Technical University of Denmark

DTU

IPTV traffic management using topology-based hierarchical scheduling in Carrier Ethernet transport networks

Yu, Hao; Yan, Ying; Berger, Michael Stübert

DTU Library
Technical Information Center of Denmark

IPTV Traffic Management Using Topology-Based Hierarchical Scheduling in Carrier Ethernet Transport Networks

Hao Yu, Ying Yan, and Michael S. Berger
Department of Photonics Engineering
Technical University of Denmark
2800 Kgs. Lyngby, Denmark
{haoyu, yiya, msbe}@fotonik.dtu.dk

Abstract—**Carrier Ethernet is becoming a favorable access technology for Next Generation Network (NGN). The features of cost-efficiency, operation flexibility and high bandwidth have a great attraction to service providers. However, to achieve these characteristics, Carrier Ethernet needs to have Quality of Service (QoS) provisioning abilities, which guarantee end-to-end performances of voice, video and data traffic delivered over networks. This paper introduces a topology-based hierarchical scheduler scheme, which controls the incoming traffic at the edge of the network based on the network topology. This work has been carried out as a part of the research project HIPT (High quality IP network for IPTV and VoIP) founded by Danish Advanced Technology Foundation.**

Keywords-traffic management; hierarchical scheduling; Carrier Ethernet; IPTV

I. INTRODUCTION

Ethernet – an incontestable technology that has dominated the Local Area Networks (LAN) for decades – is now being developed and extended to become a possible choice for Metropolitan Area Networks (MAN). The pressures from competition and changing communications, and entertainment needs of residential customers are driving network operators to upgrade their networks to be capable of voice, video and data delivery (also known as triple play services). Services carried by separate networks, such as Public Switched Telephone Network (PSTN), the cable television network, and the Internet, will be provided in a single network. The network traffic load will be greatly increased by broadcasting or multicasting video and Video on Demand (VoD) services through IP networks (known as IPTV services). To ensure that the quality of IPTV services is guaranteed without damaging Voice-over-IP (VoIP) services and high-speed Internet access, different QoS requirement of each type of traffic must be ensured by the converged network. Thus, a fine-grained traffic management scheme is demanded.

From the work of [1], the flow-based scheduling scheme using Deficit Round Robin (DRR) algorithm has been evaluated and shown to be an appropriate choice for the IPTV service in Carrier Ethernet transport networks. It is true that the flow-based scheduling scheme is capable of treating traffic flows separately and providing better protection than class-based scheduling, but it requires the network operator to upgrade the entire network with flow-based scheduling nodes. Under economic consideration, network operators consider not only the capability of the network, but also the corresponding cost to deploy such a network. It has been discussed in [2] that the Carrier Ethernet can greatly reduce the consequences of the complexity associated with the large scale of carriers' networks by being a cost-effective replacement for SONET/SDH. To keep this preferable feature and to reduce the required deploying period, the topology-based hierarchical scheduling scheme is proposed in this work. Under different conditions of traffic load and burstiness, performances of hierarchical scheduler are examined in the Carrier Ethernet transport networks in comparison with the flow-based and the class-based scheduling schemes.

The remaining parts of this paper are structured as follows. In Section II, we provide an overview of Carrier Ethernet, introducing two key technologies and the concepts of control plane and transport plane. In Section III, different scheduling algorithms are compared and the advantages of DRR scheduling are explained. In Section IV, we discuss the benefit of hierarchical scheduling and demonstrate the concept. In Section V, simulation results are presented and discussed.

II. CARRIER EHTERNET OVERVIEW

The Metro Ethernet Forum (MEF) has provided a clear definition of Carrier Ethernet. Based on the description from MEF, Carrier Ethernet is defined as an omnipresent, standardized, carrier-class service and network defined by five attributes that distinguish Carrier Ethernet from LAN based Ethernet:

- Standardized Services
- Scalability
- Reliability
- Quality of Service
- Service Management

To use Ethernet as a transport technology, which requires customer separation and manageability, Traffic Engineered Provider Backbone Bridging (PBB-TE) and MPLS Transport Profile (MPLS-TP) have been developed and proposed as Carrier grade Ethernet transport network solutions.

