# IMPROVING NETWORK ROUTING PERFORMANCE IN DYNAMIC ENVIRONMENTS

A Dissertation

by

YONG LIU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2006

Major Subject: Computer Engineering

# IMPROVING NETWORK ROUTING PERFORMANCE IN DYNAMIC ENVIRONMENTS

A Dissertation

by

YONG LIU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

| | |
|---|---|
| Chair of Committee, | A. L. Narasimha Reddy |
| Committee Members, | Pierce E. Cantrell |
| | Riccardo Bettati |
| | Marina Vannucci |
| Head of Department, | Costas N. Georghiades |

December 2006

Major Subject: Computer Engineering

# ABSTRACT

Improving Network Routing Performance in Dynamic Environments.

(December 2006)

Yong Liu, B.S., Peking University;

M.E., Peking University

Chair of Advisory Committee: Dr. A. L. Narasimha Reddy

In this dissertation, we study methods for improving the routing performance of computer communication networks in dynamic environments. The dynamic environments we considered in this work include both network topology changes and traffic demand changes.

In the first part, We propose a novel fast rerouting scheme for link state routing protocols. Link state routing protocols are widely used by today's ISPs on their backbone networks. The global update based rerouting of link state protocols usually takes seconds to complete which affects real time applications like Voice over IP. In our scheme, usually, only routers directly connected to failed links are involved in rerouting. For other cases, only a small number of neighboring routers are also involved. Since our scheme calculates rerouting paths in advance, rerouting can be done faster than previous reactive approaches. The computation complexity of our scheme is less than previous proactive approaches.

In the second part, we study Multihoming Route Control (MRC) that is a technology used by multihomed stub networks recently. By selecting ISPs with better quality, MRC can improve routing performance of stub networks significantly.

We first study the stability issue of distributed MRC and propose two methods to avoid possible oscillations of traditional MRC. The first MRC method is based

on “optimal routing”. The idea is to let the stub networks belonging to a same organization coordinate their MRC and thus avoid oscillations. The second method is based on “user-optimal routing”. The idea is to allow MRC devices to use multiple paths for traffic to one destination network and switch traffic between paths smoothly when path quality or the traffic matrix changes.

A third MRC method we propose is for MRC of traffic consisting of TCP flows of different sizes on paths with bottlenecks of limited capacity. Based on analysis of quality characteristics of bottleneck links, we propose a greedy MRC approach that works in small timescales. Simulation results show that the proposed MRC method can greatly improve routing performance for the MRC sites as well as the overall routing performance of all sites in the network.

To Father, Mother and Geqing

# ACKNOWLEDGMENTS

I would like to take this opportunity to give my heartfelt gratitude to my advisor, Dr. A. L. Narasimha Reddy. This work would not have been possible without his consistent encourage and support. I greatly appreciate his sage guidance for my research and tremendous effort to help me to finish this work. I also want to give my thanks to Dr. Reddy for guiding me to master the methodology for academic research which will be beneficial to me for the rest of my life.

I also need to give my heartfelt thanks to my committee members: Dr. Pierce Cantrell, Dr. Riccardo Bettati and Dr. Marina Vannucci, for their valuable suggestions, comments and constructive criticism to this work.

My thanks also go to other students in Dr. Reddy's group including: Seong Soo Kim, Sumitha Bhandarkar and Sukwoo Kang, for their valuable discussion on research with me and help during my Ph.D. study.

I also want to give many thanks to my fellow lab-mates including: Zhili Zhao, Xiaonan Ma, Zhuo Li, Haiyun You, Xiang Lu, Chuan He, Wentao Zhao and Le Zou, who gave me countless help during my Ph.D. study and made life in College Station warm and joyful.

I am in debt to my parents and my brother and sister, for their selfless love and support from my birth until today. I am also in debt to my parents-in-law for their trust, love and support. Finally, I want to say "thank you" to my wife, Geqing Liu, for her love, support and encouragement.

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION . . . . . 1

A. A fast rerouting scheme for ISP networks . . . . . 2
B. Fractional Multihoming Route Control (MRC) for stub networks . . . . . 4

II A FAST REROUTING SCHEME FOR OSPF/IS-IS NETWORKS 8

A. Fast rerouting scheme . . . . . 8
1. Overview . . . . . 8
2. Calculation of Rerouting Paths (RP) . . . . . 9
3. Identification of affected traffic . . . . . 11
4. Rerouting operations and setup of RP . . . . . 16
a. Operations of local router . . . . . 16
b. Operations of routers on a RP . . . . . 16
c. Modified forwarding operations . . . . . 19
B. Evaluation . . . . . 20
1. Number of signaling hops . . . . . 20
2. Path elongation . . . . . 21
3. Complexity of algorithms . . . . . 22
C. Future work . . . . . 23
D. Conclusions . . . . . 23

III FRACTIONAL MULTIHOMING ROUTE CONTROL . . . . . 25

A. Introduction . . . . . 25
B. Measurement study of qualities of alternate paths provided through multihoming . . . . . 28
1. Measurement method . . . . . 28
2. Measurement results . . . . . 30
C. Simulation methods . . . . . 32
1. Topologies and path characteristics . . . . . 32
2. Traffic demands . . . . . 34
3. Queuing delay models . . . . . 35
D. Basic multihoming route control and possible oscillations . 36

CHAPTER Page

1. The greedy method: best-path-only multihoming route control . . . 36
2. A less greedy method: threshold based load-balancing 37
3. Possible oscillations of basic multihoming route control method . . . 38
E. Framework of fractional multihoming route control . . . . 43
F. Implementation of fractional forwarding engine . . . . . 50
G. Related work . . . . . 53
H. Conclusions . . . . . 53

IV ROUTE OPTIMIZATION AMONG A GROUP OF MULTIHOMED STUB NETWORKS . . . . . 54

A. Introduction . . . . . 54
B. Route optimization among a group of multihomed stub networks . . . . . 54
1. Optimal routing formulation . . . . . 54
C. Evaluation . . . . . 57
1. Simulation scenarios . . . . . 57
2. Implementation of the optimal routing algorithm . . . 58
3. Simulation results . . . . . 59
a. Static analysis . . . . . 59
b. Dynamic analysis . . . . . 61
D. Conclusions . . . . . 68

V USER-OPTIMAL MULTIHOMING ROUTE CONTROL . . . . 69

A. Introduction . . . . . 69
B. Related work . . . . . 70
C. Optimal routing formulation . . . . . 71
D. User-optimal routing formulation . . . . . 73
1. User-optimal routing . . . . . 73
2. Characterization of user-optimal routing . . . . . 74
3. Characterization of optimal routing solutions . . . . . 74
4. Formulation of user-optimal routing . . . . . 75
E. User-optimal routing based MRC among a group of multihomed stub networks . . . . . 76
F. Evaluation . . . . . 78
1. Performance compared to optimal routing . . . . . 78
2. Dynamic behavior . . . . . 101

CHAPTER Page

G. Conclusions and future work . . . . . 103

VI MULTIHOMING ROUTE CONTROL OF HIGHLY DYNAMIC TCP TRAFFIC . . . . . 105

A. Introduction . . . . . 105
B. Link characteristics . . . . . 106
1. Simulation setup . . . . . 106
2. Simulation results and analysis . . . . . 108
C. MRC of highly dynamic TCP traffic . . . . . 116
1. Greedy MRC for highly dynamic TCP traffic . . . . . 116
2. Implementation of greedy MRC . . . . . 118
D. Simulation study . . . . . 119
1. Implementation of switching algorithms in simulations 119
2. Simulation setup . . . . . 120
3. Simulation result . . . . . 123
a. Smoothing of measurement result . . . . . 123
b. Single MRC sender stub network: routing performance of the MRC site . . . . . 125
c. Single MRC sender stub network: effect on non-MRC stub networks . . . . . 129
d. Multiple MRC sender stub networks: overall performance . . . . . 137
E. Conclusions . . . . . 139

VII CONCLUSIONS AND FUTURE WORK . . . . . 140

REFERENCES . . . . . 143

VITA . . . . . 150

# LIST OF TABLES


| TABLE | | Page |
|---|---|---|
| I | Notations for fast rerouting algorithms | 12 |
| II | Average completion time of traffic of MRC site, 2 ISP case: traffic matrix 1 | 126 |
| III | Average completion time of traffic of MRC site, 2 ISP case: traffic matrix 2 | 128 |
| IV | Number of TCP events experienced by traffic of MRC site, 2 ISP case, traffic matrix 1 | 132 |
| V | Number of TCP events experienced by traffic of MRC site, 2 ISP case, traffic matrix 2 | 137 |

# LIST OF FIGURES


| FIGURE | | Page |
|---|---|---|
| 1 | The sink tree of $b$ | 10 |
| 2 | Algorithm SEQ($ST(T_s, i)$) | 13 |
| 3 | Algorithm DFS_seq($ST(T_s, i)$, $n$) | 14 |
| 4 | Sequence numbers of nodes in a subtree tree | 15 |
| 5 | Algorithm RP_SETUP1($(a, b)$) | 17 |
| 6 | Algorithm RP_SETUP2($msg$) | 18 |
| 7 | Algorithm NEW_FWD(pkt) | 19 |
| 8 | Percentage of RPs with different number of signaling hops and average number of signaling hops (the number above each vertical bar) | 21 |
| 9 | Elongation ratio and value compared to optimal paths | 22 |
| 10 | Traceroute measurement | 28 |
| 11 | Average RTT differences and loss rate differences of pair 1 of alternate paths. (over 5 minute durations, Y axis: $\Delta RTT$ in milliseconds and $\Delta loss\_rate$, X axis: time since the start of measurement in hours) | 30 |
| 12 | Average RTT differences and loss rate differences of pair 2 of alternate paths. (using same parameters as Fig. 11) | 31 |
| 13 | Average RTT differences and loss rate differences of pair 3 of alternate paths. (using same parameters as Fig. 11) | 31 |
| 14 | Topology consisting of multihomed stub networks, the edge routers of their ISPs and the paths among them | 34 |
| 15 | Basic multihoming route control algorithm for a MRC device | 37 |

FIGURE Page

16 Average delay and delay variance of best-path-only MRC v.s. utilization and path quality diversity factors (asymmetric topology, Pareto type traffic) . . . . . 39

17 Average delay and delay variance of threshold based MRC v.s. utilization and path quality diversity factors (asymmetric topology, Pareto type traffic) . . . . . 40

18 Average delay and delay variance of static load-balancing v.s. utilization and path quality diversity factors (asymmetric topology, Pareto type traffic) . . . . . 40

19 Average delay and average loss rate of best-path-only MRC . . . . . 41

20 Routing vector for one pair of source and destination in best-path-only MRC . . . . . 42

21 End to end virtual delay for one pair of source and destination in best-path-only MRC . . . . . 42

22 Average delay and delay variance of best-path-only MRC v.s. utilization and path quality diversity factors (symmetric topology, Pareto type traffic) . . . . . 44

23 Average delay and delay variance of threshold based MRC v.s. utilization and path quality diversity factors (symmetric topology, Pareto type traffic) . . . . . 45

24 Average delay and delay variance of static load-balancing v.s. utilization and path quality diversity factors (symmetric topology, Pareto type traffic) . . . . . 45

25 Average delay and delay variance of best-path-only MRC v.s. utilization and path quality diversity factors (asymmetric topology, Poisson type traffic) . . . . . 46

26 Average delay and delay variance of threshold based MRC v.s. utilization and path quality diversity factors (asymmetric topology, Poisson type traffic) . . . . . 46

FIGURE Page

27 Average delay and delay variance of static load-balancing v.s. utilization and path quality diversity factors (asymmetric topology, Poisson type traffic) . . . . . 47

28 Average delay and delay variance of best-path-only MRC v.s. utilization and path quality diversity factors (symmetric topology, Poisson type traffic) . . . . . 47

29 Average delay and delay variance of threshold based MRC v.s. utilization and path quality diversity factors (symmetric topology, Poisson type traffic) . . . . . 48

30 Average delay and delay variance of static load-balancing v.s. utilization and path quality diversity factors (symmetric topology, Poisson type traffic) . . . . . 48

31 Multihoming route control system architecture . . . . . 49

32 Performance improvement ratios: topologies of 10 stub networks, each has 2 ISPs, capacities are in Mbps . . . . . 60

33 Performance improvement ratios: topologies of 10 stub networks, each has 3 ISPs, capacities are in Mbps . . . . . 62

34 Performance improvement ratios: topologies of 10 stub networks, each has 4 ISPs, capacities are in Mbps . . . . . 63

35 Performance improvement ratios: topologies of 20 stub networks, each has 2 ISPs, capacities are in Mbps . . . . . 64

36 Performance improvement ratios: topologies of 20 stub networks, each has 3 ISPs, capacities are in Mbps . . . . . 65

37 Performance improvement ratios: topologies of 20 stub networks, each has 4 ISPs, capacities are in Mbps . . . . . 66

38 Inhouse traffic end to end delay improvement: asymmetric topology, Pareto traffic. Topology 1 to 9: 4x2 4x3 4x4 10x2 10x3 10x4 20x2 20x3 20x4 . . . . . 66

FIGURE Page

39 Inhouse traffic end to end delay improvement: asymmetric topology, Poisson traffic. Topology 1 to 9: 4x2 4x3 4x4 10x2 10x3 10x4 20x2 20x3 20x4 . . . . . 67

40 Inhouse traffic end to end delay improvement: symmetric topology, Pareto traffic. Topology 1 to 9: 4x2 4x3 4x4 10x2 10x3 10x4 20x2 20x3 20x4 . . . . . 67

41 Inhouse traffic end to end delay improvement: symmetric topology, Poisson traffic. Topology 1 to 9: 4x2 4x3 4x4 10x2 10x3 10x4 20x2 20x3 20x4 . . . . . 68

42 User-optimal routing based MRC (for traffic from $i$ to $j$, where $i \in N$; $j \in N, j \neq i$ or $j \in M_i$.) . . . . . 77

43 Optimization objective, 8x2, symmetric topology: average, minimum and maximum . . . . . 79

44 Average delay, 8x2, symmetric topology: average, minimum and maximum . . . . . 80

45 Average loss rate, 8x2, symmetric topology: average, minimum and maximum . . . . . 80

46 Maximum link utilization, 8x2, symmetric topology: average, minimum and maximum . . . . . 81

47 Optimization objective, 8x2, symmetric topology: average, minimum and maximum . . . . . 81

48 Average delay, 8x2, asymmetric topology: average, minimum and maximum . . . . . 82

49 Average loss rate, 8x2, asymmetric topology: average, minimum and maximum . . . . . 82

50 Maximum link utilization, 8x2, asymmetric topology: average, minimum and maximum . . . . . 83

51 Optimization objective, 8x3, symmetric topology: average, minimum and maximum . . . . . 83

FIGURE Page

52 Average delay, 8x3, symmetric topology: average, minimum and maximum . . . . . 84

53 Average loss rate , 8x3, symmetric topology: average, minimum and maximum . . . . . 84

54 Maximum link utilization , 8x3, symmetric topology: average, minimum and maximum . . . . . 85

55 Optimization objective , 8x3, asymmetric topology: average, minimum and maximum . . . . . 85

56 Average delay , 8x3, asymmetric topology: average, minimum and maximum . . . . . 86

57 Average loss rate , 8x3, asymmetric topology: average, minimum and maximum . . . . . 86

58 Maximum link utilization , 8x3, asymmetric topology: average, minimum and maximum . . . . . 87

59 Optimization objective ratio(opt/elb), 8x2 symmetric topology: average, minimum and maximum . . . . . 90

60 Average delay ratio(opt/elb), 8x2 symmetric topology: average, minimum and maximum . . . . . 91

61 Average loss rate $\Delta$(opt-elb), 8x2 symmetric topology: average, minimum and maximum . . . . . 92

62 Maximum link utilization ratio(opt-elb), 8x2 symmetric topology: average, minimum and maximum . . . . . 92

63 Optimization objective ratio(opt/elb), 8x2 asymmetric topology: average, minimum and maximum . . . . . 93

64 Average delay ratio(opt/elb), 8x2 asymmetric topology: average, minimum and maximum . . . . . 93

65 Average loss rate $\Delta$(opt-elb), 8x2 asymmetric topology: average, minimum and maximum . . . . . 94

FIGURE Page

66 Maximum link utilization ratio(opt-elb), 8x2 asymmetric topology: average, minimum and maximum . . . . . 94

67 Optimization objective ratio(opt/elb), 8x3 symmetric topology: average, minimum and maximum . . . . . 95

68 Average delay ratio(opt/elb), 8x3 symmetric topology: average, minimum and maximum . . . . . 96

69 Average loss rate $\Delta$(opt-elb), 8x3 symmetric topology: average, minimum and maximum . . . . . 97

70 Maximum link utilization ratio(opt-elb), 8x3 symmetric topology: average, minimum and maximum . . . . . 97

71 Optimization objective ratio(opt/elb), 8x3 asymmetric topology: average, minimum and maximum . . . . . 98

72 Average delay ratio(opt/elb), 8x3 asymmetric topology: average, minimum and maximum . . . . . 99

73 Average loss rate $\Delta$(opt-elb), 8x3 asymmetric topology: average, minimum and maximum . . . . . 100

74 Maximum link utilization ratio(opt-elb), 8x3 asymmetric topology: average, minimum and maximum . . . . . 100

75 Convergence of user-optimal routing based MRC in different scenarios, simulation 1 (step size = 0.02) . . . . . 102

76 Convergence of user-optimal routing based MRC in different scenarios, simulation 2 (step size = 0.02) . . . . . 102

77 Effect of different step sizes . . . . . 103

78 Topology for link quality predictability study . . . . . 107

79 Example of Internet path quality: average utilization of 40% . . . . . 108

80 Example of Internet path quality: average utilization of 60% . . . . . 109

81 Example of Internet path quality: average utilization of 80% . . . . . 109

FIGURE Page

82 Internet path quality differences: average utilization of 60% . . . . . 110

83 Autocorrelation functions of link quality metrics: average utilization of 40% . . . . . 111

84 Autocorrelation functions of link quality metrics: average utilization of 60% . . . . . 111

85 Autocorrelation functions of link quality metrics: average utilization of 80% . . . . . 112

86 Cross correlation functions of link quality metrics: average utilization of 40% . . . . . 112

87 Cross correlation functions of link quality metrics: average utilization of 60% . . . . . 113

88 Cross correlation functions of link quality metrics: average utilization of 80% . . . . . 113

89 Autocorrelation function of Internet path quality differences: average utilization of 60% . . . . . 114

90 Cross-correlation function of Internet path quality differences: average utilization of 60% . . . . . 115

91 Simulation topology: 3 send to 1, one sender site uses MRC . . . . . 121

92 Simulation topology: 9 send to 1, all sender sites use MRC . . . . . 122

93 Performance of EWMA and last-period predictors: (1) elb-flow, (2) elb-packet, (3-10) ewma $\alpha = 0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1$, (11-16) last period, period = 5ms, 10ms, 50ms, 100ms, 500ms, 1s . 124

94 Routing performance of MRC site, 2 ISP case, traffic matrix 1: (1) elb-flow, (2) elb-packet, (3) greedy-flow, (4) greedy-packet, (5-8) greedy-flowlet, timeout: 50ms, 100ms, 250ms and 500ms . . . . . 127

95 Routing performance of MRC site, 2 ISP case, traffic matrix 2: (1) elb-flow, (2) elb-packet, (3) greedy-flow, (4) greedy-packet, (5-8) greedy-flowlet, timeout: 50ms, 100ms, 250ms and 500ms . . . . . 129

FIGURE Page

96 Statistics of normalized mean completion time(compared to elb-flow, TCP-SACK) of MRC site, 2 ISP case: (1) elb-flow, TCP-DCR; (2) elb-flow, TCP-SACK; (3) greedy-flow, TCP-DCR; (4) greedy-flow, TCP-SACK; (5) greedy-packet, TCP-DCR; (6) greedy-packet, TCP-SACK; (7) greedy-flowlet-100ms, TCP-DCR; (8) greedy-flowlet-100ms, TCP-SACK. . . . . 130

97 Routing performance of MRC site, 3 ISP cases: (1) elb-flow-TCPDCR, (2) elb-flow-TCPSACK, (3) greedy-packet-TCPDCR . . . 131

98 Routing performance of MRC site, 4 ISP cases: (1) elb-flow-TCPDCR, (2) elb-flow-TCPSACK, (3) greedy-packet-TCPDCR . . . 131

99 Routing performance of non-MRC sites, 2 ISP case, traffic matrix 1: (1) elb-flow, (2) elb-packet, (3) greedy-flow, (4) greedy-packet, (5) greedy-flowlet50ms, (6) greedy-flowlet100ms, (7) greedy-flowlet250ms, (8) greedy-flowlet500ms . . . . . 133

100 Routing performance of non-MRC sites, 2 ISP case, traffic matrix 2: (1) elb-flow, (2) elb-packet, (3) greedy-flow, (4) greedy-packet, (5) greedy-flowlet50ms, (6) greedy-flowlet100ms, (7) greedy-flowlet250ms, (8) greedy-flowlet500ms . . . . . 134

101 Statistics of normalized mean completion time(compared to elb-flow, TCP-SACK) of non-MRC site, 2 ISP case: (1) elb-flow, TCP-DCR; (2) elb-flow, TCP-SACK; (3) greedy-flow, TCP-DCR; (4) greedy-flow, TCP-SACK; (5) greedy-packet, TCP-DCR; (6) greedy-packet, TCP-SACK; (7) greedy-flowlet-100ms, TCP-DCR; (8) greedy-flowlet-100ms, TCP-SACK. . . . . 135

102 Routing performance of non-MRC site, 3 ISP cases: (1) elb-flow-TCPDCR, (2) elb-flow-TCPSACK, (3) greedy-packet-TCPDCR . . . 136

103 Routing performance of non-MRC site, 4 ISP cases: (1) elb-flow-TCPDCR, (2) elb-flow-TCPSACK, (3) greedy-packet-TCPDCR . . . 136

FIGURE Page

104 Overall routing performance when all sites using MRC, asymmetric topologies, cases: (1) elb-flow, (2) elb-packet, (3) greedy-ewma0.1-packet, (4) greedy-ewma0.2-packet, (5) greedy-ewma0.4-packet, (6) greedy-TCPSACK-flowlet50ms, (7) uopt-0.01, (8) uopt-0.05, (9) uopt-0.1, (10) uopt-0.5, (11) uopt-1 . . . . . 138

# CHAPTER I

## INTRODUCTION

As the Internet is becoming the most important communication infrastructure of our world, Internet routing has drawn increasing research interests. While the Internet routing performance has been improved greatly as the Internet evolves, the increasing demands keep bringing new challenges to the networking research community. Today's Internet is expected to provide data, voice and video services at the same time. Accordingly, the Internet needs to provide higher availability and quality then ever before.

A challenge to improve Internet routing performance today is to make Internet routing software respond to dynamic environments more effectively. The dynamic environments include both topology and routing changes on the Internet, as well as highly dynamic traffic demand changes. Accidental link failures may cause service disruption to Internet users [1]. Inter-domain route changes may affect service availability for some Internet address prefixes [2]. Traffic demand changes may cause Internet links to be over utilized, thus result in higher delay and loss rate to Internet traffic. Routing software's capability to effectively handle such network events is a key to improve service availability and quality of the Internet.

Current Internet consists of many Autonomous Systems (ASes). An AS is either a transit network, like an ISP network, or a "stub network", like a university network or an enterprise network. ASes interconnect and form the Internet. Internet is hierarchical where ISP networks compose the core, while stub networks compose the edge. See [3] for a formal study of the hierarchy of the Internet. Accordingly, Internet rout-

---

The journal model is *IEEE Transactions on Automatic Control*.

ing is also hierarchical. An AS usually runs one or more Interior Gateway Protocols (IGPs) to route traffic within its network. This is called intra-domain routing. ASes use Exterior Gateway Protocols (EGPs) to exchange routing information for traffic across AS boundaries.

This dissertation aims to solve two important problems for improving Internet routing performance in dynamic environments. The first contribution of this dissertation is a fast rerouting scheme for Internet Service Providers (ISP) networks. It helps to improve the service availability of ISP networks. The second contribution is a set of Multihoming Route Control (MRC) methods for multihomed "stub networks". These methods allow the stub networks to change their routing in respond to changes inside the Internet and on the edge of Internet close to the traffic destinations. In the remaining of this chapter, we will introduce the tow parts of this work.

## A. A fast rerouting scheme for ISP networks

The most common IGPs used by ISP networks today are OSPF [4] and IS-IS [5]. Both of them are link state routing protocols. Since our work applies to both link state routing protocols, we don't distinguish them and use "OSPF network" to represent a network running either of these two link state routing protocols. An OSPF network is divided into areas. A backbone area is used to connect all other areas. Therefore, the routing is also done hierarchically: each router just needs to know how to route a packet to an interior or border router of the same area. In this work, we focus on the routing of backbone area of intra-domain routing since the backbone area carries larger amount of traffic and its routing is more challenging.

In an OSPF network, each router knows the topologies of all the areas it belongs to. For each area topology, the router uses Dijkstra's Shortest Path First (SPF)

algorithm to calculate the shortest path tree from itself to other routers on the topology. The tree is called SPF tree. Then the router builds its routing tables from the calculated SPF tree. When the topology information on all routers in the area is synchronized, routing according to above routing tables guarantees packets are routed along the shortest path to their destinations.

When network topology changes, e.g. a link fails, the routers connected to the link send information about the event to all their neighbors. A router in the area, after receiving this information, will propagate the information to other routers other than the router from which the information comes from. With proper loop avoidance mechanisms the information is flooded through the whole area. After getting the information, every router in the area needs to update its topology database, rerun the SPF algorithm and update its routing tables and forwarding tables on line cards. The same procedure occurs, when a failed link is restored or other topology changes occur. When all the routers finish updating their routing tables after a change in topology, IGP convergence is said to occur. For a large ISP network, the backbone area can span the whole country and may consist of more than 100 routers. In such a backbone network, the convergence of a link state protocol after a link failure usually takes a few seconds or more. While some recent work shows sub-second IGP convergence can be achieved by utilizing layer 2 protection timer and fine tuning parameters of IGP protocols [6], the rerouting latency can still affect many Internet applications, e.g. real-time applications like voice and video. According to measurement studies on link failures in a backbone network [7], it is still common to have link failures on today's backbone network. Thus, it is desirable to enhance link state routing protocols in handling link failures.

A number of mechanisms are proposed by researchers to reduce the convergence latency of link state protocols. Nelakuditi et al [8] propose a pro-active fast rerout-

ing method. Using this method, a more complex algorithm compared to the SPF algorithm is used to calculate interface specific forwarding tables. Narváez et al [9] propose a localized reactive restoration algorithm for link state protocols. The algorithm requires routers on a restoration path to change the weights of links on the path to zero and recalculate their routing tables after link failures occur. The calculation of routing table and the update of forwarding tables increase the response time of the algorithm to link failures.

MPLS based approaches, such as [10], use pre-computed backup paths to route around failures immediately after the detection of link failures. However, this is usually done in a centralized manner and is not suitable for protection of all links in the network. In addition, MPLS is built on top of IGPs and relies on IGPs to propagate MPLS label information.

In the first part of this dissertation (Chapter II), we propose a hybrid fast rerouting scheme for Link State protocols that includes both pro-active and reactive components. Our scheme calculates rerouting path pro-actively, routers use the rerouting path after a link failure occurs. Since the rerouting path may involve routers not adjacent to the failed link, a short multi-hop rerouting path may need to be established after a link failure. However, no routing table recalculation is needed after a link failure, thus the response latency can be reduced.

### B. Fractional Multihoming Route Control (MRC) for stub networks

"Multihoming" means that a network has more than one external link to one or more ISPs [11]. In this work, we use "multihoming" to represent the case when the external links connect to different ISPs since it offers more benefits than multihoming to single ISP. Multihoming provides Internet connection redundancy to stub networks. When

the connection to one ISP fails or when one ISP encounters problems, the stub network can use other ISPs to connect to the Internet.

The current de facto inter-domain routing standard is BGP [12]. Since today's Internet is very large and consists of many relatively independent domains, BGP needs to maintain scalability, stability as well as the business relationship between different ASes. Studies of BGP performance have shown that the inter-domain routing protocol may converge slowly after inter-domain route changes [2]. Mis-configuration of BGP could cause persistent route oscillations [13].

In recent years, Multihoming Route Control (MRC) technology, e.g. [14], has been used by multihomed stub networks to get around inefficiency of Inter-domain routing. Because alternate paths via different upstream ISPs may have different qualities at a given time, route control devices can improve the performance of stub networks by selecting the best available path according to active or passive measurement results. Measurement experiments on the Internet show that multihoming route control can significantly improve the routing performance of stub networks [15].

There are two possible causes for the quality diversity of alternate paths: first, the paths provided by different ISPs may have different "distance" or propagation delays to a given destination; second, different ISPs and other ASes (Autonomous Systems) along the alternate paths may experience different degrees of congestion. While the AS paths are relatively static, the congestion on the Internet is dynamic. MRC divices can choose paths that have short "distance" and are less congested thus improve the routing performance of stub networks.

Most current route control devices choose the best path based on their own view of the qualities of alternate paths and switch traffic to one ISP based on changes in path quality. This type of distributed route control by a number of stub networks may interact with each other and cause oscillations, as we will show in Chapter III.

In the second part of this dissertation (Chapter IV, V and VI), we study advanced multihoming route control algorithms that can avoid oscillations.

First, we propose a global coordination method for multihoming route control among a group of multihomed stub networks (Chapter IV). In this method, volume of traffic demand and measured path delays and loss rates are exchanged among all stub networks in the group; each stub network calculates the "optimal routing" [16] solution using the data exchanged and change routing assignment according to the optimal routing solution. Through coordination, oscillations of traditional MRC can be avoided.

Second, we propose a "user-optimal" routing based distributed multihoming route control scheme (Chapter V). This approach avoids exchanging information about traffic demands and measured path quality and is fully distributed. Therefore, this scheme can be used by independent multihomed stub networks for general Internet traffic. In this approach, traffic is switched smoothly during network environment changes and tus can avoid oscillations possible with traditional multihoming route control technologies. We compare performance of this user-optimal routing based method to network optimal routing method. Through extensive simulations, we show the user-optimal routing achieves similar performance as network optimal routing. We also study the dynamic performance of our method under different network events.

Both the coordinated route optimization and the user-optimal routing based distributed MRC are evaluated using UDP type traffic. The traffic matrices are predetermined and do not change as avalable bandwidth of paths changes. This assumption is realistic for Internet traffic when network links are under-utilized. Otherwise, the results of the simulations do not have direct physical meaning. However, in the later case, higher loss rates indicate that the throughput of traffic on the path is

limited by the link capacity and data transfers take more time to complete.

In the Chapter VI, we study MRC for highly dynamic TCP traffic, i.e. TCP traffic on paths with bottlenecks of limited capacity. The traffic we studied in this chapter is different from Chapter IV and Chapter V in following ways: (1) the volume of TCP traffic adapts to changes of qualities of Internet paths; (2) the traffic volume changes more rapidly because of TCP's congestion control mechanism and increased burstiness because smaller traffic volume. In this chapter, we propose a small time scale greedy MRC for highly dynamic TCP traffic based on study of characteristics of bottleneck links for this type of traffic. We evaluate the approach using ns-2 [17] packet level simulations.

# CHAPTER II

## A FAST REROUTING SCHEME FOR OSPF/IS-IS NETWORKS

In this chapter, we propose such a Fast Rerouting scheme for Link-state protocols. In our approach, when a link fails, the affected traffic is rerouted along a pre-computed Rerouting Path. In case rerouting cannot be done locally, the local router will signal minimal number of upstream routers to setup the Rerouting Path for rerouting. We propose algorithms that simplify the rerouting operation and the Rerouting Path setup. With a simple extension to the current Link State protocols, our scheme can route around failures faster and involves minimal number of routers for rerouting.

### A. Fast rerouting scheme

In this section we first briefly describe how our scheme works, then give algorithms used in each step.

#### 1. Overview

In this paper, we assume that all links are point to point, bi-directional and with equal weights on both directions, which is generally true for backbone networks. We also assume there is at most one link failure at a time. This is because individual link failures account for nearly 70% of all unplanned failures [7] and rerouting around multiple failures requires more complex algorithms. Instead of using a complex algorithm, our scheme relies on the original IGP mechanism to handle multiple failures.

Our scheme uses the *nearest Feasible Next Hop* (with regard to number of hops) of *affected* traffic for Fast Rerouting. *Feasible Next Hop* (FNH) is defined as a router whose shortest paths to the destination of a packet do not include the failed link. Throughout this paper, *affected* means all the paths to the destination of a packet

from the local router include the failed link. When there is more than one nearest FNH, our scheme chooses the one with minimum distance from the *exit node* to affected destinations, where *exit node* is defined as the node one hop away from the nearest FNH. We define *Rerouting Path* (RP) as the path from the router adjacent to the failed link to the selected FNH.

For example, in Fig. 1, $a$ may have $e, f, h, i$ as its FNHs for traffic affected by the failure of link $(a, b)$ and path $(a, e, h)$, etc. as the corresponding RPs. Our scheme will choose one from $(a, f)$ and $(a, i)$ that has shorter distance from $a$ to $b$ as the RP.

There are tow types of RPs: 1). Local RPs, i.e. direct FNHs; 2). RPs with one or more signaling hops. If the nearest FNH is type 1, rerouting is done locally. For example, in Fig. 1, $a$ can forward all traffic affected by link $(a, b)$ via FNH $f$. If the nearest FNH is type 2, local router should setup the RP by notifying (signaling) routers on the RP about the failure and the RP before rerouting. For example, in Fig. 1, $a$ needs to notify $g$ about the failure of $(a, b)$ and the RP, $(a, g, h)$. $g$ will route all traffic affected by the failure of $(a, b)$ to $h$. Since RPs are usually very short, as shown in Section B, the cost to setup a RP is not significant.

## 2. Calculation of Rerouting Paths (RP)

In our scheme, each router calculates a RP based on its sink tree for each of its links on behalf of its neighbor. Its neighbor will use the RP to send affected traffic when the link fails. This choice of RP calculation requires routers to exchange RP information but can avoid routers to calculate sink tree for all its neighbors.

Without losing generality, we describe the algorithm for a router, $b$, to calculate a RP for a link between itself and one of its neighbors, $a$, as shown in Fig. 1.

In the sink tree of $b$, we use BFS (Breadth First Search) in the sub-tree rooted at $a$ to search for a RP for $a$ and link $(a, b)$. We call this sub-tree as the *upstream*

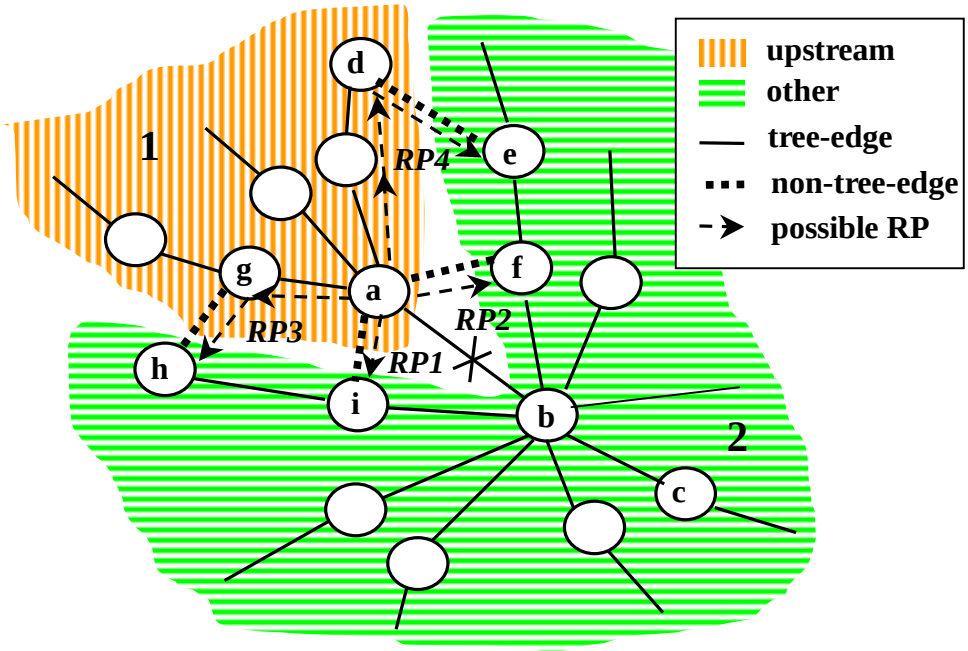


Fig. 1.: The sink tree of $b$

area for link $(a, b)$. During the search, at each node we check every neighbor of it. If the neighbor is not an upstream router, it is a FNH. As mentioned before, our scheme choose the *nearest FNH* for rerouting. When there are ECMPs (Equal Cost Multi-Paths) between $a$ and the *exit* node we use all of them as part of the RP. We can always find a RP for a given link as long as the network is not partitioned. (See Appendix for detailed algorithm.)

Using this method we actually find a rerouting path for traffic with destination of $b$. But as Theorem 1 shows, this type of RPs can be safely used for all traffic originally forwarded to $b$ by $a$.

**Theorem 1.** *Once a packet originally forwarded from $a$ to $b$ is rerouted to a router outside the upstream area (the 1st part in Fig. 1), the packet will be routed along a shortest path without link $(a, b)$ to its destination.*

*Proof.* It is obvious for packets that have $b$ as its destination.

For a packet heading for a router below $b$ , say $c$, it is also true. This can be proved as below: As shown in Fig. 1, the shortest path from a nearest FNH, $e$, to $b$, $(e, f, b)$, must be shorter than the shortest path from it via $a$ to $b$, $(e, a, b)$. So the shortest path from $e$ to $a$ to $b$ to $c$, $(e, a, b, c)$, is longer than the shortest path from $e$ to $b$ to $c$, $(e, f, b, c)$, i.e. the shortest path from $e$ to $a$ to $b$ to $c$ is not the shortest path from $e$ to $c$. Therefore, the shortest path from $e$ to $c$ must not include edge $(a, b)$. And it is obvious that the path from $e$ to $c$ must not include edge $(b, a)$. □

## 3. Identification of affected traffic

In case there is no local RP, a router on the RP needs to efficiently identify traffic affected by a link failure.

Because there are ECMPs, when a link fails, it is also possible that a destination is not reachable via one next hop but reachable via another next hop. So our scheme decides whether a destination is reachable via a given next hop.

In our scheme, a router identifies affected traffic using a simple range checking. It relies on an algorithm that assigns sequence numbers for all nodes in each *first level subtree* (i.e. the subtree under a first level child node) of the local routing tree. Sequence numbers for each subtree are independent. For each subtree, the sequence numbers are from 0 to the number of nodes in the subtree. The algorithm ensures that the sequence numbers of all nodes affected by a node failure and the sequence number of the failed node itself are continuous and thus can be represented as a simple range. In other words, when a node fails, only the nodes with sequence numbers within the *affected range* become unreachable from the root of the subtree. The start of the *affected range* of a node is its sequence number. The end of the *affected range* is the largest sequence number of all nodes affected by this node. We call the end of the *affected range* of a node as the *seq_end* number. We store the sequence number and

the *seq_end* number along with each node in the subtree.

A link failure either causes the downstream node of the link to become unreachable from the root of the subtree or does not affect reachability of any destination. So the destinations affected by a link failure can also be identified using above sequence numbers. We will discuss this further in Section 4.

For a subtree without ECMPs, we can use DFS (Depth First Search) traversal order as the sequence numbers. This is because: in a tree without ECMPs, all descendant nodes are the nodes affected by this node; and they are traversed during the period when this node is traversed.

However, for a subtree with ECMPs, if a node becomes unreachable from the root of the subtree, some of its descendants may be still reachable, because there may be more than one path from the root to the later. We have developed a modified DFS algorithm to assign sequence numbers for nodes in a general subtree (with or without ECMPs). The algorithm, *SEQ*, is shown in Figs. 2 and 3. Notations used in this and following algorithms are listed in Table I.

Table I. Notations for fast rerouting algorithms

| | |
|---|---|
| $V$: | set of all vertices of the topology |
| $E$: | set of all edges of the topology |
| $(a, b)$: | link from $a$ to $b$ |
| $RP(a, b)$: | Rerouting Path of link $(a, b)$ |
| $T_s$: | routing tree of $s$ |
| $ST(T_s, i)$: | subtree of $T_s$ rooted at $i$ |
| $P(T_s, n)$: | set of parents of $n$ in $T_s$ |
| $C(T_s, n)$: | set of children of $n$ in $T_s$ |

Algorithm *SEQ* can be described as follows: during the DFS traversal, whenever

$ST(T_s, i).seq\_count \Leftarrow 0$

**foreach** $n \in ST(T_s, i)$ **do**

  $n.enqueued \Leftarrow False$

  $n.visited \Leftarrow False$

$DFS\_seq(ST(T_s, i), i)$

Fig. 2.: Algorithm SEQ($ST(T_s, i)$)

we encounter a child node having more than one parent, we find the nearest single upstream node whose failure will cause the child node unreachable from the root. We call this upstream node as the *nearest ancestor* of the child node. (The algorithm to find the *nearest ancestor* is a reverse DFS traversal from the node to the root. See Appendix for detailed algorithm.) We enqueue the child node to the *deferred DFS queue* of its *nearest ancestor*. After that we continue the DFS traversal for other children. After traversing all its children, we call this modified DFS algorithm for the nodes in the local deferred DFS queue one by one. Similar to subtree without ECMPs, the sequence number of each node is its traversal order. Fig. 4 shows the result of algorithm *SEQ* on a simple topology.

As Theorem 2 shows, the sequence numbers and the checking ranges assigned by the above modified DFS algorithm can be used to identify the unreachable nodes after a node fails.

