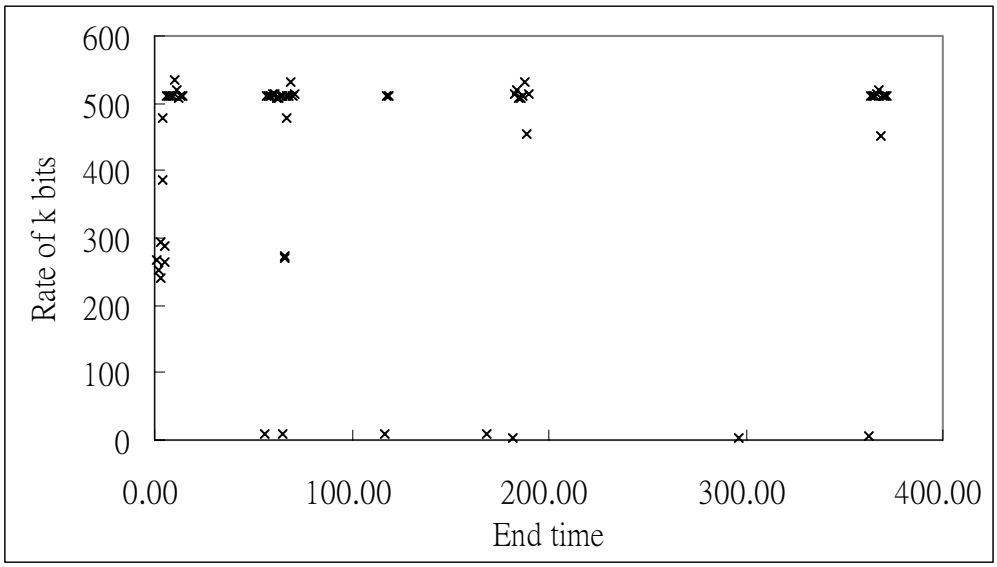


(b)Mean absolute deviation

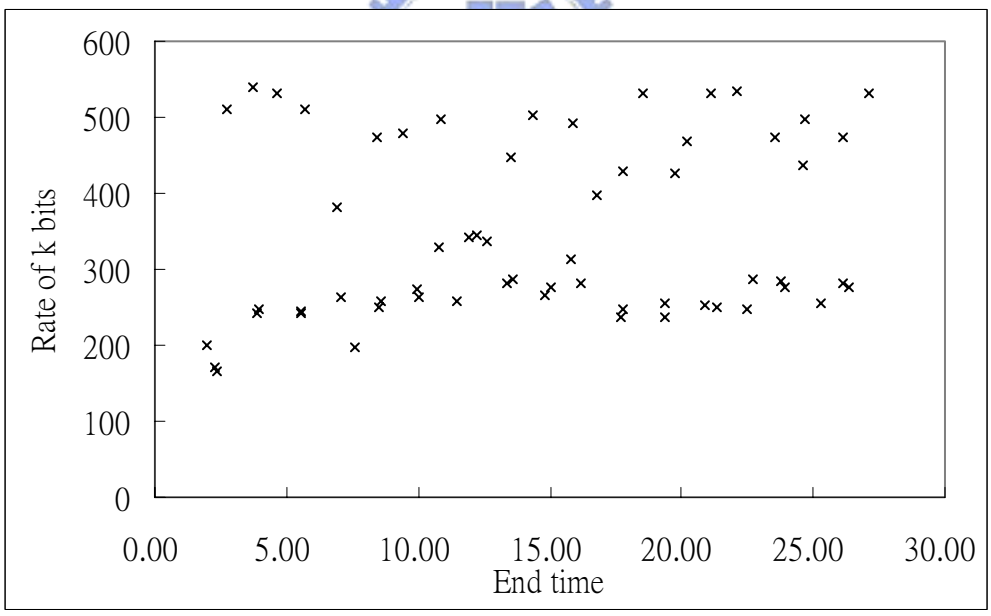
Figure 9 (a) Number of connections on each link (b) Mean absolute deviation of different users' throughput

Figure 8(b) shows the EBU of different algorithms. The EBU of weighted algorithms is found to be lower than the algorithms with the "best" dispatching scheme at every workload. This lower EBU situation of weighted algorithms is caused by the total transmission time, T_c in Equation (10). Equation (10) shows that

under the same total service rate $\sum_{j=1}^{k} \text{bandwidth}(Lj)$ and total bandwidth required $\sum_{i=1}^{n} \text{bandwidth\_required}(Ci)$, T_c dominates the EBU calculation.



(a) WMORBF



(b) FRRTDF

Figure 10 Per session finishing time and throughput in (a) WMORBF (b) FRRTDF

The time needed to complete the data transfer, given by T_c, depends on the flow control behavior. When the traffic arrival rate is higher than the service rate of the link,

packets are lost and retransmitted. TCP flow control thus starts to reduce the end-to-end transmitting speed, lengthening T_c.

