
Design and Implementation of an Improved Two Rate Three Color Meter (Itrtcm) Algorithm for Quality of Service Admission Control Using Udp as Traffic Agent

Oyetunji Moses Oyekunle¹, Emuoyibofarhe Ono Justice², Oladosu John Babalola³, Oladeji Florence Alaba⁴

¹ Department of Computer Science and Mathematics, Mountain Top University, Ibafo, Nigeria.

² Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria.

³ Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria.

⁴ Department of Computer Science, University of Lagos, Akoka, Lagos, Nigeria.

ARTICLE INFO

Keywords:

TRTCM, ITRTCM, Quality of Service, Peak Information Rate, Transmission Control Protocol (TCP), User Datagram Protocol

ABSTRACT

This paper designed an Improved Two Rate Three Color Meter (ITRTCM) algorithm to get more packets operating above PIR across the Internet for better QoS assurance. The design and simulation of network model for implementing ITRTCM algorithms on User Datagram Protocol (UDP) as traffic agents on the Internet was done using NS-2. The throughput, loss rate, fairness rate and one-way packet delay were used as performance metrics. The results for UDP as traffic agent revealed that ITRTCM and TRTCM had throughput of 50463 and 50460; fairness rate of 0.2726 and 0.2632; loss rate of 74.59 and 74.70; and one-way packet delay of 0.8584ms and 0.8669ms respectively. The proposed ITRTCM algorithm outperforms the existing TRTCM algorithm in terms of aforementioned metrics

1. Introduction

The deployments of multimedia applications over the Transport Control Protocol/Internet protocol (TCP/IP) suite require high bandwidth, bursty data transmission and stringent delay constraints. These applications also have challenges associated with quest for differential services. Therefore, there is need to provide a predictable Internet support to accommodate the emergence of newly multimedia applications over the TCP/IP protocol suite for quality of service (QoS) assurance [1]. It uses the principle of best effort service model where the incoming packets are enqueued in first in first out (FIFO) order at the router. The router transmits as much as its capacity could allow and thus gives no assurance as to packets actual delivery or timeliness [2].

The best effort services can only accommodate the earlier Internet applications such as File

transfer Protocol (FTP), World Wide Web (WWW), Electronic Mail etc; because some of these applications have high tolerance for delay, packet loss and low throughput. Most of the recent applications integrated on the Internet are multimedia in nature carrying voice, video and data signals [2]. These applications require some qualities such as differential services, timely delivery and sensitivity to packet loss on transit and packet delay. In view of this, it becomes necessary for the TCP/IP model to be modified or extended [2] (Forouzan, 2013).

The research efforts through Internet Engineering Task Force (IETF) have concentrated on the extension and modification of the TCP/IP protocol for QoS assurance in order to accommodate the aforementioned real time multimedia applications on the Internet. This modification resulted in integrated service (IntServ) architecture and differentiated service (DiffServ) architectures. The IntServ architecture is characterized by resource reservation for each session or flow by the router along the path of the traffic to the destination. This means resources must be reserved for each source at the intermediate routers. This could lead to scalability problem when thousands of applications are requesting for reservation at the same time [3-5]. This limitation of IntServ architecture brought about the development of DiffServ architecture which makes provision for traffic flows demanding same treatment from the network to be aggregated at the routers [6].

DiffServ is based on a model which attempts to move complexity of the network core to the edge router and keep the functionality of the core network as simple as possible [7]. Core router does only packet forwarding based on the choice of treatment indicated at the packet header [8]. The admission control mechanism is introduced at the boundary of the DiffServ network or domain to check whether a service request is to be granted or rejected [9]. The DiffServ model uses this mechanism at the network edge (edge routine) to measure, mark, shapen or police packets if necessary. This mechanism operates on Service Level Agreement (SLA) between the network (Internet Service Provider) and the traffic sources (customers).

In SLA, the user or source of traffic specifies which network quality or treatment it prefers (service level specification) while the network indicates the amount of traffic to inject by the user (called traffic profile) if that specification is to be granted. Before traffic flows enter the network core from different edge domains, they need to be marked either as in-profile or out-of-profile at their respective edge networks. Traffic meter algorithms measure the properties of the stream of packets that arrive to the router against a set traffic profile. It guides in regulating the injection of traffic to the network [10] Traffic streams that are considered after measurement as in-profile or compliant are given the desired treatment by the network while those measured as out-of-profile have violated the SLA and are exposed to any kind of treatment depending on the network load.