**Theorem 2.** *In a routing tree, where each node is assigned a sequence number and an affected range using algorithm* SEQ *if a node fails, all and only the nodes with sequence numbers in the affected range of the node are affected.*

*Proof.* During the traversal, whenever a child node is found to have more than one parent it is added to the deferred search queue of its *nearest ancestor*. It is equivalent to removing the links of this node to its current parent and linking it to its *nearest*

```
/* n:  DFS start node */
n.visited ⇐ True
if |P(ST(T_s, i), n)| > 1 and n.enqueued = False then
    p ⇐ DFS_nearest_ancestor(ST(T_s, i), n)
    enqueue(p.deferred_dfs_queue, n)
    n.enqueued ⇐ True
else
    n.seq[i] ⇐ ST(T_s, i).seq_count
    ST(T_s, i).seq_count ⇐ ST(T_s, i).seq_count + 1
    foreach m ∈ C(ST(T_s, i), n) do
        if m.visited = False then
            DFS_seq(ST(T_s, i), m)
    while n.deferred_dfs_queue ≠ ∅ do
        m ⇐ dequeue(n.deferred_dfs_queue)
        DFS_seq(ST(T_s, i), m)
    n.seq_end[i] ⇐ ST(T_s, i).seq_count − 1
/* Notes:  visited ensures each node having single parent is
   visited only once (consider ECMPs).  Nodes having multiple
   parents are visited twice:  one for enqueuing, one for real
   traversal.)  enqueued ensures every node is enqueued only once.
   It is used to distinguish the tow times we visited a node having
   multiple parents.   */
```

Fig. 3.: Algorithm DFS_seq($ST(T_s, i)$, $n$)

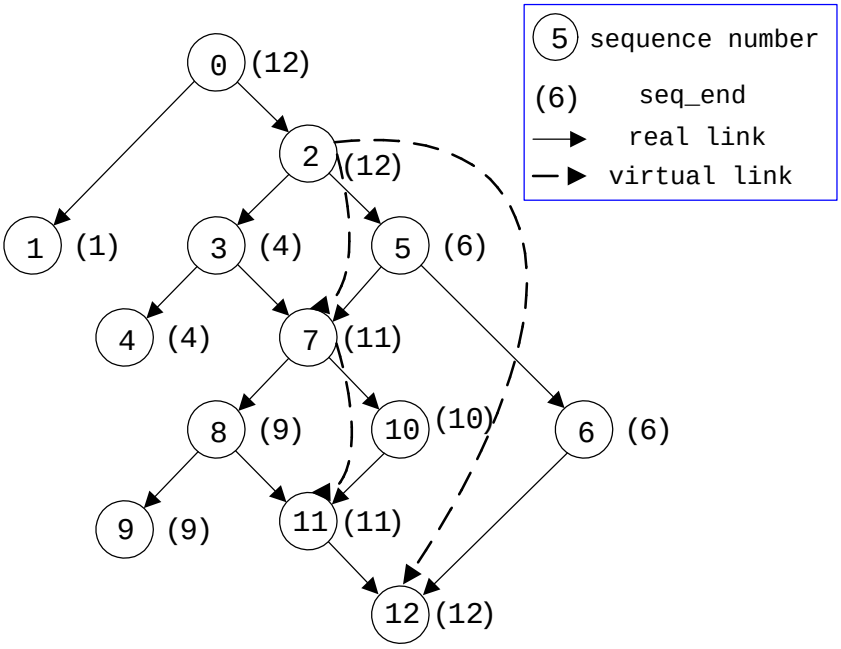


Fig. 4.: Sequence numbers of nodes in a subtree tree

*ancestor* as a child via a *virtual link*.

We claim that when we move this child node to its new parent (i.e. its *nearest ancestor*), the DFS traversal has not left its new parent. This is because:

When we move the node($C$) to its new parent($A$), the currently traversed node($B$) is one of $C$'s original parents. We must arrive at $B$ via a path from the root to it. The path may consist of real links and *virtual links* we created before. According to the DFS algorithm, all the nodes on this path has not been left at this time.

The *nearest ancestor* of node $C$ must be one of the nodes on this path or a node in the real topology between two ends of a virtual link on this path. However, the failures of nodes in the real topology between the two ends of a virtual link do not *affect* the reachability of $C$, since the failures do not *affect* the reachability of the downstream end of the virtual link. Thus the *nearest ancestor* of node $C$ must be one of the nodes on this path. Hence the claim is correct.

The above claim also ensures that all nodes in the tree are traversed by the DFS

traversal.

Therefore, the algorithm transforms the routing tree into a simple tree(without ECMPs) and the traversal is equal to a DFS traversal on the transformed tree. In the transformed tree(after removing links and adding virtual links), a failure of a node will affect all and only its descendants. According to properties of DFS traversal, the theorem is proved.

□

## 4. Rerouting operations and setup of RP

In this subsection we give detailed description of the rerouting operations of routers after a link failure detected.

### a. Operations of local router

The operations of a local router is given by algorithm $RP\_SETUP1$ shown in Fig. 5. After detection of a link failure, the local router first marks the next hop unreachable. As a result, for traffic having other ECMP next hops, the router will avoid using this next hop. If the RP for the failed link is a local FNH, the router set the *rerouting flag* and the *rerouting next hop*. If the RP includes some upstream nodes, the router send messages to them to notify the link failure and the RP before rerouting traffic along the RP.

### b. Operations of routers on a RP

The operations of a router on a RP are given in algorithm $RP\_SETUP2$ shown Fig. 6. When a router receives a RP setup request, it marks *affected* interfaces and sets the rerouting flags and affected range for the interface. We call an interface *affected* when some packets forwarded through the interface cannot reach their destinations

```
/* called by a for failed link (a, b) */
interface(b).failure ⇐ True
if interface(b).local_reroue = False then
    foreach RP ∈ RPs(a, b)
    /* There are multiple RPs only when there are ECMPs between a
       and the exit node. */
    do
        msg.RP ⇐ RP
        msg.failed_link ⇐ (a, b)
        foreach i = RP.num_of_hops, ··· , 1 do
            msg.position ⇐ i
            SEND_MSG(RP.nodes[i], msg)
reroute_next_hop ⇐ interface(b).reroute_next_hop
rerouting ⇐ True
```

Fig. 5.: Algorithm RP_SETUP1($(a, b)$)

```
/* called by a router on the RP, n */
(a, b) ⇐ msg.failed_link
foreach m ∈ C(T_n, n) do
    f ⇐ interface(m)
    if (a, b) ∈ ST(T_n, m) and b.seq[f] ∈ [a.seq[f], a.seq_end[f]] then
        f.affected ⇐ True
        f.affected_start ⇐ b.seq[f]
        f.affected_end ⇐ b.seq_end[f]
if msg.RP.num_of_hops > msg.position then
    reroute_next_hop ⇐ msg.RP.nodes[msg.position + 1]
else
    reroute_next_hop ⇐ msg.RP.next_hop
rerouting ⇐ True
```

Fig. 6.: Algorithm RP_SETUP2($msg$)

```
(dest_node, next_hops) ⇐ routing_table_lookup(pkt)
if rerouting = False then
    normal_forward(next_hops, pkt)
else
    valid_next_hops ⇐ ∅
    foreach n ∈ next_hops do
        f ⇐ interface(n)
        if f.affected = False then
            valid_next_hops ⇐ valid_next_hops ∪ n
        else if dest_node.seq[f] ∉ [f.affected_start, f.affected_end] then
            valid_next_hops ⇐ valid_next_hops ∪ n
    if valid_next_hops ≠ ∅ then
        normal_forward(valid_next_hops, pkt)
    else
        reroute(reroute_next_hop, pkt)
```

Fig. 7.: Algorithm NEW_FWD(pkt)

because of the link failure.

c. Modified forwarding operations

After the RP is setup, the router adjacent to the failed link will reroute all traffic routed to the failed link along the RP; other routers on the RP check traffic to be forwarded via affected interface and reroute affected traffic along the RP. The algorithm, *NEW_FWD*, is shown in Fig. 7.

Theorem 3 ensures the correctness of algorithms shown Figs. 6 and 7.

**Theorem 3.** *For a router on the RP, when link* $(a, b)$ *fails, if and only if* $(a, b)$ *is an edge of the* first level subtree *below an interface (assume* $a$ *is the upstream node of the link) and the sequence number of* $b$ *is within the affected range of* $a$ *for the interface, then* $b$ *becomes unreachable via that interface.*

*Proof.* If $b$ is within the affected range of $a$, then the failure of $a$ causes $b$ unreachable, i.e. the traffic to $b$ must pass $a$. Because our scheme uses the nearest FNH to reroute, $a$ does not have ECMPs to $b$, otherwise $a$ will reroute locally. So the failure of link $(a, b)$ causes $b$ unreachable via the interface.

If link $(a, b)$ is not an edge of the subtree, no traffic via the interface will pass the link. If link $(a, b)$ is an edge of the subtree but the sequence number of $b$ is not within the affected range of $a$, then the failure of $a$ will not affect traffic to $b$, i.e. there is a ECMP not including link $(a, b)$ from the root of the subtree to $b$. □

## B. Evaluation

We have evaluated our rerouting scheme on random topologies generated using BRITE topology generator [18]. We have generated topologies of 25-200 nodes with average degree of 4, 6 and 8. The link weights are uniformly distributed between 100 and 300. For each configuration we have generated 5 random topologies.

### 1. Number of signaling hops

First, we have measured the number of signaling hops of rerouting paths. (0 means local rerouting, 1 means the *exit* node is 1 hop away, and so on)

As shown in Fig. 8, the maximum number of signaling hops is 2 for all topologies we used. For topologies with average degree of 6 and 8, most RPs are local FNH.

The small value of number of signaling hops is beneficial for our rerouting scheme,

since it means short RP setup time, i.e. the response time of our rerouting scheme. For signaling hops of 0, the response time of our scheme is near zero. For signaling hops of $n$, the response time of our scheme is the time it takes to send a message to a router $n$ hops away in the network plus the processing time of routers.

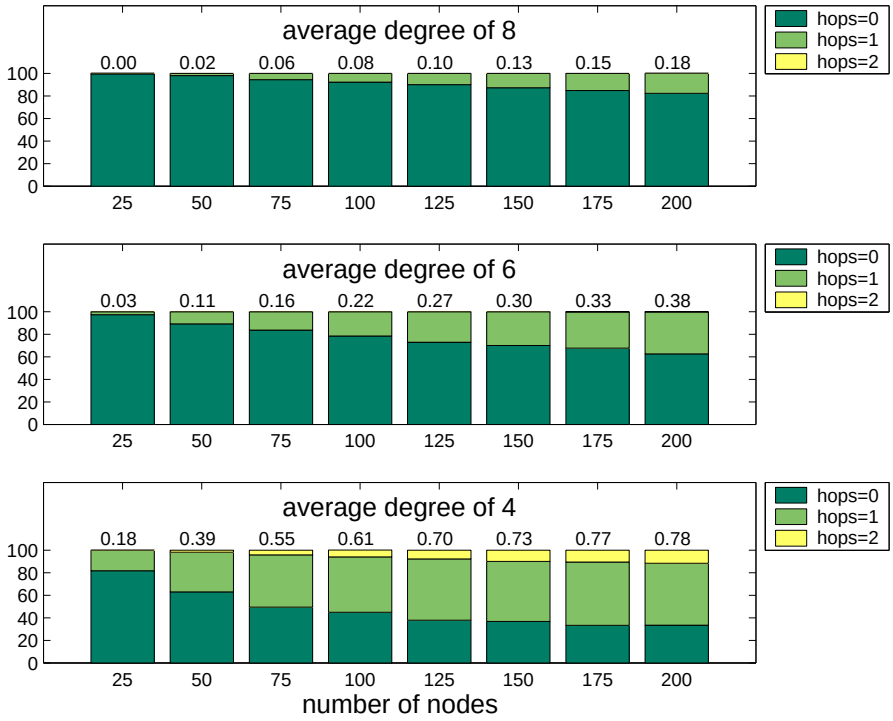


Fig. 8.: Percentage of RPs with different number of signaling hops and average number of signaling hops (the number above each vertical bar)

## 2. Path elongation

We have measured the distance elongation of our rerouting scheme compared to the optimal shortest path routing.

We define the *elongation ratio* (*elongation value*) as the ratio (difference) of the distance between an affected pair of nodes under our rerouting scheme and the optimal distance after global routing table recalculation.

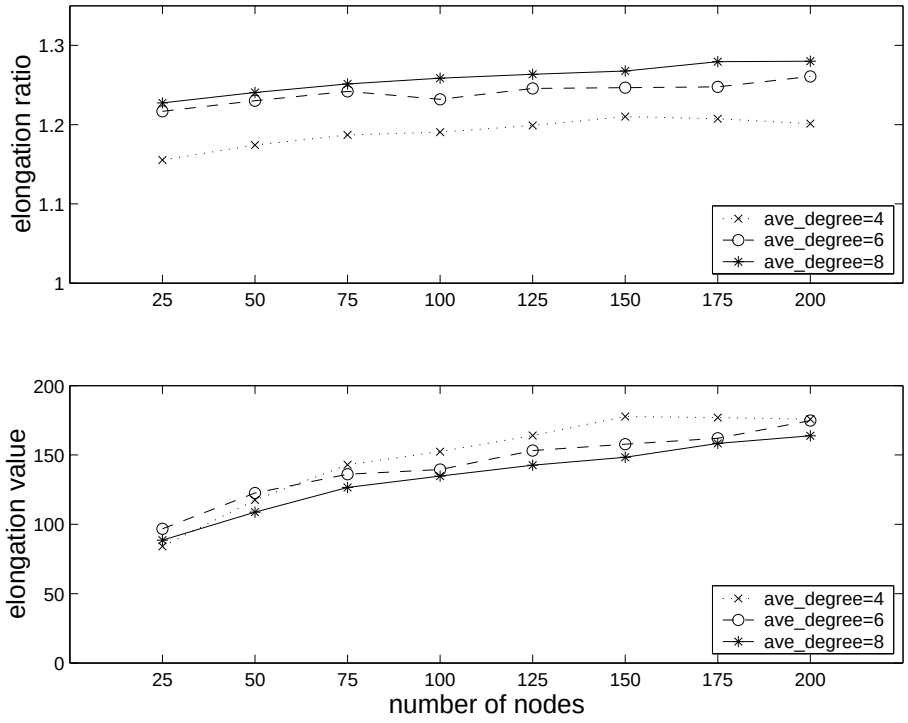


Fig. 9.: Elongation ratio and value compared to optimal paths

Since our main objectives are to achieve fast response to link failures and simple rerouting operations, our scheme does not prioritize the optimization of the rerouting path. But as we see in Fig. 9, the path elongation is not significant. While the average elongation ratio is about 1.2[1], the elongation value value is about 100 to 160, i.e. less than twice of the minimum link weight, 100. This means the average elongation of our scheme is less than two hops.

## 3. Complexity of algorithms

The most complex algorithm in our scheme is Algorithm *SEQ* shown in Fig. 2 that calculates sequence numbers for all nodes in each first level subtree of the local routing

[1]The elongation ratio is decided by the elongation value and the average distance between pairs. While the increase of average degree reduces the elongation value, it also reduces the average distance. And we can see it increases the elongation ratio as shown in Fig. 9

tree.

In a routing tree without ECMPs, the complexity is same as DFS, i.e. $O(|V|)$ where $|V|$ is the number of nodes in the routing tree.

In a routing tree with ECMPs the complexity can be estimated as $O(n|V| + n|V'||E'|)$, where $n$ is the maximum degree of the root node, $|V'|$ is the maximum number of nodes in a first level subtree that have more then one parent, $|E'|$ is the maximum number of unique edges traversed in a call of $DFS\_nearest\_ancestor$, which is at most $|V|$. $O(n|V|)$ represents the complexity of the DFS search procedures for all first level subtrees. $O(n|V'||E'|)$ represents the complexity of the calculation of all *nearest ancestors* that is loosely bounded by $O(n|V|^2)$. However, in a real-world routing tree the number of nodes that have more than one parent($|V'|$) is much smaller than $|V|$; accordingly, $|E'|$ is in order of the diameter of the topology Therefore the complexity of the algorithm is small. Moreover, using incremental implementation by storing the $deferred\_dfs\_queue$ with the nodes in first level subtrees, the complexity of this algorithm can be further reduced.

## C. Future work

First, we plan to use more than one RPs to split rerouted traffic for load balancing. Second, we plan to enhance our algorithm to detect multiple link failures and node failures. In such cases we should avoid fast rerouting and rely on the original IGP convergence mechanism.

## D. Conclusions

We have proposed a Fast Rerouting scheme for OSPF/IS-IS networks in this paper. We have developed efficient algorithms for calculation of Rerouting Path, and identi-

fication of affected traffic. The rerouting operation for each packet is comparable to basic IP forwarding. Simulation results show that, assuming there is one link failure at a time which accounts for a large portion of network failures, our scheme achieves fast response to link failures and the path elongation compared to optimal path is not significant.

# CHAPTER III

# FRACTIONAL MULTIHOMING ROUTE CONTROL

## A. Introduction

When a network has more than one external link to one or more ISPs, it is said to be multihomed [11]. Multihoming has been traditionally used by stub networks for improving service availability. In recent years, Multihoming Route Control (MRC) technology [19] has been employed by multihomed stub networks to improve their Internet access performance. In the discussion of multihoming route control, multihoming specifically means that a stub network connects to multiple ISPs. A MRC device chooses the best ISP(s) for traffic to (and from, for NAT [20] based of MRC) an IP address prefix that is usually equivalent to a destination network according to measured qualities of alternate paths via different ISPs. Measurement based analysis of the benefits of multihoming [15] showed that MRC may improve Internet access performance significantly for both enterprises and large data centers.

There are two types of multihoming: NAT(Network Address Translation) [20] based and BGP [12] based. NAT based multihoming is usually used by small to medium size stub networks because it does not require the stub network to have an independent IP block and maintain a BGP router. BGP based multihoming is usually used by a large stub network that has an independent IP address block(s) and maintains a BGP router. Accordingly, multihoming route control devices can be classified into NAT based and BGP based categories, see [21] for a survey.

In this work, we study MRC for large stub networks that employ BGP based multihoming. We also assume the stub network advertises its IP address block(s) to all its ISPs. While it is possible to control the incoming traffic direction through

selective advertisement of addresses to different ISPs, such a control is only possible over longer timescales and may leave the stub network vulnerable to network failures. In this case, the stub network can send outgoing traffic via either of its ISPs, but it cannot control which ISP the ingress traffic comes from. In summary, the task of MRC for BGP based multihoming is to map egress traffic onto available paths provided through BGP based multihoming. This is not a problem when multihomed receivers also have MRC devices deployed since all traffic is controlled by the MRC devices of the originating stub networks. When the receivers are single-homed or do not employ MRC, we can assume the ingress traffic is statically routed. In this case, MRC can still improve performance of stub networks that deploy MRC, but the improvement is limited to optimizing egress traffic. However, the control of egress traffic alone is good enough to attract content providers to deploy MRC, since the volume of their egress traffic is usually larger than the volume of their ingress traffic. For networks where MRC of ingress traffic is desirable, NAT based MRC should be considered which is beyond this work.

Multihoming route control is usually done in a distributed manner: each stub network adaptively changes the ISP of its traffic to (and from, for NAT based MRC) a destination network according to its own view of the quality of alternate paths via different ISPs. When (1) the controlled traffic accounts for a significant portion of the total load on the bottleneck links and (2) the traffic controlled by multiple MRC devices shares bottleneck links, the MRC by different stub networks may interact with each other. In such situations, greedy MRC approaches may cause oscillations.

Beginning in this chapter, we start to study the possible oscillations of MRC and propose new MRC schemes to avoid such oscillations. In this dissertation, we focus on MRC among a group of multihomed stub networks, for example, the networks of branches of an enterprise that are multihomed and exchange considerable traffic

regularly among themselves. Since access links of stub networks are more likely to be the bottlenecks along end to end Internet paths, MRC among a group of multihomed stub networks is more likely to cause oscillations. Although we focus on MRC among a group of multihomed stub networks, the distributed MRC scheme we proposed in Chapter V can be also used as a general MRC method.

In this chapter, we explain the problem we want to solve, give the fractional MRC framework of our solutions and explain the method of our study. In the next three chapters, we will propose MRC schemes under the framework and study their performance and implementation issues. In Chapter IV, we propose an optimal routing based global coordination method for MRC among a group of multihomed stub networks. In Chapter V, we propose a user-optimal routing based distributed MRC approach that can be used for both MRC among a group of multihomed stub networks and more general MRC of stub networks. While in Chapter IV and V, we assume traffic is UDP type traffic whose traffic volume does not change because of network condition changes, in Chapter VI, we study MRC of highly dynamic TCP traffic consisting of TCP flows of different sizes.

The remaining of this chapter is organized as follows. In Section B, we describe an Internet measurement experiment we have done to study the quality differences between alternate paths through multihoming and present the results. In Section E, we propose the framework of fractional MRC to avoid oscillations of MRC. This framework is an extension of traditional MRC and is the foundation of our MRC study in next three chapters. In Section C, we describe the simulation methods we used in this chapter and the next two chapters. In Section D, we introduce two greedy MRC algorithms and show the possibility of oscillations of these two MRC approaches. In Section G, we discuss related work on MRC. Conclusions are drawn in Section H.

## B. Measurement study of qualities of alternate paths provided through multihoming

### 1. Measurement method

We conducted measurements on the Internet to study the dynamics of the quality differences of the alternate paths available through multihoming. This is important for designing an effective MRC scheme.

The experiment consisted of the following steps.

(1) We identified a number of multihomed stub networks by analyzing the AS graph generated using multiple BGP RIBs (Routing Information Bases) from the Route Views website [22]. The method is similar to the one used in [3].

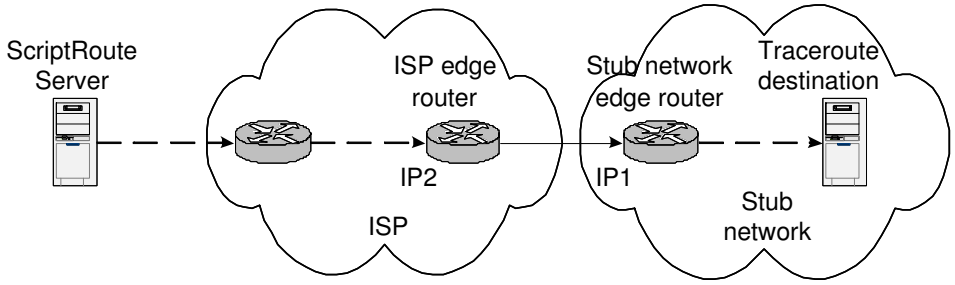


Fig. 10.: Traceroute measurement

(2) Based on the IP address ranges of Autonomous Systems (ASes) from the BGP RIBs, we use the traceroute utilities on a number of Scriptroute servers [23] to identify the edge routers of these stub networks. Traceroute utility generates a list of IP addresses of routers along the forwarding path from the originating host to the destination host. The address of a router along the path is usually the IP address of the incoming interface of the traceroute packet. For example, in a traceroute measurement shown in Fig. 10, we get an IP address of the ISP edge router, IP2, and an IP address of the stub network edge router, IP1. Usually IP1 is the last IP address belonging to the ISP along the list of addresses generated by the traceroute measurement.

(3) We measure alternate paths to a stub network by "pinging"(using "traceroute" (UDP) packets) the IP1s of different ISPs of the stub network. We measured the alternate paths to one stub network from a number of Scriptroute servers.

In this way, we measure the alternate "round trip paths" from a Scriptroute server to a multihomed stub network. We compare the qualities of two "round trip paths" by calculating the differences of the average RTTs and average loss rates of the "ping" packets on the two paths. Without access to hosts in the remote stub networks, it is hard to measure quality of one way paths from the Scriptroute servers to remote multihomed stub networks. That is the reason we conduct measurement of quality of round trip paths. Although it is not a direct measurement of quality difference between alternate "one way paths" through multihoming, in most cases, it still reflects the dynamic quality differences of alternate paths through multihoming.

In our measurement, we find that the RTTs and loss rates of the "ping" packets to the stub network edge routers sometimes fluctuate significantly and have a daily pattern. We suspect that it is because the access links of the remote networks are busy, so the characteristics of the paths may be concealed by the queuing delay and packet losses on the access links. To get rid of the queuing delay on access links of the stub networks, we calculate the RTT of a "route trip path" from a Scriptroute server to a stub network edge router as $RTT1_{min} + RTT2 - RTT2_{min}$, where $RTT1$ is the RTT of a "ping" packet to the stub network edge router, e.g. IP1 in the measurement shown in Fig. 10, $RTT2$ is the RTT of a "ping" packet to the ISP edge router, e.g. IP2 in the measurement shown in Fig. 10. The minimum of $RTT1$ and the minimum of $RTT2$ are expected to be the propagation delay (without queuing delays on links) of the "round trip paths". We use the loss rates of the packets to ISP edge routers as the loss rates for the "round trip path".

## 2. Measurement results

We carried out measurement for more than 2 days in June, 2004. Here, we give some results from our measurement of 33 pairs of alternate paths in the United States.

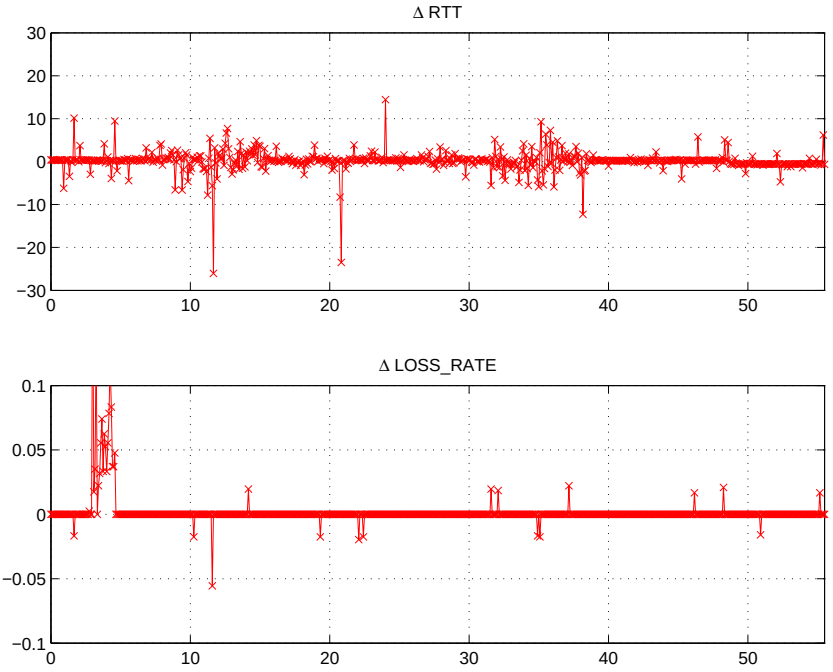


Fig. 11.: Average RTT differences and loss rate differences of pair 1 of alternate paths. (over 5 minute durations, Y axis: $\Delta RTT$ in milliseconds and $\Delta loss\_rate$, X axis: time since the start of measurement in hours)

Figs. 11, 12 and 13 show the RTT differences and loss rate differences of 3 pairs of alternate paths. The RTTs and loss rates are averaged over 5 minute durations. From the figures, we see significant performance differences between a pair of alternate paths for both RTTs and loss rates. The differences change over time. A better path may become worse some time later. The RTT differences persist over both small time scales (e.g. 5 minute) and over long time scales (e.g. a few hours). Similar differences are observed for more than half of the paths we measured. These observations indicate: (1) route control is possible to improve the performance of

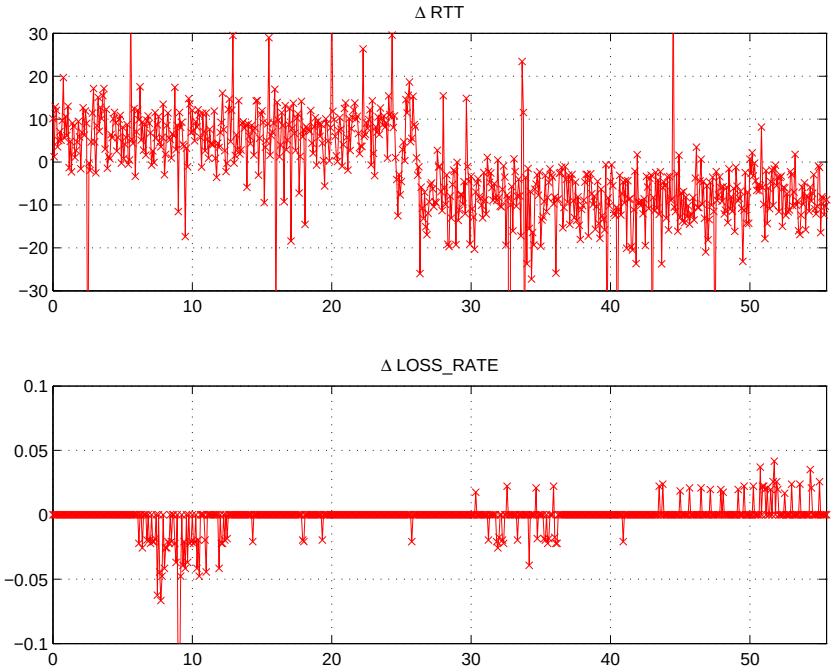


Fig. 12.: Average RTT differences and loss rate differences of pair 2 of alternate paths. (using same parameters as Fig. 11)

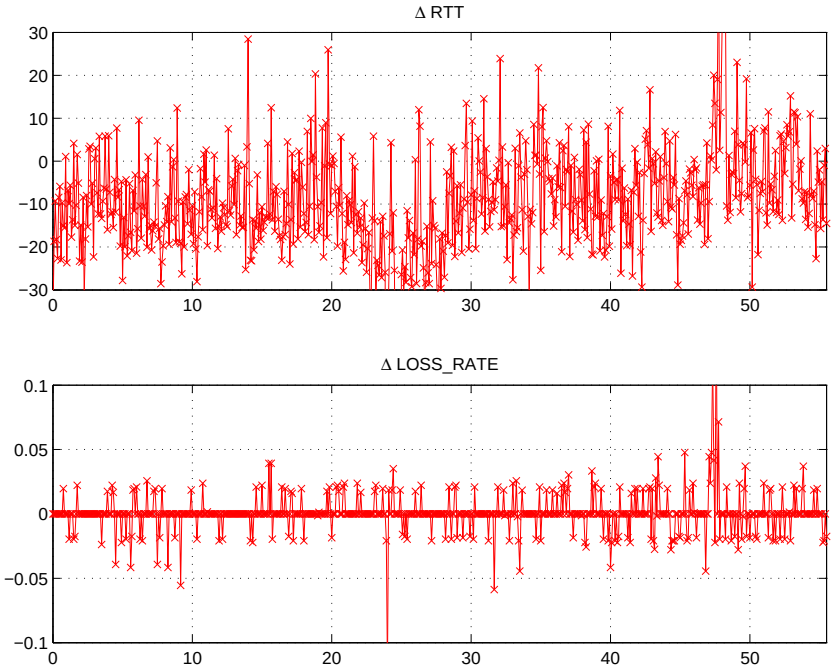


Fig. 13.: Average RTT differences and loss rate differences of pair 3 of alternate paths. (using same parameters as Fig. 11)

multihomed network significantly, and it should be implemented in a dynamic manner; (2) both small time scale route control and large time scale route control have the potential to improve the routing of multihomed stub networks.

## C. Simulation methods

In this chapter and next two chapters, Chapter IV and Chapter V, we use flow level simulations to study the performance of different MRC algorithms. In Chapter VI, we use network simulator, ns-2 [17] to study our MRC approach proposed in Chapter V. In this section, we describe the common methods we used in these simulations. Other methods will be discussed in corresponding chapters.

### 1. Topologies and path characteristics

In our simulations, we make some simplifications to network topologies. We assume that the traffic controlled by the stub networks accounts for only a small part of the total traffic on any link of the backbone networks which is usually true. Under this assumption, the MRC of traffic of the stub networks won't affect the quality (or level of load) of backbone links. Therefore, we can abstract a network path between the ISP edge routers of two stub networks as a directed "virtual" link with given quality that may change overtime. We also abstract paths from ISP edge routers of a stub network to an Internet destination as "virtual" links. Based on previous analysis of heavy-tailed Internet traffic distributions, e.g. [24], we apply fractional MRC only for a number of Internet destinations that account for a large amount of total egress Internet traffic of a stub network. Since the routing inside a stub network is not relevant to our work we also abstract each stub network as a node.

Therefore, a network we are studying consists of: (1) nodes representing stub

networks in the group; (2) nodes representing ISP edge routers of the stub networks; (3) nodes representing a number of top Internet destinations of the stub networks; (4) links representing access links of the stub networks; (5) virtual directed links between ISP edge routers of different stub networks. (6) virtual directed links between ISP edge routers of the stub networks and the Internet destinations. To simplify the simulation, we assume that stub networks in the group are multihomed to same number of ISPs and each has same number of representative Internet destinations.

Depending on whether the group of networks multihome to the same set of ISPs, there are two types of topologies for traffic among the stub networks: (1) Symmetric topology: When all the nodes multihome to the same set of ISPs, the path from a stub network $i$ via ISP $k$ to any other stub network, say $j$, normally reach $j$ via ISP $k$. Thus, the alternate paths between these two stub networks are "parallel", they merge only inside the stub networks we considered. (2) Asymmetric topology: When nodes multihome to different set of ISPs, the alternate paths from a stub network to another stub network are not necessarily "parallel". Two paths to a stub network may merge in one ISP of the stub network or in other AS between the two stub networks. This is decided by the BGP relationship of ASes between the two stub networks. We generate asymmetric paths for inhouse traffic from a node, say $i$, to a node, say $j$, by connecting each ISP of $i$ to a randomly selected ISP of $j$ . Each ISP of $j$ has equal probability to be selected.

Fig. 14 shows the partial topology of a "4x2" network ("AxB" means the topology has $A$ stub networks and each stub network has $B$ ISPs) . The ISPs of different stub networks are different, i.e. asymmetric topology for inhouse traffic. For clarity, we only draw paths from stub network 2, 3, and 4 to stub network 1. Other paths among the stub networks and paths to and from Internet destinations are ignored.

We generate path characteristics as follows: we randomly map the stub networks

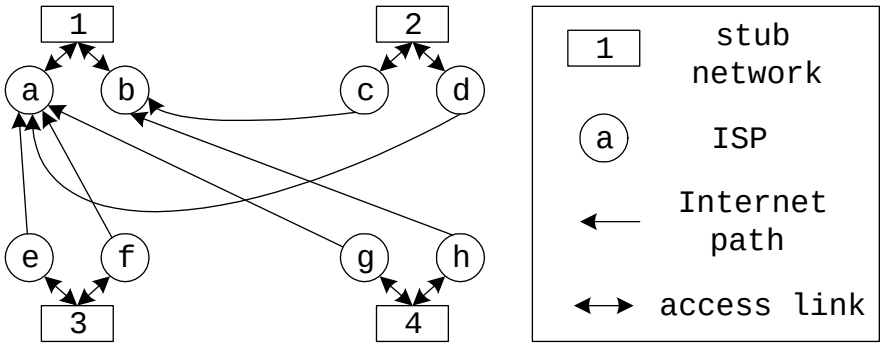


Fig. 14.: Topology consisting of multihomed stub networks, the edge routers of their ISPs and the paths among them

onto 17 major cities of the United States. We generate propagation delay of a path for inhouse traffic between two ISP edge routers by multiplying half of the RTT between the the two cities on the AT&T backbone [25] by a random factor, uniformly distributed from $\alpha - \delta$ to $\alpha + \delta$, where $\alpha$ and $\delta$ are constant for each topology. $\alpha$ defines the mean end to end delay of the path. We set $\alpha = 1.3$ in our simulations. We call $\delta$ as the *diversity factor*, because it reflects the average performance difference between alternate paths via different ISPs. We use different diversity factors from 0.2 to 0.6 in our simulations. If more than one stub network is mapped to the same city, we assigned the propagation delay of a path for inhouse traffic between these stub networks as 1.5 millisecond multiplied by the above random factor.

### 2. Traffic demands

Traffic demands of networks are usually not available to the public. Therefore, in most of our simulations, we generate demand matrices using a simplified version of the "gravity model" [26]. This model is shown to work well in traffic matrix estimation [27]. In Chapter IV, we also use traffic matrices constructed from Internet traces. We will describe the method in that chapter.

Using the simplified "gravity model", to generate inhouse traffic between nodes, we assign two uniformly distributed random numbers to each node $i$, $O_i, D_i \in [0, 1]$. Then the traffic demand from node $i$ to node $j$ is $\alpha O_i D_j$, where $\alpha$ is a parameter, $O_i$ and $D_i$ model how active node $i$ is as a sender and as a receiver.

Similarly, to generate Internet traffic, we assign two uniformly distributed random numbers to each node $i$, $O'_i$ and $D'_i$, where $O'_i, D'_i \in [0.5, 1]$ in Chapter IV, $O'_i, D'_i \in [0, 1]$ in Chapter V. The two choices do not have significant impact on our conclusions. The egress and ingress Internet traffic of the node are $\beta O'_i$ and $\beta D'_i$, where $\beta$ is a parameter, $O'_i$ and $D'_i$ model how active node $i$ is in sending and receiving Internet traffic. In Chapter IV, the above aggregate of Internet traffic is enough for our simulations. When we consider MRC of egress traffic to Internet destinations in Chapter V, the egress Internet traffic is randomly distributed to 5 Internet destinations(each has a random weight uniformly distributed in [0,1]); the ingress Internet traffic is randomly distributed on all ingress access links(each has a random weight uniformly distributed in [0,1]).

In our simulations we choose $\alpha$ and $\beta$ to make the expected volume of inhouse traffic 50% of the total traffic.

### 3. Queuing delay models

We wrote a flow level simulator for our simulations in this chapter and next two chapters. In flow level simulations, queuing delays and loss rates on links are calculated according to queuing models.

In this chapter and Chapter IV, like most previous related work [16, 28], we consider only queuing delays. We use following two representative queuing models: 1) M/D/1 queuing model [29] : fixed sized packets arrive according to Poisson process; 2) P/M/1 queuing model [30] : exponentially sized packets arrive according to Pareto

process. The queuing delay function of M/D/1 is: $q(x) = \frac{0.5}{\mu - x} + \frac{0.5}{\mu}$, where $\mu$ is the link capacity, $x$ is the load on the link. In our simulations, we assume packet size is 1000 bytes. Because there is no close form of queuing delay function for P/M/1 queue, we use a piece-wise linear function to approximate the function according to the numerical result of [30] (with $\beta = 1.5$). Because both of these two models are defined in $x \in [0, 1)$ while the demand in our simulation may exceed the capacity, similar to previous work [28], we extend the queuing delay of the two models linearly for utilization over 99%.

## D. Basic multihoming route control and possible oscillations

### 1. The greedy method: best-path-only multihoming route control

According to literature on multihoming route control [31, 32], most MRC devices greedily choose the "best" path for traffic to a destination prefix according to measurement of quality of alternate paths. Since this algorithm is a special case of the next algorithm we will study, we express it using the same algorithm as shown in Fig. 15. For this algorithm, $H_1 = 1, H_2 = 0$ (Note that $H_2$ is not used in this algorithm and we set $H_2 = 0$ here to make "delay $x < 0$" always false). In this algorithm, each node selects a path among alternate paths according to path quality measurements after every $T$ seconds, where $T$ is a random number uniformly distributed in $[0.5T_0, 1.5T_0]$ ($T_0$ = is the mean period over which route control decisions are made.) to avoid update synchronization that may increase oscillation probability. $H_1$ specifies that only the path with minimum delay can be used.

An implicit assumption of this type of greedy route control is that the quality of the "best" path will not get much worse after traffic being switched to it from alternate paths. However, this assumption does not always hold. When the assumption is not

**while** *true* **do**

  **foreach** *destination, d* **do**

    $x_{min} \leftarrow$ min delay of alternate paths to $d$ ;

    equally split traffic to $d$ along paths with delay $x$: $x \leq H_1 \cdot x_{min}$ or $x \leq H_2$ ;

  wait for random time $T \in [0.5T_0, 1.5T_0)$ ;

Fig. 15.: Basic multihoming route control algorithm for a MRC device

true, route control may cause oscillations. For the routing of traffic among a group of multihomed networks, access links may have high utilization and traffic controlled by MRC devices may account for non-trivial volume relative to the total capacity of an access link. Thus this assumption could be easily violated. Simulation results in Section 3 will illustrate the affect of possible oscillations in the greedy multihoming route control approach.

### 2. A less greedy method: threshold based load-balancing

The fundamental reason for the above MRC approach to cause oscillations is its "coarse" control over traffic in a network where access links have limited capacity. Naturally, a less greedy MRC approach that has finer control of traffic routing may lower the probability of oscillations.

A possible such approach is to split traffic among a few paths that have relatively better quality according to the algorithm shown in Fig. 15. The meanings of $t, T_0$ are the same as described in the best-path-only algorithm. Each round the MRC device chooses all the paths satisfying some QoS requirement: "end_to_end_delay $< H_2$" or "$< H_1 \cdot$ minimum_end_to_end_delay" ($H_1 = 1.4$, $H_2 = 50ms$ in our simulations) and begin to equally split traffic among these paths.

This approach is less greedy and has a finer control over traffic. However, we have found that this approach may still cause oscillations in some situations as we show in Section 3.

## 3. Possible oscillations of basic multihoming route control method

We study the performance of above basic multihoming route control methods using simulations. The simulation configurations are given in Section C. For comparison, we also study the corresponding performance of static load balancing( splitting traffic to one destination along all alternate paths).