PBB-TE is the most recent development after several years of work by the IEEE and ITU-T aiming at improving and enhancing Ethernet technology for use in carrier networks. PBB-TE reuses current implementations of double-tagging and VLANs and combines them with the network separation and layering principles of PBB [3]. MPLS-TP, the former T-MPLS, is now developed under the cooperation of ITU-T and IETF. It promises a solution that provides familiar and reliable packet-based technology, i.e. MPLS, in a way that is aligned with circuit-based transport networks. Both technologies aim at providing a connection-oriented packet switching transport network, where traffic is tunnelled and delivered to the destinations.

In Fig. 1, the Carrier Ethernet contains two separate and independent domains, the control plane and the transport plane. The specification of the control plane implementation is not yet finished in the process of standardization. The main functions of the control plane include, however, QoS mapping, label distribution, Connection and Admission Control (CAC) [4]. In the transport plane, traditional switches should be updated with advanced functionalities in order to provide carrier grade services and to guarantee the QoS performance, especially for the real-time traffic like IPTV.

III. SCHEDULING ALGORITHMS

In order to achieve the QoS requirement, limited resources in the network need to be fairly allocated to the users by scheduling algorithms. Various scheduling approaches have been proposed and classified into two main categories: timestamp-based (also sorted-priority) scheduling and frame-based scheduling.

In timestamp-based schedulers, a global virtual time is maintained to approximate a fluid model, which is the Generalized Processor Sharing (GPS) [5]. Timestamps are generated through this virtual machine for each arriving packet, and the packets are inserted into a priority queue based on their timestamp values and transmitted in order of increasing timestamps [6]. Examples of timestamp-based schedulers are

Figure 1.   Carrier Ethernet transport plane

Weighted Fair Queuing (WFQ), Self-Clocked Fair Queuing (SCFQ), Start-time Fair Queuing (SFQ), and Worst-case Fair Weighted Fair Queuing (WF^{2}Q). They vary in the manner of calculating the global virtual time function. These timestamp-based schedulers generally provide good fairness and low latency, but these methods are not very efficient owing to the complexity involved in computing the system virtual time and sorting the packets based on the timestamps [7].

On the other hand, in frame-based schedulers, like Deficit Round Robin (DRR) and Elastic Round Robin (ERR), packets are served in a round robin fashion [8]. These schedulers do not need to calculate the virtual time function. Thus, the packet processing work is low in complexity and the design of such frame-based schedulers is fairly simple. Deficit Round Robin (DRR), or sometimes called Weighted Deficit Round Robin (WDRR) [9], is one of the early frame-based scheduling algorithms proposed to overcome the unfairness. DRR shows near-perfect throughput fairness with very low complexity of packet processing, in comparison with other fair queuing algorithms. The WFQ and WF^{2}Q scheduling schemes require $O(N)$ time to complete a scheduling decision, where N is the number of sessions sharing the outgoing link of the router/switch. The SCFQ approach reduces the time complexity but still holds the $O(log(N))$ bottleneck. DRR, by contrast, has a low complexity of $O(1)$, which is much simpler than other timestamp-based algorithms. In [9], Shreedhar and Vaghese have concluded that the DRR provides near-perfect isolation at low implementation cost and can be combined with other fair queuing algorithms to offer better latency bounds.

It has been discussed in [1] that the advantages of DRR, such as low complexity and near-perfect isolation, are significant when a large amount of packets need to be processed in a very short period. Thus, DRR is selected as a basic scheduling algorithm in this work. DRR can be implemented together with advanced buffer management and thus can support sufficient QoS differentiation between connections and ensure that the maliciously behaving connections do not adversely impact the bandwidth and the QoS performance of other conforming traffic flows. Incoming packets from different flows are sorted and stored in queues. Based on the arrangement of queues, which determines how the flows are treated, schedulers can process packets based on the class of traffic or flow identification. Different queuing schemes result in different bandwidth sharing and QoS provision. Furthermore, in this work, we enhance our work in [1] and propose an intelligent topology-based scheduler using flow-based DRR, which takes into account the network topology.