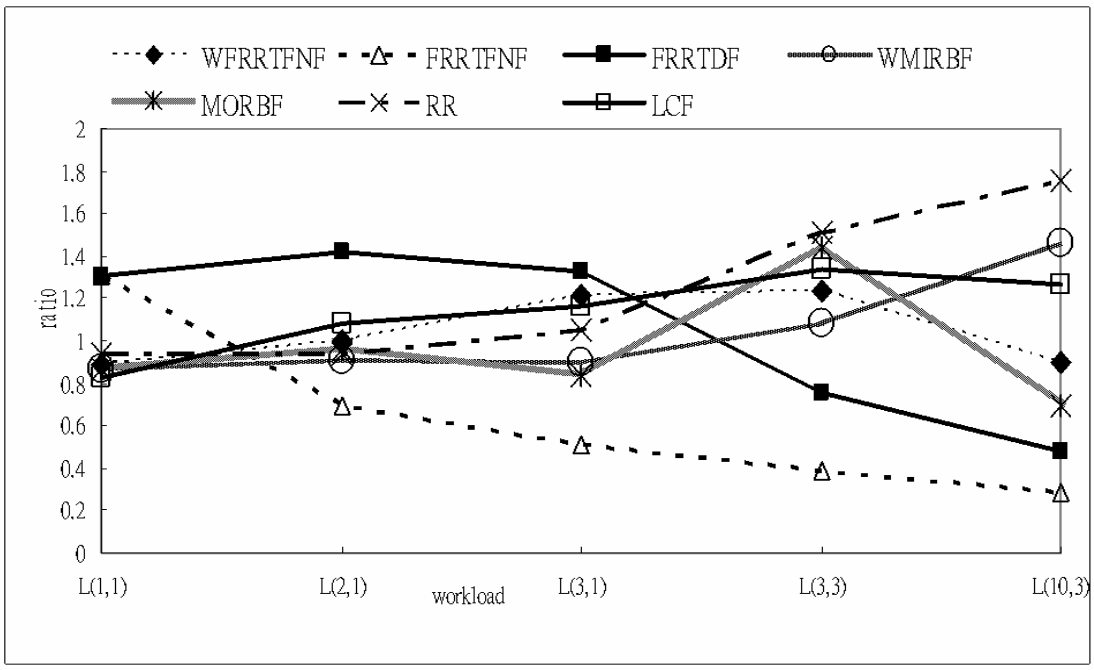
Figure 10(a) illustrates the finished times of every session and their throughput using WMORBF at workload L(10,3). Figure 10(b) shows FRRTDF in the same situation. The diagram shows that the transmission time of some sessions in WMORBF is very long, lowering the EBU.

3.4.2.2 Comparison of remote congestion

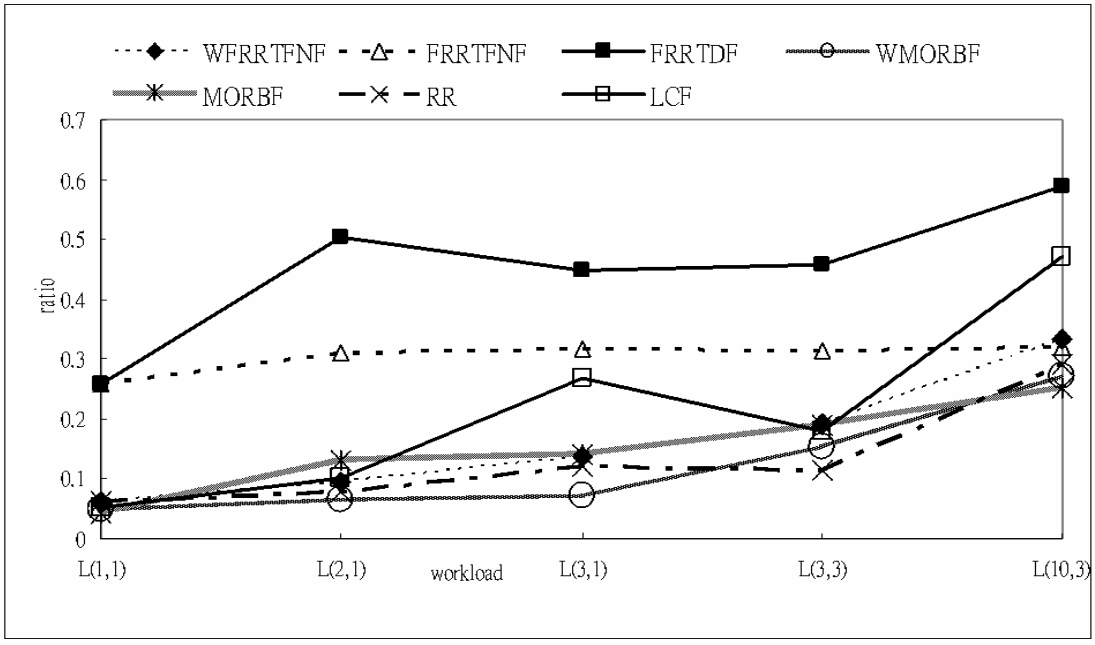
Figure 11(a) compares the reaction of these algorithms against congestion at a remote distance. The traffic congested condition is set at the second hop of BL_2 and is detected by all the MRT-type algorithms, but not by the ST-type algorithms.

The situation of the MRT-type and CSN-type algorithms is similar to that shown in Figure 8(a), where congestion occurs at the last mile. Using ST-type algorithms such as WMORBF and MORBF, at a light workload (L(1,1), L(2,1)) or a workload near the bandwidth capacity (L(3,1), the comparable throughput falls far behind that of CSN-type algorithms. This phenomenon occurs, because when traffic is dispatched to the congestion path in a remote area, flow control reduces the traffic, increasing the capacity at the last mile. If traffic is dispatched along the path with the largest available bandwidth at the last mile, the algorithm treats that path as usable and sends more sessions there. As shown in Figure 9, when dispatching traffic at workload L(3,1), more sessions are dispatched to the congested path using ST-type algorithms such as WMORBF and MORBF than when using CSN-type algorithms .

Figure 11(b) shows almost all the graph plots are similar to those in Figure 8(b) when congestion occurs at the last mile, except for MORBF. The bottleneck could not be measured for MORBF, which has a low EBU.



(a) Throughput



(b) EBU

Figure 11 (a) Mean throughput of remote congestion (b) EBU of remote congestion

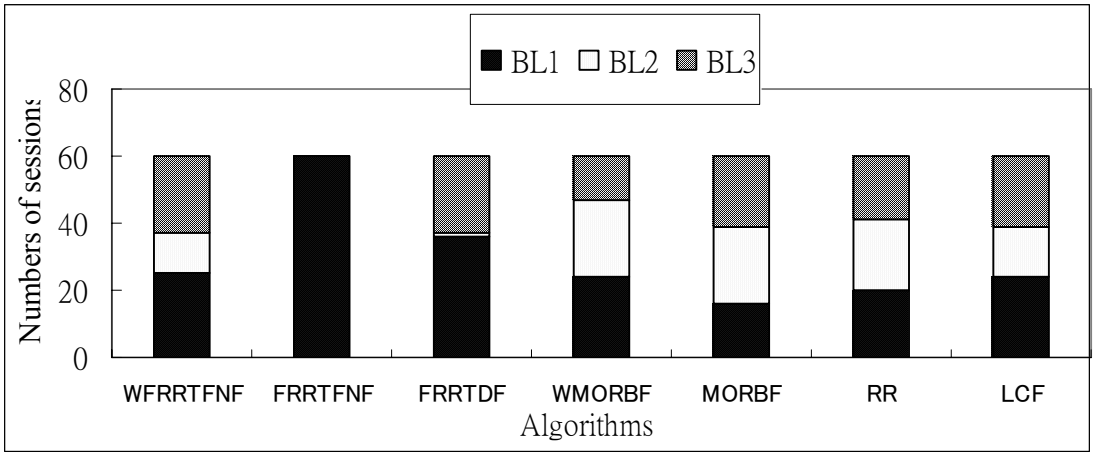


Figure 12 Number of connections on each link at workload L(3,1)