In each simulation, we initialize the routing allocations according to the static load-balancing approach and evaluate the routing dynamics of routing approaches from the 100th second to the 1000th second. For each MRC approach, we plot the time average and variance of mean end to end delay of inhouse traffic under different network configurations. The average end to end delay of inhouse traffic is a metric representing the overall performance of the routing approach. The variance of end to end delay of inhouse traffic is a metric reflecting the stableness of the routing approach. Because we assume the network situation and demand matrix do not change in the simulations, a routing approach should ideally get 0 variance after it converges.

Figs. 16, 17 and 18 are the simulation results for an asymmetric topology of 8 stub networks where each stub network connects to 3 ISPs, makes a routing decision every second ($T_0 = 1$), and with the P/M/1 queuing model.

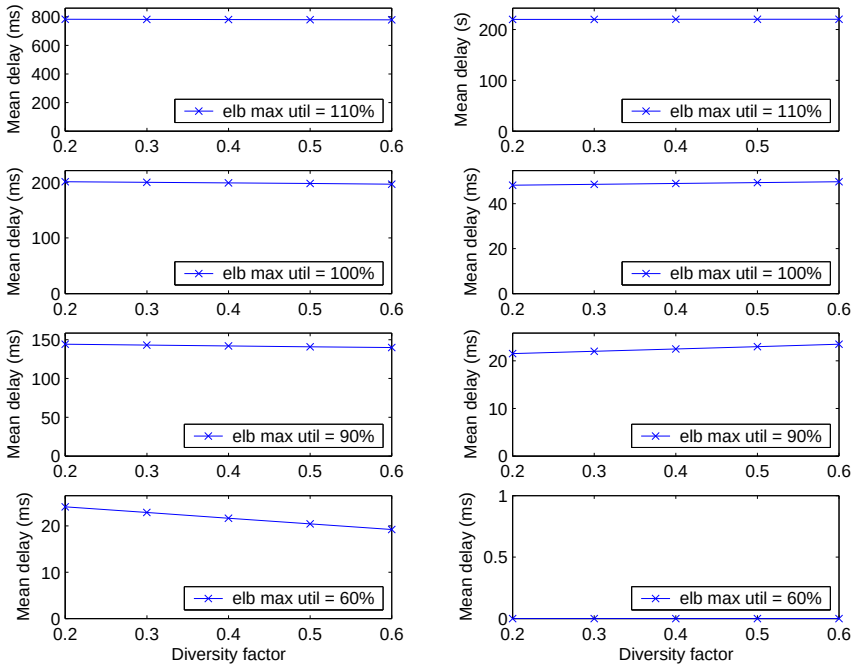


Fig. 16.: Average delay and delay variance of best-path-only MRC v.s. utilization and path quality diversity factors (asymmetric topology, Pareto type traffic)

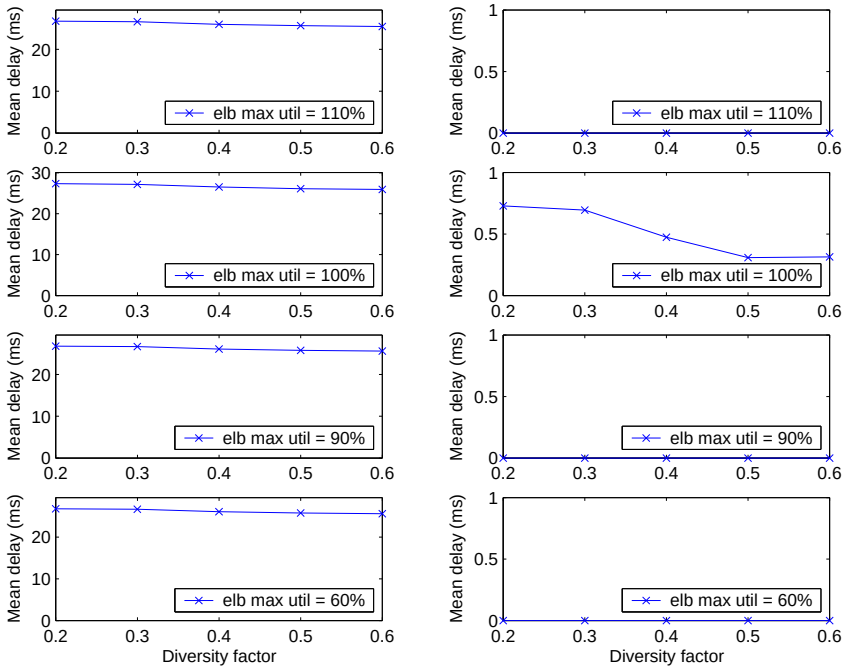


Fig. 17.: Average delay and delay variance of threshold based MRC v.s. utilization and path quality diversity factors (asymmetric topology, Pareto type traffic)

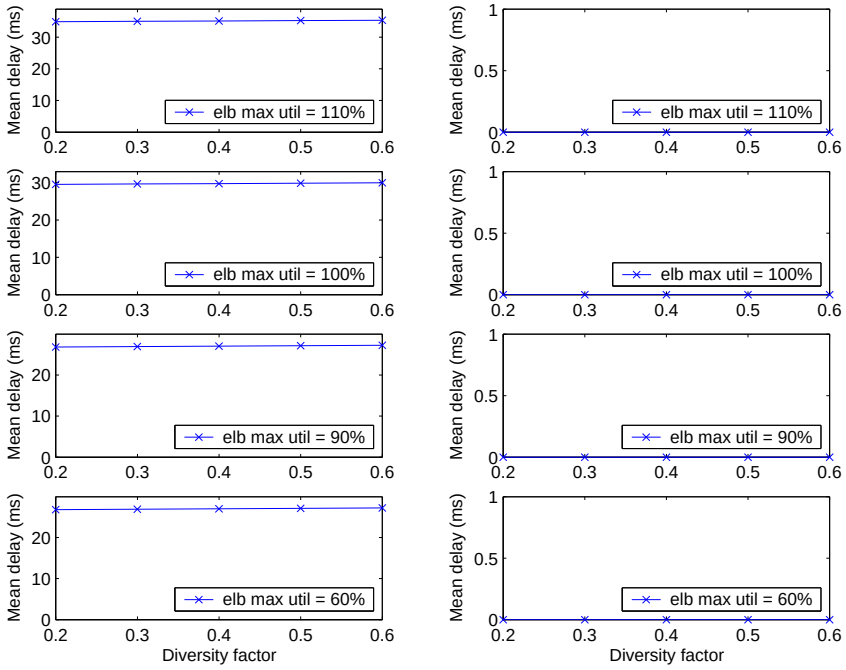


Fig. 18.: Average delay and delay variance of static load-balancing v.s. utilization and path quality diversity factors (asymmetric topology, Pareto type traffic)

To show the cause of the delay variances, we plot some details of one simulation point in Fig. 16. We plot the time sequences of (1) average virtual delay and average loss rate; (2) routing vector for from one site to one destination; (3) end to end virtual delay of paths from the site to the destination, in Figs. 19 to 21, for following configuration: diversity factor = 0.4; traffic matrix is scaled such that the maximum utilization for static load-balancing is 100%; asymmetric topology; Pareto type traffic.

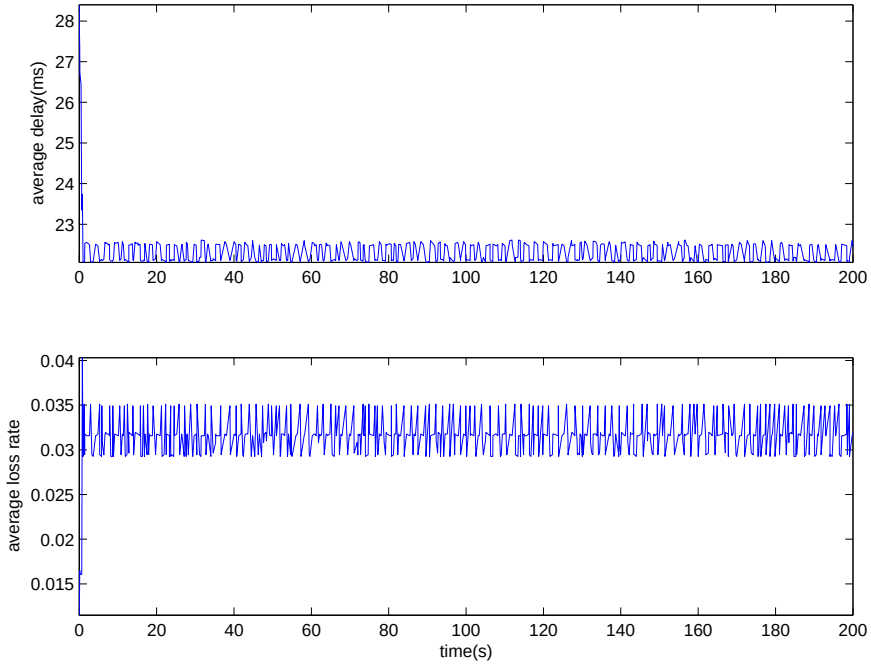


Fig. 19.: Average delay and average loss rate of best-path-only MRC

As we can see from these figures:

1. When average utilization equals 60% or higher, the variances of delays along time of both best-path-only MRC and the threshold based MRC are larger than $10^5$, which indicates that there are severe oscillations. At the same time the traffic experiences larger mean delay compared to static load-balancing.

2. When average utilization equals to 40%, the threshold based MRC can avoid majority of oscillations (indicated by very small delay variance) and achieves

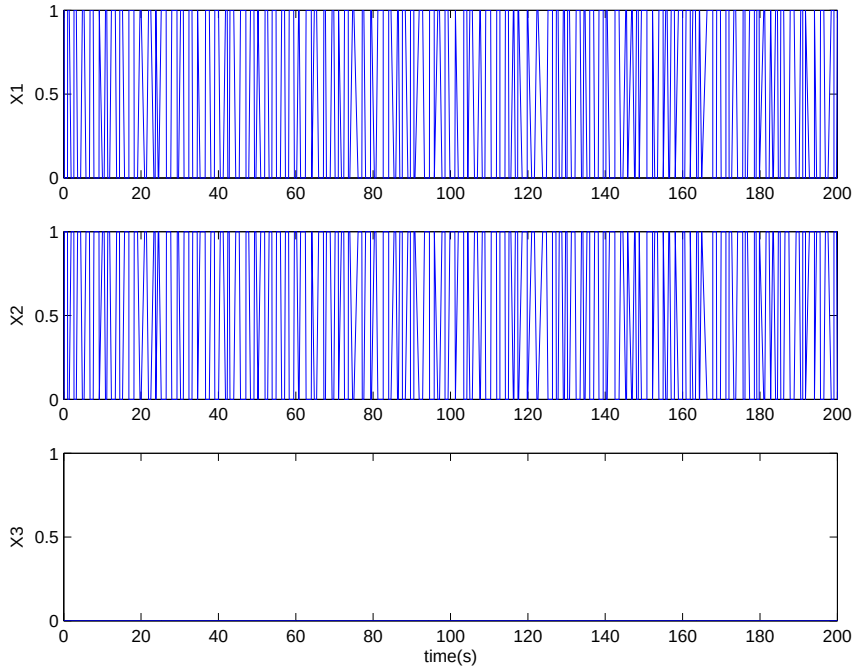


Fig. 20.: Routing vector for one pair of source and destination in best-path-only MRC

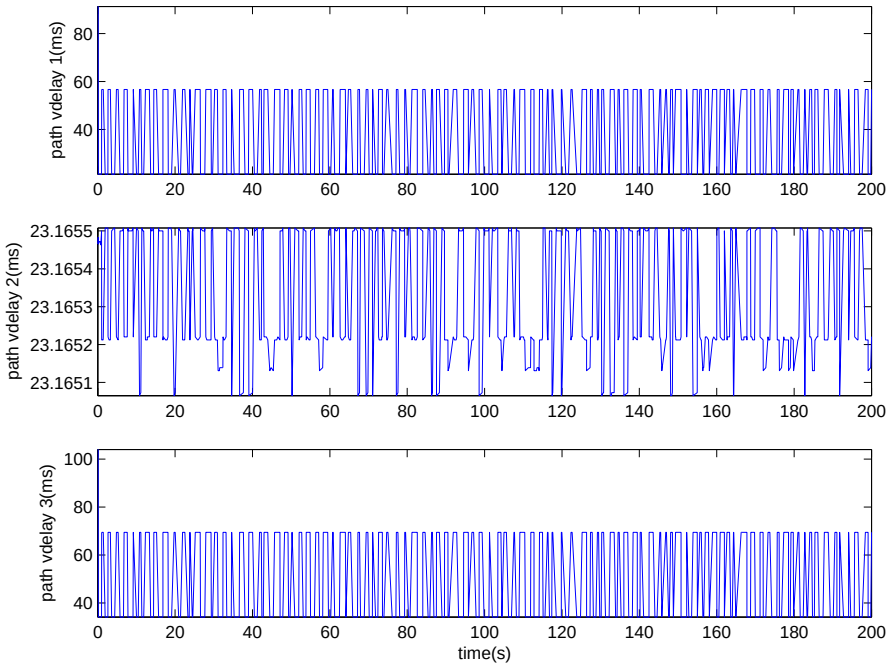


Fig. 21.: End to end virtual delay for one pair of source and destination in best-path-only MRC

smaller delay than static load-balancing, which means the approach can exploit the diversity of alternate paths.

3. For the same average utilization of 40%, the best-path-only MRC cause some oscillations (indicated by medium delay variance) and achieve larger average delay than the other two approaches.

4. The best-path-only MRC do achieve smaller average delay when path quality diversity factor becomes larger when average utilization is 40% and oscillations are not severe(medium delay variance along with time). The average delay for threshold based MRC does not change as the diversity factor changes because it ignore path quality changes smaller than the threshold.

Simulation results for symmetric topology and Poisson traffic are shown in Figs. 22 to 30. Similar results are observed. In summary, using greedy MRC approach for traffic among a group of multihomed stub networks may cause oscillations. And a better approach is needed for traffic routing among a group of multihomed stub networks.

## E. Framework of fractional multihoming route control

Existing BGP based MRC schemes usually work as follows [19]: A MRC device measures quality of alternate paths from the local stub network to an IP prefix via different ISPs. Based on the measurement results, the MRC device directs a local BGP border router to select an ISP for traffic to that IP address prefix. Because BGP uses single route for an address prefix at any time, MRC is also restricted to use a single route. However, the coarse granularity of control of single path MRC increases possibility of oscillations.

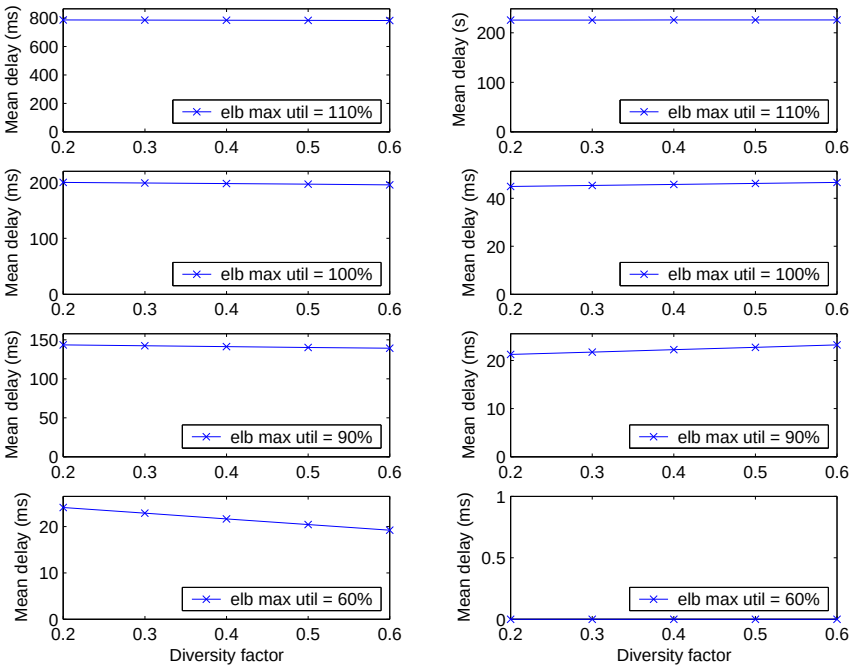


Fig. 22.: Average delay and delay variance of best-path-only MRC v.s. utilization and path quality diversity factors (symmetric topology, Pareto type traffic)

While it is critical for ISP networks to use single path for routing of inter-domain traffic to make BGP scalable, it is not a problem for stub networks to use multiple paths for their egress traffic. In this work, we assume stub networks can use multiple paths for egress traffic. The desired percentages of traffic to one destination network on each alternate path are decided by the MRC device. We call MRC that uses multiple paths simultaneously for traffic to one destination network as *fractional MRC*. In this chapter and Chapter IV and V, we study fractional MRC for UDP type traffic. In Chapter VI, we study MRC for TCP traffic, the proposed scheme is also under the Fractional MRC framework.

Fractional MRC device can move traffic from one path to another path smoothly. Proper implementations of fractional MRC can avoid oscillations, as we will show in in Chapter IV and Chapter V and Chapter VI. Fractional MRC is a form of multipath

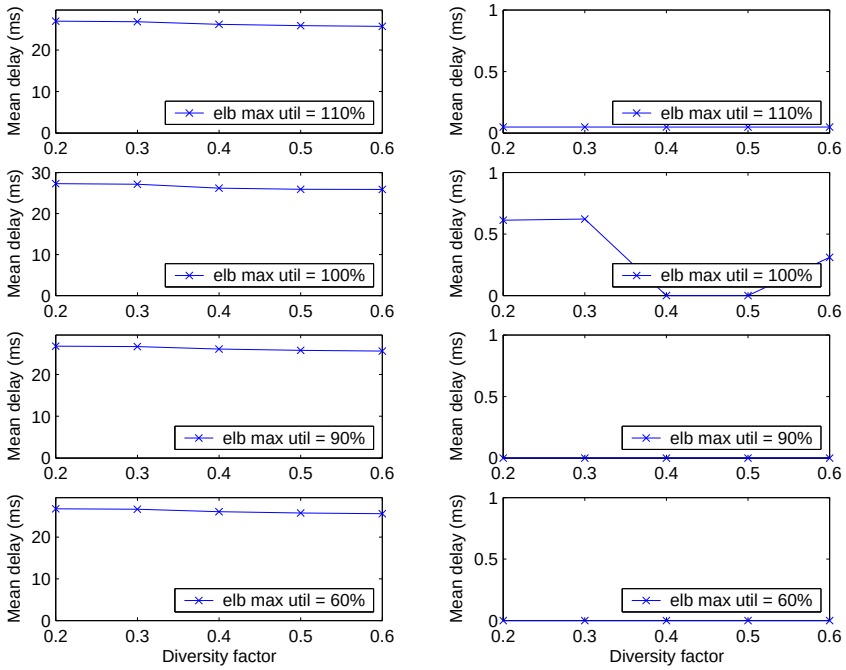


Fig. 23.: Average delay and delay variance of threshold based MRC v.s. utilization and path quality diversity factors (symmetric topology, Pareto type traffic)

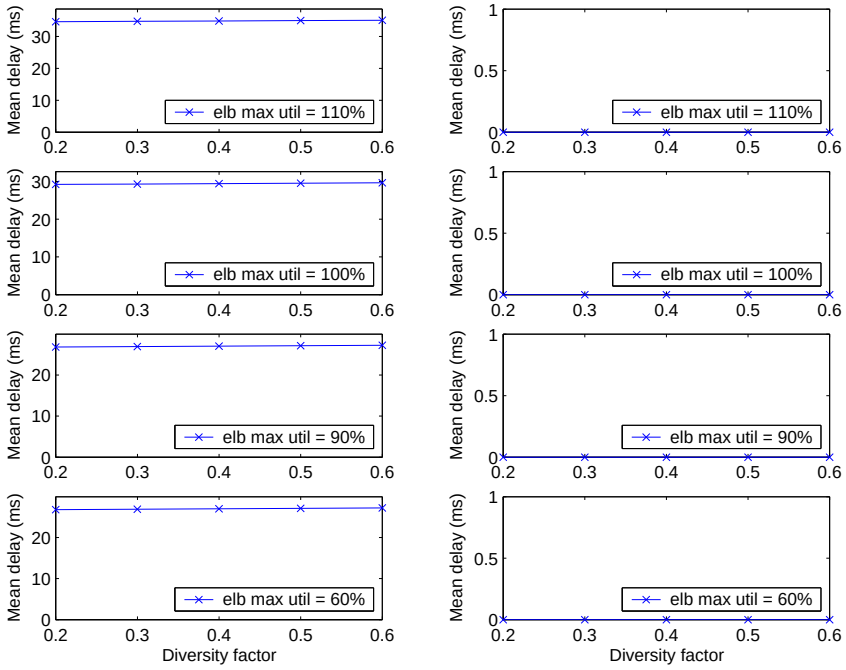


Fig. 24.: Average delay and delay variance of static load-balancing v.s. utilization and path quality diversity factors (symmetric topology, Pareto type traffic)

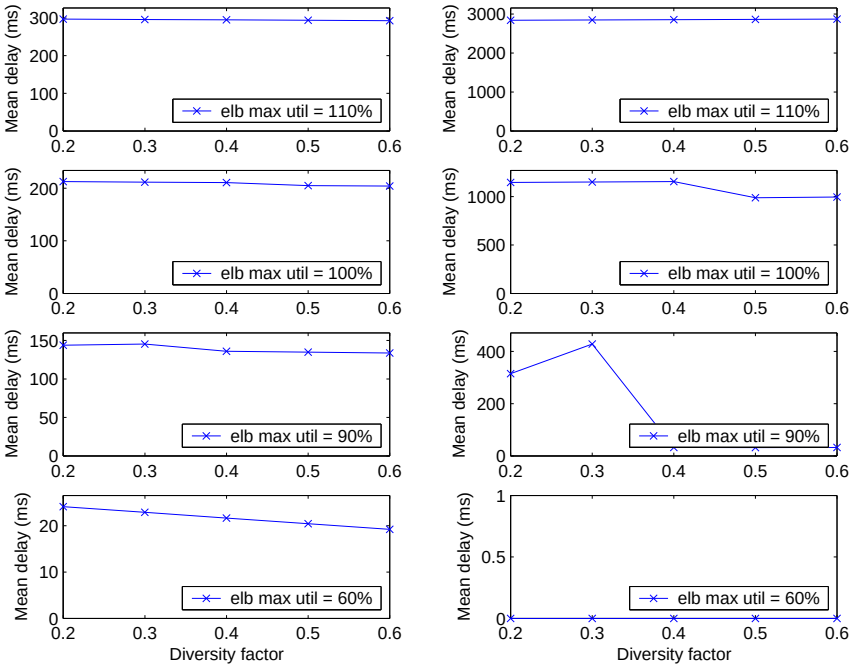


Fig. 25.: Average delay and delay variance of best-path-only MRC v.s. utilization and path quality diversity factors (asymmetric topology, Poisson type traffic)

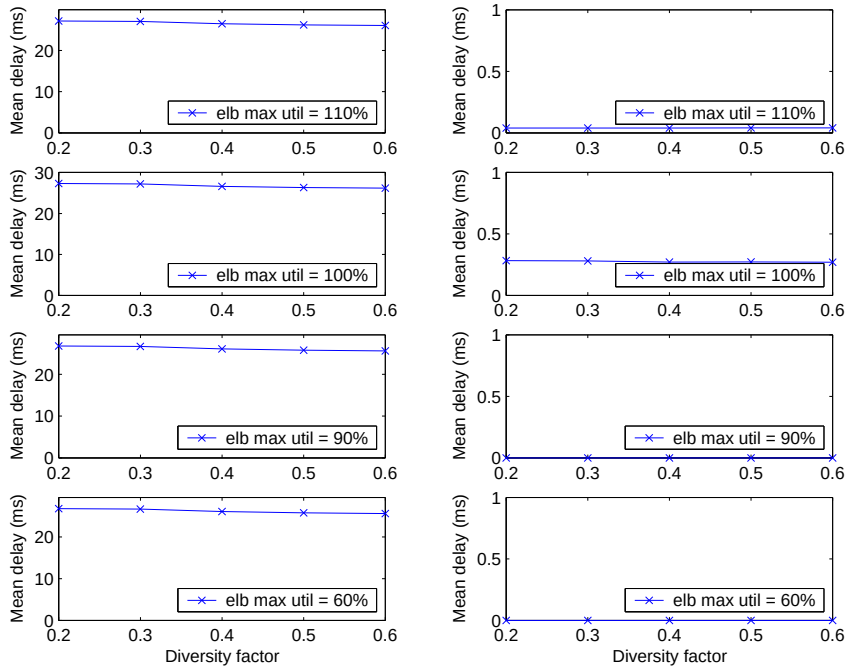


Fig. 26.: Average delay and delay variance of threshold based MRC v.s. utilization and path quality diversity factors (asymmetric topology, Poisson type traffic)

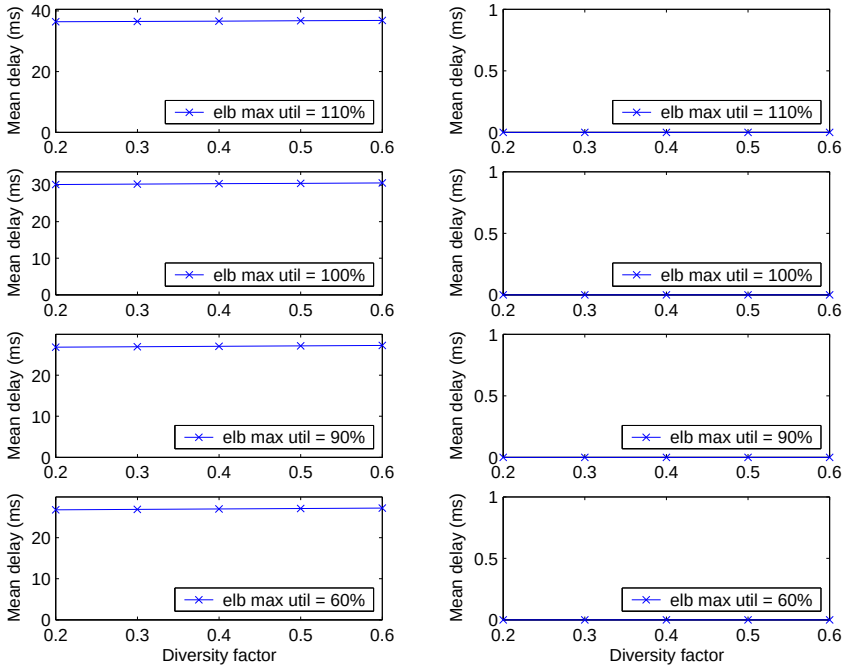


Fig. 27.: Average delay and delay variance of static load-balancing v.s. utilization and path quality diversity factors (asymmetric topology, Poisson type traffic)

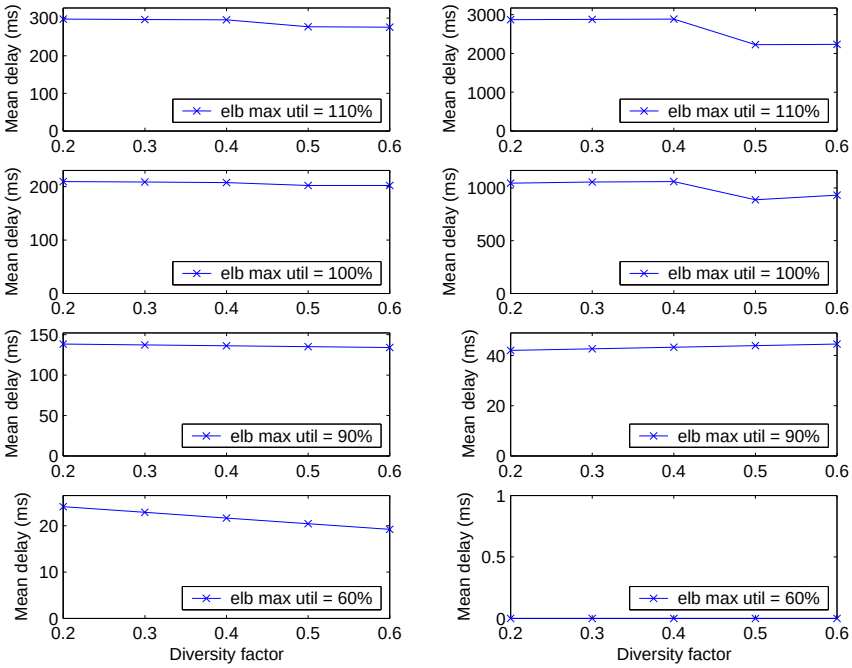


Fig. 28.: Average delay and delay variance of best-path-only MRC v.s. utilization and path quality diversity factors (symmetric topology, Poisson type traffic)

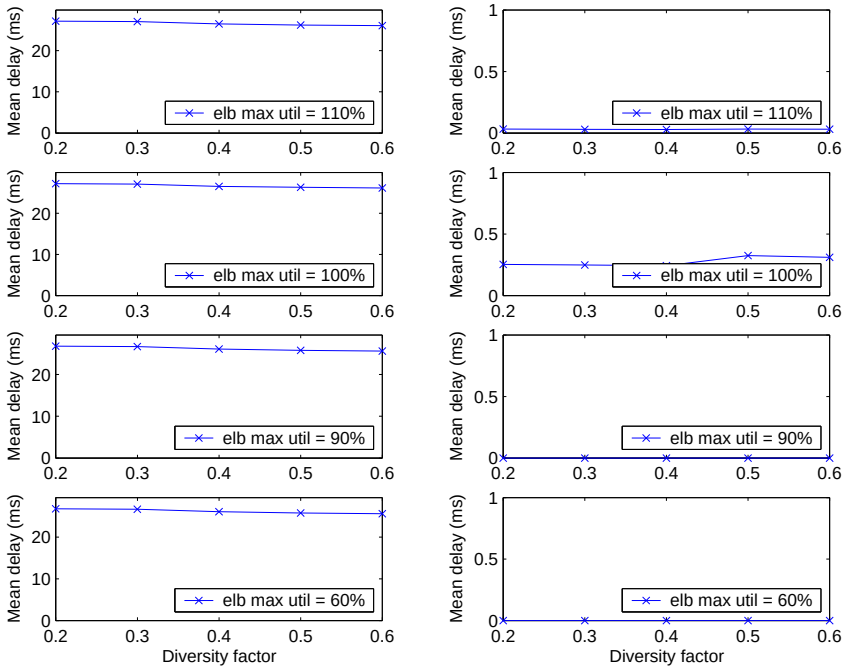


Fig. 29.: Average delay and delay variance of threshold based MRC v.s. utilization and path quality diversity factors (symmetric topology, Poisson type traffic)

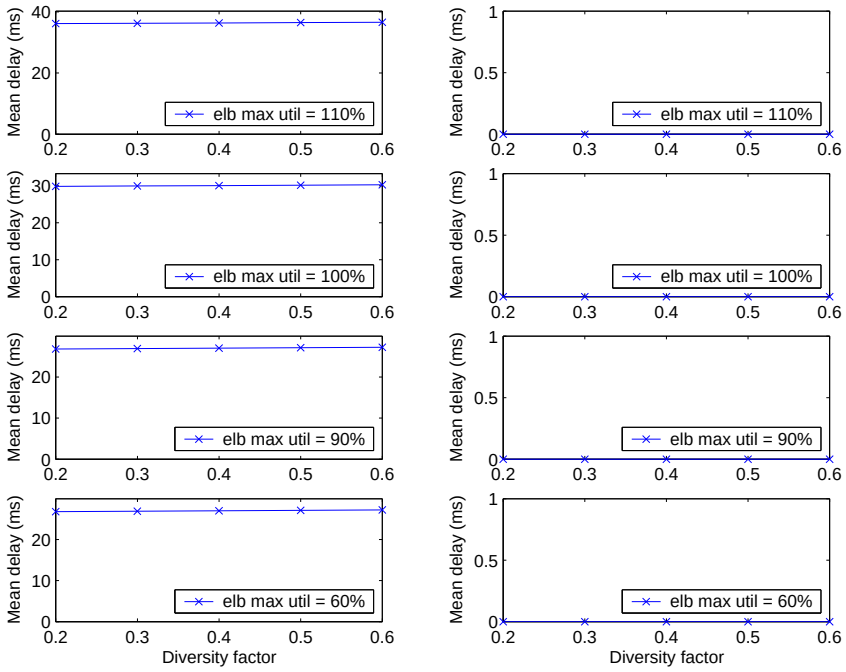


Fig. 30.: Average delay and delay variance of static load-balancing v.s. utilization and path quality diversity factors (symmetric topology, Poisson type traffic)

routing. It inherits other advantages of multipath routing, e.g. ability to route broader range of traffic matrices than single path routing without causing congestion.

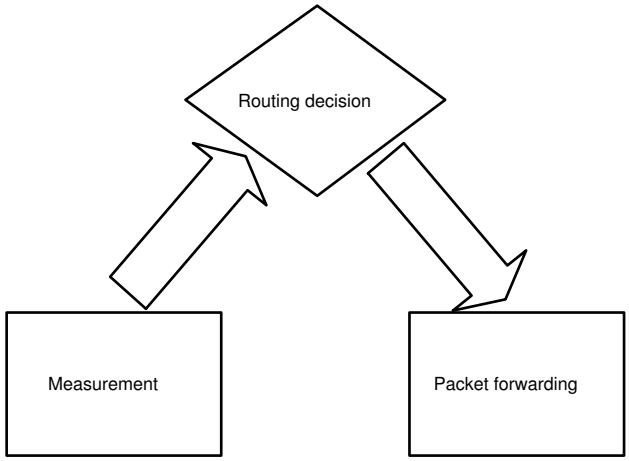


Fig. 31.: Multihoming route control system architecture

A basic fractional MRC system consists of three components as shown in Fig. 31: (1) measurement module, (2) routing decision module and (3) packet forwarding module. The measurement module performs active or passive measurement of qualities of alternate paths to specified destination networks.

The measurement results are sent to the routing decision module. The routing decision module keeps current routing vectors (We call the fractions of traffic to one destination as a "router vector".) for destination networks, and will change the routing vectors according to path quality measurement results. The packet forwarding module forwards packets in such a way that the traffic load on alternate paths match the routing vectors.

To implement the measurement module, there are a number of metrics of quality we can use, e.g. delay, loss rate and available bandwidth (ABW). Different applications have different quality requirements. For example, VoIP applications prefer low delay, low loss rate paths, bulk transfer applications, such as FTP, prefer high ABW paths. In this and next three chapters, we study MRC devices that measure delays

and loss rates of forwarding paths. These two metrics are measurable when we have access to both measurement sources and destinations. They can usually be estimated when we don't have access to measurement destinations. These two metrics are easier to measure than ABW and have relatively mature techniques available, e.g. [33]. The usage of ABW measurement in MRC is a direction for future study.

Different algorithms for calculating exact percentage of traffic sent on each ISP can be used in the decision module. In next two chapters, we propose two such algorithms for MRC among a group of multihomed stub networks.

Techniques to implement the forwarding engine is discussed in next section.

## F. Implementation of fractional forwarding engine

To implement fractional MRC, multiple routes need to be assigned for interested destination networks. The packet forwarding module forwards packets according to the routing vectors decided by the routing decision module.

There are a number of choices for designing this forwarding module. The straightforward one is packet level switching. Packet level forwarding treats packets from different flows in the same way. Weighted fair queuing [34] or deficit round robin [35] can be used as the packet level forwarding algorithm to maintain the target fraction of traffic routed on each path.

A drawback of packet level forwarding is packet reordering that could affect performance of TCP. Most currently used TCP implementations treat duplicated ACKs as a sign of congestion. A TCP sender reduce congestion window by half after it receives the third duplicated ACK packet from its peer. This mechanism causes TCP throughput degradation in multipath routing environments. See [36] for detailed description.

A few types of approaches have been proposed to address the packet reordering issues of multipath routing schemes:

1. Reordering robust TCP variants.

   In recent years, a number of TCP variants have been proposed to address the above weakness of TCP, e.g. TCP-DCR [37]. In this work, we will study the use of TCP-DCR to mitigate the affect of packet reordering caused by our MRC approach. TCP-DCR is designed to work in both wireless and wired environments. It is a sender side modification. Therefore, there are other possible benefits of deploying TCP-DCR in addition to solve packet reordering issue. Of course, other TCP variants can also be used. We will not study other packet reordering robust TCP variants because we don't have access to their ns-2 source code at the time this work is being done.

2. Flow level switch and "Flowlet" [38].

   To totally avoid packet reordering, "flow level" switching can be used. Packets belong to a flow can be identified by the (source address, destination address, protocol type, source port, destination port) tuple, where the two port fields are applicable for UDP and TCP packets. In this approach, a flow is assigned to a path at the moment the first packet of the flow is forwarded on a router. All subsequent packets of the flow will be routed along the same path. This method does not cause packet reordering, but has limited ability to switch traffic from one path to another path and loses benefits of adaptive routing. Anther limitation of this method is that this approach cannot guarantee accurate fractional routing that is required by the MRC scheme we proposed in Chapter IV and V.

   Flow level hashing can be used to approximately maintain the target fraction while reduce number of reordering. Hashing based load-balancing methods

are studied in [39], where hash values of packet headers are used to decide the outgoing links. The authors of [39] also show that "dynamic hashing", i.e. hashing bins are reassigned to outgoing links periodically to maintain the target fractions, can significantly improve the splitting accuracy. A number of algorithms for dynamic hashing are proposed and studied in [40]. However, to support a dynamic MRC where traffic is constantly switched, hashing based splitting methods are still not perfect. Some existing flows need to be switched to maintain higher splitting accuracy, thus packet reordering may happen for such flows. Otherwise, target allocations cannot be achieved accurately.

"Flowlet" switching [38] is proposed to overcome the above drawback of hashing based splitting methods. "Flowlet" switching is based on the fact that a TCP flow normally consists of a sequence of bursts that are separated by intervals of 50 milliseconds or so. If a flow is switched right after such an interval, it will not cause packet reordering assuming the delay difference between the two paths are smaller than the interval.

The most important parameter of Flowlet switching algorithm is the timeout value. An idea value should be larger than the maximum delay difference between alternate paths so that Flowlet switching won't cause any out of order delivery.

Using "Flowlet" switching, higher splitting accuracy can be achieved for dynamic multipath routing schemes like our MRC approach and cause much less packet reordering event than packet level switching approaches.

## G. Related work

The benefits of MRC are studied using both Internet measurements [15] and emulation [32]. Tao et al [31] have measured quality of alternate paths among between three campus networks. They show packet losses are bursty and no path is consistently better than others. We also measure the quality of alternate paths provided through multihoming in this work. Using a number of hosts in the PLANETLAB [41], we measured more paths than [31]. With only access to tools like traceroute, we can measure only round trip delays and loss rates. But it still reflects the quality differences between alternate paths provided by multihoming. Goldenberg et al [42] have studied the optimization of cost and performance for multihoming. All the above work does not consider interaction between MRC of different stub networks which is the focus of this work.

## H. Conclusions

In this chapter, we have presented our Internet measurement experiment results that show the dynamics of quality differences between alternate paths through multihoming. The results indicate the benefits of multihoming route control and that both large time scale and small time scale MRC have potential to improve performance. We have also proposed the generic fractional multihoming route control method for avoidance of oscillations. Our simulations of two greedy MRC approaches have shown the possibility of oscillations which motivates our fractional MRC approaches in following chapters.

# CHAPTER IV

# ROUTE OPTIMIZATION AMONG A GROUP OF MULTIHOMED STUB NETWORKS

## A. Introduction

In this chapter, we propose an "optimal routing" [16] based global coordination method for MRC among a group of multihomed stub networks under the fractional MRC framework introduced in Chapter III. Through global coordination, our approach can avoid oscillations which may be caused by uncoordinated route control.

In Section B, we describe our global optimization based MRC approach for traffic among a group of multihomed stub networks. In Section C, we evaluate the performance of the approach for static traffic matrices and dynamic traffic matrices. Conclusions are drew in Section D.

## B. Route optimization among a group of multihomed stub networks

### 1. Optimal routing formulation

The problem we are studying can be formulated as a optimal routing [16] problem. In a general optimal routing problem, routing traffic on a link incurs some cost that is a function of the total load on the link and the optimal solution maps all traffic on all "physically possible" paths such that the overall cost is minimized. In our problem, only a limited number of paths are given and we need only to map traffic to these paths.

The cost function used in optimal routing is usually a continuous non-decreasing convex function. A common cost function used in earlier optimal routing work is the delay on the link weighted by the traffic volume on the link. The objective is to

find a routing solution that minimizes the average delays experienced by all traffic. Using this cost function, other performance metrics, packet loss rate and congestion are partly considered because these metrics are correlated with queuing delay of a link.

In the optimal routing formulation of MRC among a group of multihomed stub networks, we set the objective as minimizing the sum of the "path delays" of inhouse traffic and the access link queuing delays of inhouse traffic and Internet traffic weighted by their traffic volumes.

Before giving the optimal routing formulation of MRC among a group of multihomed stub networks, we define following symbols.

