ITC19/ Performance Challenges for Efficient Next Generation Networks 1-10
LIANG X.J. and XIN Z.H.(Editors)
V.B. IVERSEN and KUO G.S.(Editors)
Beijing University of Posts and Telecommunications Press

A Distributed Bandwidth Fair Allocation Algorithm for RPR Networks [#]

Tan Liansheng, Wang Hanwu and Moshe Zukerman

The Computer Science Department of Central China Normal University, Wuhan 430079, PR. China (e-mail: l.tan@mail.ccnu.edu.cn)

The Australian Research Council Special Research Center for Ultra-Broadband Information Networks (CUBIN) an affiliated program of National ICT Australia, EEE Dept., The University of Melbourne, Victoria 3010, Australia (e-mail: mzu@unimelb.edu.au)

Abstract: Despite the rapid development of the resilient packet ring (RPR) technology, the available bandwidth allocation algorithms for RPR networks do not provide satisfactory solutions to meet their performance requirements. To attack this problem, this paper proposes an algorithm, which we call the distributed bandwidth fair allocation (DBFA) algorithm, that achieves the key RPR performance goals, i.e., fairness, high utilization and spatial reuse. The algorithm is distributed and uses a simple proportional control mechanism to allocate bandwidth among competing flows in a weighted manner. In order to realize global coordination on the entire RPR ring, one global control packet traverses around the ring to make every node on the ring dynamically adjust its sending rate, so it eventually achieves its fair share of the ring bandwidth. As a result, global fairness as well as high utilization and maximal spatial reuse over the RPR ring can be achieved. Simulation results verify the satisfactory performance of the DBFA algorithm.
Keywords: Resilient packet ring (RPR) network, Spatial reuse, Proportional controller, Control packet, Distributed bandwidth fair allocation (DBFA) algorithm, Fairness Control Field (FCF)

1. INTRODUCTION

The Resilient Packet Ring (RPR) [1], [2], [3] IEEE 802.17 standard is a new network structure and data transport technology for the ring metropolitan area networks (RMANs). The RPR network is based on a dual-ring topology. RPR is the successor of earlier technologies such as SONET, Gigabit Ethernet, DQDB and FDDI, which have been previously used in metropolitan networks. The predecessors of RPR have had their own respective shortcomings, which have hindered their use as metropolitan networks. RPR, however, overcomes many of their shortcomings and inherits their advantages. RPR is efficient, robust and economically attractive, so it is also considered the primary technology

[#] This research has been partially supported by National Natural Science Foundation of China under Grant No. 60473085 and partially by the Australian Research Council.

for the next metro IP networks.

Unlike token ring [4], the data transmission through an RPR node does not depend on whether it holds the token or not and the RPR provides destination release of the data traffic. Therefore, spatial reuse [5]-[8] can be achieved in RPR to better utilize the ring bandwidth. Unfortunately, the ability of the most current RPR algorithms [9 - 11] to achieve their desired objectives (fairness, high utilization and spatial reuse) simultaneously is limited.

A known problem in aforementioned Medium Access Control (MAC) protocols is that a node may starve its downstream nodes. To solve this problem, the RPR standard defines a fairness policy that the upstream nodes must inject traffic at a rate according to the downstream congestion situation. Two well-known modes, namely Aggressive Mode (AM) [6], [10], [11] and Conservative Mode (CM) [9], [11], are used in most RPR algorithms. However, the mechanism for assigning rates at each of the modes is not optimal and may lead to severe oscillations and hence performance degradation. Inheriting the mechanisms of AM and CM, the distributed virtual-time scheduling in ring (DVSR) [7, 8, 12] scheme achieves better performance than AM and CM by better rate allocation assignment.

All these algorithms have one common mechanism that the rate adjustments are controlled by feedback based on reaction to congestion somewhere, so why not make the adjustment as early as possible instead of waiting for the congestion to occur. Obtaining such information earlier will reduce oscillations and achieve better performance [13]. In this paper, we implement a controller into our algorithm so that the rate adjustment is based on the congestion state of the system and may further satisfy stability requirements by choosing the proper parameters. This detail is, however, been omitted due to space limit.