The existing TRTCM algorithm for Quality of Service (QoS) admission control in the Internet fails to provide any guarantee for packet operating above Peak Information Rate (PIR) to accommodate the requests of real-time applications. This research focused on the design and simulation of an improved variant of token bucket meter algorithm for quality of service admission control on the Internet. The throughput, fairness rate, loss rate and one-way packet delay were used as performance metrics.

2. Related Works

The TRTCM measures traffic stream based on four traffic conditioning parameters: Committed Information Rate (CIR), Peak Information Rate (PIR) and two burst sizes namely Committed Burst Size (CBS) and Peak Burst Size (PBS) with 3 drop precedence (one complaint and two non-compliant packets). TRTCM uses two rates (CIR and PIR) and it is also identified in terms of two buckets:

C and P. The capacity of C is CBS and that of P is PBS. For clarity sake, its algorithm in [11, 12]. Originally, both token buckets are full, i.e. $T_c(0) = CBS$ and $T_p(0) = PBS$.

If $(T_c < CBS)$ then $T_c \square T_c + 1$ else

If $(T_p < PBS)$ then $T_p \square T_p + 1$ else neither T_c nor T_p are incremented

When a packet of size B arrives at time t, the following happen:

If $T_c(t) - B \geq 0$, the packet is marked as green and $T_c \square T_c - B$

If $T_c(t) - B < 0$ and $T_p(t) - B \geq 0$, the packet is marked yellow and $T_p \square T_p + 1$

If neither of the previous cases is valid, the packet is marked red. T_c and T_p remain unchanged [12].

A different level of assurance of transmission (priority) is then given to packets which are green, yellow and red. Green packet received the highest level of assurance, followed by yellow packet and red packets received the lowest level of assurance.

3. Methodology

This research adopted simulation approach to carry out design and simulation of network model to implement the algorithms of an improved two rate three color meter (ITRTCM) algorithm on UDP as traffic agents with exponential ON/OFF as traffic generator. The improved algorithm was compared with the existing two rate three color meter using throughput, fairness rate, loss rate and one-way packet delay as performance metrics. The network model was simulated using a software simulator called network simulator-2 (NS-2).

3.1. Experimental Design

The network model design shown in Figure 1 represents the model used for this research to implement the algorithms of ITRTCM and TRTCM meters. In the network model, the same parameter settings were used for the two meter algorithms simulated (TRTCM and ITRTCM) to create platform for comparison among the traffic meter algorithms considered. The packet sizes used for the simulated experiments were 1000bytes and 2000bytes for the first and second scenarios respectively for each traffic meter algorithm to justify the results of the simulation experiments. The packet sizes were varied because traffic meter algorithms deal with measurement of the packet size to check its conformity with traffic profile.

The network model was designed to consist of 21 nodes (eight nodes are for sources, three nodes are for edge routers, two nodes are core routers and the remaining eight nodes are for destinations). The node-to-node network links from sources to destinations were configured with bandwidth of 10Mbps and link delays of 5ms except from the core router C2 to edge router E3 which was configured as 5Mbps of bandwidth and 5ms of link delay. The core to edge router configuration was set to 5Mbps intentionally to study the effect of congestions at the core routers. The sources (S1, S2, S3, S4) generated traffic streams with Exponential ON/OFF using UDP traffic agent and send them to ingress edge router (E1).

Likewise, the sources (S5, S6, S7, and S8) also generated traffic streams via the same medium using UDP as traffic agent and send them to edge router (E2). At the ingress routers (E1 and E2), the incoming traffic streams are admitted, classified and conditioned with traffic conditioning functions such as metering, marking and policing using associated differentiated services code point (DSCP) to carry out traffic profiles enforcement before sending them to the core router for forwarding. Each meter algorithm measured the traffic stream to ensure whether the traffic stream is in-profile or out-of-profile for proper traffic admission control. The core router buffers the packets into respective queues using priority scheduling discipline to forward them to respective destinations (D1, D2, D3, D4, D5, D6, D7 and D8) through the egress router (E3). Network simulator-2 (NS-2) was used for the simulation which provides tools for visualization- network animation (NAM) unlike others simulators that have no facility for network visualization.