IV. HIERARCHICAL SCHEDULING

A. Drives for Hierarchical Scheduling

Although it could bring the operator a network which provides QoS guarantee by replacing all the switches with more advanced ones, at the same time it places a considerable burden on the network operator, especially when the size of the network is fairly large. Distributing intelligence, in terms of large size of memory, advanced scheduling algorithm, flow control ability and so forth, to all the nodes in the network will inevitably needs a management platform that can manage and
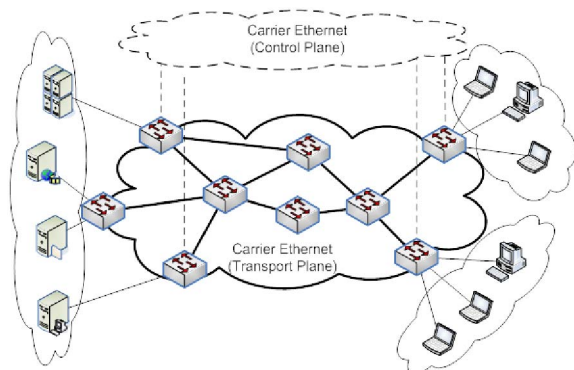
configure the switches efficiently. This could be one of the drawbacks of distributed intelligence. Besides, the resources, e.g. the size of queue buffer, which the operator brings to each of the nodes, may not be fully utilized. And the deploying time of a large network will be relatively long. Thus, distributing intelligence, i.e. replacing all the switches with advanced ones, will not be cost-efficient enough.

Another solution is to introduce an intelligent switch with the knowledge of the network topology and to place it at the edge of the network. This node should be able to manage the traffic on behalf of other nodes lacking of traffic engineering ability, and thus avoids traffic congestion in the network. Topology-based hierarchical scheduling scheme is an attractive candidate to such a situation. In order to effectively manage the packet flows, the topology-based hierarchical scheduling switch has the knowledge of the network topology and the capacity of nodes on each level. By mentioning the term "capacity of nodes", we mean the output link rate. Based on the information, the scheduler could guarantee that no traffic congestion will occur in the nodes on each lower level, and at the same time allows the operator to manage the network in an efficient way. For the sake of simplicity, we use "hierarchical scheduling" instead of the full name "topology-based hierarchical scheduling".

B. Overview of Flow-based Scheduling

Owing to the advantages of flow-based scheduling scheme, such as per flow scheduling, flow protection, and the low complexity of DRR mentioned in the previous section, hierarchical scheduler should take advantage of the flow-based algorithm, which has been investigated in [1].

The structure of a flow-based scheduler is demonstrated in Fig. 2. The flow-based scheduler consists of a central scheduler and several subschedulers. Within each queue block of the same traffic class, the subscheduler runs DRR algorithm on backlogged subqueues. The central scheduler executes DRR algorithm on backlogged subschedulers and is in charge of allocating bandwidth to each traffic class. As the name indicates, the scheduler provides per flow level packet scheduling instead of traditional per class level scheduling. The advantages of such a flow-based scheduler include (1) protecting flows against any other misbehaving flows, (2) eliminating delay and jitter (delay variation) spreading across flows and (3) low in complexity.

C. Architecture of Hierarchical Scheduling

Hierarchical scheduler uses the concept demonstrated in Fig. 2 as a basic component and combines multiple DRR schedulers based on the topology of the network. Since the architecture of hierarchical scheduler depends on the actual network topology, it is reasonable to provide an example network first and then shows the corresponding structure of the hierarchical scheduler.

We choose a simple binary tree topology as an example shown in Fig. 3. Tree topology can be found in many networks for IPTV dissemination, and is a natural way to multicast or broadcast IPTV traffic to end users. The node labeled with *HS* represents the hierarchical scheduling switch. It is located at the edge of the network and sends the received traffic flows into the network. The network connected to *HS* is divided into 4 levels, i.e. A, B C and E. The output link rate is the same for nodes on the same level, but is reduced by half from Level A to Level E. The nodes on Level A, B, and C represent simple switches which lack the functionality of traffic management. The 8 nodes on Level E represent the end nodes.

Traffic flowing from the external source, i.e. *IPTV source*, to the 8 end nodes, will first enter into *HS* and then further down to the rest of the network. For the sake of simplicity, we define a flow based on the *<source, destination, type of traffic>* pair. 8 flows of the same traffic type from the same source are sent to 8 end nodes respectively. Since the nodes on Level A, B, and C do not have the ability to manage traffic, they will simply discard packets if the temporary storage buffer overflows. Thus it is *HS*'s responsibility to take care of the traffic. Knowing the capacity of each node, *HS* will not release more traffic than the simple switches can handle so that traffic
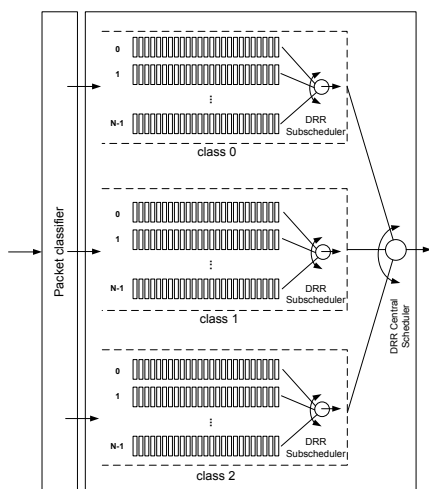