3.4.2.3 Discussion of experiment results

The experimental results show that during light traffic, the mean throughput is entirely dependent on the measurement function, but during heavy traffic the weighted dispatching scheme significantly influences the mean throughput more than the measurement function does. The measurement can minimize the variability among users and increase the EBU. Algorithms which can detect bottlenecks and which use weighted dispatching can maintain the mean throughput value at both light and heavy workloads, as shown in figures 8(a) and 11(a). The graph produced by WFRRTNF is close to the graph of the average of all the algorithms. Additionally, when the location of the congesting path does not fall on the measurable scope, the overall performance drops.

Chapter 4 Enhanced load-balancing algorithms for end-to-end traffic condition

Section 4.1 discusses the issues of timely end-to-end measuring and dispatching schemes. Section 4.2 provides the WSDM solution. Section 4.3 compares the operations of various measurement approaches.

4.1 Issues

4.1.1 Timely measurement

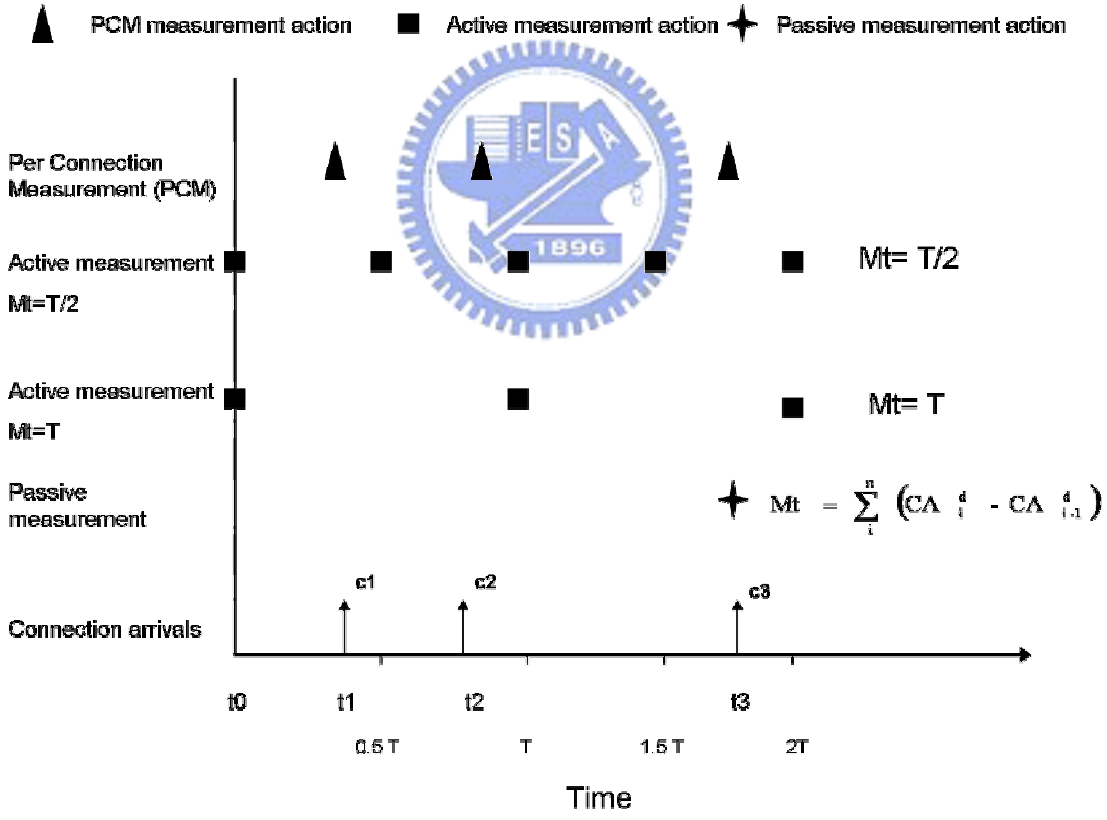


Figure 13 Time slots of connection arrivals and end-to-end measurement actions

Akella's measurement scheme mentioned in the above section is not designed for per-connection timely measurement**.** A gap occurs between the connection arrival

time and the measurement time. A larger gap implies more difficulty for the routing path selection process to reflect the real network traffic situation.

Figure 13 depicts the time slots of connection arrivals and end-to-end measurement actions. There are three connections C1, C2, and C3 that arrive at times t1, t2, and t3, respectively, to the same destination. Mt is the measuring period of the end-to-end measurement for a destination.

Using the passive measuring method, Mt is dominated by the inter-arrival time of connections to the same destination which is not determinable. With N Internet links, the algorithm has to wait N connections to the same destination in order to obtain traffic conditions for all of the links. Let CA_i^d denote the connection arrival time of the ith connection to destination d, and Mt for passive measurement is given by:

$$Mt = \sum_{i}^{n} \left(CA_i^d - CA_{i-1}^d \right). \qquad (12)$$

When using the active measurement, Mt equals a fixed duration *T*. Therefore, time *T* can be controlled to minimize the measuring interval in order to provide more up-to-date traffic conditions. As shown in Figure 13, when using Mt= *T/2*, connection C2 has a fresher end-to-end measuring result (at 0.5 *T*) than when using Mt=*T*.

When the multihoming system continuously generates measurement packets in a shorter duration, service requirements constantly influence the destinations in the destination list maintained by active measurement. Moreover, the multihoming system has to handle many processes for each destination that sends and receives the measuring packets. A multihoming system which has 3 links and 300 destinations in the destination list always has to process 900 end-to-end measurements for every Mt. A smaller Mt could make the system be always under a busy situation.