Buffer size is assumed to be finite because using infinite buffer can worsen congestion [13] and the server is the bottlenecked core router with a deterministic capacity (bandwidth). The transmission mode used between sources and destinations were full duplex transmission modes while simplex transmission mode was used between edge router and core router and also between core router and core router. The edge and the core nodes used RED buffering scheme to detect incipient congestion. The simulation experiments were carried out for 80 seconds for each traffic meter algorithm implemented in the network model using sources with eight different queues and the data generated in the course of the simulation were traced into files. The scheduler used for the simulation was priority scheduler.

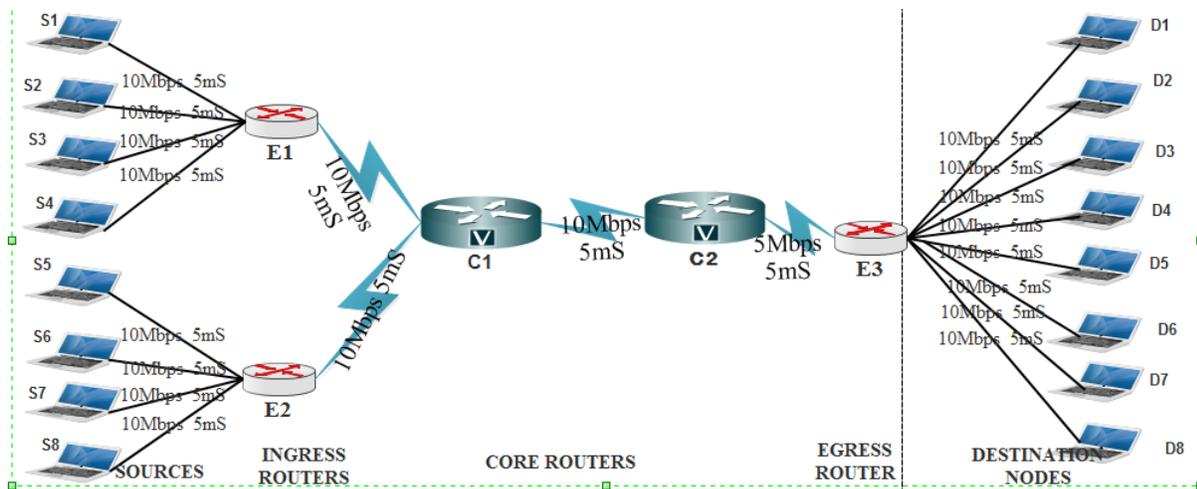


Figure 1. Simulation network model.

3.2. Traffic Meter Considered for Improvement

TRTCM is unable to provide level of assurance differentiation for packets in data streams operating above peak information rate (PIR). Packets that are marked red are usually dropped instantly if there is congestion in the network. There is absolutely no second chance for any packets that exceeded PIR in this critical condition which was demonstrated in the course of the simulation. It is in this vein, that a modification is suggested to TRTCM logic so as to buffer more packets that exceed PIR temporarily in case there is low congestion in the network. Packet that would have been dropped during congestion are buffered in order get them across to the destination. It would be appreciated if more of its packets are helped on transit to get to the destination.

3.2.1. Design of an Improved Two Rate Three Color Meter (ITRTCM) Algorithm

The improved meter (ITRTCM) algorithm used the same network model design discussed in Figure 1 to implement its algorithm save the packet sizes that were 1000 bytes and 2000 bytes respectively for the two simulation scenarios. Based on the deficiency of TRTCM to provide level of assurance differentiation for packets in data streams operating above peak information rate, its algorithm was modified such that when CIR and PIR are exceeded, some decisions are made not to drop all the non-compliant packets (all packets that are marked yellow and red). The essence of

coming up with an improved version of TRTCM is to give second chance to dropped packets when PIR is exceeded. Priority is randomly given to some of the non-compliant packets in such a way that not all the packets are dropped in a case of serious congestion in order to increase the number of packets that gets to the destination (i.e. network throughput).

When PIR is exceeded, packets arriving are held briefly in the buffer. A clock tick counting is started up to 100 after which a number (N) between 0 and 100 is generated. The decisions are based on the generated numbers. If the number is less than 21 the packets should be dropped immediately. If the number is between 21 and 40, the packet should be sustained further in the yellow queue for a set interval; if there is no token, it is dropped. If the number is greater than 40, the packets are sent to join the green queue for transmission.