Figure 2.   Detailed schematic structure of flow-based scheduler for one output port. The dashed block represents a queue for one traffic type. Within each type, several logical subqueues are generated, each of which stores packets coming from one flow.

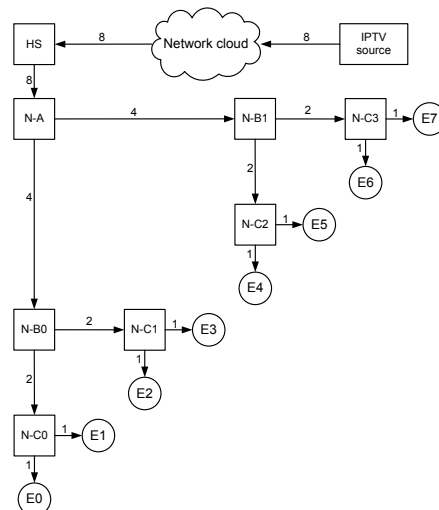

Figure 3.   An simple binary tree topology network, with link rate and number of flows reduced by half from Level A to Level E. HS node is connected with N-A at the edge of the network.

congestion can be avoided.

Based on the example shown in Fig. 3, the structure of hierarchical scheduler can be formed by a combination of several DRR schedulers as shown in Fig. 4. Packets enter into the scheduler from the top and leave from the packet memory. The packet classifier generates a token for each arriving packet and forwards it to one of the token queues based on the flow mark *<source, destination, type of traffic>*. The token carries information such as packet length, flow mark and a packet location pointer. The packet is then stored in the packet memory based on the packet location pointer. Token schedulers denoted X_{Ei} represent nodes Ei, and they schedule packets for the links between N-Ci and Ei. X_{Ei} schedulers run DRR algorithm on all backlogged token queues and correspond to the central schedulers in Fig. 2. Since only one type of traffic from one source is provided in Fig. 3, subschedulers are not shown. If the traffic profile is more complex, corresponding subqueues and subschedulers will be created as shown in Fig. 2.

The similar operation applies to the token schedulers in Level C. An X_{Ci} scheduler manages traffic for the links between N-Bi and N-Ci. It runs DRR algorithm on all backlogged X_{Ei} schedulers. A backlogged scheduler means that the scheduler has targeted a token and will transmit it once a grant is received. In Level B, token schedulers X_{Bi} execute DRR algorithms on backlogged X_{Ci} schedulers, and in Level A the master token scheduler X_A runs DRR on backlogged X_{Bi}. The X_A selects a token scheduler in Level B and grants permission. The permission will be passed down to one of the schedulers in Level C and then to Level E. Once the permission is received by a X_{Ei}, a token path is established and the previously targeted token will be immediately passed to X_A. The all the deficit counters along the path will be updated. The X_A scheduler examines the received token and releases the corresponding packet from the packet memory. A packet transmission including token selection is thus completed. The token transmission rate of each token scheduler corresponds to the actual like rate of each node in the network. It is used by
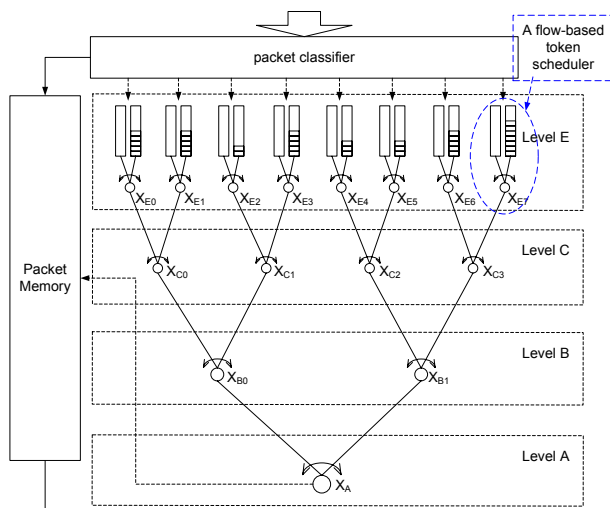

Figure 4. The structure of the topology-based hierarchical scheduler based on a binary tree topology.

token schedulers to control the outgoing traffic rate in order to avoid congestion in the network. If the release of tokens is not controlled, it is very likely that more tokens are passed from one scheduler to another, which will inevitably cause packet overflow in the corresponding node.