We introduce a new algorithm termed *distributed bandwidth fair allocation* (DBFA), which is designed to achieve the RPR key performance requirements. The algorithm, which is performed at all RPR nodes in a distributed way, uses a simple proportional control mechanism to allocate the link bandwidth among all the competing flows crossing this link in a weighted manner. In order to realize global coordination on the entire RPR ring, we propose to use a certain control packet that runs around the ring to collect load information on every node. The collected information is written into the control packet in a common field which we call *fairness control field* (FCF). In our RPR dual-ring topology, we propose to have two opposite directional control packets - one on each ring. The control packet on one ring controls the data traffic on the other ring. As the control packet propagates along the ring, each node on the ring uses the information from the FCF and dynamically adjusts its sending rate and eventually reaches its fair share of the ring bandwidth. As a result, we achieve global fairness as well as high utilization, maximal spatial reuse and stability.

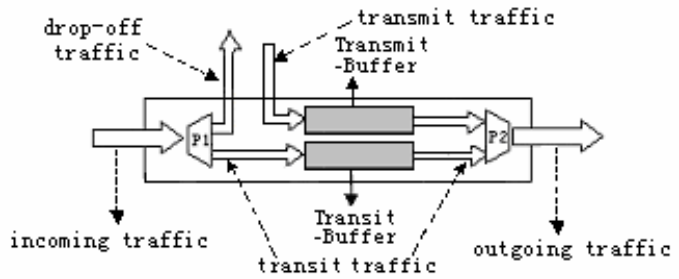

Figure 1. The data traffic process in the RPR node

The remainder of this paper is organized as follows. In Section 2, we analyze the characteristics of RPR networks and show how data traffic is processed in RPR nodes. In Section 3, we present the DBFA algorithm in detail. Simulation results that demonstrate the superiority of DBFA are presented in Section 4. Finally we conclude the paper in Section 5.

2. BASIC CONCEPTS

In an RPR network, each node can act either as a source node that sends data, or as a destination node that receives (removes) data or as a switch node to transit the data traffic, and buffers are available in the RPR nodes for temporary storage of these data traffic streams. Figure 1 describes the data traffic process for an RPR node.

In Figure 1, traffic originated from a source node is called *transmit traffic*, traffic removed by this node is called *drop-off traffic*, and traffic transited by the node is called *transit traffic*. The corresponding buffers, namely, *Transmit-buffer* and *Transit-buffer* are used for *transmit traffic* and *transit traffic*, respectively. These traffic streams are to be processed by processors P_1 and P_2; the incoming traffic is to be processed by P_1, as a result, the drop-off traffic will be separated from the transit traffic and removed by this node. The transit traffic and the transmit traffic are processed by processor P_2 before they enter the outgoing link of the node.

One notable issue related to fair bandwidth allocation at a certain link in RPR networks is the desire to fairly control the so-called IA (ingress-aggregated) flow, which is the aggregation of the individual flows originating from the same source node while ending at different destinations. Note this is different from the previous flow-based fairness criterion, in which the traffic granularity for fairness determination on a certain link is the single flow. The other issue is efficiency requirements. While we have full allocated bandwidth on a single link fairly, it must also be done with the entire network in the most efficient way, so as to achieve the so-called spatial reuse to ensure the high utilization on all links. The RPR fair algorithms should be designed to achieve fairness among all the IA flows and maximal spatial reuse over the entire RPR ring.

3. THE DBFA ALGORITHM

The goal of our DBFA algorithm is to achieve fairness and high utilization. The algorithm relies on explicit rate feedback and adopts a known control theory method to adjust the sending rates of flows passing by each link. The operation is distributed and each node performs the bandwidth allocation separately. For each node, the bandwidth on its outgoing link is allocated to the different IA flows that pass by the link in a weighted manner. We use the per- destination controllers to determine the sending rates of the individual flows originating from the same source node. Furthermore, to realize global control and coordination, we use one global control packet, containing the FCF with load information of all the RPR nodes, that rotates around the ring. Specifically, we divide the FCF in the control packet into distinguished FCF sub-fields so that each node has its own FCF sub-field in the

control packet. Each such FCF sub-field has two functions for this node: (i) the one where the node writes its load information, and (ii) the other where the node get the control message to adjust its sending rate. Every time the control packet passes by an RPR node, the relevant FCF sub-field is updated and also the sending rate of this node is adjusted. Let us consider one of the RPR ring nodes, say node *n*, as an example, to analyze our DBFA algorithm and the operation process of which is illustrated in Figure 2.