This same method was applied when CIR was exceeded. If the number is less than 5 the packets should be dropped immediately. If the number is between 5 and 40, the packet should be sustained further in the yellow queue for a set interval; if there is no token, it is dropped. If the number is greater than 40, the packets are sent to join the green queue for transmission as shown in Figure 2.

3.2.2. Improved Two Rate Three Color Meter (ITRTCM) Algorithm

The algorithm stated below improved on the existing TRTCM algorithm stated in section 3.2. Instead of rejecting packets marked as yellow or red completely when CIR and PIR are exceeded, the following modifications were integrated in the algorithm to get more packet across to the destination and also improve on its quality of service. The algorithm was integrated into the C++ backend codes of ns-2 as a new traffic meter algorithm:

If $Tc(t) - B \geq 0$, the packet is marked as green
and $Tc \leftarrow Tc - B$ else

If $Tc(t) < B$, generate a number (N) after 100 clock tick
if $N \leq 5$, the packet is red, else

If $5 < N < 40$, the packet is yellow, is decremented by B else
if $N \geq 40$ the packet is green and both Tp and Tc are decremented by B,
the packet is green and both Tp and Tc are decremented by B.
else

If $Tc(t) - B < 0$ and $Tp(t) - B \geq 0$, the packet is marked yellow and $Tp \leftarrow Tp + 1$ else

If $Tp(t) < B$, generate a number (N) after 100 clock tick
if $N \leq 21$, the packet is red, else

If $21 < N < 40$, the packet is yellow, is decremented by B else
if $N \geq 40$ the packet is green and both Tp and Tc are decremented by B.

3.3. Ranking System

Ranking system was used for results evaluation of the performance metrics to come up with the best meter in terms of each aforementioned metric for the meter algorithms considered. This was done because each one of them had one metric or the other where they performed better. Also, the higher the throughput and fairness rate metrics, the better the performance of the meters. On the contrary, the lower the packet loss rate and the one-way packet delay metrics, the better the performance of the meter. These two conditions informed the usage of ranking system to come out with the best traffic meter algorithm based on throughput, fairness rate, loss rate and one-way packet delay on TCP and UDP as traffic agents.

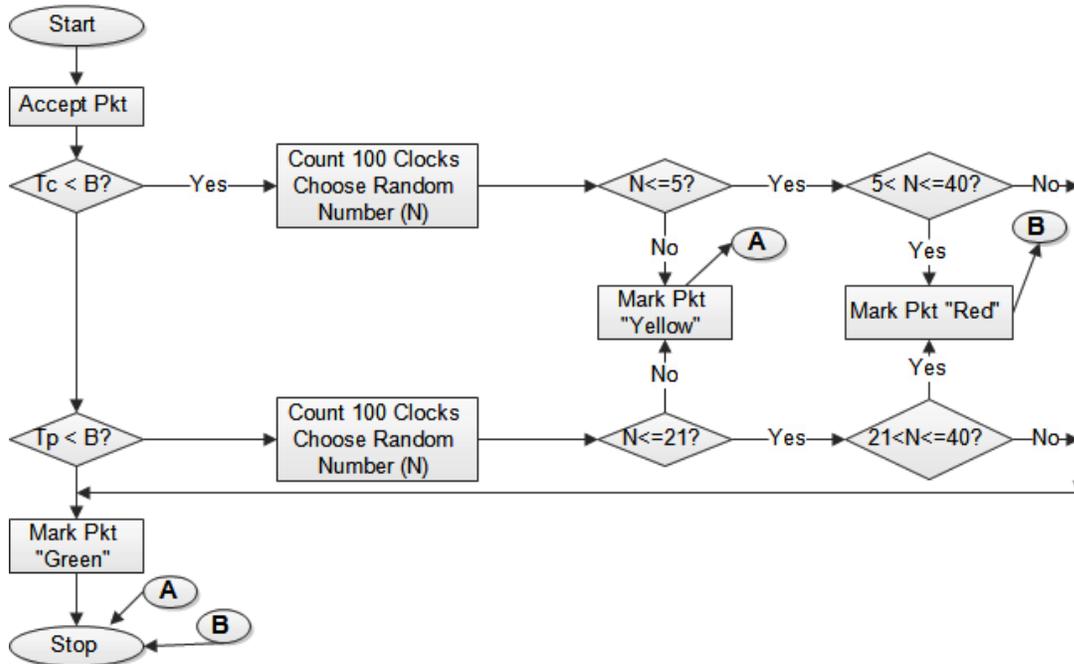


Figure 2. Flow chart diagram for Improved TRTCM (ITRTCM).