- $N$: set of nodes representing stub networks;
- $K_i$: set of nodes representing ISP edge routers of $i \in N$;
- $P_{ij}$: set of virtual directed links representing valid paths between ISP edge routers of $i$ and ISP edge routers of $j$, where $i, j \in N, i \neq j$;
- $(i, j)$: link from $i$ to $j$, where $i \in N, j \in K_i$ or $i \in K_j, j \in N$;
- $d_{ij}(x)$: virtual delay function of directed link $(i, j)$, where $x$ is the load on link $(i, j)$, $i \in N, j \in K_i$ or $i \in K_j, j \in N$;
- $d_{ijw}$: virtual delay of virtual link $w$, where $w \in P_{ij}$, $i, j \in N, i \neq j$;
- $r_{ij}$: traffic demand from $i$ to $j$, where $i, j \in N, i \neq j$;
- $x_{ijw}$: fraction of $r_{ij}$ routed along path $w \in P_{ij}$, where $i, j \in N, i \neq j$;
- $u_{ij}$: load of ingress Internet traffic on link $(i, j), i \in K_j, j \in N$;
- $v_{ij}$: load of egress Internet traffic on link $(i, j), i \in N, j \in K_i$ ;

- $S_{ik}$: set of paths (virtual links), egress traffic on which passes link $(i,k)$, where $i \in N$, $k \in K_i$, i.e. $\{(m,n)|m = k, n \in K_j, j \in N, j \neq i, (m,n) \in P_{ij}\}$;

- $S_{ki}$: set of paths (virtual links), ingress traffic on which passes link $(k,i)$, where $i \in N$, $k \in K_i$, i.e. $\{(m,n)|m \in K_j, n = k, j \in N, j \neq i, (m,n) \in P_{ij}\}$.

The optimal routing formulation of our problem is as follows.

Minimize:

$$C(x) = \sum_{i \in N, j \in (N \setminus i) \cup M_i, w \in P_{ij}} x_{ijw} r_{ij} d_{ijw} + \sum_{i \in N, k \in K_i} t_{ki} d_{ki}(t_{ki}) + \sum_{i \in N, k \in K_i} t_{ik} d_{ik}(t_{ik}) \quad (4.1)$$

Subject to:

$$x_{ijw} \geq 0, \qquad (i \in N, j \in (N \setminus i), w \in P_{ij}) \quad (4.2)$$

$$\sum_{w \in P_{ij}} x_{ijw} = 1, \qquad (i \in N, j \in (N \setminus i)) \quad (4.3)$$

$$t_{ik} = \sum_{w \in S_{ik}} x_{ijw} r_{ij} + v_{ik}, \qquad (i \in N, k \in K_i) \quad (4.4)$$

$$t_{ki} = \sum_{w \in S_{ki}} x_{ijw} r_{ij} + u_{ki}, \qquad (i \in N, k \in K_i) \quad (4.5)$$

where (4.1) is the objective function, i.e. delays of Internet paths (virtual links) experienced by traffic controlled by MRC devices plus virtual delays on access links weighted by traffic volumes; (4.2) is non-negativity of routing vectors; (4.3) is to ensure that all traffic are routed; (4.4) and (4.5) are traffic volume on access links.

Packet losses are not considered in the above formulation. However, we can take loss rates into account by substituting $p_{ijkl}$ with a "virtual delay function", $vdelay(p_{ijkl}, r_{ijkl})$, where $r_{ijkl}$ represents the loss rate on the path. We will use virtual delay functions in next chapter.

Our optimal routing based MRC works as follows: The demand matrices of inhouse traffic and Internet traffic and path qualities are measured in a distributed manner. After a fixed period, say one minute, or when there are significant changes of the demand matrix or path qualities, the measured path characteristics and the demand information are exchanged among all the stub networks. After receiving the updated information, each stub network predicts the demand matrix and the path characteristics using a prediction model (see section 1 for details), and calculates the optimal routing solution using an optimal routing algorithm for all the stub networks. The routing solution is adopted until next update. We also assume that the queuing delay function can be measured and is known to the optimization algorithm. As long as the queuing delay is a non-decreasing convex function, our algorithm can find an optimal solution [16].

## C. Evaluation

### 1. Simulation scenarios

We analyze the performance of our approach in two aspects:

1. Static analysis: we study the performance of our approach for a set of model-based random demand matrices on a set of randomly generated topologies with different path characteristics. We compare the average end to end delay of inhouse traffic and the average queuing delay of Internet traffic with a static load-balancing approach that distributes egress inhouse traffic evenly on each link. In this analysis, we assume the demand matrix and path characteristics are static. This analysis gives an upper bound of the performance of our approach. We assume egress Internet traffic is distributed evenly on all egress links. Both our approach and the static load-balancing approach do not have control over

the routes of ingress Internet traffic. We generate 20 random demand matrices for each combination of number of stub networks and number of ISPs for each stub network.

2. Dynamic analysis: We also study the performance of our algorithm with changing demand matrices. Specifically, we evaluate the performance of our algorithm with a time series of demand matrices generated from an Internet trace. In this analysis, the path characteristics are fixed. We study a simple demand prediction method, i.e. using the average demands of the last period as the prediction of demands for the next period. Empirical study [43] shows this simple prediction model is as good as other complex models for prediction period in order of minutes. We study optimization periods of 1, 2, 3, 5 and 10 minutes. We compare the performance of our approach using predicted demand matrices with the ideal performance, i.e. when the demand matrix of current period is available to the algorithm.

   For dynamic analysis, we generate time series of demand matrices using the Leipzig-II trace from NLANR [44] website. We generate a time series of demand matrices by classifying packets with same source (or destination) IP addresses into one of a number of flows by a probability that is proportional to the expected volume of the flow.

### 2. Implementation of the optimal routing algorithm

Optimal routing problems are a type of non-linear optimization problems. They are usually solved using the gradient projection methods [16]. In this chapter, to speed up the calculation, we solve the optimal routing problem using a linear programming approximation method. Specifically, we use piece-wise linear function $f_{ki}(t_{ki})$

and $f_{ik}(t_{ik})$ to approximate the item $t_{ki}d_{ki}(t_{ki})$ and $t_{ik}d_{ik}(t_{ik})$ in (4.1). This linear approximation method is similar to the one used in [45]. In our implementation, we use the GNU Linear Programming Kit (GLPK) [46] to solve the linear programming problems.

## 3. Simulation results

### a. Static analysis

Fig. 32 shows the "performance improvement ratios" of our approach compared to the static load-balancing approach for topologies of 10 stub networks where each stub network has 2 ISPs. Here, we define "performance improvement ratio" of our approach compared to the static load-balancing approach as $(L_{lb} - L_{opt})/L_{lb}$, while $L_{lb}$ is the average delay for the static load-balancing approach, $L_{opt}$ is the average delay for our global optimization approach. The ratios are calculated for both the access link queuing delay of Internet traffic and end to end delay of inhouse traffic under different link utilization. We fix a demand matrix and change the access link capacities to change the average utilization. The four curves show the effects of queuing model and type of network topologies.

The main observations are:

1). The improvement for Pareto queuing model is more evident than Poisson queuing model. This is because the queuing delay of Poisson model is lower than the queuing delay of Pareto model. While our approach gets improvement mainly by exploiting path diversity for the Poisson queuing model case, it can get more improvement by balancing load on access links for the Pareto queuing model case.

2). The improvement for asymmetric topologies is more evident than symmetric topologies. This is because for symmetric topologies, inhouse traffic can be balanced

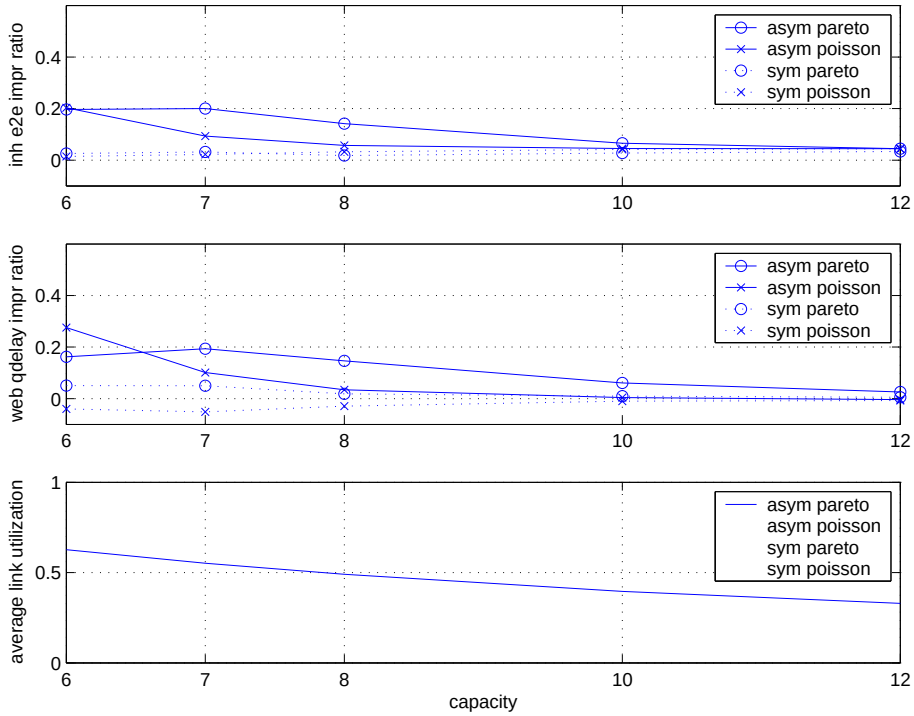


Fig. 32.: Performance improvement ratios: topologies of 10 stub networks, each has 2 ISPs, capacities are in Mbps

on links using the static load-balancing.

3). The improvement is larger at higher average utilization. This is because the queuing delay of both Poisson and Pareto model is dramatically increased as utilization approaches 100%. (The utilization shown in the figure is average utilization. Utilization is higher on some links in the topologies.)

4). The average access link queuing delays of Internet traffic are also improved, except for symmetric networks under Poisson queuing model where the queuing delay is increased by 5% at most.

Results for 10 stub networks with 3 ISPs and 4 ISPs are shown in Figs. 33 and 34. As the number of ISPs increases, the improvement ratios become larger than the above 2-ISP case. The explanation is that more ISPs provide more opportunities for global optimization. Results for 20 stub networks with 2, 3 and 4 ISPs for each stub network are shown in Figs. 35 to 37. The above observations for 10 stub network topologies are still true for 20 stub network topologies.

b. Dynamic analysis

Figs. 38 to 41 show the average improvement ratios of end to end delay of inhouse traffic of our approach compared to the static load-balancing approach for symmetric/asymmetric topology and Pareto/Poisson type traffic. Results for topologies of 9 different sizes are shown in groups of bars from left to right. The sizes of topologies can be represented by $x - y$, where $x = 4, 10, 20$, is the number of stub networks, $y = 2, 3, 4$, is the number of ISPs of each stub network. In this set of simulations, the resulting average link utilization is about 50%.

The results for asymmetric topology and Pareto type traffic show that the performance improvements are larger over shorter prediction periods. Shorter periods enable more accurate prediction of demand and path characteristics and hence pro-

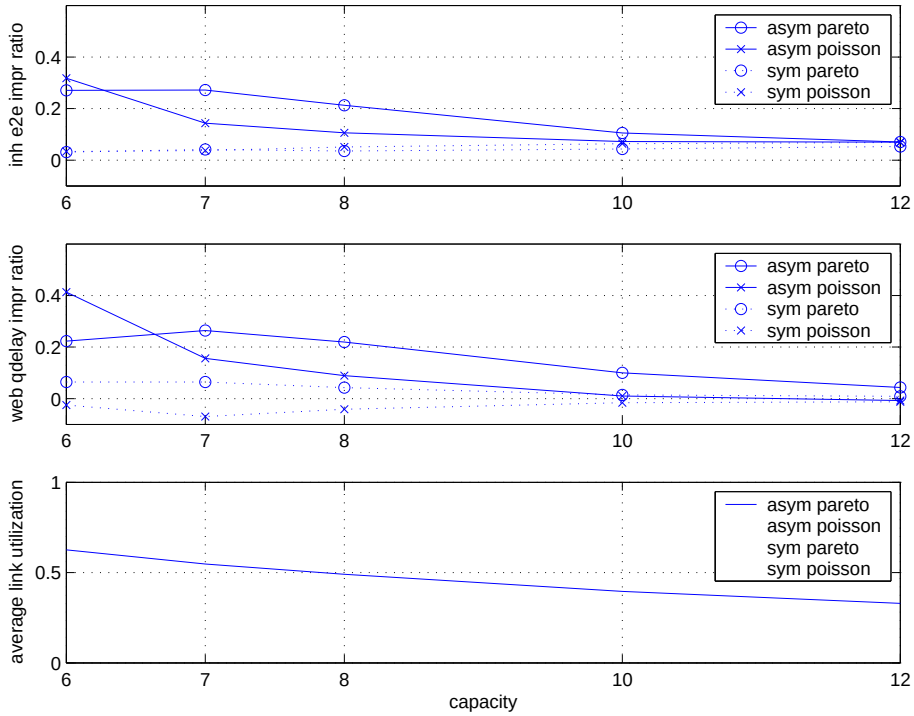


Fig. 33.: Performance improvement ratios: topologies of 10 stub networks, each has 3 ISPs, capacities are in Mbps

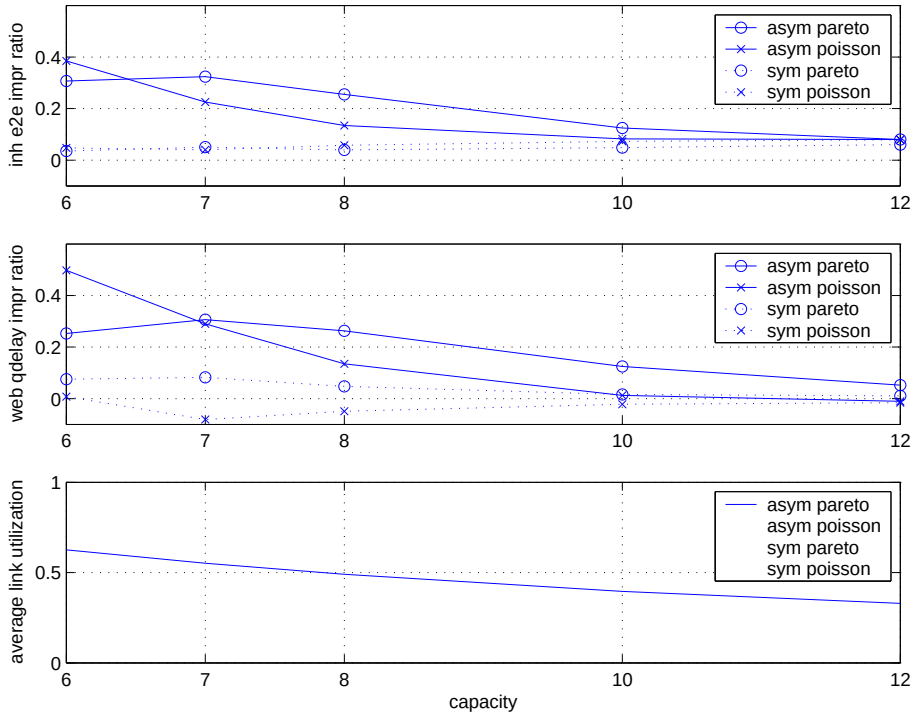


Fig. 34.: Performance improvement ratios: topologies of 10 stub networks, each has 4 ISPs, capacities are in Mbps

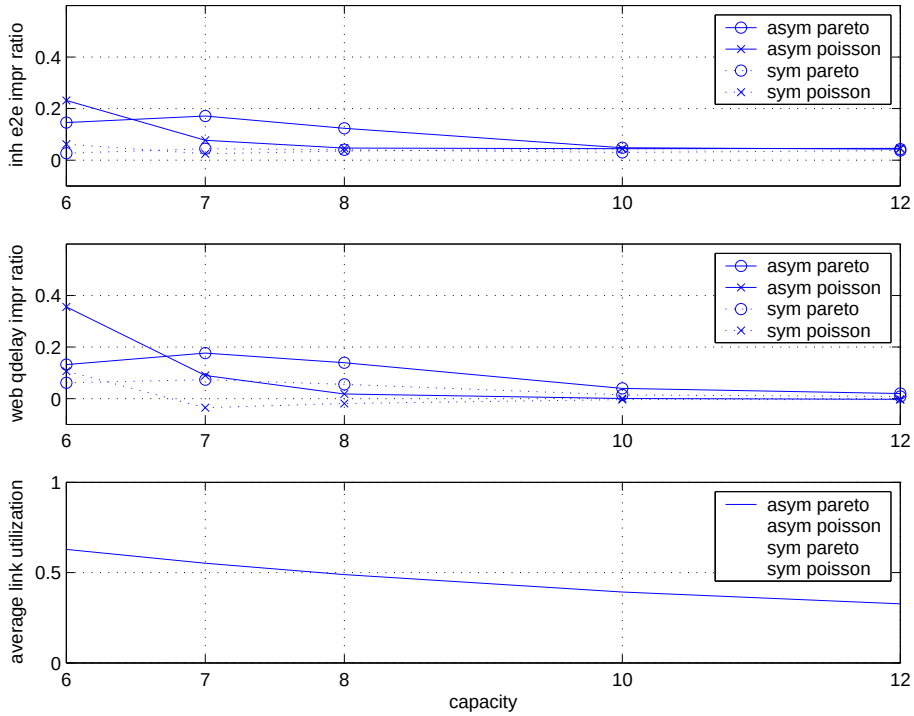


Fig. 35.: Performance improvement ratios: topologies of 20 stub networks, each has 2 ISPs, capacities are in Mbps

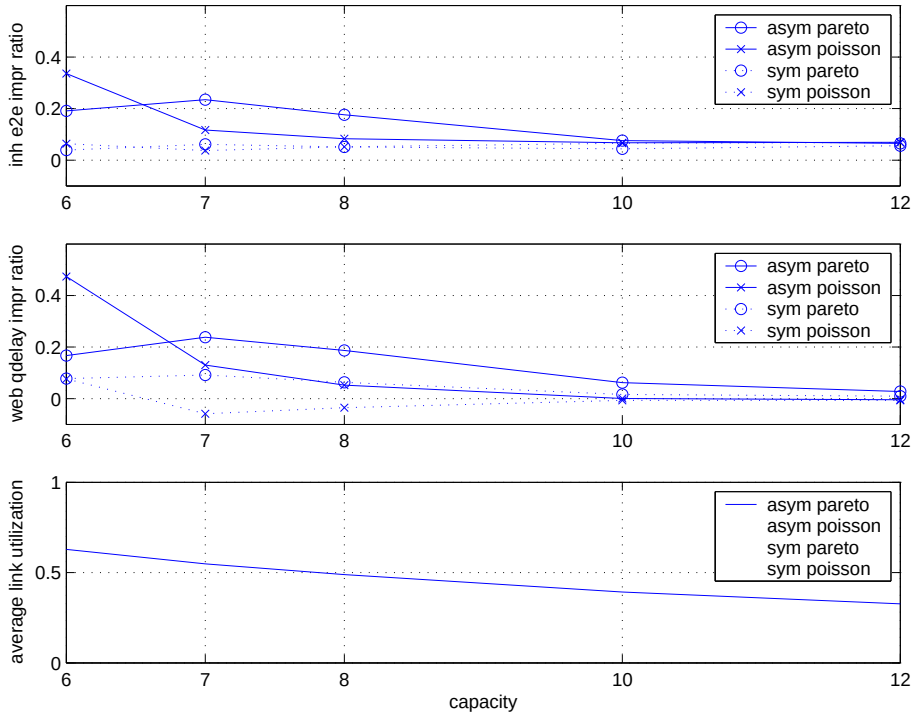


Fig. 36.: Performance improvement ratios: topologies of 20 stub networks, each has 3 ISPs, capacities are in Mbps

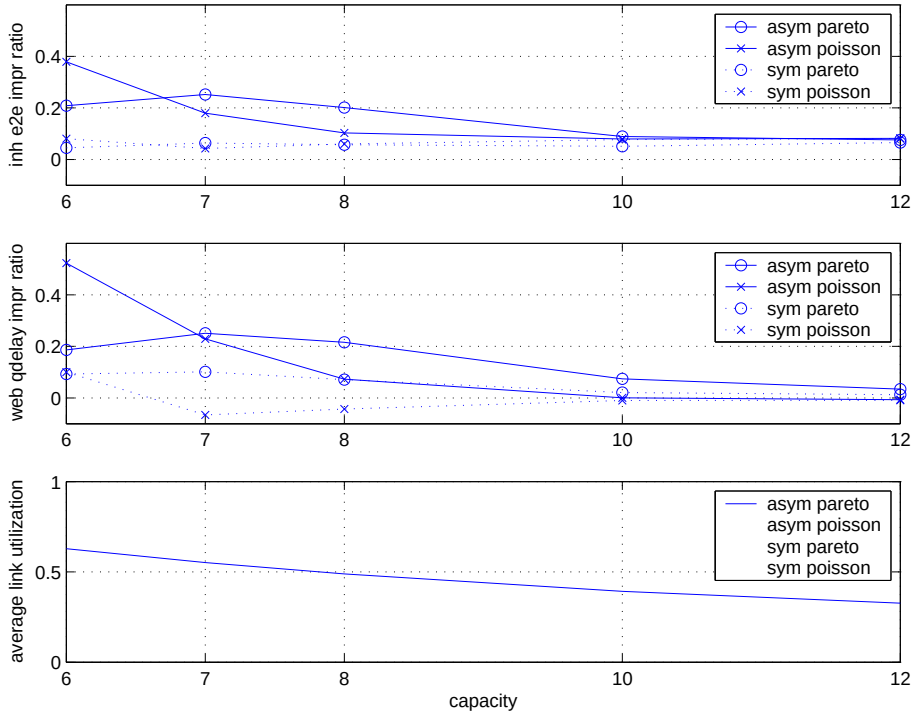


Fig. 37.: Performance improvement ratios: topologies of 20 stub networks, each has 4 ISPs, capacities are in Mbps

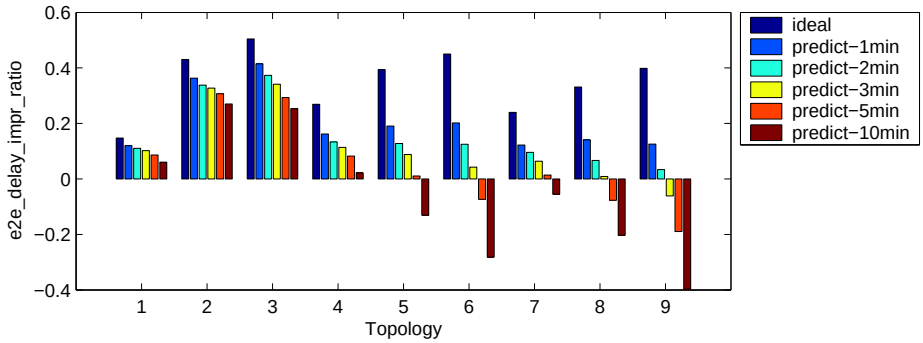


Fig. 38.: Inhouse traffic end to end delay improvement: asymmetric topology, Pareto traffic. Topology 1 to 9: 4x2 4x3 4x4 10x2 10x3 10x4 20x2 20x3 20x4

vide larger performance gains. We also found that as the number of stub networks increases, the performance gain decreases more rapidly as prediction period increases. This is primarily due to the artifact of how the demand matrices are generated in our simulations. In our simulations, the total expected volume of inhouse traffic on a link is fixed, as the number of stub networks increases, the volume of traffic from a stub network to another becomes smaller making the demand less predictable.

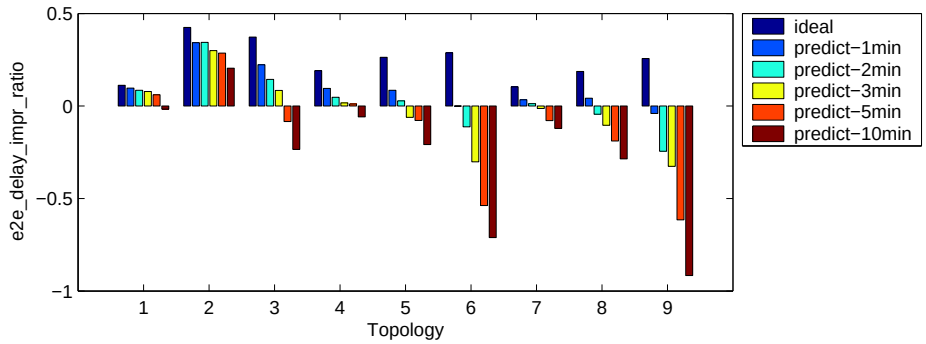


Fig. 39.: Inhouse traffic end to end delay improvement: asymmetric topology, Poisson traffic. Topology 1 to 9: 4x2 4x3 4x4 10x2 10x3 10x4 20x2 20x3 20x4

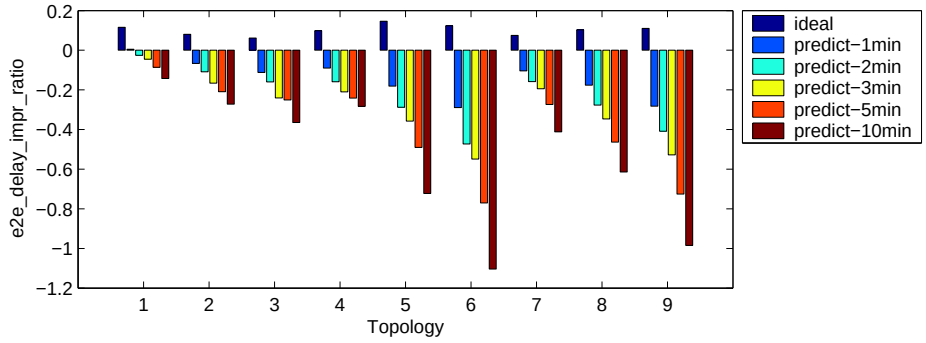


Fig. 40.: Inhouse traffic end to end delay improvement: symmetric topology, Pareto traffic. Topology 1 to 9: 4x2 4x3 4x4 10x2 10x3 10x4 20x2 20x3 20x4

From Figs. 38 to 41, we observe that symmetric topologies and Poisson-arrival queuing model show smaller performance gains even performance degradation(the negative values in the figures) for dynamic simulations in dynamic environment, which

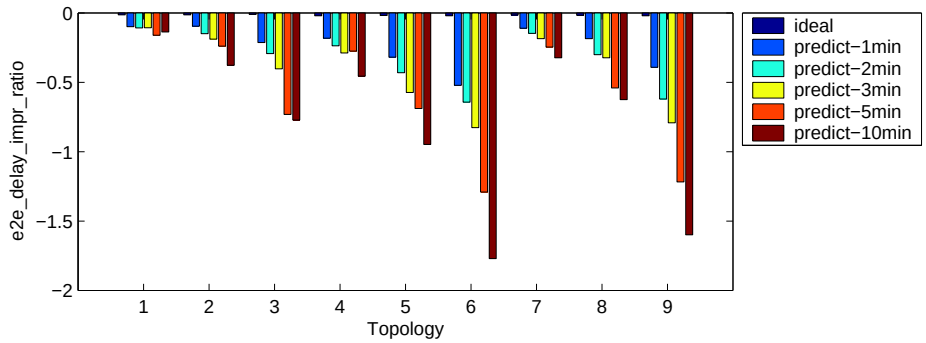


Fig. 41.: Inhouse traffic end to end delay improvement: symmetric topology, Poisson traffic. Topology 1 to 9: 4x2 4x3 4x4 10x2 10x3 10x4 20x2 20x3 20x4

is consistent with the static analysis. Since the volume of the demand we generated from the real trace is small, we scale it by 36 to get the average load on a link to be about 3.5 Mbps. In the real environment, because volume of demand is larger, we expect the demand matrix becomes more predictable as traffic volume increases.

## D. Conclusions

In this chapter, we proposed an approach for coordinating the route control of traffic among a group of multihomed stub networks. We studied the static and dynamic performance of our approach using simulations. The results show that our approach can significantly improve the performance of route control among a group of multihomed stub networks for both static traffic demand and dynamic traffic demand.

# CHAPTER V

# USER-OPTIMAL MULTIHOMING ROUTE CONTROL

## A. Introduction

In this chapter, we continue our study on MRC among a group of multihomed stub networks. As we pointed out in Chapter III, when the access links of the stub networks are not over-provisioned, traditional MRC schemes that use a single path at a time for traffic to one destination network may cause oscillations. In Chapter IV, a global optimal routing based coordination method is proposed to avoid such possible oscillations. In this chapter, we propose a distributed "user-optimal routing" [47] based MRC scheme to solve the above problem. The basic idea is to use multiple paths provided by multihoming simultaneously and move traffic gradually during changes in network environment. Specifically, our scheme calculates "user-optimal routing" using the gradient projection method that is originally used in solving optimal routing problems. User-optimal routing is simpler to implement. Moreover, as we will show in Section F, it can achieve similar performance as global optimal routing for this problem.

While this approach is mainly designed for MRC among a group of multihomed stub networks, it can also be applied to Internet traffic to a number of "top" Internet destination networks that account for a large portion of the Internet traffic of the stub networks. In this study, we assume there are a few top Internet destinations for each stub network and we use the same MRC algorithm for these Internet destinations. We call the traffic among such a group of stub networks as *inhouse* traffic and call the traffic between a stub network and networks not in the group as *Internet* traffic.

The rest of this chapter is organized as follows: Related work is discussed in

Section B. The optimal routing formulation of MRC (modified from Chapter IV to take Internet traffic into account) is given in Section C. In Section D, we introduce the user-optimal routing formulation of our MRC problem. In Section E, we give our user-optimal routing based MRC algorithm. In Section F, we compare the performance of our scheme with the optimal solution and show the dynamic characterization of our scheme using simulations. Conclusions and future work are discussed in Section G.

## B. Related work

The performance of user-optimal routing, also called selfish routing, has been studied analytically [48] and using simulations [28]. Qiu et al [28] show selfish routing achieves performance close to optimal routing for intra-domain routing. Similarly, our work shows selfish routing based MRC achieves performance close to optimal routing based MRC.

Centralized and distributed gradient projection algorithms for optimal routing have been proposed in [16, 49, 50]. A measurement based multipath optimal routing algorithm has been proposed in [51]. However, the algorithm needs an overlay network infrastructure to measure the first derivatives of the cost functions of all links in the network. In our approach no such information exchange is required.

Our MRC approach uses measurement based gradient projection algorithm to calculate the user-optimal solution. It is similar to [51, 52] but does not require overlay infrastructure to exchange information, thus is simpler to implement.

In summary, our work leverages previous work on optimal routing and user-optimal routing and targets a new problem on multihoming route control.

C. Optimal routing formulation

We modify the optimal routing formulation in Chapter IV to take MRC of Internet traffic into account. The modified symbols used in this chapter are defined as follows.

- $N$: set of nodes representing stub networks;
- $M_i$: set of nodes representing Internet destinations of $i \in N$;
- $K_i$: set of nodes representing ISP edge routers of $i \in N$;
- $P_{ij}$: set of virtual directed links representing valid paths between ISP edge routers of $i$ and ISP edge routers of $j$, where $i, j \in N, i \neq j$, or paths from ISP edge routers of $i$ to Internet destination $j$, where $i \in N, j \in M_i$;
- $(i, j)$: link from $i$ to $j$, where $i \in N, j \in K_i$ or $i \in K_j, j \in N$, or virtual link from $i$ to $j$, where $i \in K_n, j \in M_n, n \in N$ or $i \in K_m$, $j \in K_n, (i, j) \in P_{mn}, m, n \in N, m \neq n$;
- $d_{ij}(x)$: virtual delay function of directed link $(i, j)$, where $x$ is the load on link $(i, j)$, $i \in N, j \in K_i$ or $i \in K_j, j \in N$;
- $d_{ijw}$: virtual delay of virtual link $w$, where $w \in P_{ij}$, $i, j \in N, i \neq j$, or $i \in N, j \in M_i$;
- $r_{ij}$: traffic demand from $i$ to $j$, where $i, j \in N, i \neq j$ or $i \in N, j \in M_i$;
- $x_{ijw}$: fraction of $r_{ij}$ routed along path $w \in P_{ij}$, where $i, j \in N, i \neq j$ or $i \in N, j \in M_i$;
- $u_{ij}$: load of ingress Internet traffic on link $(i, j), i \in K_j, j \in N$;

- $S_{ik}$: set of paths (virtual links), egress traffic on which passes link $(i,k)$, where $i \in N$, $k \in K_i$, i.e. $\{(m,n)|m = k, n \in K_j, j \in N, j \neq i, (m,n) \in P_{ij}\} \cup \{(m,n)|m = k, n \in M_i\}$;

- $S_{ki}$: set of paths (virtual links), ingress traffic on which passes link $(k,i)$, where $i \in N$, $k \in K_i$, i.e. $\{(m,n)|m \in K_j, n = k, j \in N, j \neq i, (m,n) \in P_{ij}\}$.

The optimal routing formulation of our problem is as follows.

Minimize:

$$C(x) = \sum_{i \in N, j \in (N \setminus i) \cup M_i, w \in P_{ij}} x_{ijw} r_{ij} d_{ijw} + \sum_{i \in N, k \in K_i} t_{ki} d_{ki}(t_{ki}) + \sum_{i \in N, k \in K_i} t_{ik} d_{ik}(t_{ik}) \quad (5.1)$$

Subject to:

$$x_{ijw} \geq 0, \qquad (i \in N, j \in (N \setminus i) \cup M_i, w \in P_{ij}) \quad (5.2)$$

$$\sum_{w \in P_{ij}} x_{ijw} = 1, \qquad (i \in N, j \in (N \setminus i) \cup M_i) \quad (5.3)$$

$$t_{ik} = \sum_{w \in S_{ik}} x_{ijw} r_{ij}, \qquad (i \in N, k \in K_i) \quad (5.4)$$

$$t_{ki} = \sum_{w \in S_{ki}} x_{ijw} r_{ij} + u_{ki}, \qquad (i \in N, k \in K_i) \quad (5.5)$$

where (5.1) is the objective function, i.e. virtual delays of Internet paths (virtual links) experienced by traffic controlled by MRC devices plus virtual delays on access links weighted by traffic volumes; (5.2) is non-negativity of routing vectors; (5.3) is to ensure that all traffic are routed; (5.4) and (5.5) are traffic volume on access links.

In this chapter, we consider both queuing delays and loss rates on links. We use half of the expected TCP hand shake time [53] as the virtual one way delay function, i.e. $delay + T_s \frac{loss\_rate}{1-2loss\_rate} (loss\_rate < 0.5)$, where $T_s$ is the TCP SYN timeout, initially

three seconds [54]. Here, we assume the reverse path has the same delay and loss rate as the forward path. TCP handshake round trip time is used in comparing qualities of alternate paths by previous studies on multihoming, e.g. [15]. When loss rates are small, e.g. less than 1%, the sum of virtual delays of links along a path is roughly the same as the virtual delay of the path.

In our simulations, we use piecewise linear approximation models of queuing delay and packet drop rate built from samples of ns-2 [17] simulations. The two models are Poisson queuing model, M/M/1, and a Pareto ($\beta = 1.5$) queuing model, P/M/1. The parameters of the ns-2 simulation are as follows: The average packet length is 558 bytes(calculated from a backbone trace); The link capacity is 100 Mbps; The buffer size of each link is equal to the product of 250 milliseconds and link speed according to the rule-of-thumb of router buffer sizing [55]). The resulting virtual delay function of a link is a continuous, non-decreasing, convex function, which ensures that the optimal routing has a unique solution [16].

## D. User-optimal routing formulation

In this section, we first introduce the concept of user-optimal routing. After discussing the characterizations of optimal routing and user-optimal routing, we give the user-optimal routing formulation of our MRC problem.

### 1. User-optimal routing

"User-optimal routing" [47] is optimal from the point of view of each user. It is also called "selfish routing". Like previous work on "user-optimal routing" [28], we assume traffic consists of a lot of "infinitesimal flows" and each user controls such an infinitesimal flow. At equilibrium of user-optimal routing, each flow is routed along a

path with minimum end to end delay. Thus no user can reduce the delay of its traffic by changing the routing of its own traffic unilaterally.

Previous work [28] showed that "selfish routing" can achieve similar performance as "optimal routing" [16] for intra-domain routing. As we will show in Section F, for MRC, the performance of user-optimal routing based approach is also close to optimal routing based approach. One of the advantages of user-optimal routing is that it is distributed in nature and easier to implement.

Because the similarity of the characterization between network-optimal routing and user-optimal routing, user-optimal routing can be solved using algorithms for network-optimal routing with a specific cost function [56].

### 2. Characterization of user-optimal routing

The characterization of user-optimal routing is that user-optimal routing allocation is positive only on paths with minimum end to end delay [57]. In our MRC problem, it is as follows:

$$x^*_{ijw} > 0 \quad \Rightarrow \quad \begin{cases} d_{ijw'} + d_{ik'}(t_{ik'}) + d_{l'j}(t_{l'j}) \geq d_{ijw} + d_{ik}(t_{ik}) + d_{lj}(t_{lj}), \\ (i \in N, j \in (N \setminus i), w, w' \in P_{ij}, w = (k, l), w' = (k', l')); \\ x^*_{ijw} > 0 \Rightarrow d_{ijw'} + d_{ik'}(t_{ik'}) \geq d_{ijw} + d_{ik}(t_{ik}), \\ (i \in N, j \in M_i, w, w' \in P_{ij}, w = (k, j), w' = (k', j)) \end{cases} \tag{5.6}$$

### 3. Characterization of optimal routing solutions

We give the characterization of optimal routing here, because it is useful to design an algorithm for user-optimal routing as we will see in Section 4.

According to [16], the characterization of the optimal solution $x^*$ is:

$$x^*_{ijw} > 0 \Rightarrow \frac{\partial C(x^*)}{\partial x_{ijw'}} \geq \frac{\partial C(x^*)}{\partial x_{ijw}}, \ \ (i \in N, j \in (N \setminus i) \cup M_i\}; w, w' \in P_{ij}) \tag{5.7}$$

In other words, at the optimal point, for any source-destination pair, shifting a small amount of traffic from one path to an alternate path that is not used by the source-destination pair will increase the total cost, and shifting a small amount of traffic to an alternate path that is used by the source-destination pair won't change the total cost. In summary, at the optimal point, changing routing solution $x$ won't lower the total cost.

## 4. Formulation of user-optimal routing

Specifically, we need to define the cost function of a link, $l$, as

$$D_l(x) = \int_0^x d_l(t)dt \tag{5.8}$$

Therefore, the user-optimal routing problem can be formulated as the following optimal routing problem and can be solved using the algorithms for optimal routing. Minimize:

$$D(x) \ = \sum_{i \in N, j \in (N \setminus i) \cup M_i, w \in P_{ij}} x_{ijw} r_{ij} d_{ijw} + \sum_{i \in N, k \in K_i} D_{ki}(t_{ki}) + \sum_{i \in N, k \in K_i} D_{ik}(t_{ik}) \tag{5.9}$$

subject to: (5.2), (5.3), (5.4), (5.5)

The solution for this optimal routing problem has the characterization given by (5.7). Thus

$$x^*_{ijw} > 0 \quad \Rightarrow \quad \frac{\partial D(x^*)}{\partial x_{ijw'}} \geq \frac{\partial D(x^*)}{\partial x_{ijw}} \tag{5.10}$$

$$\Leftrightarrow \quad \begin{cases} d_{ijw'} + d_{ik'}(t_{ik'}) + d_{l'j}(t_{l'j}) \geq d_{ijw} + d_{ik}(t_{ik}) + d_{lj}(t_{lj}), \\ (i \in N, j \in (N \setminus i), w, w' \in P_{ij}, w = (k, l), w' = (k', l')) \, ; \\ d_{ijw'} + d_{ik'}(t_{ik'}) \geq d_{ijw} + d_{ik}(t_{ik}), \\ (i \in N, j \in M_i, w, w' \in P_{ij}, w = (k, j), w' = (k', j) \end{cases} \tag{5.11}$$

That is equivalent to (5.6), i.e. the solution has the same characterization as the user-optimal routing solution. Because the user-optimal solution is unique when the delay function of a link is continuous and nondecreasing [57], we actually get the user-optimal solution.

E. User-optimal routing based MRC among a group of multihomed stub networks

An important class of algorithms for solving optimal routing problems are the gradient projection methods[16]. They are also suitable for distributed implementation. In this work, we implement a distributed asynchronous gradient projection algorithm [16] to solve the user-optimal routing problem formulated in Section 4. Two minor modifications are made to improve the convergence speed: normalization of delay difference (changing $(l_{ijw} - l_{ij\bar{w}})$ to $(l_{ijw} - l_{ij\bar{w}})/l_{ijw}$) and randomized waiting time (changing $T = T_0$ to $T \in [0.5T_0, 1.5T_0)$). This algorithm is more practical than other projection methods because it requires simpler calculations at each iteration. Because important information needs to be exchanged during optimal routing calculation, the sum of first derivatives, are end to end delays of alternate paths that can be measured directly, the distributed algorithm can be implemented more easily than normal optimal routing algorithm. The algorithm for each node is shown in Fig. 42.

**while** *true* **do**

Measure all one-way end to end virtual delays, $l_{ijw} : w \in P_{ij}$ ;

Find the minimum virtual delay to $j$, $l_{ij\bar{w}}$ ;

**foreach** $w \in P_{ij}, w \neq \bar{w}$ **do**

$x_{ijw}^{t+1} = max\{0, \quad x_{ijw}^{t} - \alpha^t(l_{ijw} - l_{ij\bar{w}})/l_{ijw}\}$ ;

$x_{ij\bar{w}}^{t+1} = 1 - \sum_{w \in P_{ij}, w \neq \bar{w}} x_{ijw}$ ;

Wait for random time $T \in [0.5T_0, 1.5T_0)$ ;

Fig. 42.: User-optimal routing based MRC (for traffic from $i$ to $j$, where $i \in N$; $j \in N, j \neq i$ or $j \in M_i$.)

Three types of route control strategies, static, greedy and user optimal can be expressed using the same algorithm shown in Fig. 42. For static route control, the step size of modification to the routing factor is 0. For greedy route control, the step size is $\infty$. For user optimal route control the step size is a positive number.