V. PERFORMANCE EVALUATION

In this section, we evaluate the statistical multiplexing performance of the hierarchical scheduling scheme by simulations in OPNET [10]. Flow-based and class-based scheduling schemes are compared with the hierarchical scheduling in terms of traffic delay and delay variation (jitter).

A. Traffic Parameter and Network Setup

Three networks of the same binary tree topology shown in Fig. 3 are created, each of which uses one scheduling scheme, i.e. class-based, flow-based, and hierarchical scheduling. All three networks are connected to the same IPTV traffic generator and provided with 16 identical traffic flows simultaneously. Each flow is configured to be sent to one end node and has an average bandwidth of 7 Mbps. Maintaining the input traffic flows unchanged, we reduce the output link rate of the edge node. For instance in Fig. 3, the output rate of the link between *HS* and *N-A* is reduced. We use the input-output rate ratio as the x-axis. Since the output rate is reduced gradually, the ratio increases from the initial value of 1.0.

B. Simulation Results

Becasue of the burstiness of the input traffic and the aggregation of flows, the capacity of the link can be saved by using statistical multiplexing to reduce the link rate. As the link rate decreases, the average end-to-end delay and jitter will increase if the traffic is not highly bursty. Fig. 5 provides the average end-to-end delay comparison between the class-based, flow-based and hierarchical scheduling under various input-output rate ratios. Fig. 6 shows the jitter comparison under the same range of input-output rate ratio.

In Fig. 5, hierarchical scheduling has improved the performance on average end-to-end delay. The curve of hierarchical scheduling is below the other two. To achieve the same end-to-end delay, hierarchical scheduling can endure more link capacity being reduced. As the input-output rate ratio increases, average end-to-end delay increases for all three schemes but the slope of hierarchical scheduling curve becomes lower than the class-based.

In Fig. 6, the three scheduling methods, i.e. class-based, flow-based and hierarchical scheduling have shown alike performance in terms of traffic jitter under different input-output rate ratio. As the input-output rate ratio increases, the jitter values of all three schemes become greater. Hierarchical scheduling has little improvement on the jitter performance.

From the results it can be obtained that the improvement of statistical multiplexing gain (SMG) factor by the hierarchical scheduling scheme is limited. The traditional way of users browsing websites allows the operator to reduce the required bandwidth for the aggregated flows. If there are 1000 users, for instance, and each is guaranteed 10 Mbps download bandwidth, the operator can assign 200 Mbps bandwidth for the aggregated traffic to satisfy the requirement since not all the

users need the resource at the same time. The SMG is thus become 50 under this circumstance. When IPTV services are introduced to a network, the SMG factor will begin to decrease because the traffic is low in burstiness, but high in bandwidth consuming. The traffic characteristic will be very different from normal website browsing or file downloading.

The advantage of the hierarchical scheduling scheme is that it can provide the nearly same performance as the distributed intelligence fashion. By learning the network topology through the management plane or manual configuration, the scheduler at the edge of the network forms a mapping structure with virtual token schedulers. The cooperation between each token scheduler is far more efficient than the cooperation between different nodes. The centralized intelligence way of traffic management can be considered as a solution.

VI. CONCLUSION

In this paper, we present a topology-based hierarchical scheduling algorithm. By mapping the network topology into
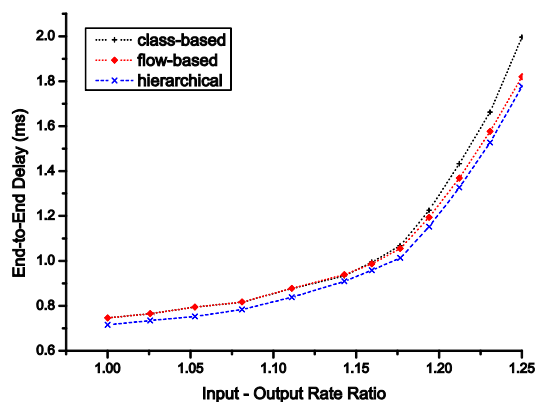