The number of measurement actions and destinations must often be restricted to alleviate the burden on the destinations and the multihoming system resulting from measurement operations. This compromise of restriction could incur serious problems in a link failure condition. Link failure in a link to a destination causes transmission failure for all the connections transferred to that destination during two measurement actions. If that destination is not in the destination list, then the connection transferred to that destination, in the worst case, could always fail, leading to a starvation condition. One solution to this problem is to set:

Mt = every connection arrival.

The active measurement proceeds at every connection arrival. The ongoing connection would wait for the active measurement to give it the optimum path. This measurement scheme, called "Per-Connection Measuring (PCM)", ensures that an initial connection does not choose a failure link for its destination. The commercial products, mentioned in section 2, provide similar schemes: "proximity" [5] and "fastway" [7]. Both the "proximity" and "fastway" process end-to-end measurement for every connection arrival. As drawn in Figure 13, PCM processes end-to-end measurement at every connection arrival.

The PCM still gives the system a heavy loading at a busy connection arrival. In a busy station, the measurement is triggered as many times as the number of new arrivals. The multihoming equipment with three ISP links has 1000 connections arriving within one second which requires 3000 end-to-end measurements. This end-to-end measurement induces process burden and may delay subsequent connections. Obtaining a timely end-to-end traffic condition and eliminating the process burden of end-to-end measuring on both the multihoming system and the destinations represents a trade-off situation.

4.1.2 Dispatching scheme

After measuring the traffic load over ISP paths, the dispatch schemes are required to assign a path to a specific TCP/IP session. Aside from using a "best" dispatching scheme that dispatches each session to the path with the best measuring result (round trip time as in Akella's approach; available bandwidth is discussed in a later section), "weighted" is another scheme that can be chosen. The "best" scheme might cause a "self-congested condition" where all sessions are dispatched to the best path in a measuring period as discussed in section 3.1.3. The weighted dispatching method as described in Equation (4) and Equation (5) is applied to dispatch traffic.

4.2 Weighted Self-Detected Measurement (WSDM)

A Weighted Self-Detected Measurement (WSDM) approach is proposed to minimize the end-to-end measurement cost, achieve a fresh traffic condition, and gain higher bandwidth utilization. To minimize the gap between the connection arrival time and the measurement time, WSDM also resembles the PCM mentioned in 4.1 to proceed with measuring every connection arrival. To minimize the measuring cost, WSDM does not send measuring packets to obtain an end-to-end round trip time. WSDM manipulates the NAT and routing mechanism in the connection cache and utilizes the TCP connection setup phase to detect the end-to-end traffic condition. A higher bandwidth utilization of WSDM can be achieved by using the "weighted" dispatching approach. The NAT and routing mechanism in the connection cache are discussed in 4.2.1, and the WSDM algorithm is discussed in Section 4.2.2. The end-to-end measuring at the scoop of the TCP protocol is also limited as in Akella's [8] proposed scheme.

4.2.1 Connection cache and NAT mechanism

NAT equipment contains a cache to record every connection. This record can direct packets belonging to the same connection to maintain a consistent NAT address.

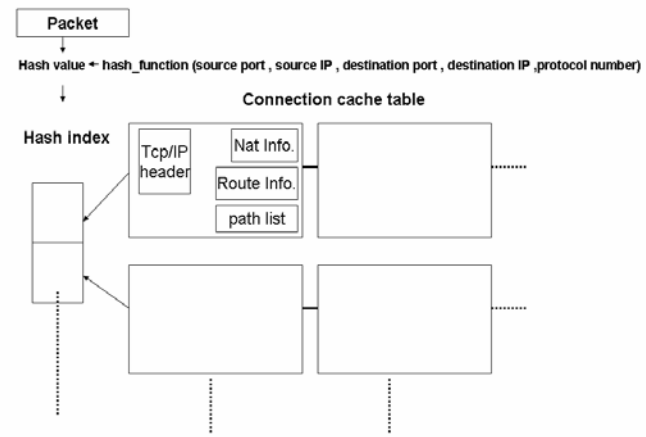


Figure 14 Illustration of connection cache mechanism

Figure 14 illustrates the connection mechanism. Every TCP/IP packet applies the value of TCP/IP header such as destination IP, destination Port, source IP, source port, and IP protocol number, through a hash_function in order to produce a hash value.

The first packet of a connection is used to construct a record in the connection cache table by its hash value. The hash_function may produce the same hash value from different TCP/IP headers, requiring that the implementation of the connection cache table apply a linked list array to place the records with the same hash value in different places.

The connection cache has three important data members: the NAT address, the routing information of the next hop, and a path list of selectable ISP links of this connection. By manipulating these data members using the WSDM algorithm provided in the later section, one connection obtains multiple chances to choose ISP links based on their end-to-end traffic conditions.

4.2.2 Algorithm

WSDM utilizes the TCP connection setup phase to perform path selection. For the first SYN packet at a TCP connection that is generated by client application, WSDM uses the available bandwidth during the last miles to specify the weight on each ISP link and transmits the connections by the ratio of each link. Therefore, the routing and NAT information are also written in the connection record.

When no ACK response is obtained from the remote site after a SYN timeout period in TCP protocol, the client's TCP protocol stack generates another SYN retransmission packet, which is a signal informing a bad traffic situation. At this time, the WSDM removes the NAT and routing information and marks the path used by the first SYN packet as the failure state on the path list in the connection record. A path is then chosen from the path list, which excludes the path with a failed mark for the retransmitted SYN packet.

The benefit for using the available bandwidth of the last mile as a path selection is its fast calculation. Accumulation activity only needs to be provided in the multihoming system. The calculation of the last-mile available bandwidth between two positions of time of an ISP link is provided in Equation (6). The utilization of TCP retransmissions saves the measuring packets from the multihoming system and also determines the end-to-end traffic condition.