As we can see from Figure 2, flow 1, flow 2, ⋯, flow *m* are the *m* transit flows that pass through node *n*, and flow *n* is the transmit flow originating from node *n*, note all these flows belong to the IA flow. The control packet is to collect load information and update the relevant FCF sub-field for each node and node *n* reads its own FCF sub-field when the control packet reaches it. Recall that there are two opposite directional control packets each of which is in the direction opposite to the data transmission traffic it controls. Buffer-a is used for the transmit traffic and Buffer-p is used for the transit traffic, the P-Controller is used to adjust the rate of the transmit traffic and the transit traffic according to their buffer occupancies. The bandwidth on the outgoing Link *L* of this node is shared by these flows in a weighted manner. We distinguish the transit traffic from the transmit traffic that pass through the same node and give certain priority to the former, to ensure the data transmission along the ring is successfully done. Specifically, when the occupancy of Buffer-p exceeds its target

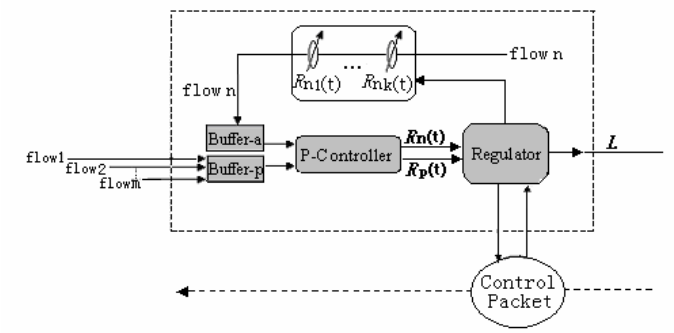

Figure 2. The operating process of the DBFA algorithm

value, we adjust (reduce) the sending rates of the transit traffic through the explicit rate feedback to the source nodes, and reallocate more bandwidth on link L to the transit traffic streams simultaneously. This way, we ensure that the transit traffic streams that pass through this node without loss as well as without starving the node itself. We adopt the proportional controller to control the rate of the transit/transmit traffic according to their buffer occupancies, which can be described as follows

$$R(t) = \mu - k(Q(t-\tau_b) - Q_0) \tag{1}$$

The above formulation is used to adjust the sending rate for one flow crossing a certain link according to the maximal bandwidth *u* allocated to it and its buffer occupancy *Q*(*t*) on the link, τ_b is the backward delay from the link to the source of the flow, *k* is the control parameter and *R*(*t*) is the allowed sending rate for this flow.

The complete pseudocode of the DBFA algorithm is provided in Figure 3. We first design the following data structure for each IA flow on the ring

Structure IA_flow

```
{    int     number;   // the number of single flows within an IA flow
     float   Rate;     // the total rate for the IA flow
     float   rate;     // the average rate for the single flow
}
```

Where *IA_flow* defines the type for an IA flow crossing some link on the RPR ring, both $flow_i$ and c_flow_i in the algorithm belong to this type and denote the IA flow originating from node *i*. Specifically, the entity c_flow_i is used as the FCF sub-field for node *i* in the control packet. As a policy, we treat the single flows within an IA flow equally so that each flow is assigned the same weight. The variables in the DBFA algorithm are specified as follows: $Q_p(t)$ is the occupancy of Buffer-p for the *m* transit flows passing through node *n* and $Q_n(t)$ is the occupancy of Buffer-a for the transmit flow generated at node *n*; Q_p and Q_n are the relevant target values for these two buffers. *U* denotes the capacity of the outgoing link of node *n*, $U_a(t)$ and $U_p(t)$ are the available bandwidth for the transmit flow and the transit flows, respectively; *f* is the decreasing factor for $U_a(t)$. Rn_1, Rn_2, …, Rn_k are the rates of the individual flows originating from node *n* while terminating at different nodes.