Our algorithm works as follows:

(1) Each node measures the end to end delays of alternate paths from it to a remote node every $t$ seconds, where $T$ is uniformly distributed in $[0.5T_0, 1.5T_0]$ to avoid update synchronization. Each measurement consists of a number of samples to filter noise. Because our algorithm is based on the difference of delays of alternate paths, it does not require clock synchronization of different nodes.

(2) After the node obtains measured end to end delays, it updates the routing vector $x$ for this destination according to the algorithm shown in Fig. 42. This is according to the gradient projection algorithm [16]. $\alpha^t$ is the step size, for distributed implementation it is usually a constant.

In this algorithm, we assume that the traffic demand and path quality does not change too rapidly compared to its convergence speed. This is true when the network

traffic consists of a lot of small flows. The algorithm converges to user-optimal routing given $\alpha$ is small enough [49].

## F. Evaluation

In this section, we evaluate our user-optimal routing based MRC scheme. The evaluation consists of two parts: (1) Since our scheme is based on user-optimal routing, it is important to ensure it will not cause network wide performance degradation. We perform a number of simulations to compare the performance of user-optimal routing based MRC with the optimal solution. (2) We study the dynamic behavior of our algorithm in various dynamic network environments.

### 1. Performance compared to optimal routing

We calculate the global optimal routing("gopt"), user-optimal routing("uopt") and static load-balancing("elb") MRC solutions for randomly generated topologies and traffic matrices. The "static load-balancing" here is to split traffic evenly among all alternate paths. The topologies are of size "4x2", "4x3", "8x2" and "8x3". For each size, we generate 1 symmetric topology, 6 asymmetric topologies and 5 traffic matrices. (See Chapter IV, Section 1, for definitions of types of topology.) We scale the traffic matrices to make the maximum link utilization 60%, 95%, 110% and 125% assuming basic load-balancing routing is used.

For each simulation input(topology, queuing model, traffic matrix), we calculate the total routing cost, average delay, average loss rate and maximum link utilization for "gopt", "uopt" and "elb". The average, minimum and maximum values for simulations of each configuration (same topology type, same queuing model, same network size, traffic matrices that resulted same maximum link utilization for "elb")

are plotted in Figs. 43 to 58.

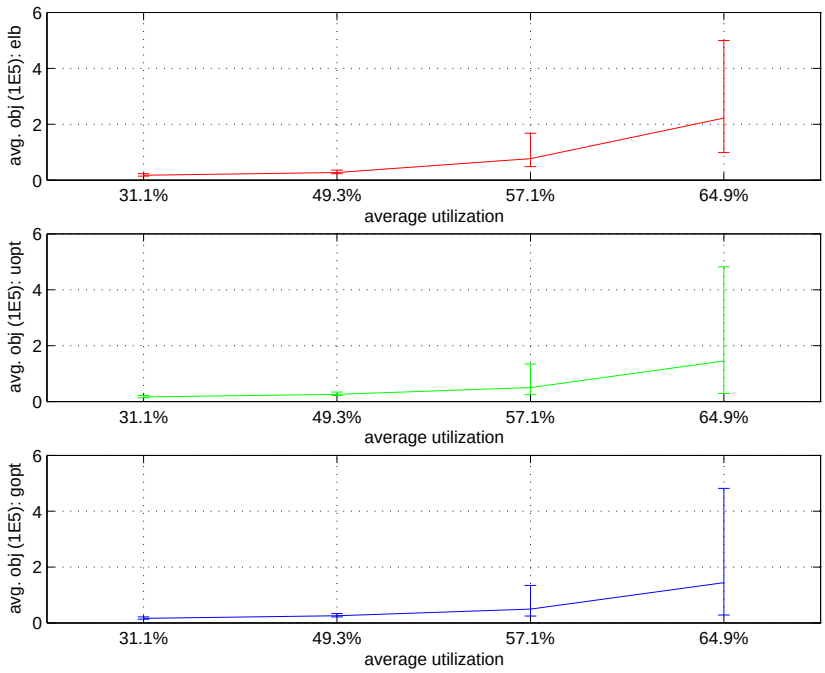


Fig. 43.: Optimization objective, 8x2, symmetric topology: average, minimum and maximum

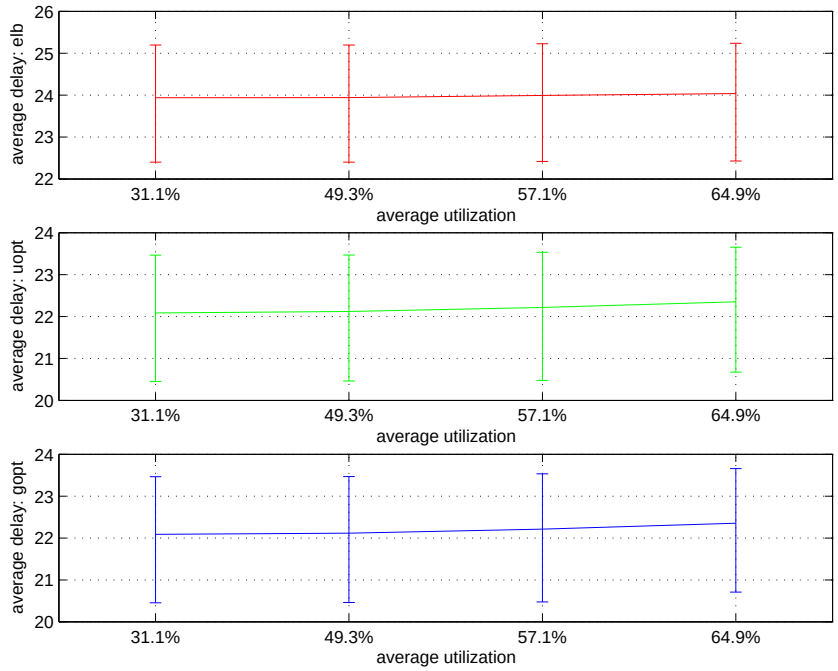


Fig. 44.: Average delay, 8x2, symmetric topology: average, minimum and maximum

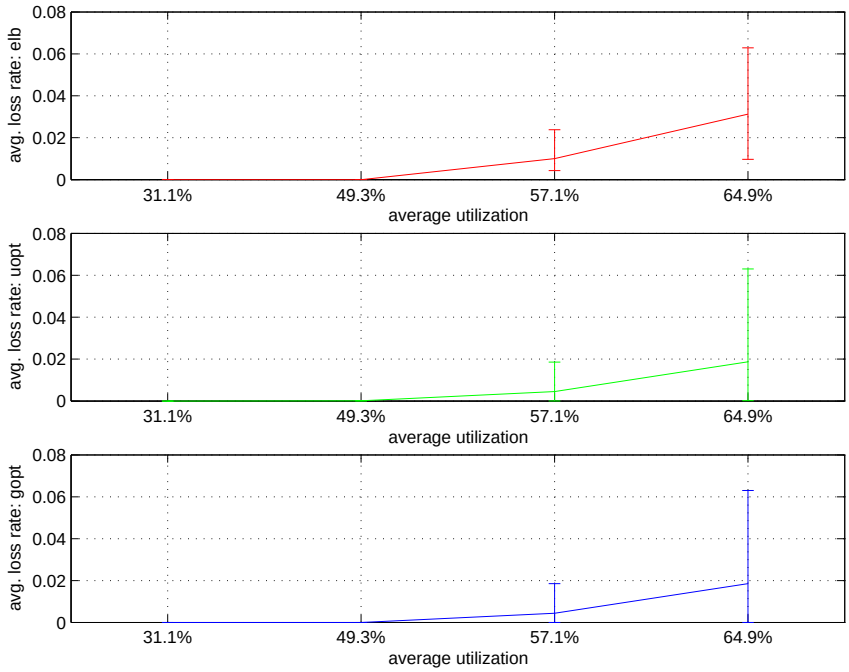


Fig. 45.: Average loss rate, 8x2, symmetric topology: average, minimum and maximum

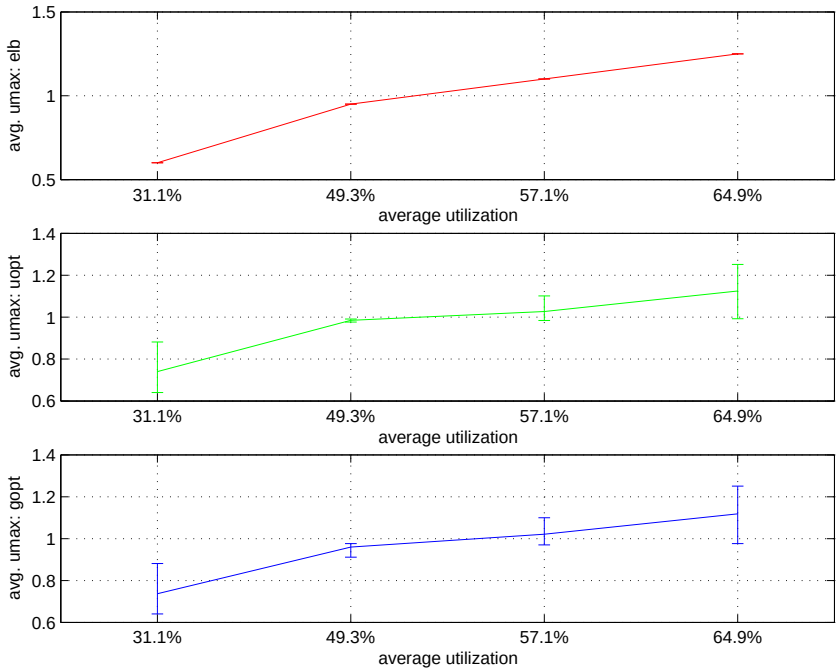


Fig. 46.: Maximum link utilization, 8x2, symmetric topology: average, minimum and maximum

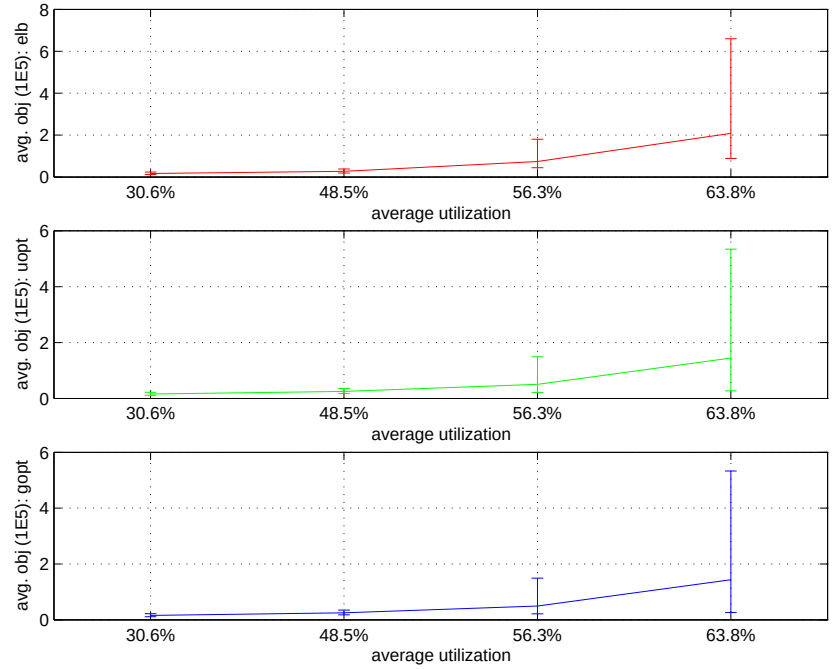


Fig. 47.: Optimization objective, 8x2, symmetric topology: average, minimum and maximum

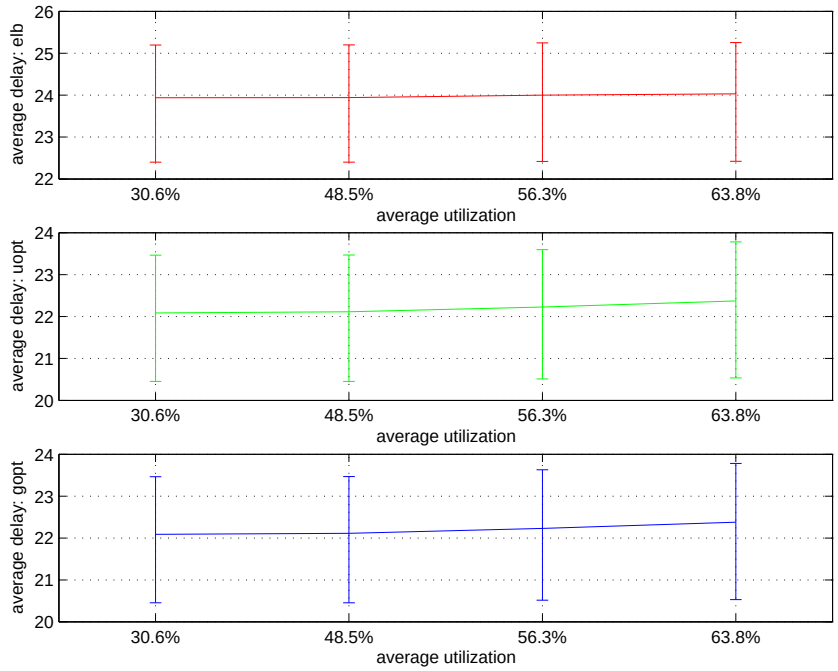


Fig. 48.: Average delay, 8x2, asymmetric topology: average, minimum and maximum

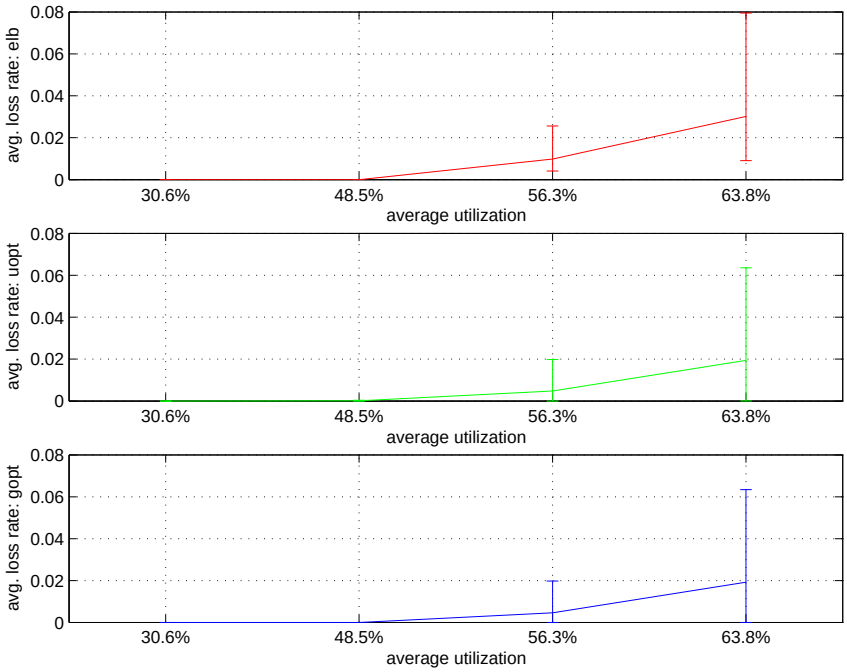


Fig. 49.: Average loss rate, 8x2, asymmetric topology: average, minimum and maximum

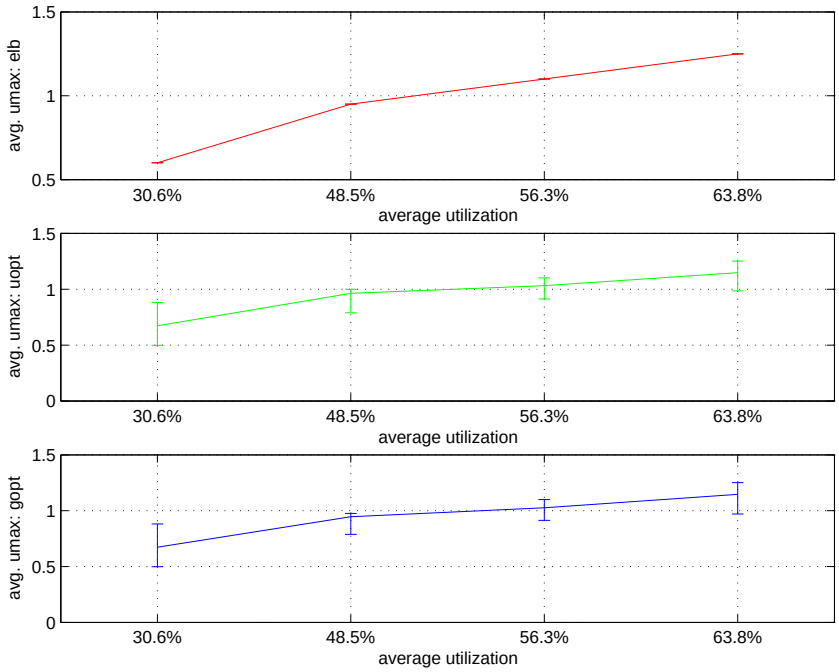


Fig. 50.: Maximum link utilization, 8x2, asymmetric topology: average, minimum and maximum

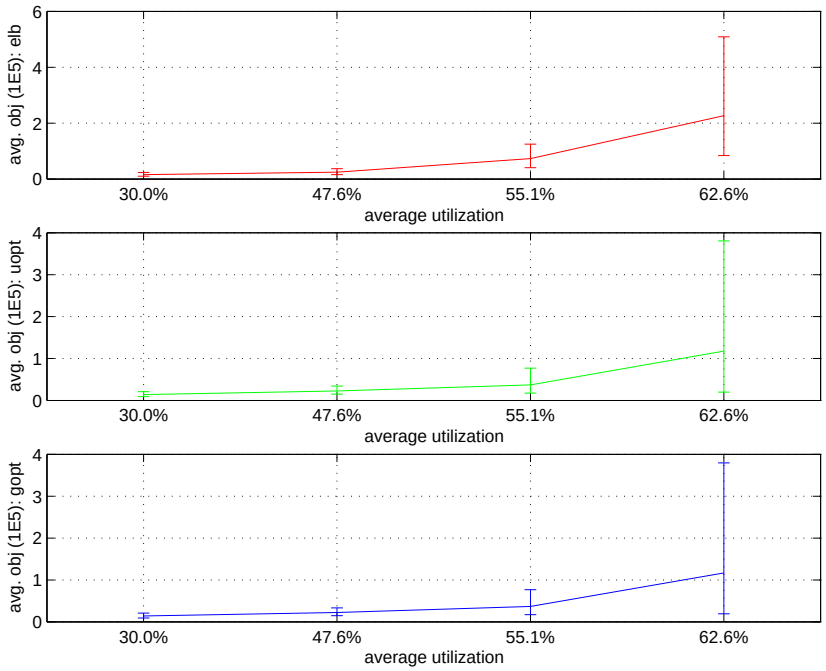


Fig. 51.: Optimization objective, 8x3, symmetric topology: average, minimum and maximum

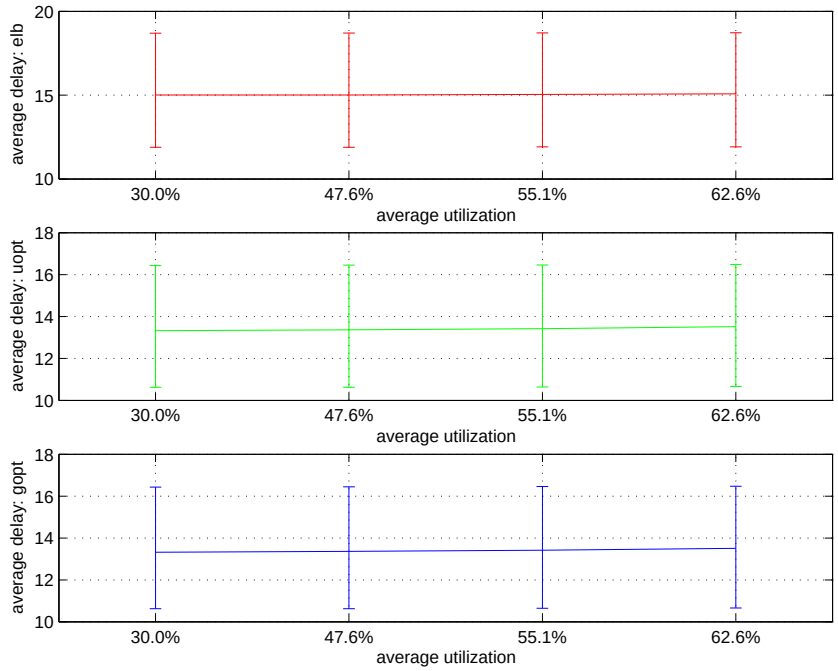


Fig. 52.: Average delay, 8x3, symmetric topology: average, minimum and maximum

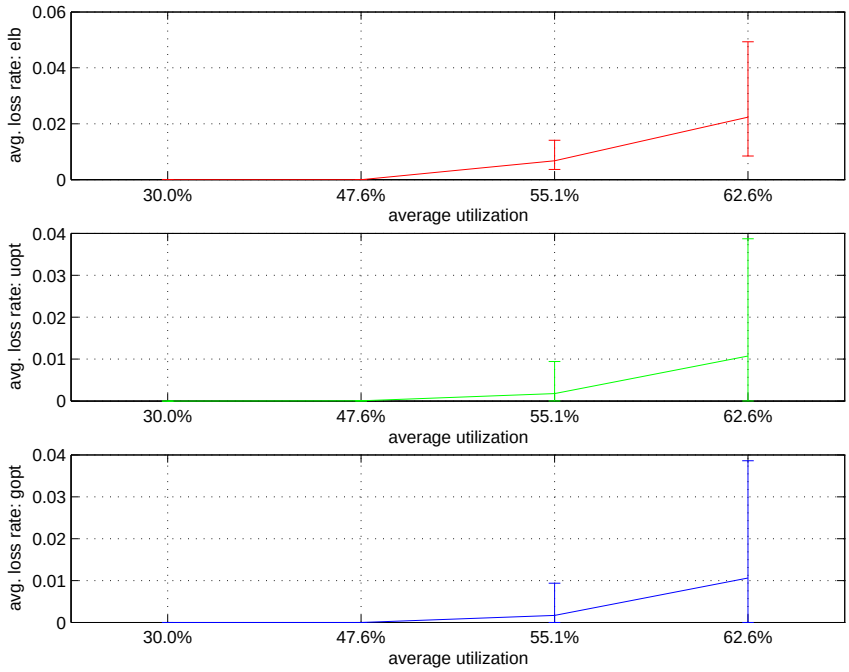


Fig. 53.: Average loss rate , 8x3, symmetric topology: average, minimum and maximum

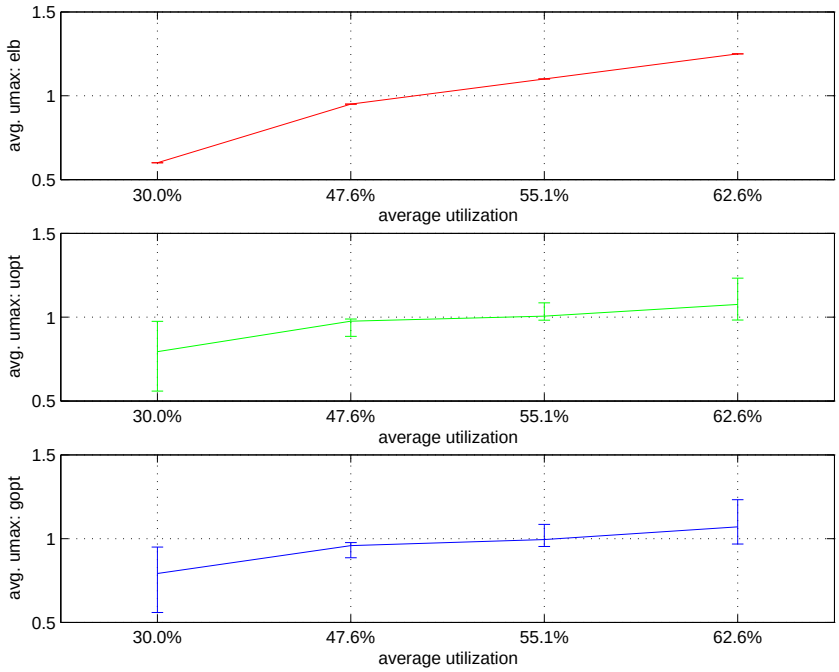


Fig. 54.: Maximum link utilization , 8x3, symmetric topology: average, minimum and maximum

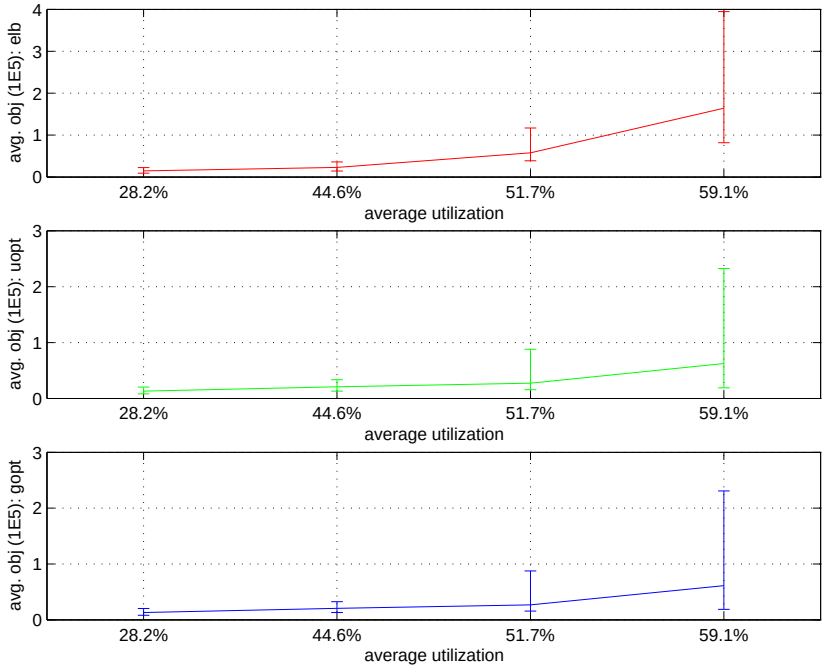


Fig. 55.: Optimization objective , 8x3, asymmetric topology: average, minimum and maximum

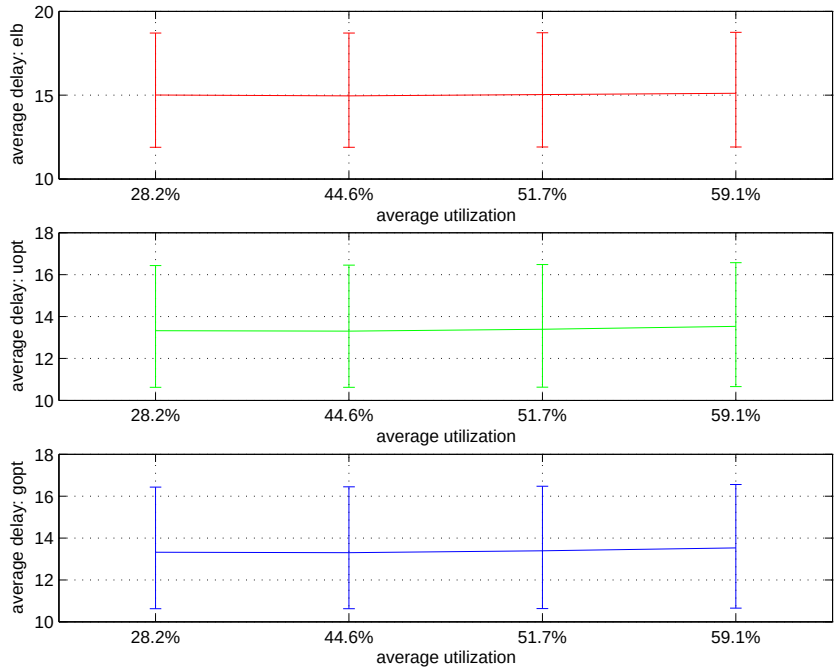


Fig. 56.: Average delay , 8x3, asymmetric topology: average, minimum and maximum

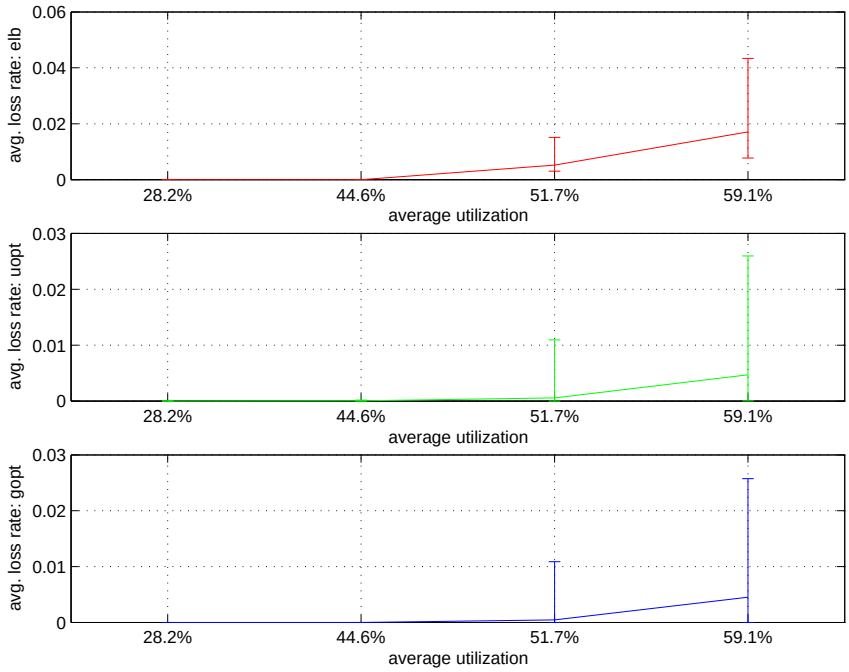


Fig. 57.: Average loss rate , 8x3, asymmetric topology: average, minimum and maximum

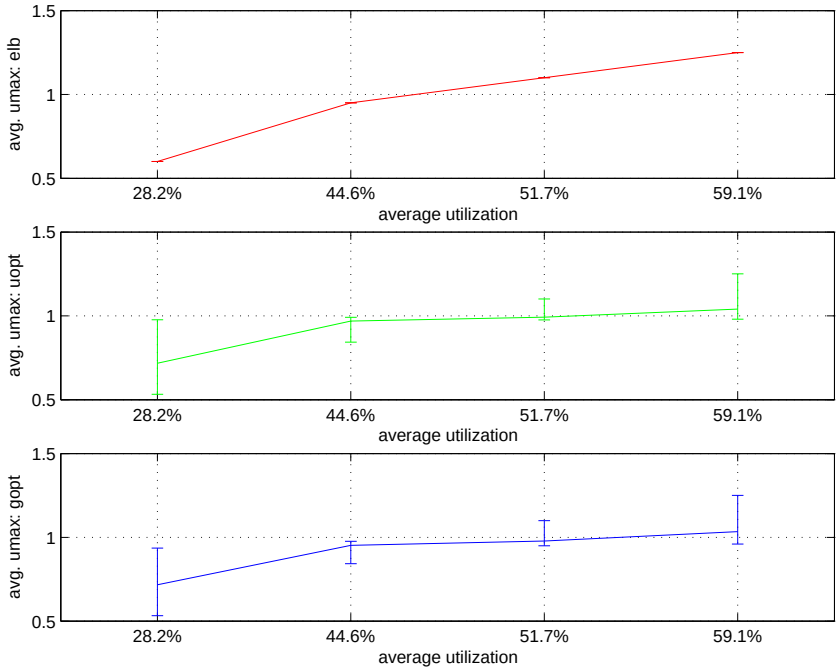


Fig. 58.: Maximum link utilization , 8x3, asymmetric topology: average, minimum and maximum

From the simulation results, we observe that the topology type (symmetric/asymmetric), queuing model type (Pareto/Poisson) do not make significant difference to the simulation results. The explanations are as follows: (1) Although packet arrival process under Pareto queuing model is more bursty than Poisson queuing model, the delay and loss rate are significant only near and above 100% utilization. When link is over-utilized, the delay and loss rate for both types of packet arrivals become very close. That is why we see similar performance for both kinds of queuing models in our simulations. (2) Because the Internet traffic on the links is not balanced, optimal routing and user-optimal routing can improve routing performance for symmetric topologies by balancing inhouse traffic.

For the queuing models we studied, queuing delay is very small from utilization of 0 up to utilization of 90%. Therefore, from Figs. 44, 48, 52 and 56, we can see the average delays for all three routing approaches change very little when average utilization increases from 30.6% to 63.8%. However, because of the differences in propagation delays of alternate paths, "uopt" and "gopt" can reduce the average delay by nearly 2 milliseconds.

From the figures of loss rates and maximum utilization, i.e. Figs. 45, 46, 49, 50, 53, 54, 57 and 58, we can see: (1) When average utilization is high (56.3% and 63.8%), the "uopt" and "gopt" based MRC reduce cost by reducing loss rate in networks. This is equivalent to reducing the maximum link utilization in networks. (2) When average utilization is low (30.6% and 48.5%), the maximum link utilization in networks of "uopt" and "gopt" is higher than "elb". The explanation is: for queuing models we are studying, increasing link utilization up to 90% will not cause packet losses or excessive queuing delays. Therefore, the delay and loss rate based optimal routing and user-optimal routing may cause higher maximum link utilization but lower than 100%.

The conclusions we draw here are: (1) The performance of both the approaches("uopt" and "gopt") are very similar, which is consistent with previous work [28]. (2) Because the delay and loss rate increase significantly as the link utilization approaches 100% or exceeds 100%, the performance gains of user-optimal routing based MRC increase with link utilization. (3) User-optimal routing based MRC can achieve lower delays and lower loss rates at the same time.

We also compare the routing performance of "gopt", "uopt" and "elb" for each simulation (one topology, one traffic matrix). For each simulation, we calculate following values: $\frac{\text{total cost of gopt(uopt)}}{\text{total cost of elb}}$, $\frac{\text{average delay of gopt(uopt)}}{\text{average delay of elb}}$, "total cost of gopt(uopt)" − "total cost of elb" and $\frac{\text{average max link utilization of gopt(uopt)}}{\text{average max link utilization of elb}}$. The average, minimum and maximum values for each configuration(same topology type, same queuing model, same network size, traffic matrices that resulted same maximum link utilization for "elb") are plotted in Figs. 59 to 74. From the figures, we observe that user-optimal routing based MRC improves performance in most cases while it is unable to improve performance for some topologies and traffic matrices.

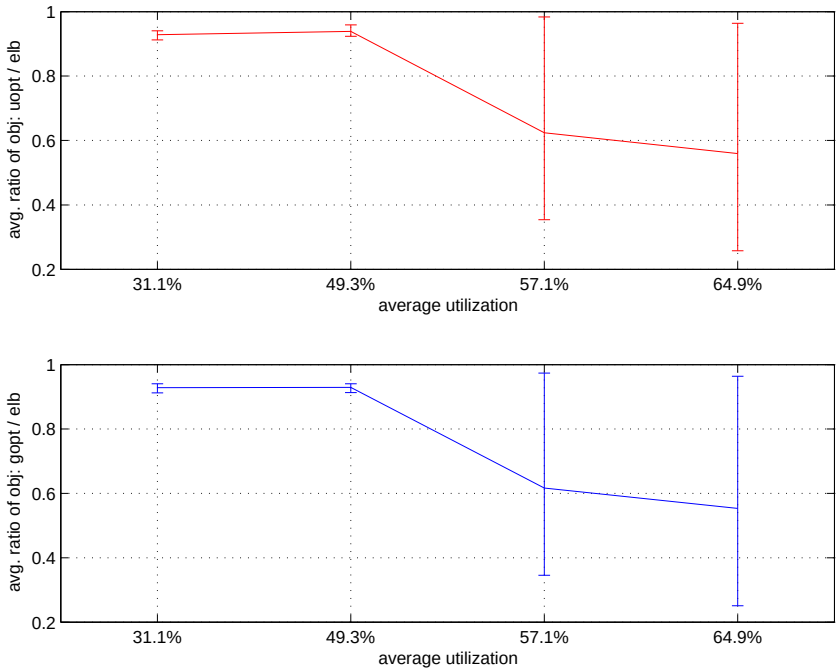


Fig. 59.: Optimization objective ratio(opt/elb), 8x2 symmetric topology: average, minimum and maximum

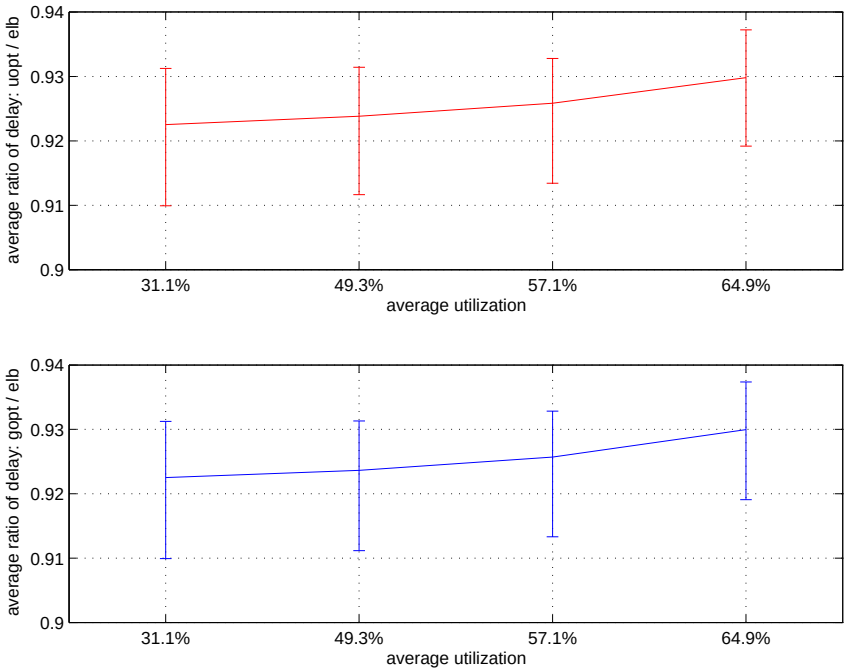


Fig. 60.: Average delay ratio(opt/elb), 8x2 symmetric topology: average, minimum and maximum

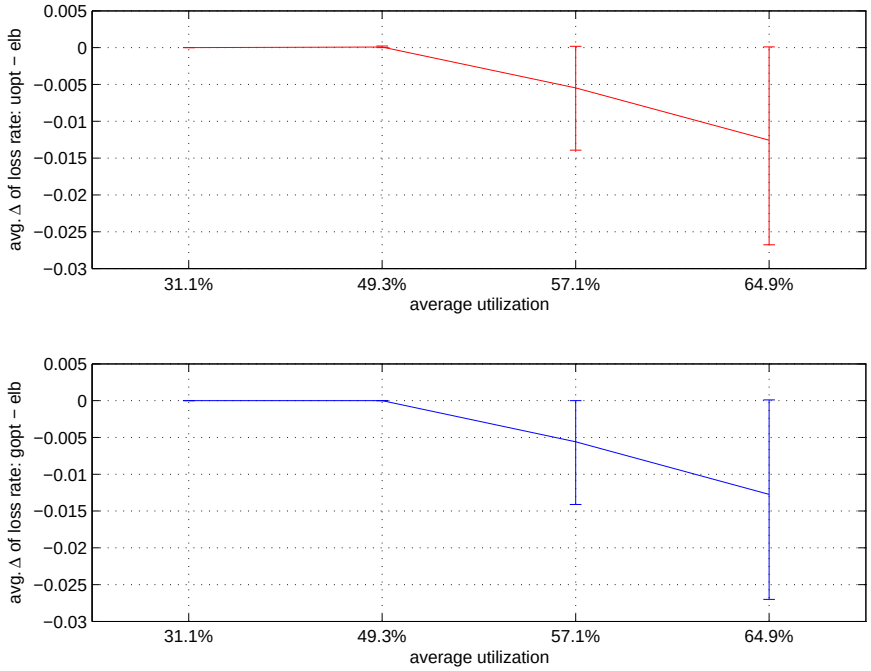


Fig. 61.: Average loss rate Δ(opt-elb), 8x2 symmetric topology: average, minimum and maximum

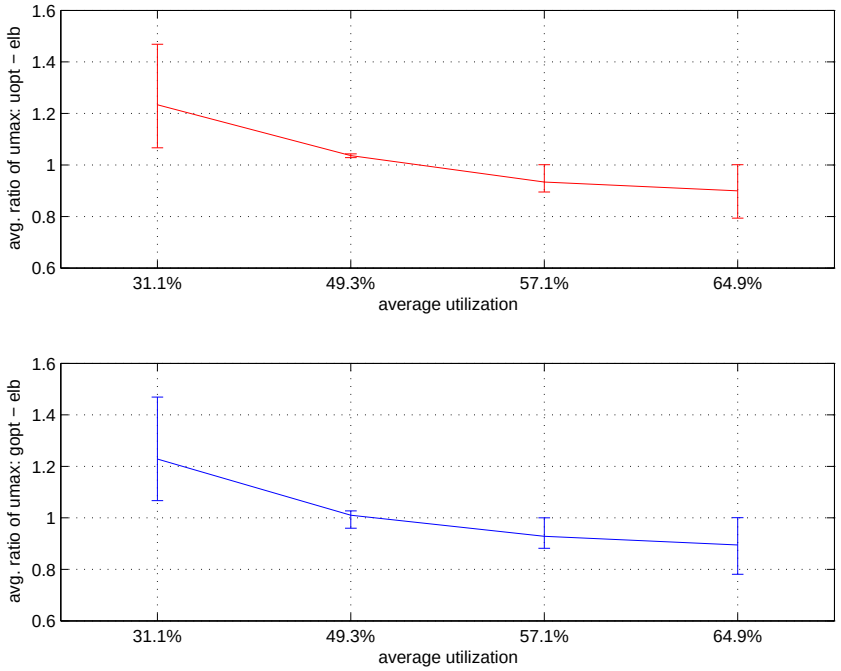


Fig. 62.: Maximum link utilization ratio(opt-elb), 8x2 symmetric topology: average, minimum and maximum