Figure 14 displays a complete WSDM algorithm which deals with the first and a retransmission of the SYN packet of a connection differently in the WSDM procedure. Here, C_i is a connection record as shown in Figure 13. The Select_Path procedure selects the path by using the Select_AB_Weight procedure, and NAT and routing information are written in the connection record. The Selecting_AB_Weight is used to process Equations (4), (5), and (6), in order to provide a candidate path.

```
< WSDM Algorithm>
WSDM  procedure
Accept(Ci ->SYN[k]) /* receive kth SYN packet in
                          connection i  , where Ci is a
                    connection record as shown in Fig.1*/
If(k is equal to 1)      /* the first packet */
  Select_Path(Ci)
Else if (k is greater than 1)  /* receive retransmission*/
  Remove_route_info(Ci-> routeinfo)
  Remove_NAT_info(Ci-> natcache)
  Mark_failed_path (Ci-> pathlist, Ci->lastpath)
  Select_Path(Ci)

Select_Path   procedue
Pathj = Select_AB_Weight(Ci-> pathlist)/* select the path
By Eq. (6), (7) , (8)*/
  Set_route_info(Ci-> routecache, Pathj)
  Set_NAT_info(Ci-> natcache, Pathj)
  Send(Ci ->SYN[k], Pathj)       /*send syn packet to
                                the chosen path */
  Store_last_path(Ci->lastpath , Pathj)
                                /* store the latest
                                 chosen path */

Select_AB_Weight procedure
  LastMax = Tmpmax= 0
  For each path i
    TmpMax = ABi(tr,d) /* use Eq.(8) for calculation */
      If (Tmpmax is greater than LastMax)
         LastMax = Tmpmax
  For each path i
     Wi = Caculate_weight( ABi(tr,d),LastMax)
          /* use (7) to calculate weight for each path*/

  Sel_path = Min_calulate(Wi on each path)
          /*use (6) to determine path */

 Return Sel_path
```

Figure 15 WSDM algorithm

4.3 Comparison of operations

Table 4 Comparison of operations

	WSDM	PCM	Active	Passive
End-to-end detecting method	Use retransmission	Use duplicated connection	Use extra measuring connection	Use ongoing connection
Dispatching scheme	Weight	Best	Best	Best
Measuring timing	connection arrival	connection arrival	fix duration T	multiple connection arrivals
Measuring for every destination?	Yes	No	No	No

Table 5 Comparison of resource usages

	WSDM	PCM	Active	Passive
Need TCP/IP socket handler?	No need	need	need	No need
Need destination list?	No need	No need	need	need
Need extra bandwidth usage?	No need	need	need	No need
Need destination reply?	No need	need	need	need

Table 4 compares the operations of WSDM, active measuring, passive measuring, and PCM schemes mentioned in the last section. Here, WSDM can perform a measurement to every destination at every arrival connection. WSDM uses

the TCP retransmission to detect end-to-end traffic conditions and uses the weighted dispatching approach to dispatch traffic.

Table 5 compares various resource requirements of different algorithms. By comparison, WSDM does not need to handle extra TCP/IP sockets for measuring, which not only reduces the measuring cost of an end-to-end measuring system, but also reduces the burden to the link and the end host for not receiving extra measuring packets. WSDM does not need a destination list, either.

4.4 Emulation results

The performance of WSDM is evaluated by comparing the throughput and failover rate to PCM. The PCM has a higher sensitivity of traffic condition as depicted in Table 4. The PCM is provided by Deansoft's [7] multihoming equipment which performs end-to-end measurements at every connection arrival. WSDM is implemented using the same hardware (VIA CPU) and OS kernel (Linux).

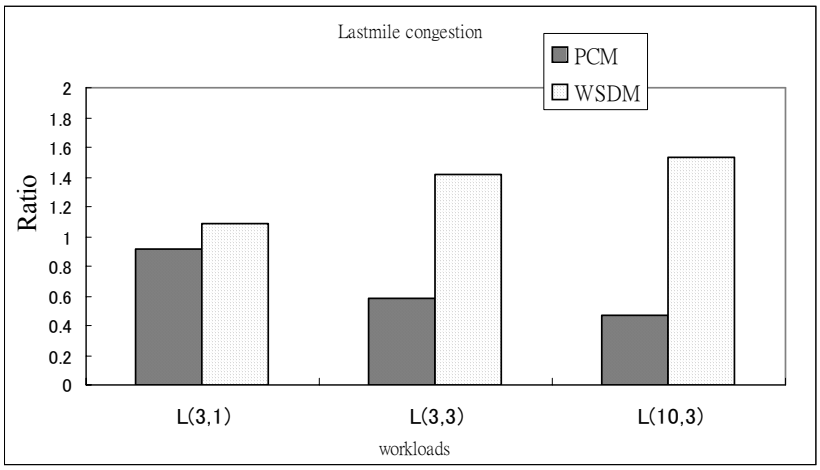
Two traffic conditions are introduced: congestion and outage. They determine the impact towards measurement operations.

4.4.1 Congestion

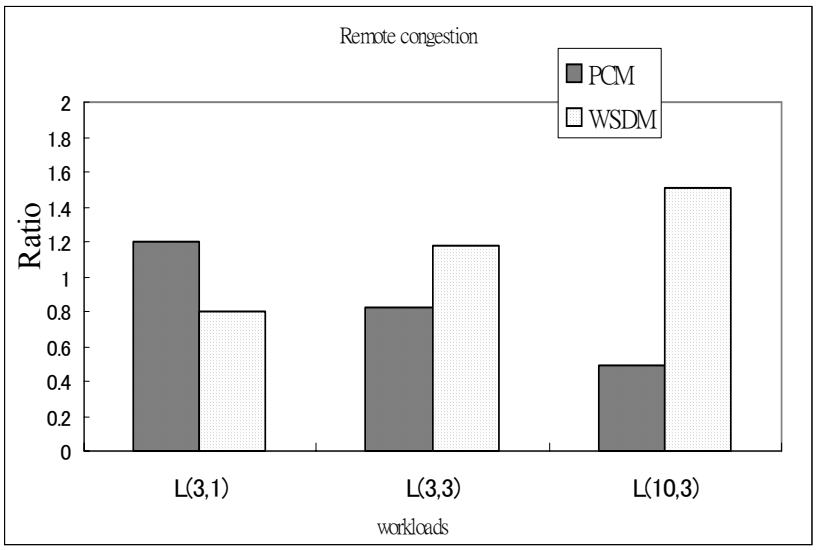
Various workloads are created to determine the effect of dispatching schemes and measurement methods used by WSDM and PCM. The emulation environment of the congestion situation is described as in section 3.5.