Algorithm DBFA operates in each RPR node in a distributed manner and the running mechanism in each node is the same as in node *n*. The proportional controller is adopted to adjust the sending rates for the relevant flows. Moreover, the global control packet containing the FCF for all the nodes propagates around the ring and accesses each node at stated periods. Upon receiving the control packet, node *n* updates its FCF sub-field in the control packet for the other nodes that have flows passing through it, and also adjusts its own sending rate according to its own FCF sub-field.

Under DBFA, bandwidth is firstly allocated among the transit and transmit flows according to their own weights and the link's capacity, but this allocation is not fixed and should be changed in terms of the congestion state or occupancy in the transit buffer, Buffer-p, when $Q_p(t)$ exceeds its target, we should reduce the bandwidth alloc-

ALGORITHM DBFA

```
Begin
for each flow_i                          // Initialize the FCF in the control packet;
   c_flow_i.Rate ← ∞;
   c_flow_i.number ← 0;
end for
   U_n ← U* w_n /( w_n + Σw_i);          // pre-allocate bandwidth for node n
   U_a(t) ←U_n;                          // assign the available bandwidth for node n
if (Q_p (t)>Q_p)                         // if the transit buffer becomes congested
   U_a(t) ←U_n – f * (Q_p (t) – Q_p ));  // re-allocate bandwidth for node n
end if
   U_p(t) ←U – U_a(t) ;                  // allocate the available bandwidth for transit traffic
   R_n (t) ←U_a(t) – k_n * (Q_n (t) – Q_n ) ;   // calculate the allowed rate for node n
   flow_n .Rate ←R_n (t);
   R_p (t) ←U_p(t) – k_p * (Q_p(t) – Q_p);     // calculate the allowed rate for transit traffic
for each flow_i
   flow_i.Rate ←R_p (t) * w_i / Σw_j;          // assign the share for the transit flow_i
   flow_i.rate ← flow_i.Rate / flow_i.number;
```

```
end for
When the control packet reaching node n      // for further adjusting
for each flow_i                              // update the FCF sub-filed    for flow_i
      c_flow_i.number ←max(flow_i.number, c_flow_i.number);
      c_flow_i.rate ←min( flow_i.rate, c_flow_i.rate);
end for
for node n                          //adjust node n's sending rate(per destination)
if (flow_n .Rate<c_flow_n .Rate)
   Rn_1, Rn_2, …Rn_k ← flow_n .rate;
                                    //get the sending rate for each single flows within flow_n
else
   if (flow_n .number>c_flow_n .number)
      Rn_2 Rn_3 …Rn_k ←c_flow_n .rate;
                                    //get sending rate for the (k-1) downstream single flows
      Rn_1 ←U - c_flow_n .Rate - ∑c_flow_i.Rate;
                                    //get sending rate for the up single flow
   else
      Rn_1, Rn_2, …Rn_k ←c_flow_n .rate;
   end if
end if
End
```

Figure 3. The complete pseudocode of DBFA algorithm

ation for the transmit flow by the deflation factor f in order to increase the bandwidth allocated to the transit traffic streams, this is specified by the determination of the two parameters $U_a(t)$ and $U_p(t)$. On the other hand, we should adjust the sending rate of the transiting traffic streams using the explicit rate feedback to their source nodes. Under this scheme, we can ensure that the data loss of the transit traffic streams passing through the node can be prevented and in the meantime the node itself will not starve.

As we can see from the DBFA Algorithm, $flow_i$ is the local entity at node n to perform the bandwidth allocation for node i while c_flow_i is the global entity to achieve the final fair share for node i. Accordingly, the rate determined by proportional controller in an RPR node is only suitable for the local area, so to reach the global coordination, node n should further adjust its sending rate according to its own FCF sub-field in the control packet when it arrives at this node; also the FCF sub-fields in the control packet for the other relevant nodes should be updated by this node. As the control packet rotates around the RPR ring, all the nodes can drag its sending rate to the desired fair value, as a result we can achieve the global fairness for all flows and realize the high utilization on the links and the maximal spatial reuse over the entire RPR network.

4. SIMULATION RESULTS

In this section, we construct two simulation models to test the performance of our DBFA algorithm. These two models are as shown in Figure 4 and Figure 5. The model of Figure 4

assigns the same weight for each node, while the model of Figure 5 assigns different weights for each node. Our purpose is to test whether the DBFA algorithm can achieve global fairness, high efficiency and maximal spatial reuse.