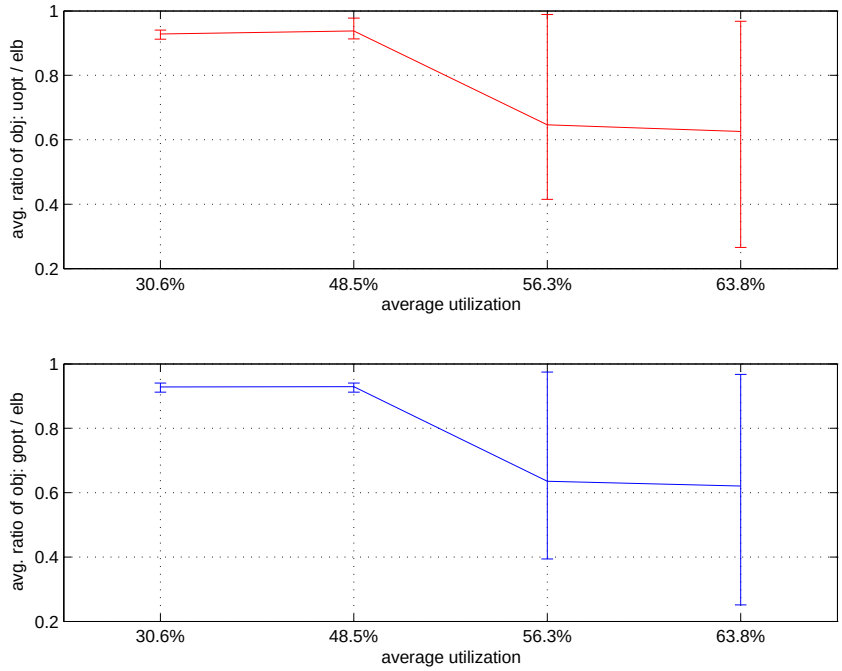


Fig. 63.: Optimization objective ratio(opt/elb), 8x2 asymmetric topology: average, minimum and maximum

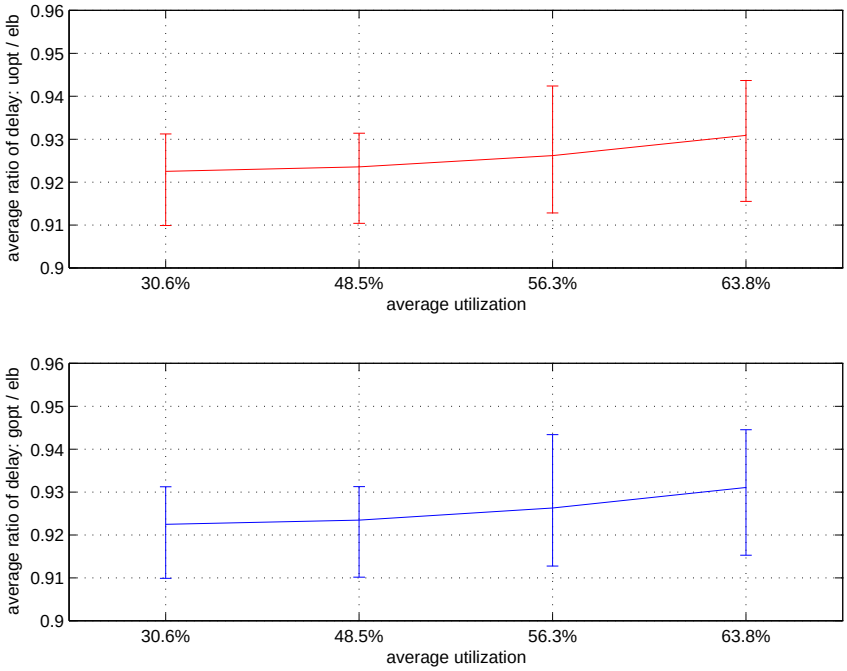


Fig. 64.: Average delay ratio(opt/elb), 8x2 asymmetric topology: average, minimum and maximum

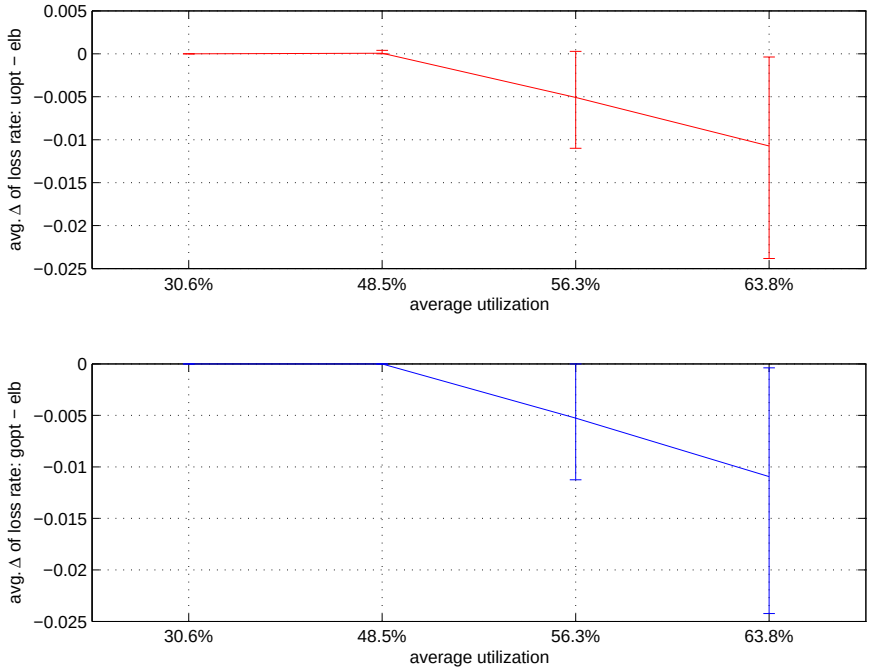


Fig. 65.: Average loss rate Δ(opt-elb), 8x2 asymmetric topology: average, minimum and maximum

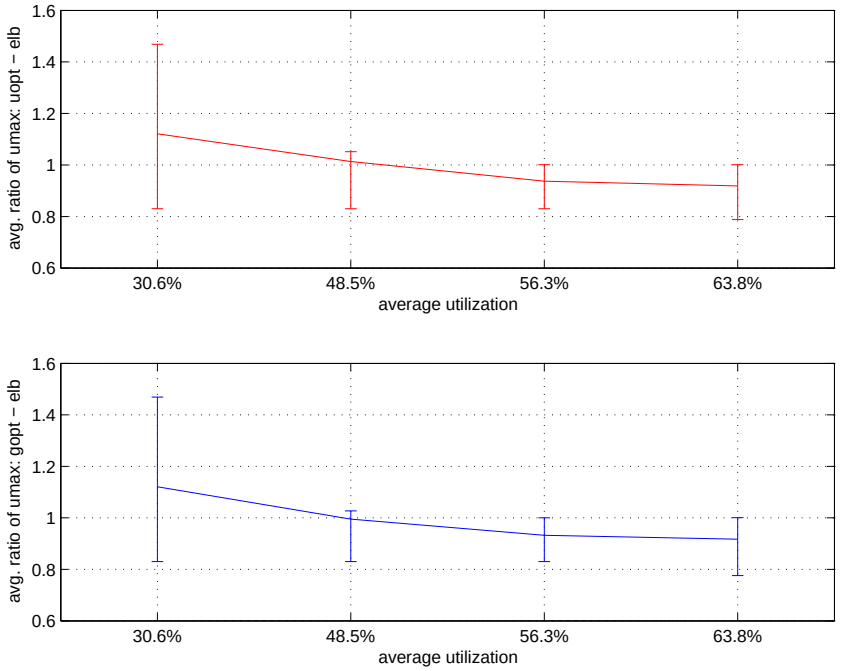


Fig. 66.: Maximum link utilization ratio(opt-elb), 8x2 asymmetric topology: average, minimum and maximum

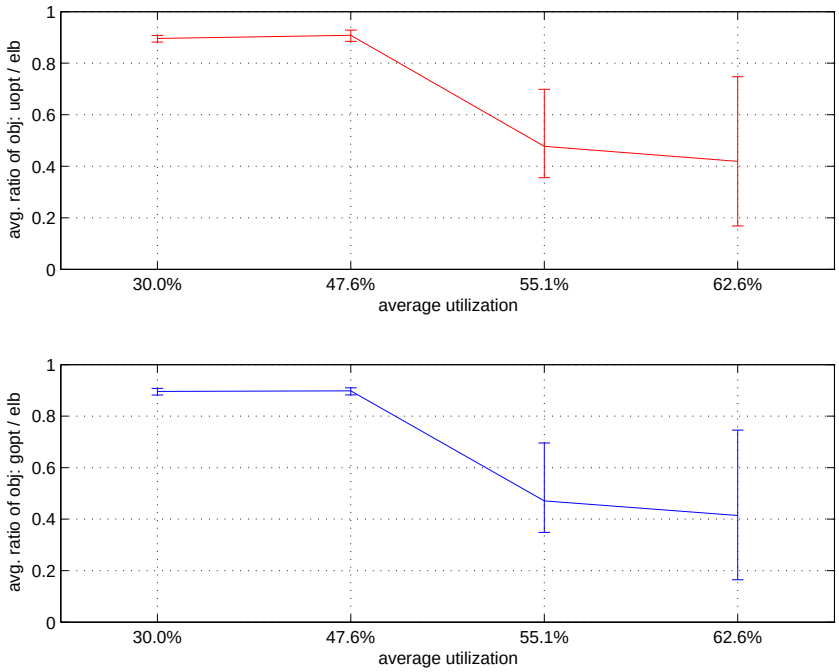


Fig. 67.: Optimization objective ratio(opt/elb), 8x3 symmetric topology: average, minimum and maximum

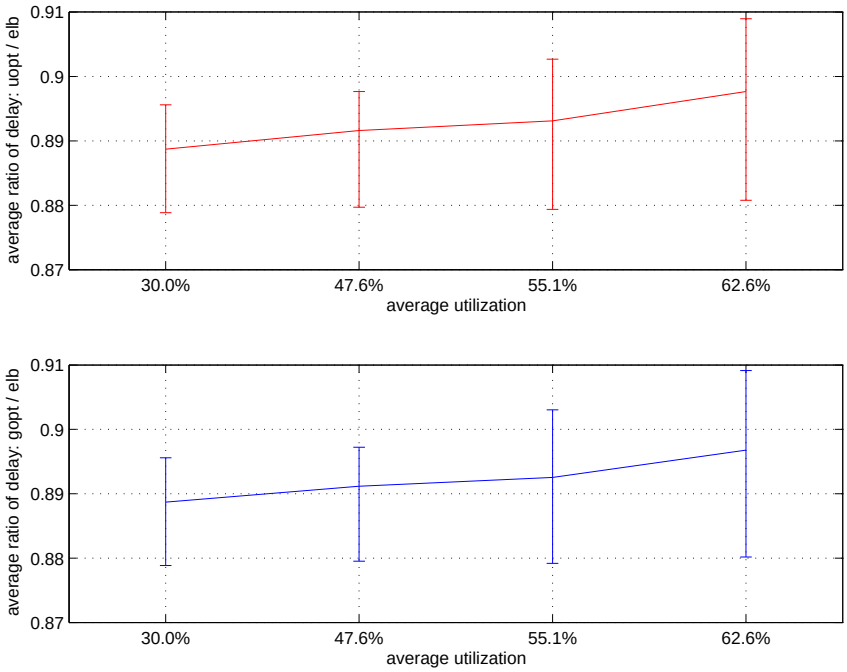


Fig. 68.: Average delay ratio(opt/elb), 8x3 symmetric topology: average, minimum and maximum

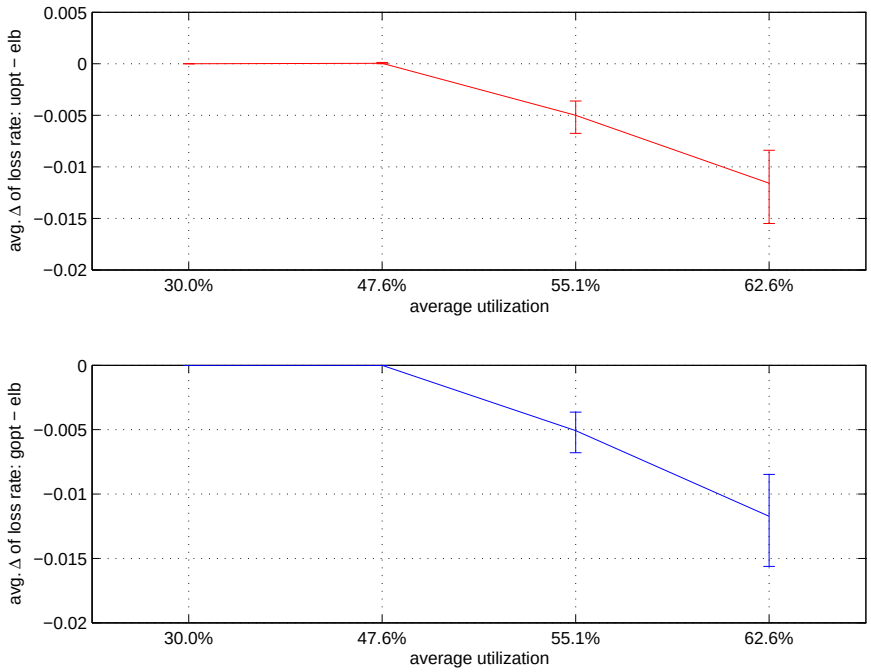


Fig. 69.: Average loss rate Δ(opt-elb), 8x3 symmetric topology: average, minimum and maximum

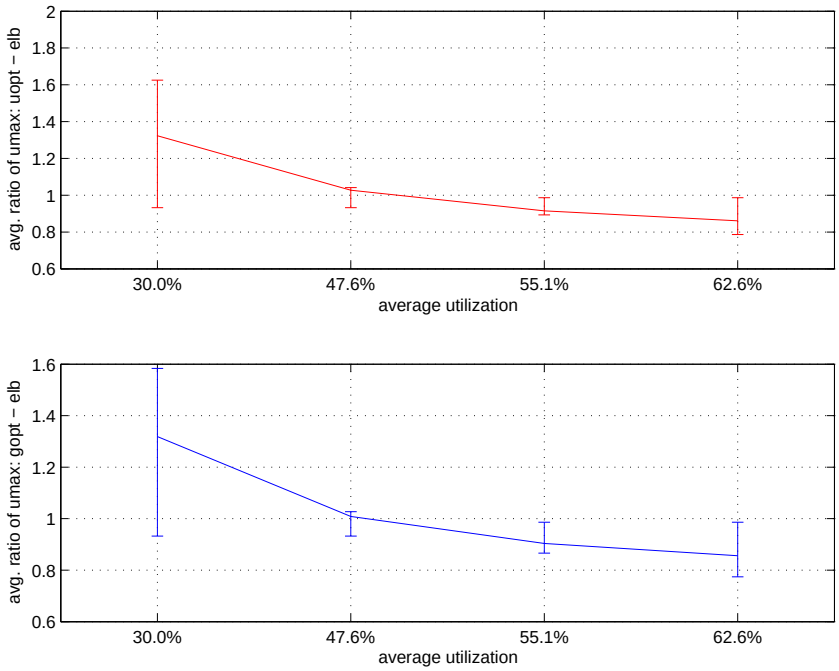


Fig. 70.: Maximum link utilization ratio(opt-elb), 8x3 symmetric topology: average, minimum and maximum

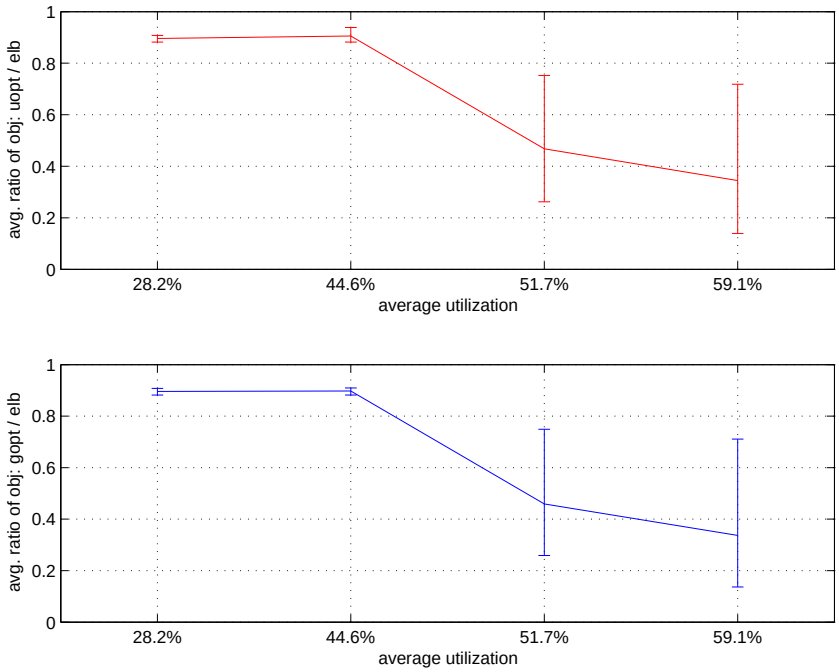


Fig. 71.: Optimization objective ratio(opt/elb), 8x3 asymmetric topology: average, minimum and maximum

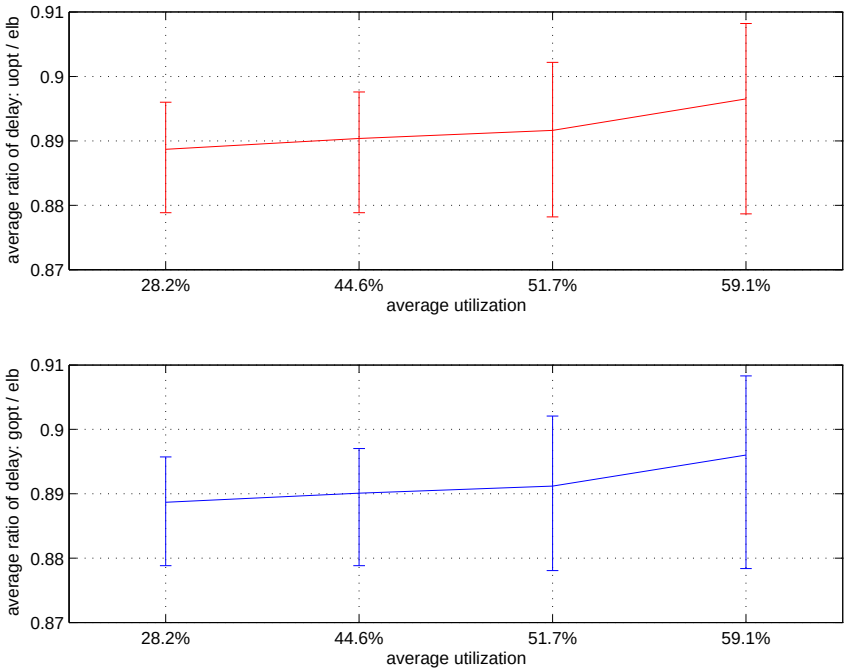


Fig. 72.: Average delay ratio(opt/elb), 8x3 asymmetric topology: average, minimum and maximum

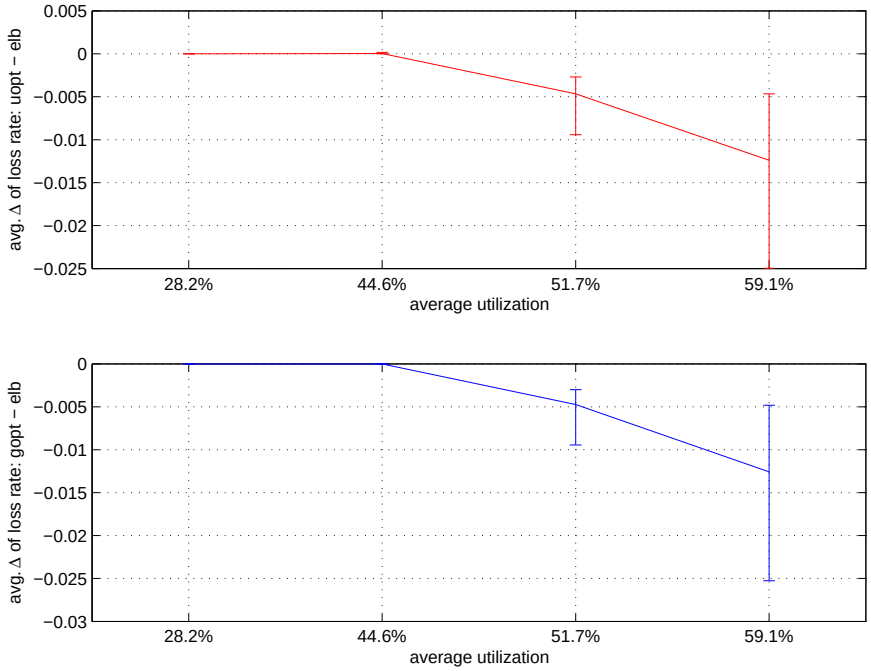


Fig. 73.: Average loss rate $\Delta$(opt-elb), 8x3 asymmetric topology: average, minimum and maximum

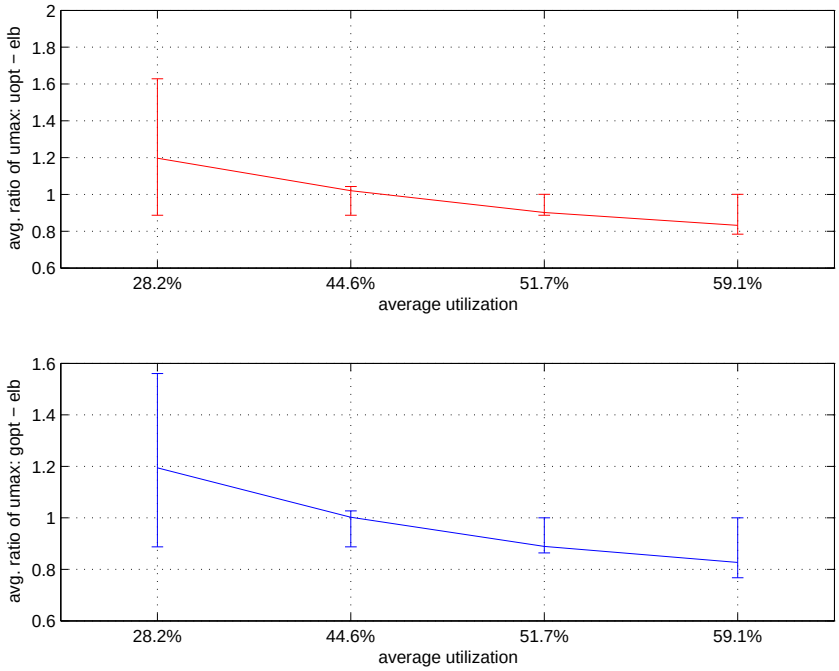


Fig. 74.: Maximum link utilization ratio(opt-elb), 8x3 asymmetric topology: average, minimum and maximum

### 2. Dynamic behavior

We study the convergence of our algorithm under following three types of scenarios: (1) We generate a random 8x3 topology and a random traffic matrix, assign initial routing allocation according to the "static load balancing" algorithm and let the algorithm converge to user-optimal equilibrium; (2) At the equilibrium point, we select 30% of paths among the stub networks and increase the delay of them by 50 ms and let the algorithm converge; (3) At the equilibrium point, we select 30% of paths among the stub networks and mark the path as disconnected and let the algorithm converge. Here, we define convergence as the state when the maximum difference of virtual delays of alternate paths used by traffic to one destination are not larger than 5 milliseconds. To converge to a state where the maximum difference is not larger than 1 millisecond or less, it takes more time but gets similar overall performance. In the simulations, the traffic matrix is scaled such that the maximum utilization for the "elb" routing approach is 110%.

Results of two sets of simulations are shown in Figs. 75 and 76. The different sub-figures in each figure correspond to the three scenarios (1), (2) and (3) explained above. We can see that the algorithm converges fast to a near-equilibrium point, in a few seconds. The convergence time for the "link failure" scenario is shorter than other scenarios because the algorithm responds to large virtual delay difference faster. The link failures cause traffic to be switched to other paths immediately and cause high virtual delay on some other paths.

We also study the effect of different step sizes on the convergence of the algorithm. The convergence from "static load balancing" to "user-optimal equilibrium" of one previous simulation (as shown in Fig. 75) is shown in Fig. 77. We can see that the algorithm converges quickly for several step sizes, converging faster with larger step

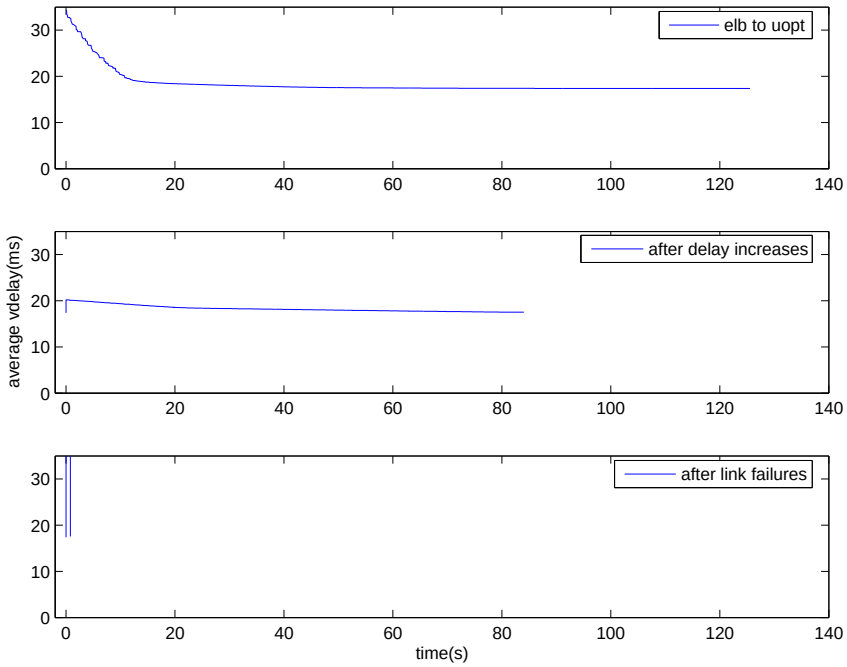


Fig. 75.: Convergence of user-optimal routing based MRC in different scenarios, simulation 1 (step size = 0.02)

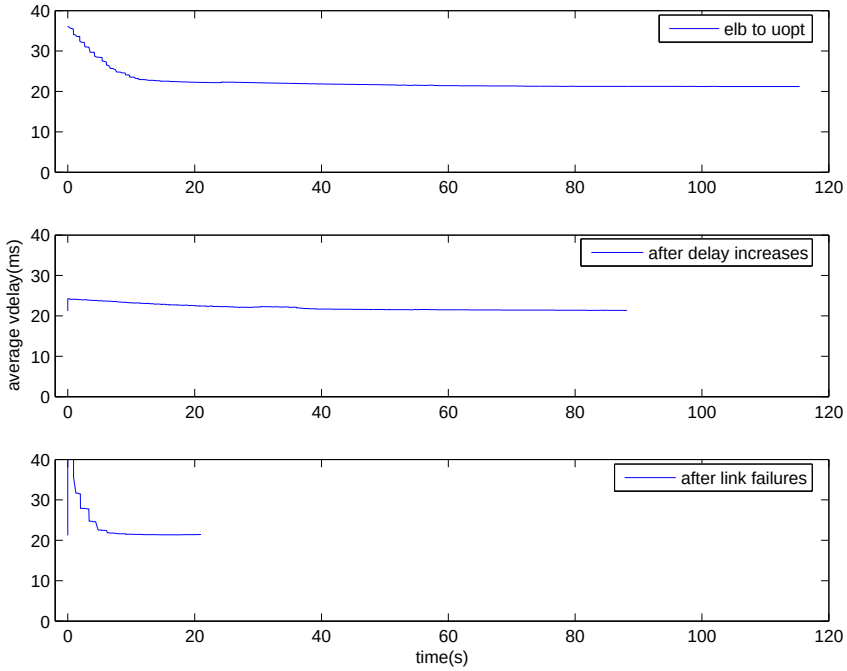


Fig. 76.: Convergence of user-optimal routing based MRC in different scenarios, simulation 2 (step size = 0.02)

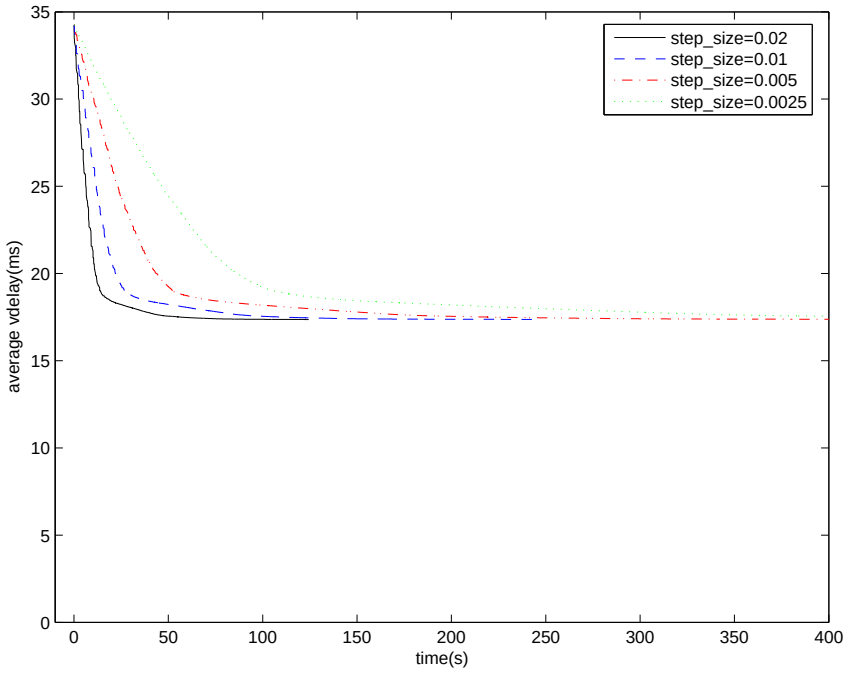


Fig. 77.: Effect of different step sizes

sizes.

## G. Conclusions and future work

In this paper, we have studied Multihoming Route Control(MRC) among a group of multihomed stub networks. We proposed a user-optimal routing based distributed Multihoming Route Control scheme that is simple to implement. We have shown through extensive simulations that the proposed MRC scheme improves performance in various network conditions without any oscillations. We have also shown that the user-optimal routing algorithm converges reasonably fast and achieves performance close to that of global optimal routing.

In this work, we assume the errors of measurement can be smoothed out. A stochastic approximation [58] analysis of the convergence of our algorithm when measurement results are corrupted by errors is desirable.

Although fixed step size is good for implementation of distributed MRC, properly decided adaptive step size may increase the convergence speed of the user-optimal routing based MRC.

# CHAPTER VI

# MULTIHOMING ROUTE CONTROL OF HIGHLY DYNAMIC TCP TRAFFIC

## A. Introduction

In the previous chapter, we proposed a user-optimal routing based multihoming route control algorithm that can avoid possible oscillations caused by uncoordinated multihoming route control. In that study, we assumed network traffic is highly multiplexed, therefore the packet inter-arrival processes can be modeled using a heavy-tailed distribution, e.g. Pareto distribution. We also assumed the traffic is UDP type traffic, i.e. the traffic demand is fixed regardless of routing changes. The above assumptions are based on previous work on Internet traffic analysis [30] and are used by network routing research community in previous work [28].

In this chapter, we study the case where access links have limited capacity and traffic consists of TCP flows, which is a more accurate model of traffic on access links with limited capacity. In this case, the instant traffic volume changes more rapidly because of TCP's congestion control mechanism, which increases the difficulty in designing an effective MRC approach. To study MRC in such situation accurately, we adopt packet level simulations in this study using the ns-2 [17] simulator. Using packet level simulations, we will also be able to study implementation issues of our MRC approach. The two implementation issues we will study in this chapter are: (1) measurement errors and the delays in observing the effect of routing changes and (2) effect of packet reordering caused by multi-path MRC.

We first study the link characteristics of bottleneck links when traffic consists of TCP flows of different sizes. The link characteristics we considered are: loss rate, queuing delay and available bandwidth. Through the simulations, we want to get a

better understanding of the relationships between the different link characteristics. This is important for our measurement based MRC. It helps us to select measurement metrics used in our MRC approach. While there is much previous work on Internet path quality measurement and prediction [31, 33], most of them are experiments on Internet and focus on general Internet path quality.

Second, we analyze the design of MRC algorithm for highly dynamic TCP traffic based on our study of link characteristics and propose our MRC scheme for highly dynamic TCP traffic.

Third, we study the effect of different parameters and algorithm choices for measurement based adaptive multihoming route control through simulations.

## B. Link characteristics

In this section we study link characteristics on bottleneck access links since they determine the end to end routing quality. First, we study the dynamics of link characteristics that determines how often an adaptive MRC scheme should change its routing behavior. Second, we study the predictability of different quality metrics since our measurement based adaptive MRC relies on predicted path quality. Third, we show that the quality differences between two bottleneck links that determine the extent of performance improvement of adaptive MRC schemes.

### 1. Simulation setup

We perform ns-2 simulations on the topology shown in Fig. 78, where $A_1, A_2, \cdots, A_{10}$ are traffic sources, $B$ is the traffic destination, $R_i$ is the edge router of $A_i's$ ISP ($i = 1, 2, \cdots, 10$), $R_{100}$ is the edge router of $B's$ ISP. The delay and bandwidth of the access links and Internet paths are labeled in the figure. We set the delay from

$R_i(i = 1, 2, \cdots, 10)$ to $R_{100}$ as $X$ milliseconds, where $X \in [5, 30]$.

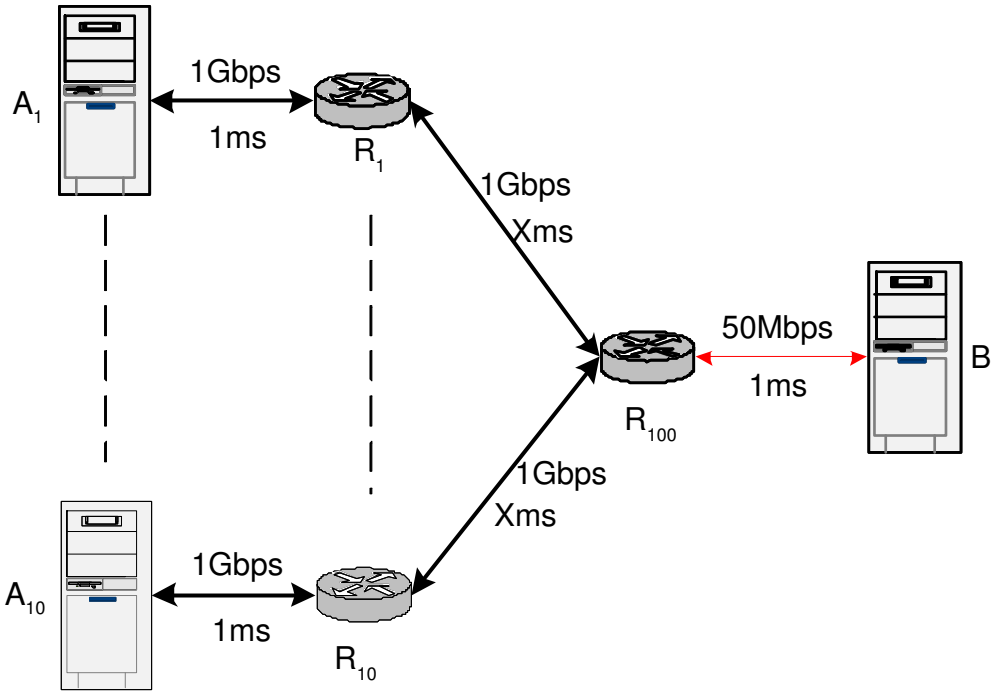


Fig. 78.: Topology for link quality predictability study

We generate TCP transfers from $A_i(i = 1, 2, \cdots, 10)$ to $B$ and study link characteristics of the access link of $B$. The flow sizes are distributed according to power-law models which are a more accurate model of Internet traffic [59]. According to previous measurement study on Internet traffic [59], the distribution of FTP-DATA "burst" sizes are heavy tailed, while flow inter-arrival interval distribution is still unknown but not Poisson. Accordingly, in our simulation, the sizes of the flows are distributed according to Pareto distribution and flow inter-arrival intervals are distributed according to Pareto-II(Lomax) distribution. The shape of the Pareto distribution of flow sizes is 1.1, while the minimum value of the distribution is uniformly distributed between 4KB and 36KB for every 100 flows. The shape of the Pareto-II distribution of flow inter-arrivals is 1.5, while the average is calculated from the target load.

## 2. Simulation results and analysis

We run each simulation for 2000 seconds. The three characteristics, queuing delay, packet loss rate and load on the access link when average link utilization is 40%, 60% and 80% are plotted in Figs. 79 to 81. From the figure, we observe the three link characteristics change rapidly over time.

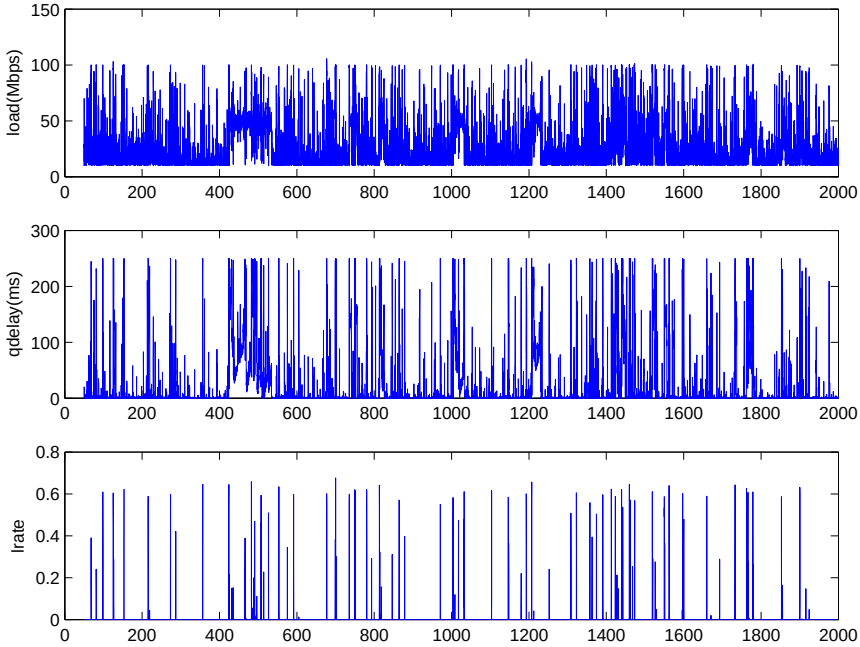


Fig. 79.: Example of Internet path quality: average utilization of 40%

We also study the dynamics of quality differences between alternate paths in a simple case, where only one site performs MRC and the background traffic are statically routed. In this case the background traffic on multiple access links can be assumed independent. We expect the "quality differences" between alternate paths change over time. To verify, we repeat the simulation with traffic load generated using a different random seed. The quality differences between two bottlenecks are plotted in Fig. 82. The results have confirmed our expectation. Therefore, an adaptive MRC has potential to adaptively switch traffic to a path with better quality. Another affect

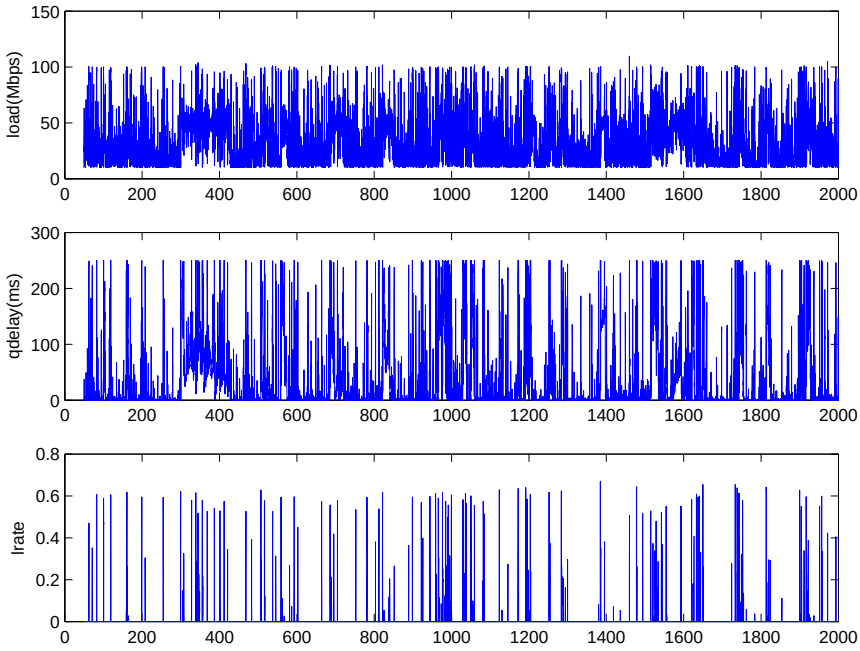


Fig. 80.: Example of Internet path quality: average utilization of 60%

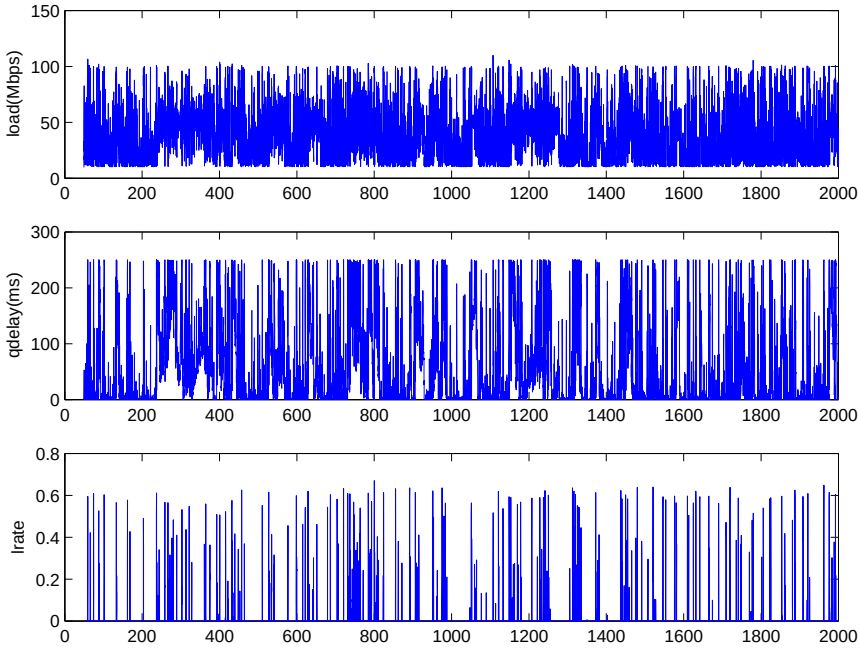


Fig. 81.: Example of Internet path quality: average utilization of 80%

of fast changes of path qualities is that we may need to switch long-lived TCP flows in order to achieve better performance. When the "background traffic" is also adaptively routed, the analysis of the dynamics of quality differences will become more complex. In this work, we will not study the later case directly, but will study the performance of MRC when there are multiple site performing MRC.