4. Results and Discussions

The results obtained from the simulations were analyzed, evaluated and interpreted in this section. The analyses of the results were based on throughput, loss rate, fairness rate and one way packet-delay as performance metrics.

Results Analyses for the improved Meter (ITRTCM) and TRTCM Meter Algorithms

The same parameter settings were used for the simulation of the improved version of two rate three color meter (ITRTCM) algorithm. The packet sizes used were 1000 bytes and 2000 bytes for the two scenarios of the simulation experiments respectively. The analyses of the improved version (ITRTCM) and the existing algorithm (TRTCM) were tabulated in Table 1 using the same performance metrics with UDP and TCP as traffic agents.

4.1. Analyses Based on Throughput

The analyses of the simulation experiments for ITRTCM and TRTCM using UDP as traffic agent for the first and second scenarios were shown in Table 1 and Figure 3. With ranking system, the higher the throughput value the better the performance of the traffic meter. The Average throughput value (with 50463) for improved meter (ITRTCM) algorithm was ranked first while the existing meter (TRTCM) algorithm (with 50460) was ranked second as shown in Table 1 and Figure 3. In terms of throughput, ITRTCM algorithm outperforms TRTCM algorithm

4.2. Analyses Based on Average Fairness

With ranking system, the higher the average fairness rate value the better the performance of the traffic meter. The average fairness rate value (with 0.2726) for improved meter (ITRTCM) algorithm was ranked first while the existing meter (TRTCM) algorithm (with 0.2632) was ranked second as shown in Table 1 and Figure 3. In terms of average fairness rate, ITRTCM algorithm outperforms TRTCM algorithm.

4.3. Analyses Based on Loss Rate

With ranking system, the lower the average loss rate percentage value the better the performance of the traffic meter. The average loss rate percentage value (with 74.595) for improved meter (ITRTCM) algorithm was ranked first while the existing meter (TRTCM) algorithm (with 74.700) was ranked second as shown in Table 1 and Figure 3. In terms of average loss rate percentage, ITRTCM algorithm outperforms TRTCM algorithm.

4.4. Analyses Based on One-Way Packet Delay

With ranking system, the lower the average one-way packet delay value the better the performance of the traffic meter. The average one-way packet delay value (with 0.85845ms) for improved meter (ITRTCM) algorithm was ranked first while the existing meter (TRTCM) algorithm (with 0.86695ms) was ranked second as shown in Table 1 and Figure 3. In terms of average one-way packet delay, ITRTCM algorithm outperforms TRTCM algorithm.

Conclusively, the improved meter (ITRTCM) algorithm would be very good for the deployment of multimedia applications because most of the multimedia applications are deployed on UDP traffic agents which will have great positive impact on the quality of service (QoS).

Table 1. Analyses for ITRTCM and TRTCM using UDP as traffic agent for first and second scenarios.

Traffic Meters		Loss Rate Percentage	Throughput	Fairness Rate	One-Way Packet Delay (ms)
ITRTCM (UDP)	1 st Scenario	74.53	50462	0.2632	0.8591
	2 nd Scenario	74.66	50464	0.282	0.8578
	Average Rank	74.595	50463	0.2726	0.85845
		1 st	1 st	1 st	1 st
TRTCM (UDP)	1 st Scenario	74.7	50462	0.2643	0.8672
	2 nd Scenario	74.7	50458	0.2621	0.8667
	Average Rank	74.7	50460	0.2632	0.86695
		2 nd	2 nd	2 nd	2 nd