Figure 16(a) illustrates the congestion during the last miles, indicating that WSDM provides a better throughput than PCM on each workload, while heavier workloads (L(3,3), L(10,3)) of WSDM have more advantages. WSDM can quickly detect traffic conditions of the last miles as mentioned in section 3.2. Moreover, the

weighed dispatching approach of WSDM can provide better bandwidth utilization than the best approach.



(a) last-mile congestion



(b)Remote congestion

Figure 16 Congestion responding (a) Last-miles (b) Remote

Figure 16(b) illustrates the congestion produced beyond the last miles. At workload L(3,1), PCM can provide a better throughput owing to its ability to detect

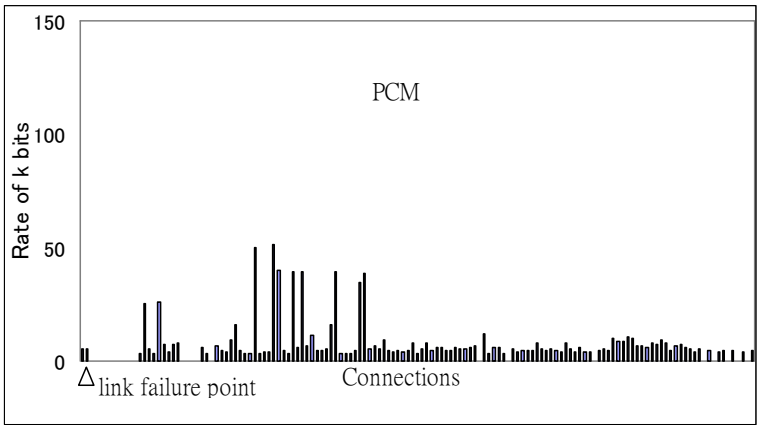
the congestion elsewhere. However, at workloads L(3,3) and L(10,3), WSDM provides a higher throughput than PCM. The throughput is higher in the weighted dispatching scheme that dispatch sessions to the congested path. Under heavy traffic loading, when using the "best" dispatching scheme only dispatch traffic to the non-congested path, the non-congested path becomes overloaded. Therefore, capacity can be increased by using congested paths as in the weighted approach. This is due to the limited total capacity.

4.4.2 Failover

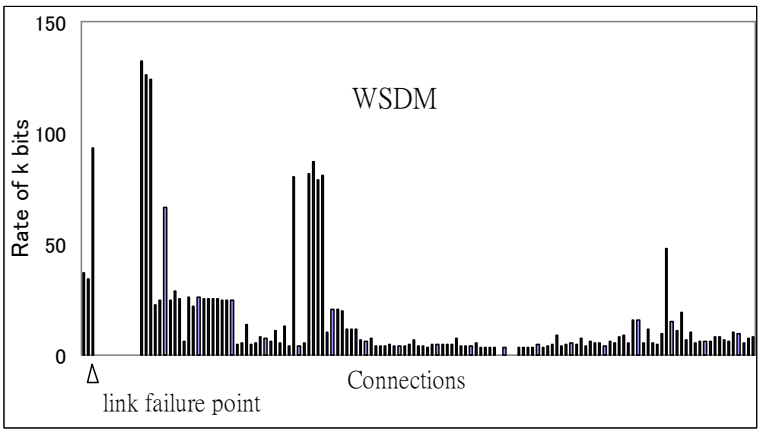
WSDM can achieve the same failover rate as PCM. PCM can measure every link at the start of every connection to keep away from the filed links. This hypothesis is verified by comparing the failover rate with the PCM and WSDM approaches through the following scenario: the emulating environment continuously produces continuous http downloads from clients and generates a disconnection at the second link after a period of time. These continuous connections select paths according to the measuring algorithms.

Figures 17(a) and 17(b) each differently display the throughput value of every connection of different measuring methods, PCM and WSDM. Figures 17(a) and 17(b) illustrate that failed transmissions occur at the former connections around the point of a link failure action, resulting in a throughput value of 0, and most of the following connections can be successful transmitted.

These failed connections face the link failure situation during the data transfer phase, meaning that they cannot utilize the connection setup phase to choose an available ISP link. Both PCM and WSDM have to utilize the connection setup phase to obtain an available path.



(a) PCM



(b) WSDM

Figure 17 The graphs depict the different algorithms' throughput of the consecutive connections that face the link failure condition. (a) Use PCM (b) use WSDM

In Figures 17(a) and 17(b) the throughput value is higher at the beginning, because as the active connection increases, the average service rate for each connection decreases. Few connections are obtained for each timeout situation of transmission.

The throughput value is higher in the former connections of 17(b) than in 17(a) after the point of a link failure. This is because at this period, the total traffic loading of using WSDM is lower than that of using PCM. WSDM may select a failed link in its first path selection, implying that the total number of active connections is smaller than that of PCM at the beginning potion of Figure 17(b) after the point of a link failure. In the experimental data of a path selection situation of 20–50 connections with WSDM and PCM approaches, WSDM has 10 connections selecting a bad link at the first selection. These connections select a successful link by SYN retransmission.

Table 6 Performance comparison of WSDM and PCM

	Failed ratio	Mean rate	Rate variance	Mean duration	Duration variance
PCM	0.09	7.5	1.6	71	25
WSDM	0.04	11.2	42.8	61	26

Table 6 compares the performance status of WSDM and PCM. WSDM performs better at a failed connection ratio, mean rate, and mean transmission duration, while PCM has a smaller variance at the mean rate and transmission duration.

Chapter 5 Conclusion

Using the load-balance mechanism in a multihoming network does not need to exchange lots of routing information to every connected ISP, and it is not necessary to conduct a lot of measuring.

In this study our contribution is to categorize the general load-balance algorithms and describe them by five generic parameters. We have reported these parameters and their effect in performance at different workloads. Therefore, when applying a load-balanced algorithm in a multihoming network, a suitable algorithm can be chosen to meet the characteristics of the specific network. If the load-balanced throughput is dominated by traffic at the last miles, the CSN-type or ST-type algorithm could be considered to decrease the measuring cost. If the traffic workload is heavy, the weighed dispatching scheme is a good choice to gain higher utilization.