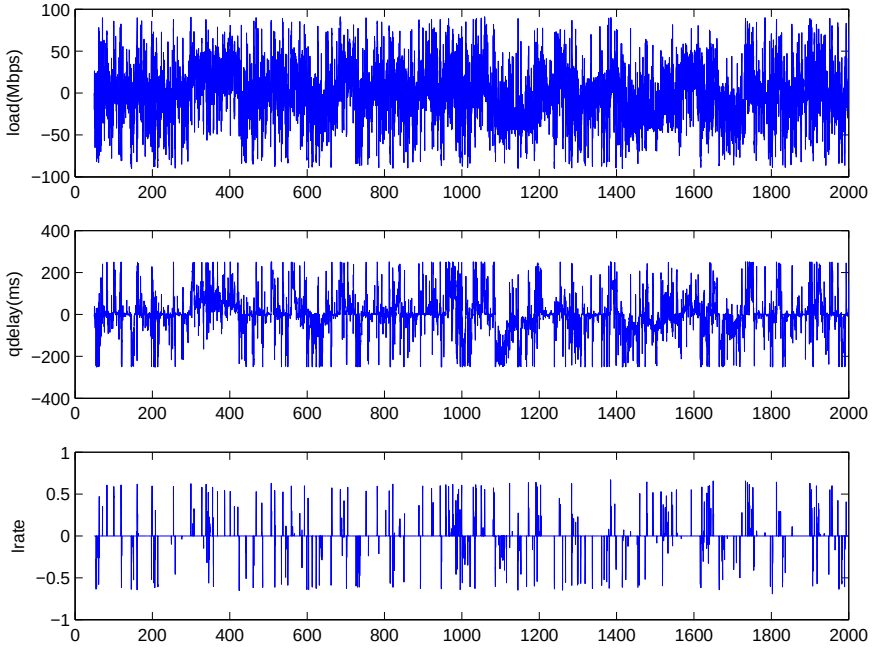


Fig. 82.: Internet path quality differences: average utilization of 60%

To quantify the dynamics of the link characteristics, we calculate the autocorrelation functions for the three quality metrics. The autocorrelation functions under different utilization are plotted in Figs. 83 to 85. We also calculate the cross-correlation functions between load and queuing delay, between load and loss rate, between queuing delay and loss rate. and plot in Figs. 86 to 88.

According to the property of autocorrelation function, the more slowly the autocorrelation function decreases as lag increases, the more slowly the characteristic changes and the more predictable the characteristic is. Similarly, a larger cross-correlation function means the characteristic is more predictable from the other char-

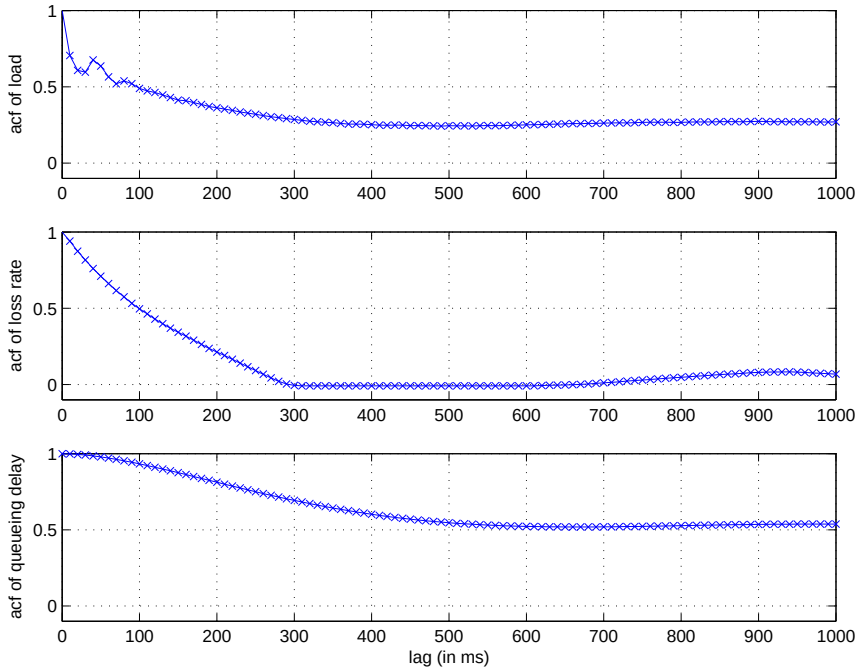


Fig. 83.: Autocorrelation functions of link quality metrics: average utilization of 40%

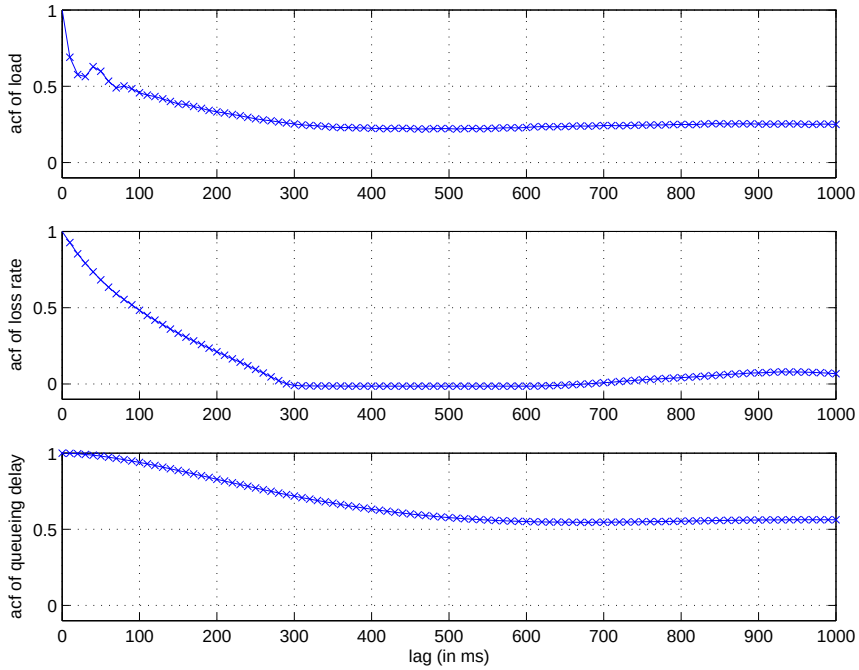


Fig. 84.: Autocorrelation functions of link quality metrics: average utilization of 60%

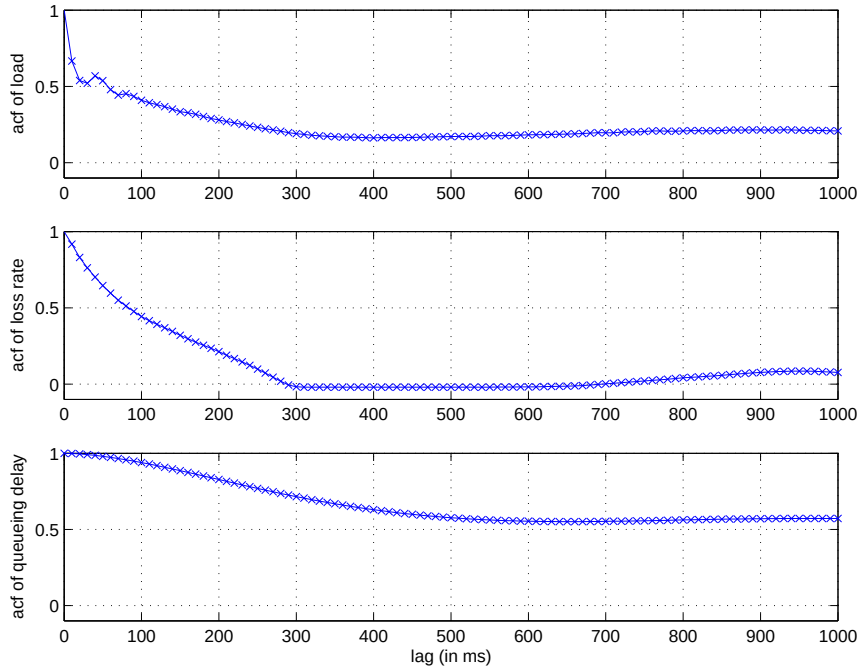


Fig. 85.: Autocorrelation functions of link quality metrics: average utilization of 80%

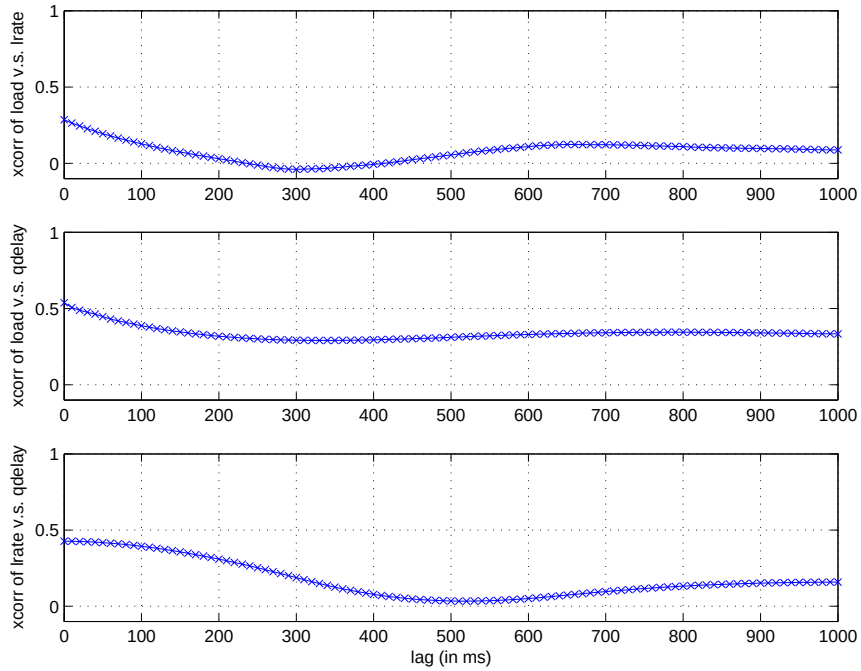


Fig. 86.: Cross correlation functions of link quality metrics: average utilization of 40%

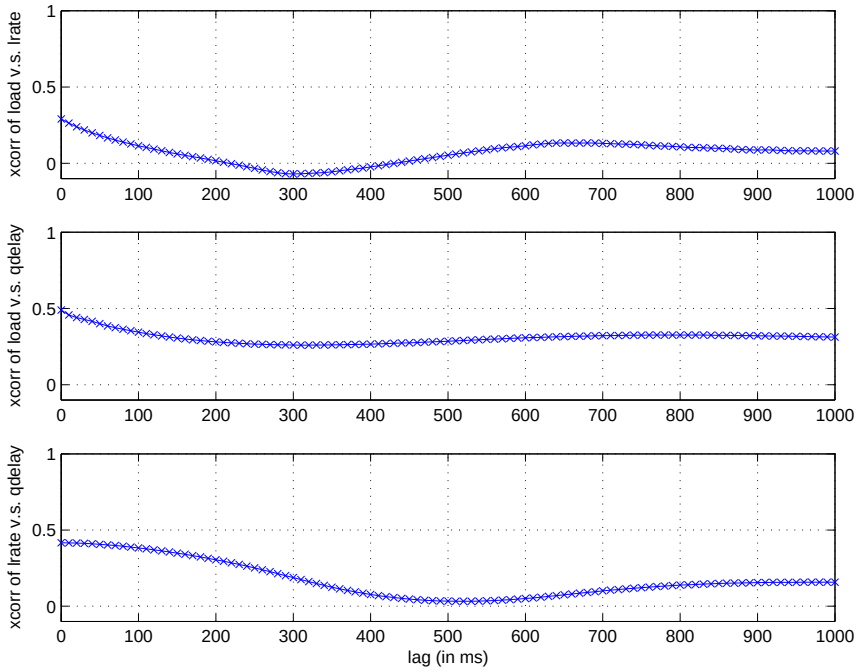


Fig. 87.: Cross correlation functions of link quality metrics: average utilization of 60%

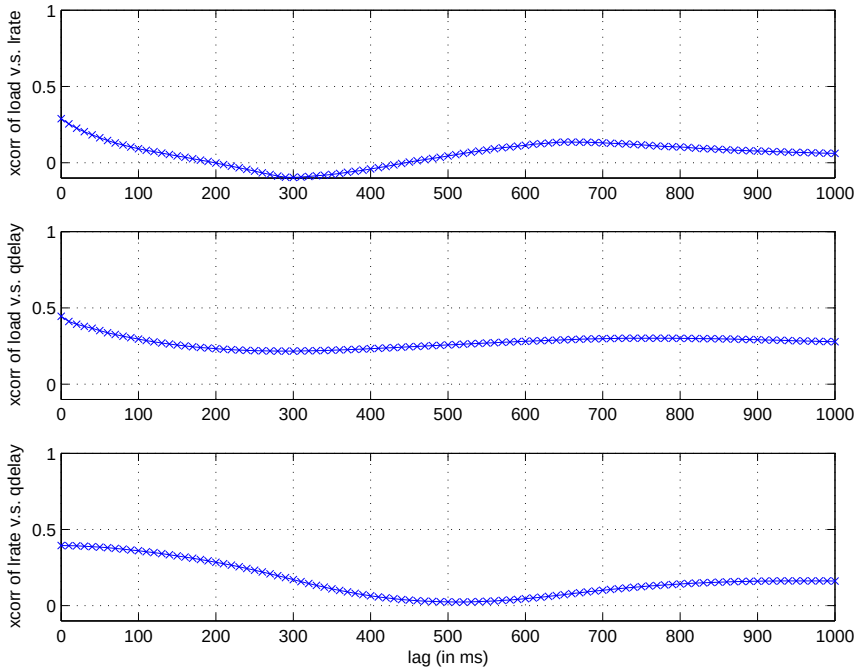


Fig. 88.: Cross correlation functions of link quality metrics: average utilization of 80%

acteristic at a different moment.

Therefore, according to the autocorrelation functions, queuing delay is the most predictable in that the value at a moment is highly correlated with its values within 100ms. load and loss rates are less predictable than queuing delay. The autocorrelation function decreases to about 0.7 within 30ms for both load and loss rate measurements. According to the cross correlation functions, we can see that load, queuing delay and loss rate are correlated, but their correlations are very weak. It is not easy to predict another metric from measurement of one metric. Similar results are observed for average utilization of 40% and 80% and when the minimum flow size is changed to 1MB.

We also calculate autocorrelation and cross-correlation functions of quality differences for utilization of 60%, as plotted in Figs. 89 and 90.

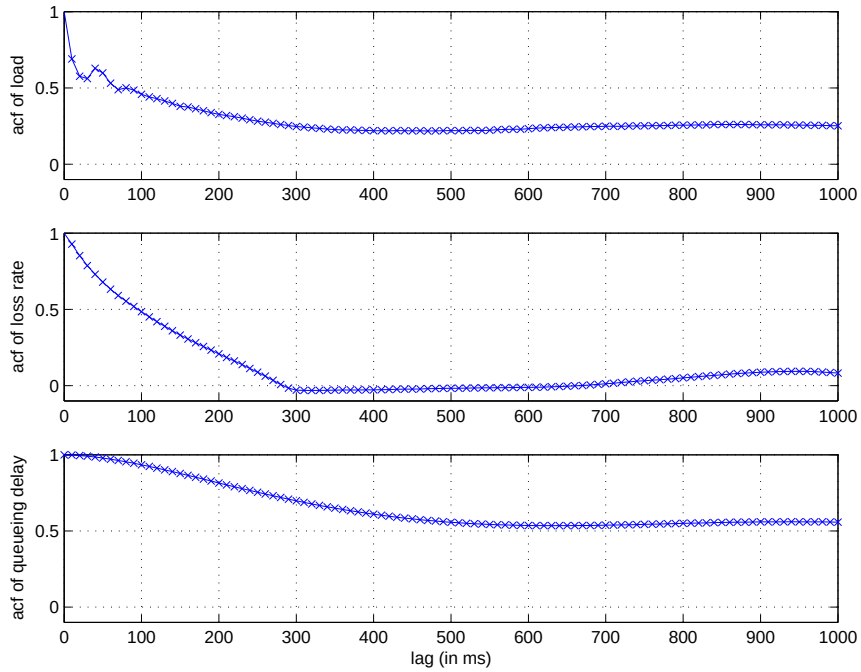


Fig. 89.: Autocorrelation function of Internet path quality differences: average utilization of 60%

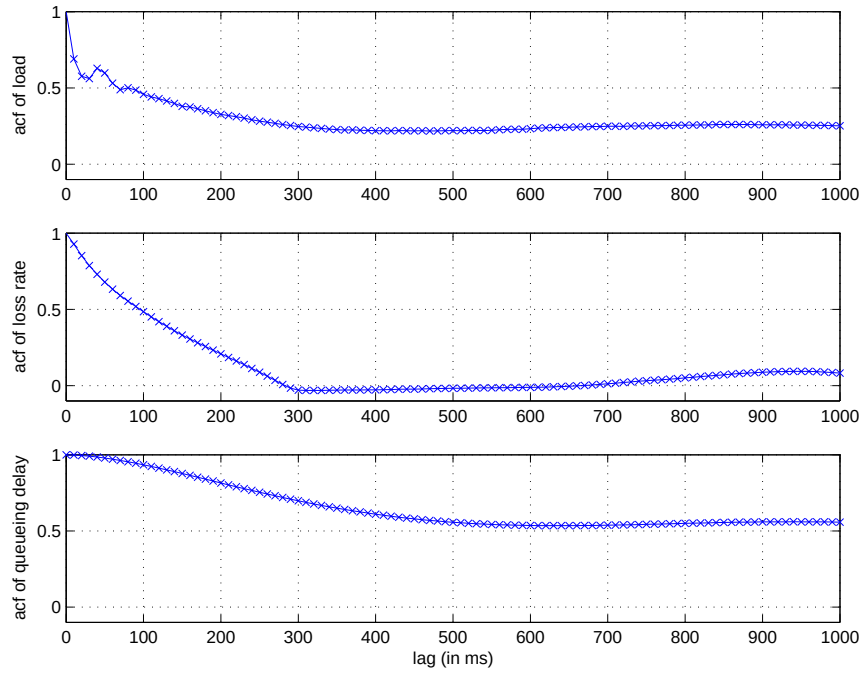


Fig. 90.: Cross-correlation function of Internet path quality differences: average utilization of 60%

From Figs. 89 and 90 we observe that the link quality differences change at similar time scale as link quality changes. It means that an adaptive MRC device should adapt it routing behavior at the same scale as the the link characteristics changes.

According to the above results, all the three characteristics are changing rapidly. Queuing delay is easier to predict than the other two metrics. Queuing delay is also more easily measurable than the other two metrics which allows faster response to link quality changes. Because of above reasons, in this work, we choose to use queuing delays as the quality metric. Specifically, we measure sum of queuing delays along a path.

To measure the sum of queuing delays along a path, ICMP timestamp request packets can be used. Using this, the one way delay can be calculated as

$remote\_node\_receiving\_time - local\_node\_sending\_time$. The sum of queuing delay can be obtained by subtracting the minimum one way delay from the current measured one way delay. This method also avoids the requirement of clock synchronization between remote and local nodes.

For inhouse traffic, each site needs only to control egress traffic. For Internet traffic, using BGP based Multihoming, we only have control of egress traffic. Therefore we need to measure qualities of forwarding paths.

## C. MRC of highly dynamic TCP traffic

### 1. Greedy MRC for highly dynamic TCP traffic

The results in section B show that MRC for highly dynamic TCP traffic should work at time scales of tens of milliseconds. This is a significant change compared to our study in last a few chapters where MRC is done every second or longer.

The fundamental reason that MRC causes oscillations in the situations we study in last a few chapters is: switching traffic to one path from other paths may cause some links on the first path congested, and the quality of the path may become worse than other paths. According to a concurrent study of MRC oscillations [60], the oscillations can be further classified into two categories: (1). When there is only one adaptive MRC sender and the switching traffic of this MRC device causes oscillations, the reason is "self-load" effect. In this case, the MRC device fails to consider the effect on the new path of its switched traffic. (2) When switching made by multiple MRC devices cause oscillations, the reason is "synchronization of multiple MRC devices".

In last chapter, a user-optimal routing based MRC was proposed to avoid oscillations in the above two cases. The basic idea is to switch traffic smoothly while measuring the path quality.

The change of MRC adaptation time scale in this chapter makes it necessary to reevaluate the oscillation problem for following reasons:

1. Switching periods of tens of milliseconds are comparable to queuing delays on a bottleneck link, whose maximum value could be 250 milliseconds. If a "greedy MRC device" switch traffic in such small time scales, it can detect the increase of queuing delays on the congested link and switch traffic to other paths before packet are dropped on the congested link. This way, the MRC device switches traffic so fast that it almost does not cause packet drops on any path. This actually avoid the "self-load" effect of MRC approach works in coarse timescales. Although the MRC device might keep switch traffic from path to path, it is different from the "oscillations" we studied before, where the oscillations cause packet drops and hurt end to end performance. As we will show in Section D, this greedy MRC approach can improve end to end performance.

   We classifies this approach as a multipath routing approach because it works very similarly as a multipath routing scheme. The difference is that in a multipath routing approach, a packet is the smallest unit that can be switched to alternate paths; in this greedy approach that works in timescales of tens of milliseconds, traffic routed in each period is the smallest unit that can be switched. Accordingly, the packet reordering issue of multipath routing should also be addressed in this approach. From the result of Section B, we observe that the link characteristics change rapidly. Therefore, a measurement based MRC scheme is expect to adapt the changes quickly enough, at the same time scale as the characteristics change. Even if single path routing is adopted by MRC devices, if the forwarding path is changed at 10 ms scale considering the queuing delays on access links, this approach still may cause packet reordering.

Since the traffic and path metrics change rapidly in this case, the greedy MRC is in fact multi-path MRC. Therefore, we need some mechanism to minimize the effect of packet reordering on performance of TCP. In this work, we study TCP-DCR and Flowlet.

2. When MRC is done in time scale of tens of milliseconds, the delays of observation of effects of MRC adaptations are not negligible anymore. It may actually avoid the synchronization of different MRC devices.

This greedy approach differs from BGP based greedy MRC in that it works in smaller time scales that is impossible for BGP based MRC. A fractional MRC switching engine is needed to implement this approach. The benefit of this approach is to further improve end to end performance for highly dynamic TCP traffic.

### 2. Implementation of greedy MRC

As we mentioned in Section C, greedy MRC for highly dynamic MRC works as a multipath routing scheme. Greedy MRC is simpler than user-optimal and optimal routing based MRC. Using the same framework for fractional MRC we proposed in Chapter III, we need only to change the algorithm of the "routing decision module". The new algorithm can be expressed as routing all traffic to one destination network to a path that has the best predicted quality for next period.

Similar to other multipath routing schemes, we need a mechanism to mitigate the affect of packet reordering to TCP and other communication protocols that relies on in-order delivery. In this study, we focus on performance of TCP applications which is the dominant transport protocol on the Internet.

As we introduced in Chapter III, there the choices to implement the forwarding engine of a fractional MRC are: flow level switching, packet level switching and

"Flowlet" switching. On the end hosts, packet reordering robust TCPs can be used to tolerate packet reordering caused by fractional MRC.

In this work, we will study the use of a variant of reordering robust TCPs, TCP-DCR, and different switching method in our MRC approach.

## D. Simulation study

### 1. Implementation of switching algorithms in simulations

We use a generic implementation to simulate different switching algorithms. The MRC algorithms we studied are the greedy MRC we proposed in this chapter and user-optimal routing based MRC we proposed in Chapter V. The switching algorithms we studied include flow level, packet level and "Flowlet" switching.

The implementation of switching algorithms is as follows. When a packet arrives, the router looks up the flow in a flow state table.

If the flow state is not in the table, the router assigns a path for the flow according to the routing algorithm. Then a new entry for that flow is created in the table. The flow ID and the route of the flow along with a timestamp of current time is recorded in the entry.

If the flow state is in the table, the timestamps of the entry is compared with current time. If the difference is larger than a given threshold, "active timeout", a new path is assigned to the flow according to the routing algorithm and the timestamp is updated as the current time. Otherwise, the packet is routed along the path stored in the entry.

The inactive flows are periodically removed from the table to reduce memory usage. Of course, for flows whose termination is detectable, the removal is made after the router forwarded the last packet of the flow.

By changing the "active timeout" threshold, we can simulate flow, packet and Flowlet level switching. When we set the "active timeout" as zero, the switching is packet level switching. When we set the "active timeout" as a "large" value, the switching is flow level switching. The value is decided by the maximum time gap between any two subsequent packets of the flow. When we set the "active timeout" as a value in between, the switching is "Flowlet" level switching.

The larger "active timeout" is, the more flow states need to be kept in the router. In our simulation, we set it as 1 second to simulate flow level switching.

To evaluate the performance of greedy MRC, we compare the flow completion times when $N1$ uses this approach with the flow completion times when $N1$ uses a static load balancing approach. The static load balancing approach we used is simply to assign a new flow to a path by equal probability. In our greedy approach, the routes for new flows are adjudged every 10 milliseconds according to the latest queuing delay measurement of alternate paths. The best path will be used for new flows coming in next 10 milliseconds.

## 2. Simulation setup

We study the performance of our MRC scheme on a number of many-to-one topologies using ns-2 simulations. Our simulation study consists of two parts:

In the first set of simulations, we perform simulations to study the cases where only one stub network does adaptive multihoming route control while all other stub networks use predefined static routing strategy. The analysis of this simple case gives us insight of performance of different techniques for adaptive multihoming route control. It is also the foundation for analysis of the more complex case where all stub networks do adaptive multihoming route control.

One topology we used in our study is shown in Fig. 91. In our simulation, stub

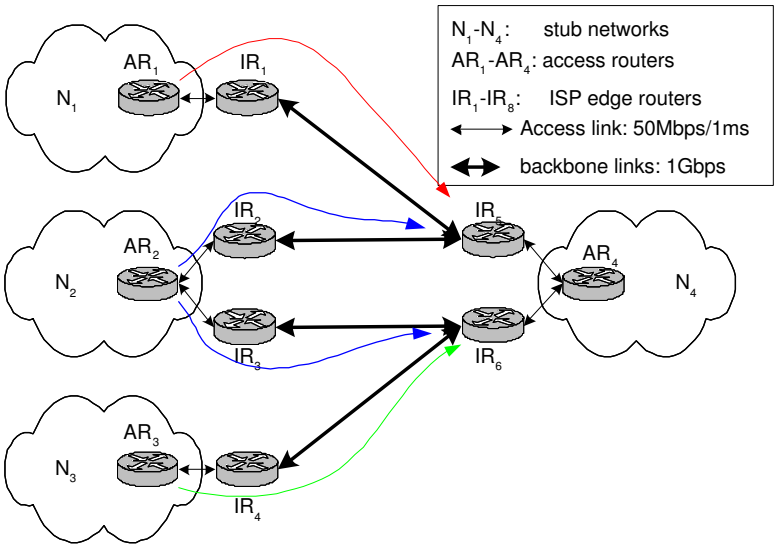


Fig. 91.: Simulation topology: 3 send to 1, one sender site uses MRC

network $N_1$, $N_2$ and $N_3$ send data using TCP protocols to stub network $N_4$. The local router of stub network $N_2$, $AR_2$, uses our greedy multihoming route control algorithm to route traffic on two alternate paths. $N_1$ and $N_3$ route traffic to $N_4$ via their single ISP routers. In other words, we use $N_1$ and $N_3$ to generate background traffic and study the performance of MRC of stub network $N_2$.

We generate Internet paths delays as $\alpha \times \beta$, where $\alpha$ models the physical distance of two stub networks and is uniformly distributed from 5ms to 30ms, $\beta$ models AS path differences of alternate paths provided by different ISPs and is uniformly distributed from 0.8 to 1.2. An Internet path here is the path from an ISP edge router of one stub network to an ISP edge router of another stub network. The propagation delays of the access links are set as 1 millisecond.

We also change the number of ISPs of the sender that adopts MRC and the receiver to 3 and 4. Accordingly, we add 1 and 2 sender stub networks, to generate background traffic on the added access links of the receiving stub network.

In the second set of simulations, we study the performance of our MRC scheme

when all sender sites use MRC. Through this set of simulations we study the interaction between MRC done by different stub networks and study the performance of MRC in such situations. The topology we used in our second part of the simulations is shown in Fig. 92. In the topology, $N_1$ to $N_9$ are sender stub networks that deploy our MRC scheme and $N_{10}$ is the receiver stub network. The Internet path delays are generated same as the first set of simulations. The topology plotted is an asymmetric topology, i.e. the path from one ISP edge router of a send stub network is randomly connected to a ISP edge router of the receiver stub network. The we also performed same simulations on symmetric topologies. There is no significant difference between the two type of topologies.

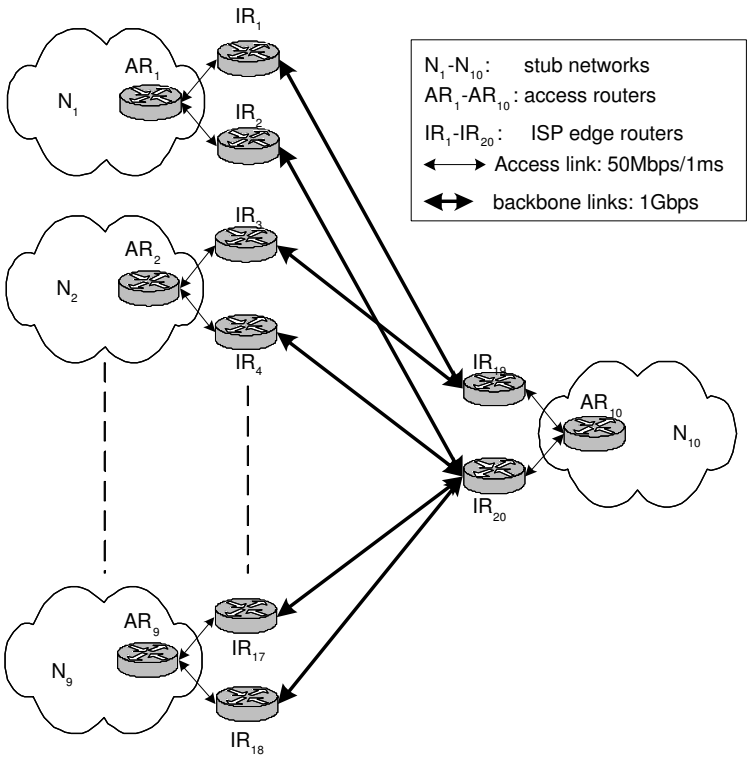


Fig. 92.: Simulation topology: 9 send to 1, all sender sites use MRC

In all our simulations, we generate traffic using the same method as we used in Section B.

To compare performance of different routing schemes, we define "average completion time" as our performance metric. Assuming the sizes of data transfers completed during a simulation are $s_i, (i = 1, \cdots, n)$ and the corresponding completion time are $t_i, (i = 1, \cdots, n)$, where $n$ is the number of flows completed, the "average completion time",

$$T = \frac{\sum_{i=1,\cdots,n} (\frac{t_i}{s_i} s_i)}{\sum_{i=1,\cdots,n} s_i} = \frac{\sum_{i=1,\cdots,n} t_i}{\sum_{i=1,\cdots,n} s_i}$$

3. Simulation result

a. Smoothing of measurement result

As we analyzed in Section B, we choose to use sum of the queuing delays along a path as the quality metric of a path. Since the measurements are noisy, we need some method to smooth the measured queuing delays. We evaluate two smoothing methods in this work: (1) using the average of measured values in the last period, i.e. $\hat{l_t} = \frac{1}{T}(l_{t-1} + \cdots + l_{t-T})$; (2) using the Exponential Weighted Moving Average (EWMA) of measured values, i.e. $\hat{l_t} = (1 - \alpha)\hat{l_{t-1}} + \alpha l_t$. We set $l_t$ as 1 second when measurement packet is dropped to take packet drops in to account.

We perform simulations using the configuration of the first set. We use packet level switching along with TCP-DCR in our simulations. The results are shown in Fig. 93.

From the results we have the following observations: (1) for the "last period" smoothing method (configuration 11 to 16), the shorter the averaging period the higher the improvement. This is consistent with our study of link quality in Section B where we show that the link characteristics change rapidly. (2) EWMA smoothing method (configuration 3 to 10) with $\alpha$ from 0.01 to 1.0 achieves similar result as the best "last period" which is what we expected since for all the $\alpha$'s the measurement

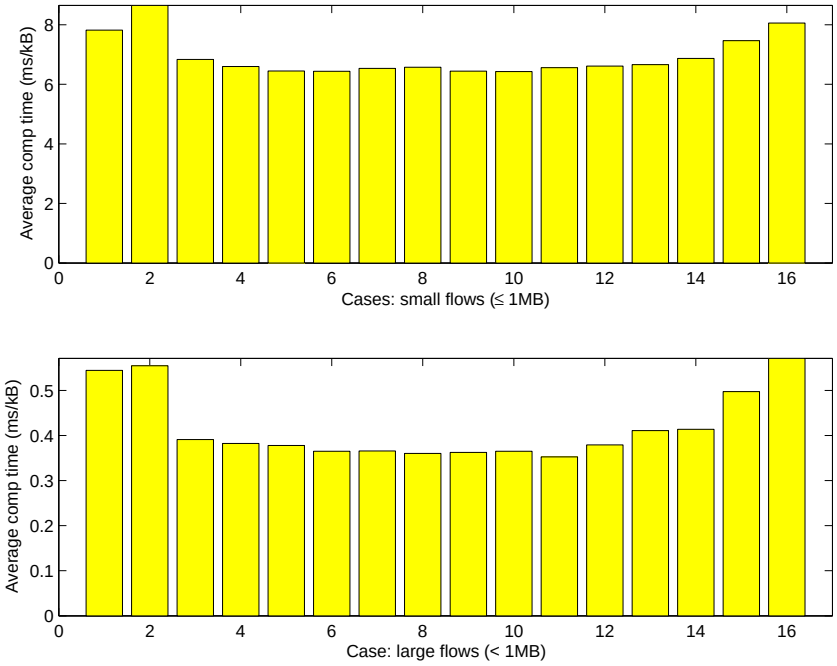


Fig. 93.: Performance of EWMA and last-period predictors: (1) elb-flow, (2) elb-packet, (3-10) ewma $\alpha$ = 0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1, (11-16) last period, period = 5ms, 10ms, 50ms, 100ms, 500ms, 1s

results in the last 10 milliseconds have higher weight, e.g. $(1-0.9)^{10} \approx 0.3487$. Since the two methods are comparable, to simplify our analysis, we use only the EWMA method in the remaining simulations.

b. Single MRC sender stub network: routing performance of the MRC site

The simulation results for routing performance of the MRC site when there is only one site performing MRC are shown in Figs. 94, 97 and 98, where the number of ISPs of MRC site and the destination site are 2, 3 and 4 respectively.

TCP works in two distinct modes: "slow start" and "congestion avoidance". TCP adapts its congestion window using different algorithms in the two modes which determine the throughput of TCP. The throughput of TCP in "slow start" phase is usually much less than its throughput in "congestion control" phase. Because of this, the throughput of small flows that spend larger proportion of their lives in "slow start" phase is usually smaller than the throughput of large flows. In this chapter, we classify flows with size less than one Megabyte as "small flows" and flows with size larger than or equal to 1 Megabytes as "large flows".

For the case when MRC devices connect to 2 ISPs, we simulate equal splitting based static routing (labeled as "elb" in the figure) and greedy MRC with different switching algorithms and two types of TCP: TCP-DCR and TCP-SACK. First, we run the simulation for 16 configurations twice with different random seeds. The results for the two runs are shown in Table II and Fig. 94 and Table III and Fig. 95.

We also record the total number of TCP events for the traffic of the MRC site in the simulations. The data 2 sets of simulations are listed in Tables IV and V

From the figure, we have the following observations:

1. For static flow level routing, i.e. "elb-flow", the "average completion time" for

Table II. Average completion time of traffic of MRC site, 2 ISP case: traffic matrix 1

| switching method | TCP-DCR | | TCP-SACK | |
|---|---|---|---|---|
| | small | large | small | large |
| elb-flow | 4.344781 | 0.436980 | 4.250966 | 0.387510 |
| elb-packet | 4.821521 | 0.353402 | 7.499323 | 2.438142 |
| greedy-flow | 2.216936 | 0.331528 | 2.154564 | 0.304395 |
| greedy-packet | 2.397582 | 0.219830 | 1.985902 | 0.470547 |
| greedy-flowlet-50ms | 1.988831 | 0.303839 | 2.081593 | 0.276129 |
| greedy-flowlet-100ms | 2.076806 | 0.320381 | 2.027103 | 0.271728 |
| greedy-flowlet-250ms | 2.126990 | 0.336444 | 2.037026 | 0.290396 |
| greedy-flowlet-500ms | 2.153087 | 0.318318 | 2.248163 | 0.294258 |

TCP-DCR is slightly larger than TCP-SACK. However, for packet level greedy MRC, TCP-DCR has obvious advantage over TCP-SACK for large flows. From the Table V, we see the reason is that the number of "fast recovery" for TCP-SACK is much larger than TCP-DCR for "packet-greedy" which is a direct result of packet reordering and reduces TCP's throughput. The confirms that TCP-DCR is more robust to packet reordering than TCP-SACK.

2. The best trade-off between benefit of switching and side effect of packet reordering is achieved by "greedy-packet" in TCP-DCR case, where the "average completion time" for large (small) flows is shortened by 49.7% (44.8%) compared to "elb-flow".

3. The performance of greedy MRC compared to static routing for small flows is better than the performance for large flows. A possible reason is that TCP flows in "congestion avoidance" phase might be more sensitive to packet reordering

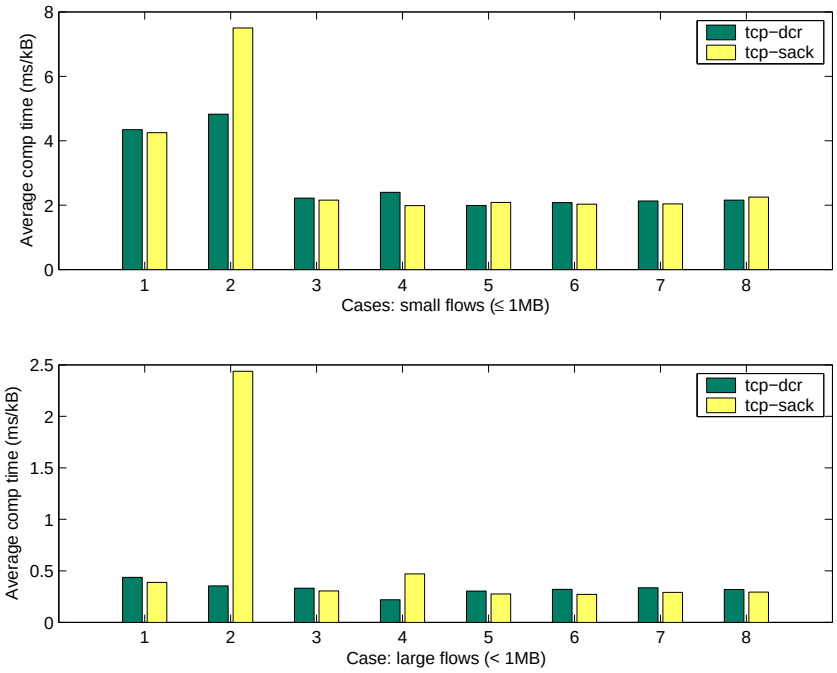


Fig. 94.: Routing performance of MRC site, 2 ISP case, traffic matrix 1: (1) elb-flow, (2) elb-packet, (3) greedy-flow, (4) greedy-packet, (5-8) greedy-flowlet, timeout: 50ms, 100ms, 250ms and 500ms

than TCP in "slow start" phase. To verify this, further analysis of TCP's behavior is needed. Unfortunately, we did not record enough trace data in our simulations for this analysis and have to leave it as future work.