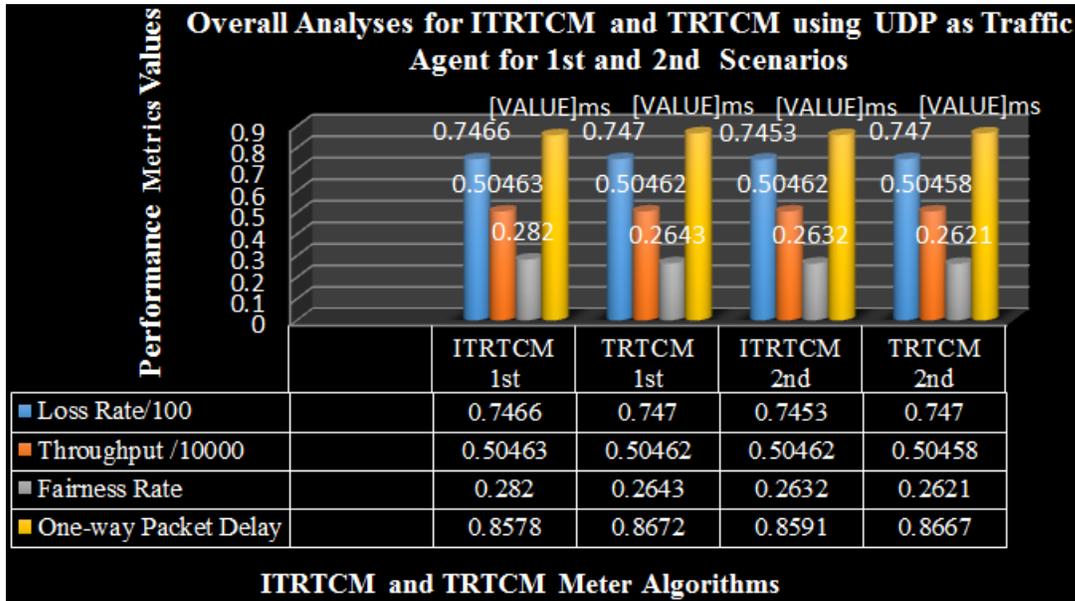


Figure 3. Overall analyses for ITRTCM and TRTCM using UDP as traffic agent for the two scenarios.

5. Conclusion

The need to support service differentiations and quality of service in the Internet has become an open research issue since 1998. The ATM network tries its best to resolve the issue through QoS mechanisms such as differential admission control, multi-queue buffering, differential traffic scheduling, quality of service routing and a host of other operations at the network switches. In order to achieve QoS, an admission control with penalty at the network router along the path to packet destination is needed. This calls for traffic measurement using meter algorithms at the routers. Research efforts towards this quest for quality of service traffic meters have come up with various algorithms like TRTCM, TBM, SRTCM, TSW3CM, TSW2CM etc. This research concentrates on the design of an improved meter algorithm for quality of service admission control in the Internet.

When TRTCM was improved upon to get more packets to the destination, ITRTCM algorithm was better than the existing TRTCM algorithm for the two packet scenarios (1000 bytes and 2000 bytes) for the four performance metrics analyses using UDP as traffic agent. Therefore, a scalable QoS admission control can be achieved using ITRTCM meter algorithm on UDP traffic agent along with the recommendation of Differentiated services for network routers as it guarantees higher network throughput.

References

Pandit, K., Quality of service performance analysis based on network calculus. 2006, TU Darmstadt.

Forouzan, B. A. and F. Mosharraf, Computer networks: a top-down approach. 2012: McGraw-Hill.

Kurose, J. and K. Ross, Computer Networking, A Top-down Approach Featuring the Internet Book. 2000: Pearson Education, Limited.

Clark, D. D. and W. Fang, Explicit allocation of best-effort packet delivery service. Networking, IEEE/ACM Transactions on, 1998. 6 (4): p. 362-373.

Braden, R., D. Clark, and S. Shenker, Integrated services in the internet architecture: an overview. 1994, rfc 1633, June.

Blake, S., et al., An architecture for differentiated services. 1998.

Blake, S., et al., An architecture for differentiated services. 1998.

Lochin, E. and P. Anelli, TCP throughput guarantee in the DiffServ Assured Forwarding service: what about the results? annals of telecommunications-Annales des télécommunications, 2009. 64 (3-4): p. 215-224.

Georgoulas, S., Admission control and bandwidth management in IP differentiated services networks. 2007, University of Surrey.

Nordström, E., Course Compendium in Network Modeling. 2006, Citeseer.

Baker, J. H. F., W. Weiss, and J. Wroclawski, 'RFC: 2597 Assured Forwarding PHP Group'. Request for Comments, IETF, 1999.

Heinänen, J. and R. Guérin, A two rate three color marker. 1999.

Jain, R., Congestion control in computer networks: Issues and trends. *IEEE Network*, 1990. 4 (3): p. 24-30.