Using multiple economic links to gain an aggregated throughput is applicable in an enterprise network. However, according to our experiments, the consideration of the flow control of a narrower bandwidth should be included in the planning process of a multihoming network.

The performances of algorithms are compared at various workloads to observe their responses to traffic congestion. Algorithms respond differently at light and heavy workloads. When the workload is light, an algorithm that can detect the bottleneck to avoid the traffic-congested path yields a better throughput. Conversely, when the workload is heavy, the highest throughput is achieved by algorithms with the weighted dispatching scheme. However, utilizing a congested path to gain throughput under a heavy workload leads to transmission delay. Finally, the algorithm with both

congestion detection and weighted dispatching yields both a better throughput and EBU at various workloads.

In this study we also propose a per-connection timely detection scheme for end-to-end transmission, called WSDM. Its resource usage efficiency and ability to keep away from the outage path have been proven. Comparing to the measuring method using extra packets to get end-to-end round trip time to do path selection for every connection, WSDM can achieve the same successful percentage of end-to-end transmission. WSDM can also provide better bandwidth utilization in a heavy workload situation owing to its weighted dispatching scheme.

The main benefit of WSDM to equipment vendors is its lower resource usage of measuring operations that can be implemented in cost effective hardware. For ISPs, WSDM consumes little bandwidth resource and does not require routing information exchange or the input traffic burden of measuring traffic. For an enterprise, WSDM provides an enhanced utilization of bandwidth.

Reference

1. A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman. A measurement-based analysis of multihoming. In Proc. of ACM SIGCOMM, August 2003.
2. D. Goldenberg, L. Qiu, H. Xie, Y.R. Yang and Y. Zhang, Optimizing Cost and Performance for Multihoming, in Proceedings of the 2004 ACM SIGCOMM Conference, August 2004
3. F. Guo, J. Chen, W. Li, T. Chiueh, "Experiences in Building a Multihoming Load Balancing System," In Proc. IEEE INFOCOM, March 2004
4. K. Egevang and P. Francis, "The IP Network Address Translator(NAT)," RFC 1631, May 1994
5. http://www.radware.com
6. http://www.f5.com
7. http://www.deansoft.com.tw/Ehome.htm
8. A. Akella, S. Seshan, A. Shaikh, "Multihoming Performance Benefits: An Experimental Evaluation of Practical Enterprise Strategy, " USENIX 2004
9. T. Bates, Y. Rekhter, "Scalable Support for Multi- homed Multi-provider Connectivity," RFC2260, January 1998
10. Y. Rekhter, T. Li, "An Architecture for IP Address Allocation with CIDR," RFC1518, September 1993
11. Y. Rekhter, T. Li, "A Border Gateway Protocol 4 (BGP-4)," 1995, RFC1771, March 1995
12. C. Labovitz, R. Malan, and F. Ahanian, "Internet routing instability," IEEE/ACM Trans. Networking, vol.6, mo. 5, pp.515-558, 1998.

13. C. Labovitz, R. Malan, and F. Ahanian, “Origins of Internet routing instability,” in Proc. IEEE INFOCOM, 1999
14. C. Labovitz, A. Ahuja,and F. Ahanian, “Experimental study of Internet stability and wide-area network failure,” in Proc. International Symposium on FaultTolerant Computing, June 1999
15. T.G. Griffin and B.J. Premore, “An Experimental Analysis of BGP Convergence Time,” in Proc. of ICNP 2001
16. D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. Felix Wu, and L. Zhang, “Improving BGP convergence through consistency assertions“, in Proc. IEEE INFOCOM, 2002.
17. Wei Li, “Inter-domain Routing: Problems and Solutions“, Technical Report, State University of New York, Feb 2003.
18. C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. “Delayed Internet routing convergence“,. Proc. ACM SIGCOMM '00, Stockholm, Sweden, pp. 175-187, 2000
19. D.G. Andersen, H. Balakrishnan, M.F. Kaashoek, and R. Morris, “Resilient Overlay Networks,” in 18th ACM Symposium on Operating Systems Principles (SOSP), October 2001.
20. N. Feamster, D.G. Andersen, H. Balakrishnan, M.F. Kaashoek, “Measuring the Effects of Internet Path Faults on Reactive Routing,” ACM SIGMETRICS, San Diego, CA, June 2003.
21. P. Srisuresh, D. Gan, “Load Sharing using IP Network Address Translation,” RFC 2391, August 1998

22. A. Feldmann, “Characteristics of TCP connection arrivals,” in: Self-similar Network Traffic and Performance Evaluation, eds. K. Park and W. Willinger, John Wiley and Sons, pp. 367-399, 2000
23. V. Paxson, and S. Floyd, “Wide-Area Traffic: the Failure of Poisson Modeling,” IEEE/ACM Transactions on Networking, June 1995
24. M. Crovella and A. Bestavros. “Self-similarity in World Wide Web traffic: Evidence and possible causes. “ IEEE/ACM Transactions on Networking, 5(6):835--846, Nov 1997.
25. A. Feldmann, A. Gilbert, W. Willinger, and T. Kurtz, “The changing nature of network traffic: Scaling phenomena, “ Computer Communications Review, vol. 28, no. 2, April 1998.

