

**TCP-FRIENDLY BUILDING BLOCKS IMPLEMENTATION:
VALIDATION OF A NEW APPROACH TO BETTER
BEST-EFFORT SERVICE ON THE INTERNET**

By

Giampiero Lo Monaco

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Approved:

Prof. Shivkumar Kalyanaraman
Thesis Adviser

Rensselaer Polytechnic Institute
Troy, New York

May 2000
(For Graduation May 2000)

© Copyright 2000

by

Giampiero Lo Monaco

All Rights Reserved

CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vi
ABSTRACT	viii
1. Introduction.....	1
1.1 Background	1
1.2 Overview	3
2. TCP Traffic Conditioning and Buffer Management	4
2.1 TCP Implementations	4
2.2 TCP Friendly Packet Marker	7
2.3 FRIO: A New Buffer Management Scheme	18
3. Performance Analysis and Experimentation	26
3.1 Performance Evaluation Model	26
3.2 Metrics	27
3.3 Experimental Configuration	28
4. Experimental Results	33
4.1 Configuration A	33
4.2 Configuration B	37
4.3 Configuration C	40
4.4 Configuration D	44
4.5 Summary of Results	47

5. Further Development	49
6. Conclusions	51
BIBLIOGRAPHY	53
A. Round Trip Time Delay Component	55
A.1 Motivations	55
A.2 Delay Component Implementation	56

LIST OF TABLES

4.1 Configuration A: 100 and 50 flows - Cases 1 through 4	33
4.2 Configuration B: 100 and 50 flows - Cases 1 through 4	37
4.3 Configuration C: 100 and 50 flows - Cases 1 through 4	41
4.4 Configuration D: 100 and 50 flows – Cases 1 through 4	44

LIST OF FIGURES

2.2.1 Algorithm 1: TCP Friendly Marker Token Allocation	10
2.2.2 Algorithm 2: TCP Friendly Packet Marking	11
2.2.3 Graphical representation of flow data structure	17
2.3.1 Algorithm 3: FRIO Buffer Management	21
3.1.1 Performance Evaluation Model	26
3.3.1 Experimental Physical Network Topology	29
3.3.2 Experimental Logical Network Topology	29
4.1.1 Number of Timeouts: Configuration A	34
4.1.2 Probability of Loss (%): Configuration A	34
4.1.3 Average per-flow Goodput: Configuration A	35
4.1.4 Goodput Coefficient of Variation: Configuration A	35
4.2.1 Number of Timeouts: Configuration B	38
4.2.2 Probability of Loss (%): Configuration B	38
4.2.3 Average per-flow Goodput: Configuration B	39
4.2.4 Goodput Coefficient of Variation: Configuration B	39
4.3.1 Number of Timeouts: Configuration C	41
4.3.2 Probability of Loss (%): Configuration C	42
4.3.3 Average per-flow Goodput: Configuration C	42

4.3.4 Goodput Coefficient of Variation: Configuration C	43
4.4.1 Number of Timeouts: Configuration D	45
4.4.2 Probability of Loss (%): Configuration D	45
4.4.3 Average per-flow Goodput: Configuration D	46
4.4.4 Goodput Coefficient of Variation: Configuration D	46

ABSTRACT

The adoption of differentiated services has been proposed in the past as a tool for improving the *best-effort* service approach of the Internet. We explore the use of a specific type of *diffserv* traffic conditioner, known as the TCP Friendly Marker, in the solution of core traffic congestion problems. We implement an open source code instance of the marker, as well as its complement buffer management technique, in order to assess the feasibility of the results as envisioned during its original design and simulation testing.

The experimentation shows significant improvements, under both Reno and SACK TCP implementations, with respect to the most commonly seen *best-effort* approach. The enhancements are seen in terms of timeouts, network utilization, predictability and fairness of service, packet loss probability, and overall net throughput. In addition we show that our approach counteracts the sensitivity of the performance to RED-like parameters, as well as being completely independent from any version of TCP implementation. We also show evidence of the scalability of our scheme by not requiring complex classification at the core of the network and by providing even larger improvement as the number of flows through the system is incremented.

We present the results obtained under our own experimentation setup, and give some arguments about their logical translation into the real Internet as well as the immediate feasibility of deployment for all the *diffserv* structures here proposed.