Figure 5. Comparison between class-based, flow-based, and hierarchical scheduling schemes in terms of average end-to-end traffic delay under different input-output rate ratio.
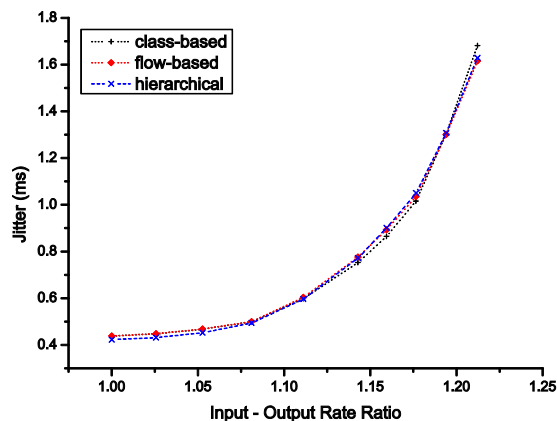

Figure 6. Comparison between class-based, flow-based, and hierarchical scheduling schemes in terms of traffic jitter under different input-output rate ratio.

the edge node, the hierarchical scheduler can manage traffic flows on behalf of other simple nodes. DRR Token schedulers are combined to form the mapping structure of the network. It is important to mention that the combination can adapt to any network topology, not limited to the binary tree topology used in the example. Tokens are generated for each arriving packet and are classified and stored in token queues based on the flow identification. Tokens are passed from the low-level scheduler to the master scheduler and then the corresponding packets are released for transmission.

The premiere simulation results from the comparison between class-based, flow-based and hierarchical scheduling show that the nearly same performance can be achieved by the hierarchical scheduler. Centralizing the intelligence to an edge node can thus provide the same performance as distributing intelligence to each node. This can be beneficial when it comes to network deployment. Instead of replacing or upgrading the simple nodes with nodes having advanced scheduling algorithm and buffer management, operators can place an intelligent node at the edge of the network. For real-time video services, it is crucial to guarantee the QoS for each traffic flow. Topology-based hierarchical scheduling provides network deployment and operation flexibility without degrading the QoS performance, which is a significant factor for IPTV service in Carrier Ethernet transport networks.

Further investigation into the topology-based hierarchical scheduling scheme will be carried on and we will continue to develop the idea the topology-based hierarchical scheduling.

REFERENCES

[1] Hao Yu, Ying Yan, and Michael S. Berger, "IPTV traffic management in Carrier Ethernet transport networks", in Proc. OPNETWORK 2008, Aug. 2008.

[2] Andy Reid, Peter Willis, Ian Hawkins, and Chris Bilton, "Carrier Ethernet", IEEE Communication Magazine, Sep. 2008.

[3] Don Fedyk and David Allan, "Ethernet data plane evolution for provider networks", IEEE Communications Magazine, vol. 46, issue 3, pp. 84-89, 2008

[4] Rong Fu, Yanmeng Wang, and Michael S. Berger, "Carrier Ethernet network control plane based on the Next Generation Network", in Proc. ITU Kaleidoscope Conference, pp. 293-298, 2008.

[5] Mouad Ben Mamoun, Jean-Michel Fourneau, Nihal Pekergin, "Analyzing weighted round robin policies with a stochastic comparison approach", Computers & Operations Research 35 (2008) 2420-2431, Jan. 2007.

[6] Anujan Varma, Dimitrios Stiliadis, "Hardware implementation of fair queuing algorithms for asynchronous trasfer mode networks", IEEE Communication Magazine, Dec. 1997.

[7] Salil S. Kanhere and Harish Sethu, "Fair, efficient and low-latency packet scheduling using nested deficit round robin", IEEE Workshop on High Performance Switching and Routing, 2001.

[8] T. Al-Khasib, H. Alnuweiri, H. Fattah, V.C.M. Leung, " Fair and efficient frame-based scheduling algorithm for multimedia networks", in Proc. the 10th IEEE Symposium on Computers and Communications, 2005.

[9] M. Shreedhar and George Varghese, "Efficient fair queuing using deficit round-robin", IEEE/ACM Transactions on Networking. vol. 4, No. 3, Jun. 1996

[10] OPNET Modeler 14.0. http://www.opnet.com.