To ensure our observed performance differences are not because of random factors in simulations, we perform simulations for a few approaches, with representative parameter setting, 10 more times with different random seeds. For each set of simulations, we normalize the mean completion time of each approach by dividing the mean completion time of the "elb" approach with TCP-SACK. The mean, maximum and minimum value of the normalized mean completion times are plotted in Fig. 96. The results shown in Fig. 96. The results are consistent with the simulation results shown in Table II and Fig. 94 and Table III and Fig. 95.

Table III. Average completion time of traffic of MRC site, 2 ISP case: traffic matrix 2

| switching method | TCP-DCR | | TCP-SACK | |
|---|---|---|---|---|
| | small | large | small | large |
| elb-flow | 4.183970 | 0.445129 | 3.978715 | 0.359663 |
| elb-packet | 4.979890 | 0.409569 | 7.592696 | 1.785910 |
| greedy-flow | 2.365486 | 0.323899 | 2.431206 | 0.309921 |
| greedy-packet | 2.255806 | 0.254252 | 1.983752 | 0.526816 |
| greedy-flowlet-50ms | 2.226852 | 0.315311 | 2.281431 | 0.291223 |
| greedy-flowlet-100ms | 2.176076 | 0.311337 | 2.173757 | 0.288538 |
| greedy-flowlet-250ms | 2.220801 | 0.326950 | 2.154934 | 0.311280 |
| greedy-flowlet-500ms | 2.283300 | 0.323854 | 2.393445 | 0.303984 |

The simulation results for 2 ISP case show that greedy MRC can significantly improve routing performance of highly dynamic TCP traffic when working together with either TCP-DCR or Flowlet switching as a mechanism to tolerate packet reordering.

However, to properly use Flowlet switching the Flowlet time-out should be set larger than the maximum delay difference between alternate paths. For our case, it should be larger than the maximum queuing delay on a bottleneck link, 250 milliseconds, the value suggested by a IETF guideline [55]. Using such a large Flowlet timeout value reduces the ability of Flowlet switching to switch traffic among alternate paths, thus the adaptiveness to path quality changes. In the rest of this work, we will focus on the study of greedy MRC when TCP-DCR is used.

We also study the performance of the greedy MRC for 3 ISP case and 4 ISP case. The results are plotted in Figs. 97 and 98. For the 3 ISP case, greedy MRC with TCP-DCR reduces the "average completion time" by 43.8% for large flows and by 46.5% for small flows compared to elb-flow with TCP-DCR. For the 4 ISP case,

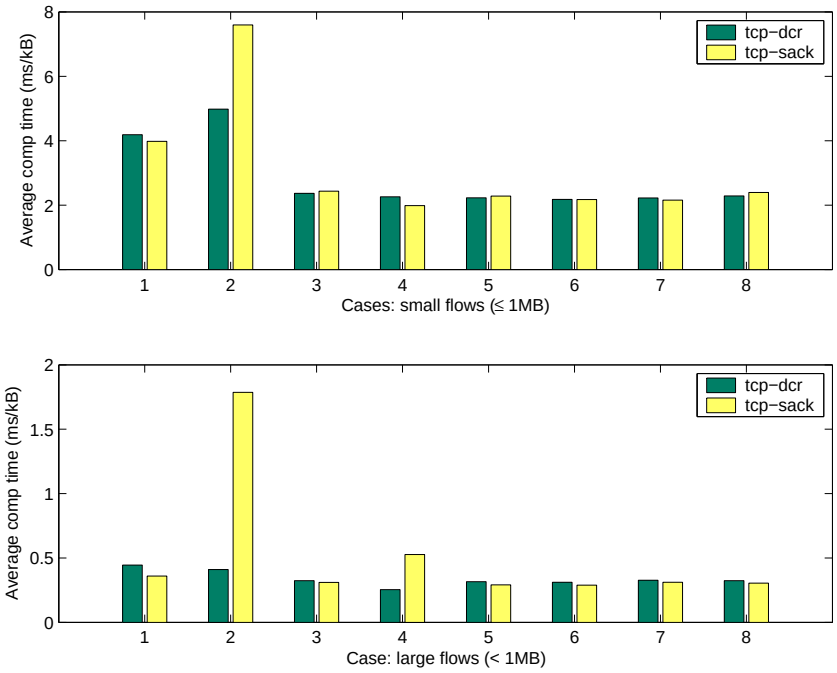


Fig. 95.: Routing performance of MRC site, 2 ISP case, traffic matrix 2: (1) elb-flow, (2) elb-packet, (3) greedy-flow, (4) greedy-packet, (5-8) greedy-flowlet, timeout: 50ms, 100ms, 250ms and 500ms

greedy MRC with TCP-DCR reduces the "average completion time" by 60.6 % for large flows and by 51.0% for small flows compared to elb-flow with TCP-DCR. We also observe that, same as in the 2 ISP case, TCP-SACK performs better than TCP-DCR for static flow level routing, i.e. "elb-flow". However, the "average completion time" of packet level greedy MRC with TCP-DCR is still around 40% to 50% less than the value of flow level static routing with TCP-SACK.

c. Single MRC sender stub network: effect on non-MRC stub networks

When MRC sites and non-MRC sites coexist, we need to make sure the MRC done by MRC sites do not hurt the routing performance of non-MRC sites. To verify that the performance improvement of greedy MRC we observed in our simulations does not

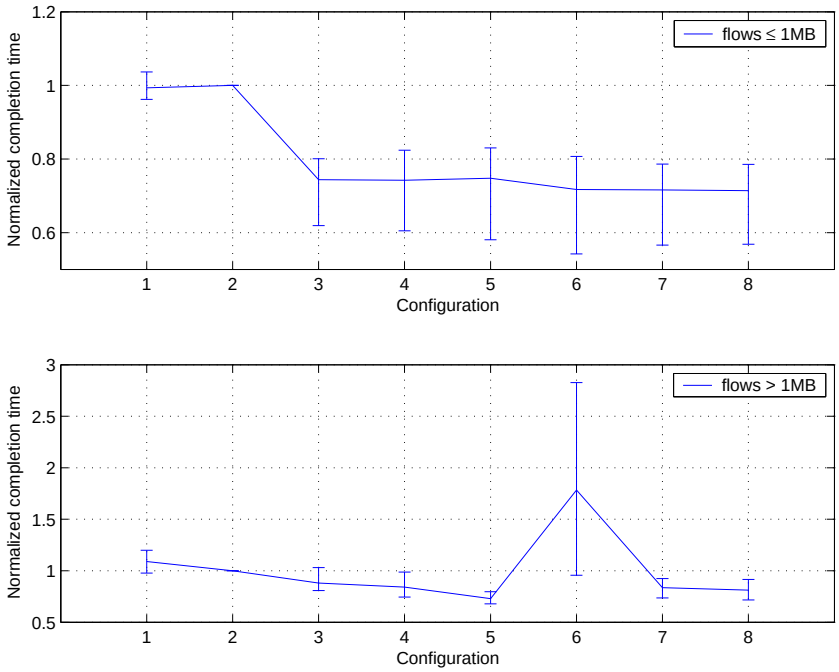


Fig. 96.: Statistics of normalized mean completion time(compared to elb-flow, TCP-SACK) of MRC site, 2 ISP case: (1) elb-flow, TCP-DCR; (2) elb-flow, TCP-SACK; (3) greedy-flow, TCP-DCR; (4) greedy-flow, TCP-SACK; (5) greedy-packet, TCP-DCR; (6) greedy-packet, TCP-SACK; (7) greedy-flowlet-100ms, TCP-DCR; (8) greedy-flowlet-100ms, TCP-SACK.

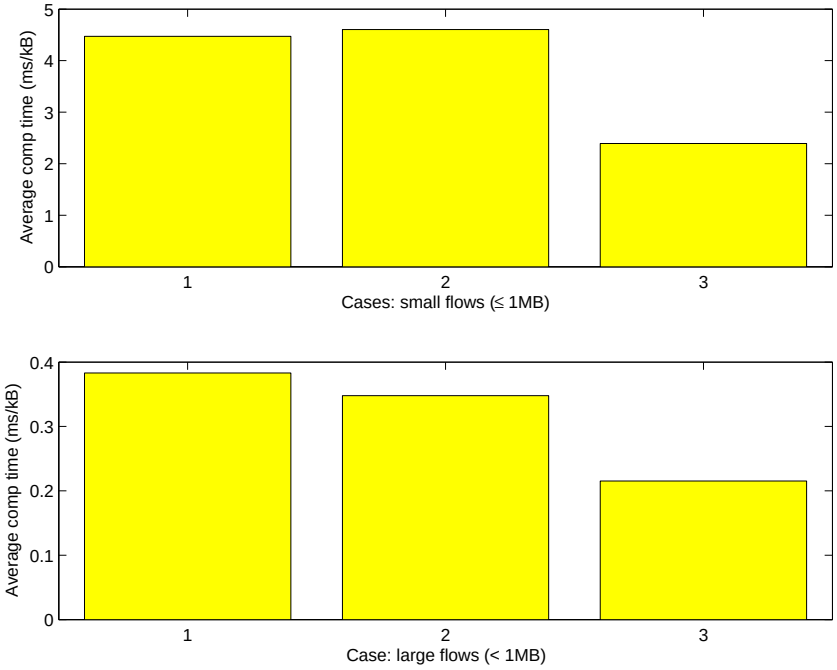


Fig. 97.: Routing performance of MRC site, 3 ISP cases: (1) elb-flow-TCPDCR, (2) elb-flow-TCPSACK, (3) greedy-packet-TCPDCR

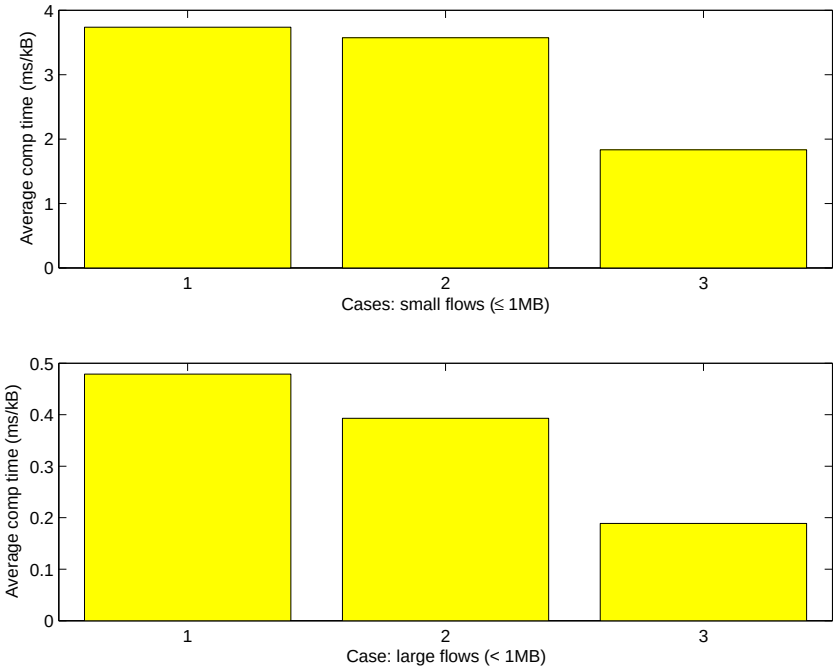


Fig. 98.: Routing performance of MRC site, 4 ISP cases: (1) elb-flow-TCPDCR, (2) elb-flow-TCPSACK, (3) greedy-packet-TCPDCR

Table IV. Number of TCP events experienced by traffic of MRC site, 2 ISP case, traffic matrix 1

| switching method | TCP-DCR | | | TCP-SACK | | |
|---|---|---|---|---|---|---|
| | timeout | slow start | fast recovery | timeout | slow start | fast recovery |
| elb-flow | 710 | 821 | 153 | 522 | 831 | 365 |
| elb-packet | 1004 | 3287 | 3082 | 970 | 16861 | 37241 |
| greedy-flow | 328 | 365 | 64 | 224 | 381 | 192 |
| greedy-packet | 395 | 504 | 177 | 112 | 1646 | 4091 |
| greedy-flowlet-50ms | 183 | 204 | 35 | 166 | 305 | 169 |
| greedy-flowlet-100ms | 189 | 206 | 32 | 176 | 316 | 166 |
| greedy-flowlet-250ms | 282 | 313 | 56 | 227 | 375 | 181 |
| greedy-flowlet-500ms | 289 | 314 | 48 | 243 | 426 | 221 |

come at the cost of hurting routing performance of other non-MRC sites, we calculate the "average completion time" for flows of non-MRC sites. The results corresponding to Figs. 94 to 98 are plotted in Figs. 99 to 103 respectively.

From Figs. 99 to 103, we observe that the routing performance of non-MRC sites are actually improved when the MRC site change from static routing to greedy MRC. These results show that the benefit of MRC does not come from hurting routing performance of non-MRC sites. In other words, the benefit of MRC in the problem we are studying comes from avoiding the use of congested paths. This improves the routing performance of the MRC site as well as the routing performance of other non-MRC sites.

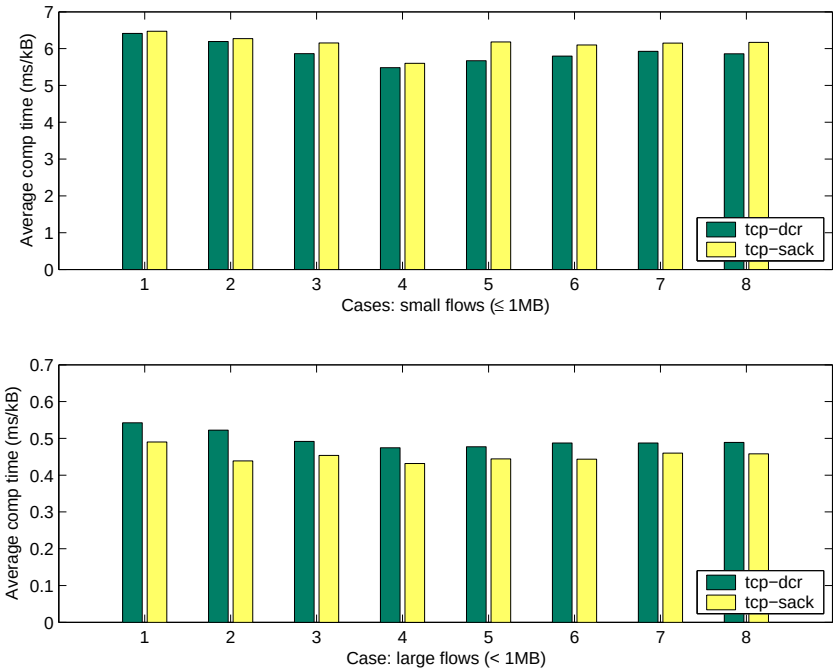


Fig. 99.: Routing performance of non-MRC sites, 2 ISP case, traffic matrix 1: (1) elb-flow, (2) elb-packet, (3) greedy-flow, (4) greedy-packet, (5) greedy-flowlet50ms, (6) greedy-flowlet100ms, (7) greedy-flowlet250ms, (8) greedy-flowlet500ms

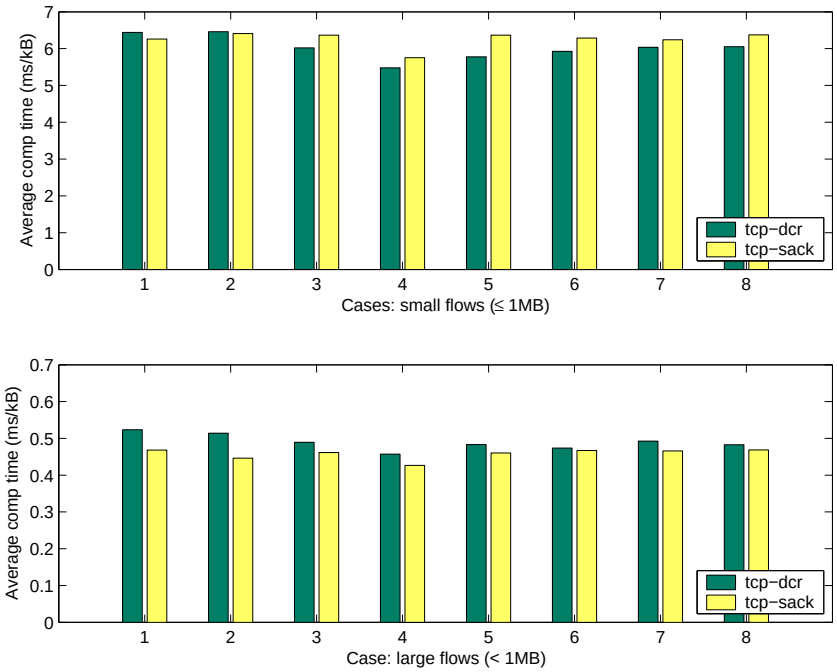


Fig. 100.: Routing performance of non-MRC sites, 2 ISP case, traffic matrix 2: (1) elb-flow, (2) elb-packet, (3) greedy-flow, (4) greedy-packet, (5) greedy-flowlet50ms, (6) greedy-flowlet100ms, (7) greedy-flowlet250ms, (8) greedy-flowlet500ms

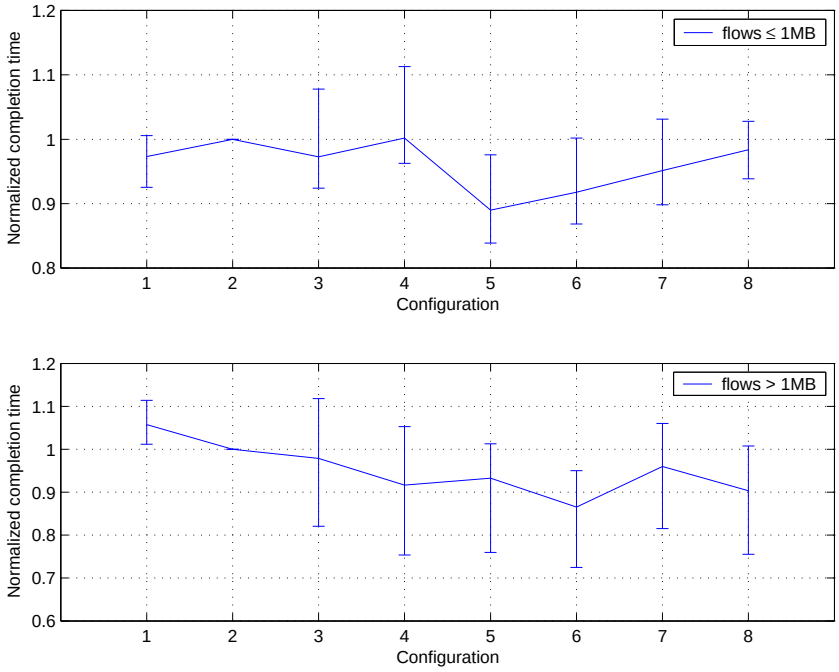


Fig. 101.: Statistics of normalized mean completion time(compared to elb-flow, TCP-SACK) of non-MRC site, 2 ISP case: (1) elb-flow, TCP-DCR; (2) elb-flow, TCP-SACK; (3) greedy-flow, TCP-DCR; (4) greedy-flow, TCP-SACK; (5) greedy-packet, TCP-DCR; (6) greedy-packet, TCP-SACK; (7) greedy-flowlet-100ms, TCP-DCR; (8) greedy-flowlet-100ms, TCP-SACK.

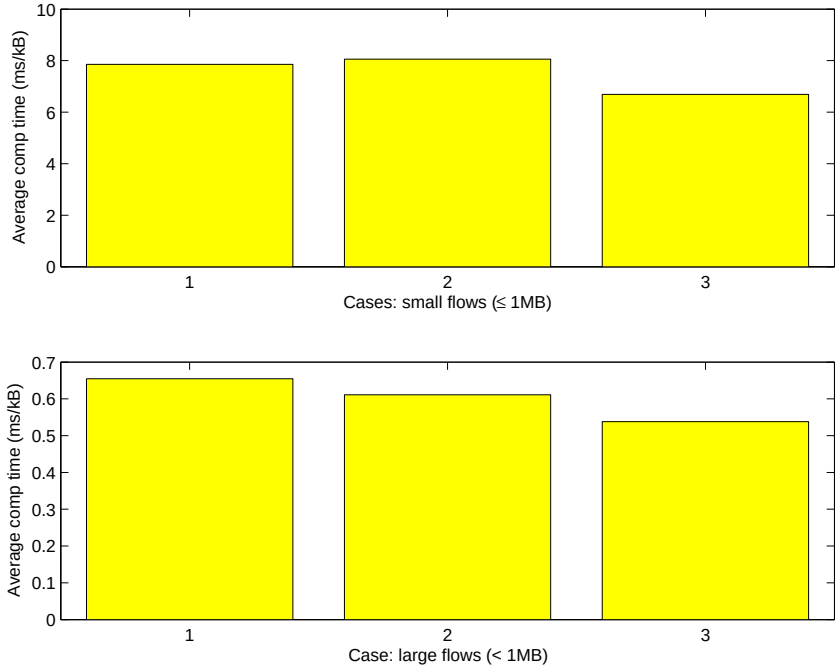


Fig. 102.: Routing performance of non-MRC site, 3 ISP cases: (1) elb-flow-TCPDCR, (2) elb-flow-TCPSACK, (3) greedy-packet-TCPDCR

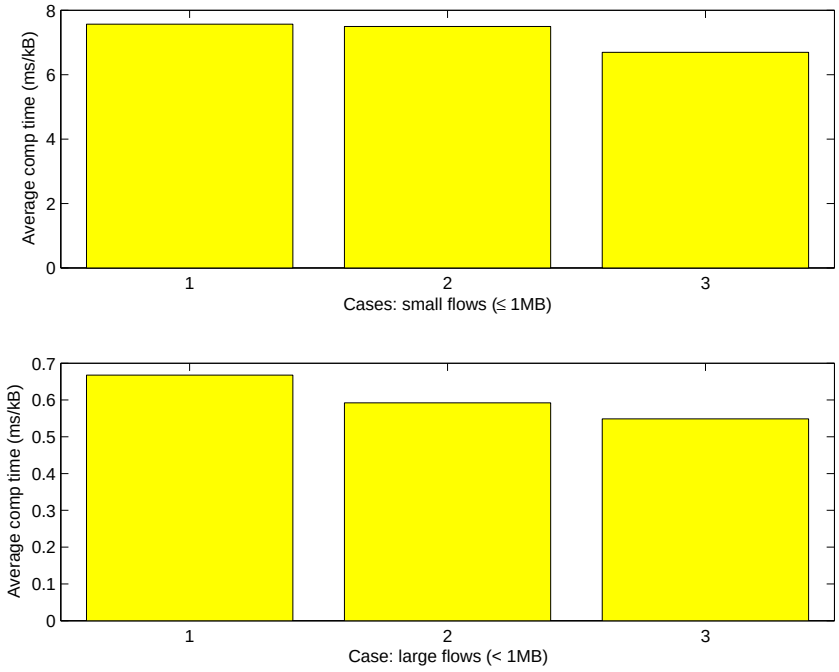


Fig. 103.: Routing performance of non-MRC site, 4 ISP cases: (1) elb-flow-TCPDCR, (2) elb-flow-TCPSACK, (3) greedy-packet-TCPDCR

Table V. Number of TCP events experienced by traffic of MRC site, 2 ISP case, traffic matrix 2

| switching method | TCP-DCR | | | TCP-SACK | | |
|---|---|---|---|---|---|---|
| | timeout | slow start | fast recovery | timeout | slow start | fast recovery |
| elb-flow | 672 | 799 | 168 | 414 | 712 | 353 |
| elb-packet | 953 | 3632 | 3658 | 766 | 17794 | 37949 |
| greedy-flow | 290 | 315 | 49 | 208 | 379 | 202 |
| greedy-packet | 369 | 493 | 221 | 135 | 1821 | 4982 |
| greedy-flowlet-50ms | 237 | 252 | 41 | 174 | 329 | 185 |
| greedy-flowlet-100ms | 177 | 192 | 35 | 142 | 254 | 135 |
| greedy-flowlet-250ms | 231 | 255 | 41 | 205 | 354 | 179 |
| greedy-flowlet-500ms | 264 | 298 | 61 | 222 | 376 | 177 |

d. Multiple MRC sender stub networks: overall performance

We also study the routing performance of MRC when multiple sites perform MRC. The routing schemes we studied include static equal-splitting based flow level and packet level routing, greedy MRC with different parameters and user-optimal routing based MRC with different steps sizes. The TCP variant used in this set of simulations is TCP-DCR. The overall routing performance of all sites are plotted in Fig. 104.

From the results, we have the following observations: (1) Compared to static routing, both greedy MRC and user-optimal based MRC can improve routing performance for highly dynamic TCP traffic; (2) The performance of Greedy MRC is not sensitive to EWMA smoothing parameter, $\alpha$; (3) Greedy MRC together with TCP-SACK and Flowlet switching achieves similar performance as greedy MRC with TCP-DCR; (4) Greedy MRC achieves similar performance as user-optimal routing

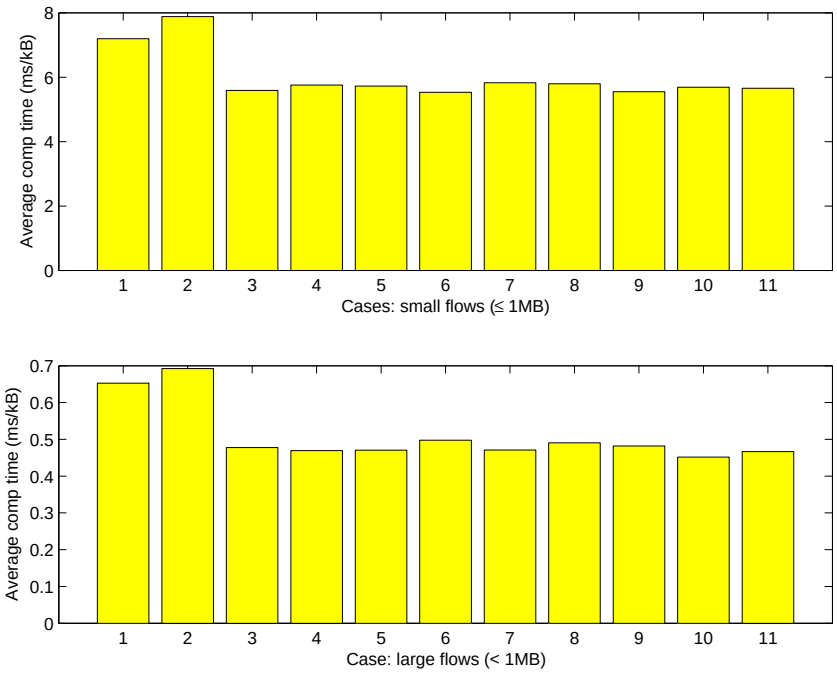


Fig. 104.: Overall routing performance when all sites using MRC, asymmetric topologies, cases: (1) elb-flow, (2) elb-packet, (3) greedy-ewma0.1-packet, (4) greedy-ewma0.2-packet, (5) greedy-ewma0.4-packet, (6) greedy-TCPSACK-flowlet50ms, (7) uopt-0.01, (8) uopt-0.05, (9) uopt-0.1, (10) uopt-0.5, (11) uopt-1

based MRC.

The results have confirmed the our analysis in Section C, i.e. greedy MRC will not cause performance degradation for highly dynamic TCP traffic when multiple MRC devices work together.

## E. Conclusions

In this chapter, we studied MRC of highly dynamic TCP traffic, i.e. traffic consisting of TCP flows of mixed sizes on paths with limited bottleneck capacity. We first analyzed the characteristics of links in this type of environment. Based on the analysis, we proposed a greedy MRC scheme for highly dynamic TCP traffic. The proposed approach works at much smaller time scales than traditional MRC to effectively route highly dynamic TCP traffic in multihomed environment. Ns-2 simulation results showed that the proposed scheme performs significantly better than static routing approaches. Both the routing performance of the MRC site and other sites are improved. The proposed greedy MRC approach achieve similar performance as user-optimal routing based MRC for routing of highly dynamic TCP traffic.

# CHAPTER VII

## CONCLUSIONS AND FUTURE WORK

In this dissertation, we have studied two types of important Internet routing problems: fast rerouting for ISP networks and multihoming route control for stub networks.

We first propose a fast rerouting scheme for link state routing protocols. Our scheme includes both pro-active and reactive components. The pro-active component avoids complex computation of rerouting paths after links fail, which reduces restoration latency. The reactive component is used to handle the cases where local rerouting cannot be done. We proved the correctness of our algorithms. Simulation results show that the rerouting after link failures usually can be done by local routers and for the remaining cases by local routers along with a small number of neighboring routers. The rerouting operations after link failures are also very simple. Therefore, our scheme can increase rerouting speed as well as limit the range of propagation of transient link failure events.

In the second part of this dissertation, we propose and analyze three Multihoming Route Control (MRC) methods for improving routing performance of multihomed stub networks in dynamic network environments.

The first MRC method is based on optimal routing. In this method, traffic matrix and information on path qualities are exchanged among a group of stub networks. Based on such information, each stub network calculates the optimal routing vector and uses it to route its egress traffic to other stub networks in the group. We study the performance of this method using both synthesized traffic matrices and traffic matrices generated from real-world Internet traffic traces. The simulation results have shown: (1) the method can improve performance of routing among a group of multihomed stub networks up to 40% compared to static equal-splitting based load balancing; (2)

The performance improvement is larger for Pareto traffic and asymmetric topologies compared to Poisson traffic and symmetric topologies; (3) smaller prediction and adjudgement periods can improve the performance of this method.

The second MRC method is based on user-optimal routing. The distributed nature of user-optimal routing makes it suitable for both inhouse traffic (traffic among a group of multihomed stub networks) and Internet traffic (from a multihomed stub network to any Internet destination). In this method, we apply the user-optimal routing theory to the MRC problem. We compare the performance of this user-optimal routing based MRC to optimal routing based MRC using extensive simulations. Results show that user-optimal routing based MRC can achieve similar performance as optimal routing based MRC. We also study the dynamics of this method using simulations. The results show that the algorithm converges fast and does not cause oscillations.

The third MRC method we proposed is for MRC of highly dynamic TCP traffic. We study MRC in this environment using ns-2 packet level simulations. We first analyze link characteristics when load consists of TCP flows of different sizes. Based on the analysis result, we propose to use greedy MRC in small timescales for MRC of highly dynamic TCP traffic. Simulation results show that the proposed method can greatly improve the routing performance of stub networks when packet reordering robust TCP is employed (we use TCP-DCR in our study). Our simulation study also show that the greedy MRC in small timescales does not hurt routing performance of non-MRC sites and can improve the overall routing performance of all sites competing for bandwidth on same set of bottlenecks.

Possible future research directions include: (1) Study of our MRC schemes in real world environment or through emulations using real world traffic and path quality data; (2) Further study of the use of alternative quality metrics, e.g. available

bandwidth, in measurement based adaptive MRC.

# REFERENCES

[1] C. Boutremans, G. Iannaccone, and C. Diot, "Impact of link failures on VoIP performance," in *Proc. NOSSDAV*, 2002, pp. 63–71.

[2] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 293–306, 2001.

[3] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *Proc. IEEE INFOCOM*, Jun. 2002, pp. 618–627.

[4] J. Moy, "OSPF version 2," RFC 2328, Apr. 1998, [Online]. Available: http://www.ietf.org/rfc/rfc2328.txt.

[5] D. Oran, "OSI IS-IS intra-domain routing protocol," RFC 1142, Feb. 1990, [Online]. Available: http://www.ietf.org/rfc/rfc1142.txt.

[6] N. Dubois, B. Fondeviole, and N. Michel, "Fast convergence project," RIPE-47 presentation, Jan. 2004, [Online]. Available: http://www.ripe.net/ripe/meetings/ripe-47/presentations/ripe47-routing-fcp.pdf.

[7] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot, "Characterization of failures in an IP backbone network," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 2307–2317.

[8] S. Nelakuditi, S. Lee, Y. Yu, and Z.-L. Zhang, "Failure insensitive routing for ensuring service availability," in *Proc. IWQoS*, 2003, pp. 287–304.

[9] P. Narváez, K.-Y. Siu, and H.-Y. Tzeng, "Local restoration algorithm for link-state routing protocols," in *Proc. ICCCN*, 1999, pp. 352–357.

[10] P. Pan, G. Swallow, and A. Atlas, "Fast reroute extensions to RSVP-TE for LSP tunnels," RFC 4090, May 2005, [Online]. Available: http://www.ietf.org/rfc/rfc4090.txt.

[11] P. Smith, "BGP multihoming techniques," NANOG 23, Oct. 2001, [Online]. Available: http://www.nanog.org/mtg-0110/ppt/smith.pdf.

[12] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, Jan. 2006, [Online]. Available: http://www.ietf.org/rfc/rfc4271.txt.

[13] A. Basu, C.-H. L. Ong, A. Rasala, F. B. Shepherd, and G. T. Wilfong, "Route oscillations in I-BGP with route reflection," in *Proc. ACM SIGCOMM*, Aug. 2002, pp. 235–247.

[14] Internap Network Services Corporation, "Route control technology," [Online]. Available: http://www.internap.com/solutions/routecontrol/page1980.html, accessed on Nov. 2006.

[15] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, "A measurement-based analysis of multihoming," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 353–364.

[16] D. P. Bertsekas and R. Gallager, *Data Networks*, 2nd edition, Upper Saddle River, NJ: Prentice-Hall, 1992.

[17] The VINT Project, "The network simulator - ns-2," [Online]. Available: http://www.isi.edu/nsnam/ns, accessed on Feb. 2006.

[18] A. Medina, A. Lakhina, I. Matta, and J. W. Byers, "BRITE: an approach to universal topology generation," in *Proc. MASCOTS*, 2001, pp. 346–353.

[19] D. Passmore, "Multihoming route optimizers," Business Communications Review, Nov. 2001, [Online]. Available: http://www.burtongroup.com/promo/columns/column.asp?articleid=64&employeeid=56.

[20] K. Egevang and P. Francis, "The IP Network Address Translator (NAT)," RFC 1631, May 1994, [Online]. Available: http://www.ietf.org/rfc/rfc1631.txt.

[21] F. Guo, J. Chen, W. Li, and T. Chiueh, "Experiences in building a multihoming load balancing system," in *Proc. IEEE INFOCOM*, Mar. 2004, vol. 2, pp. 1241–1251.

[22] University of Oregon, "The route views project," [Online]. Available: http://www.routeviews.org, accessed on Jun. 2004.

[23] University of Washington, "Scriptroute: A facility for distributed internet debugging and measurement," [Online]. Available: http://www.cs.washington.edu/research/networking/scriptroute, accessed on Jun. 2004.

[24] N. Feamster, J. C. Borkenhagen, and J. Rexford, "Guidelines for interdomain traffic engineering," *Computer Communication Review*, vol. 33, no. 5, pp. 19–30, 2003.

[25] AT&T Business, "U.S. network latency," [Online]. Available: http://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html, accessed on Feb. 2005.

[26] J. Kowalski and B. Warfield, "Modeling traffic demand between nodes in a telecommunications network," ATNAC, 1995, [Online]. Available: http://citeseer.ist.psu.edu/10026.html.

[27] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *Proc. of ACM SIGMETRICS*, 2003, pp. 206–217.

[28] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker, "On selfish routing in Internet-like environments," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 151–162.

[29] D. Gross and C. Harris, *Fundamentals of Queuing Theory*, 3rd edition, New York: Wiley, 1998.

[30] C. M. Harris, P. H. Brill, and M. J. Fischer, "Internet-type queues with power-tailed interarrival times and computational methods for their analysis," *INFORMS J. on Computing*, vol. 12, no. 4, pp. 261–271, 2000.

[31] S. Tao, K. Xu, Y. Xu, T. Fei, L. Gao, R. Guérin, J. F. Kurose, D. F. Towsley, and Z.-L. Zhang, "Exploring the performance benefits of end-to-end path switching," in *Proc. of IEEE ICNP*, 2004, pp. 304–315.

[32] A. Akella, S. Seshan, and A. Shaikh, "Multihoming performance benefits: an experimental evaluation of practical enterprise strategies," in *Proc. USENIX Annual Technical Conference, General Track*, 2004, pp. 113–126.

[33] S. Tao and R. Guérin, "On-line estimation of Internet path performance: an application perspective," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 1774–1785.

[34] A. J. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *Proc. ACM SIGCOMM*, Sep. 1989, pp. 1–12.

[35] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," *IEEE/ACM Trans. Netw.*, vol. 4, no. 3, pp. 375–385, 1996.

[36] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, chapter 20. TCP Bulk Data Flow, Addison-Wesley, 1994.

[37] S. Bhandarkar and A. L. Narasimha Reddy, "TCP-DCR: Making TCP robust to non-congestion events," in *Proc. NETWORKING*, May 2004, pp. 712–724.

[38] S. Sinha, S. Kandula, and D. Katabi, "Harnessing TCP's burstiness using flowlet switching," 3rd ACM SIGCOMM Workshop on Hot Topics in Networks, Nov. 2004, [Online]. Available: http://nms.csail.mit.edu/papers/index.php?detail=111.

[39] Z. Cao, Z. Wang, and E. W. Zegura, "Performance of hashing-based schemes for Internet load balancing," in *Proc. INFOCOM*, 2000, pp. 332–341.

[40] R. Martin, M. Menth, and M. Hemmkeppler, "Accuracy and dynamics of hash-based load balancing algorithms for multipath Internet routing," IEEE International Conference on Broadband Communication, Networks, and Systems (BROADNETS), Oct. 2006, [Online]. Available: http://ieeexplore.ieee.org.

[41] PlanetLab Consortium, "PlanetLab," [Online]. Available: http://www.planet-lab.org, accessed on Jun. 2004.

[42] D. K. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, and Y. Zhang, "Optimizing cost and performance for multihoming," in *Proc. ACM SIGCOMM*, Aug. 2004, pp. 79–92.

[43] Y. Qiao, J. Skicewicz, and P. A. Dinda, "An empirical study of the multiscale predictability of network traffic," in *Proc. HPDC*, Jun. 2004, pp. 66–76.

[44] NLANR Measurement and Network Analysis Group, "NLANR PMA: Special traces archive," [Online]. Available: http://pma.nlanr.net/Special/, accessed on Jun. 2004.

[45] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. IEEE INFOCOM*, Mar. 2000, pp. 519–528.

[46] Free Software Foundation, "GLPK (GNU Linear Programming Kit)," [Online]. Available: http://www.gnu.org/software/glpk/, accessed in 2006.

[47] J.G. Wardrop, "Some theoretical aspects of road traffic research," in *Proc. The Institute of Civil Engineers*, 1952, vol. 1, pp. 325–378.

[48] T. Roughgarden and É. Tardos, "How bad is selfish routing?," *J. ACM*, vol. 49, no. 2, pp. 236–259, 2002.

[49] J. N. Tsitsiklis and D. P. Bertsekas, "Distributed asynchronous optimal routing in data networks," *Automatic Control, IEEE Transactions on*, vol. 31, no. 4, pp. 325 – 332, 1986.

[50] W. K. Tsai, "Convergence of gradient projection routing methods in an asynchronous stochastic quasi-static virtual circuit network," *Automatic Control, IEEE Transactions on*, vol. 34, no. 1, pp. 20 – 33, 1989.

[51] T. Güven, C. Kommareddy, R. La, M. Shayman, and S. Bhattacharjee, "Measurement based optimal multi-path routing," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 187–196.

[52] A. Elwalid, C. Jin, S. H. Low, and I. Widjaja, "MATE: MPLS adaptive traffic engineering," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 1300–1309.

[53] N. Cardwell, S. Savage, and T. E. Anderson, "Modeling TCP latency," in *Proc. IEEE INFOCOM*, Mar. 2000, pp. 1742–1751.

[54] R. Braden, "Requirements for Internet hosts - communication layers," RFC 1122, Oct. 1989, [Online]. Available: http://www.ietf.org/rfc/rfc1122.txt.

[55] R. Bush and D. Meyer, "Some Internet architectural guidelines and philosophy," RFC 3439, Dec. 2002, [Online]. Available: http://www.ietf.org/rfc/rfc3439.txt.

[56] M. Florian and D. Hearn, *Network Routing*, chapter 6. Network equilibrium models and algorithms, Elsevier Science, 1995.

[57] M. Beckmann, C. B. McGuire, and C. B. Winsten, *Studies in the Economics of Transportation*, New Haven, CT: Yale University Press, 1956.

[58] H. J. Kushner and D. S. Clark, *Stochastic Approximation Method for Constrained and Unconstrained Systems*, New York: Springer-Verlag, 1978.

[59] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 226–244, 1995.

[60] R. Gao, C. Dovrolis, and E. Zegura, "Avoiding oscillations due to intelligent route control systems," IEEE INFOCOM, Apr. 2006, [Online]. Available: http://www-static.cc.gatech.edu/fac/Constantinos.Dovrolis/Papers/ruomei-infocom06.pdf.

# VITA

Yong Liu received his B.S. degree in Electronics and Information Systems from Peking University, China, in July 1998, and his M.E. degree in Communication and Information Systems from Peking University, China, in July 2001. He received his Ph.D. degree in Computer Engineering from Texas A&M University in December 2006. His research interests are in networking, especially Internet routing, and operating systems.

Mr. Liu may be reached at EMC Smarts, 44 South Broadway, White Plains, NY, 10601. His email address is yongliu2005@gmail.com.