Appendix A

Example of experimental raw data:

- Connection transferred status at workload L3(3,1)
- under traffic congestion situation
- using WMORBF algorithm

ID	Src-IP	Srcport	Start time	Setup Time	Total time	Size	Rate	State	End time
41	192.168.100.1	1041	1099857471.76	0.002725	1.465922	50250	267.8029	OK	1.47
1	192.168.100.3	1001	1099857471.76	0.021623	1.556455	50250	252.2258	OK	1.56
42	192.168.100.1	1042	1099857473.22	0.000697	1.339441	50250	293.091	OK	2.81
2	192.168.100.3	1002	1099857473.32	0.039352	1.629028	50250	240.9892	OK	3.19
43	192.168.100.1	1043	1099857474.56	0.059662	1.019458	50250	385.0851	OK	3.83
3	192.168.100.3	1003	1099857474.94	0.000635	0.819395	50250	479.1073	OK	4.01
44	192.168.100.1	1044	1099857475.58	0.04981	1.479501	50250	265.345	OK	5.31
4	192.168.100.3	1004	1099857475.95	0.008671	1.368072	50250	286.9572	OK	5.56
45	192.168.100.1	1045	1099857477.06	0.000752	0.769431	50250	510.2188	OK	6.07
46	192.168.100.1	1046	1099857478.06	0.000898	0.769562	50250	510.1319	OK	7.07
47	192.168.100.1	1047	1099857479.07	0.000672	0.767956	50250	511.1987	OK	8.07
48	192.168.100.1	1048	1099857480.07	0.000516	0.766338	50250	512.278	OK	9.07
49	192.168.100.1	1049	1099857481.07	0.000731	0.735155	50250	534.0073	OK	10.04
50	192.168.100.1	1050	1099857482.07	0.000709	0.753306	50250	521.1403	OK	11.06
51	192.168.100.1	1051	1099857483.07	0.000704	0.771498	50250	508.8518	OK	12.08
52	192.168.100.1	1052	1099857484.07	0.00067	0.769889	50250	509.9152	OK	13.08
53	192.168.100.1	1053	1099857485.07	0.000725	0.768258	50250	510.9978	OK	14.08
5	192.168.100.3	1005	1099857477.31	8.18344	50.136365	50250	7.830207	OK	55.69
6	192.168.100.3	1006	1099857527.45	0.00073	0.769694	50250	510.0444	OK	56.46
7	192.168.100.3	1007	1099857528.45	0.000619	0.768908	50250	510.5658	OK	57.46
8	192.168.100.3	1008	1099857529.45	0.000665	0.767337	50250	511.6111	OK	58.46
9	192.168.100.3	1009	1099857530.46	0.000645	0.765712	50250	512.6968	OK	59.46
10	192.168.100.3	1010	1099857531.46	0.000602	0.76411	50250	513.7717	OK	60.46
11	192.168.100.3	1011	1099857532.46	0.000683	0.772472	50250	508.2102	OK	61.47
12	192.168.100.3	1012	1099857533.46	0.000641	0.770866	50250	509.269	OK	62.47
13	192.168.100.3	1013	1099857534.46	0.000705	0.769265	50250	510.3289	OK	63.47
14	192.168.100.3	1014	1099857535.47	0.000667	0.767662	50250	511.3945	OK	64.47
54	192.168.100.1	1054	1099857486.08	8.38077	50.307779	50250	7.803527	OK	64.62
55	192.168.100.1	1055	1099857536.38	0.000693	1.430508	50250	274.4327	OK	66.06
15	192.168.100.3	1015	1099857536.47	0.006069	1.456644	50250	269.5086	OK	66.17

56	192.168.100.1	1056	1099857537.81	0.000692	0.769713	50250	510.0318	OK	66.83
16	192.168.100.3	1016	1099857537.92	0.000575	0.819893	50250	478.8163	OK	66.99
17	192.168.100.3	1017	1099857538.92	0.000635	0.769861	50250	509.9338	OK	67.94
18	192.168.100.3	1018	1099857539.93	0.000881	0.73859	50250	531.5237	OK	68.91
19	192.168.100.3	1019	1099857540.93	0.000656	0.76661	50250	512.0963	OK	69.94
20	192.168.100.3	1020	1099857541.93	0.000678	0.764944	50250	513.2116	OK	70.94
57	192.168.100.1	1057	1099857538.82	8.126867	49.616032	50250	7.912324	OK	116.67
58	192.168.100.1	1058	1099857588.43	0.000743	0.769779	50250	509.9881	OK	117.44
59	192.168.100.1	1059	1099857589.43	0.000659	0.769258	50250	510.3335	OK	118.44
60	192.168.100.1	1060	1099857590.43	8.214796	49.6147	50250	7.912537	OK	168.29
21	192.168.100.2	1021	1099857471.76	8.160236	181.6097	50250	2.161658	OK	181.61
22	192.168.100.2	1022	1099857653.37	0.000755	0.764631	50250	513.4217	OK	182.37
23	192.168.100.2	1023	1099857654.37	0.000651	0.754076	50250	520.6082	OK	183.37
24	192.168.100.2	1024	1099857655.37	0.000714	0.772282	50250	508.3352	OK	184.39
25	192.168.100.2	1025	1099857656.37	0.000703	0.770671	50250	509.3978	OK	185.39
26	192.168.100.2	1026	1099857657.38	0.000658	0.769074	50250	510.4556	OK	186.39
27	192.168.100.2	1027	1099857658.38	0.000729	0.737797	50250	532.095	OK	187.36
28	192.168.100.2	1028	1099857659.38	0.000663	0.866239	50250	453.1984	OK	188.49
29	192.168.100.2	1029	1099857660.38	0.000665	0.764224	50250	513.6951	OK	189.39
30	192.168.100.2	1030	1099857661.38	8.329544	106.42267	50250	3.688858	OK	296.05
31	192.168.100.2	1031	1099857767.81	8.372031	66.668615	50250	5.8885	OK	362.72
32	192.168.100.2	1032	1099857834.48	0.000801	0.769722	50250	510.0259	OK	363.49
33	192.168.100.2	1033	1099857835.48	0.000722	0.768225	50250	511.0197	OK	364.49
34	192.168.100.2	1034	1099857836.48	0.00067	0.766602	50250	512.1016	OK	365.49
35	192.168.100.2	1035	1099857837.48	0.000696	0.765006	50250	513.17	OK	366.49
36	192.168.100.2	1036	1099857838.48	0.000681	0.753576	50250	520.9536	OK	367.48
37	192.168.100.2	1037	1099857839.48	0.000643	0.872173	50250	450.115	OK	368.60
38	192.168.100.2	1038	1099857840.49	0.000695	0.77016	50250	509.7358	OK	369.50
39	192.168.100.2	1039	1099857841.49	0.000727	0.768564	50250	510.7943	OK	370.50
40	192.168.100.2	1040	1099857842.49	0.000636	0.766913	50250	511.894	OK	371.50