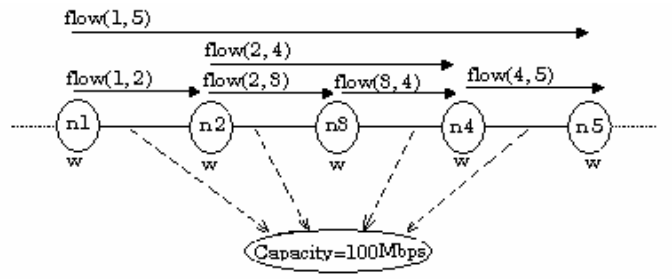

Figure 4. The configuration for all RPR nodes with the same weight

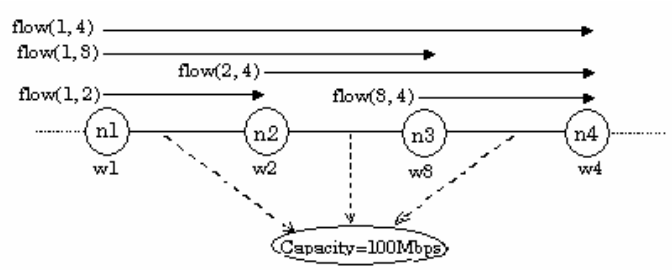

Figure 5. The configuration for each node with different weight

The capacity on each link is set to 100 *Mbps*, we denote the link between node n *i* and node n*j* as the link (*i*, *j*) and the data flow originating from node n*i* and ending at node n*j* as flow (*i*, *j*), the target value Q_p for transit buffer (Buffer-p) of each RPR node is set to be 20 packets, while the target value Q_n for the transmit buffer (Buffer-a) is set to be 30 packets. The delay on each link is 2ms and the total simulation time for each model is set to 200 ms. Suppose all the nodes are greedy, meaning that they always have data to send.

For the simulation model in Figure 4, there are six individual flows, namely flow (1, 5), flow (1, 2), flow (2, 4), flow (2, 3), flow (3, 4), flow (4, 5), and their initial rates are set to be 45 Mbps, 55 Mbps, 25 Mbps, 35 Mbps, 20 Mbps, 40 Mbps, respectively. The sending rate for each flow and the utilization on each link are shown in Figure 6 and Figure 7.

For the simulation model in Figure 5, there are five single flows, namely flow (1, 2), flow (1, 3), flow (1, 4), flow (2, 4) and flow (3, 4), and their initial rates are set to be 40 Mbps, 20 Mbps, 35 Mbps, 30 Mbps and 25 Mbps, respectively. The weights for node *n1*, node *n2* and node *n3* are set to be 30, 20 and 10, respectively. The sending rate for each flow and the utilization on each link are shown in Figure 8 and Figure 9.

As we can see from Figures 6 and 7, the three competing flows: flow (1, 5), flow (2, 4), flow (3, 4) get an equal share of the bandwidth on link (3,4), other flows such as flow (1, 2), flow (2, 3) and flow (4, 5) increase their rates to the maximums so that the bandwidth on link (1, 2), link (2, 3) and link (4, 5) is fully used. As a result, the utilization on every link reaches 100%. Therefore, the performance objectives of fairness, spatial reuse and high utilization for this model are simultaneously realized. In Figures 8 and 9, the three competing flows: flow (1,

4), flow (2, 4) and flow (3, 4) on link (3, 4) obtain their fair shares of 50 Mbps, 34 Mbps and 18 Mbps, respectively. While flow (1, 3) and flow (1, 2) obtain their fair shares of 16 Mbps and 33 Mbps, respectively. All these rate allocations lead to the full utilization of the bandwidth on each link. We have demonstrated that all RPR requirements are achieved: fairness, high utilization as well as maximal spatial reuse.

5. CONCLUSIONS

We have proposed a bandwidth allocation algorithm for RPR networks termed DBFA, which is able to achieve the key performance objectives of RPR networks, i.e., fairness, high utilization and spatial reuse. The algorithm operates at each RPR node in a distributed manner, and the sending rates of various flows passing through a node are controlled by using the well-established proportional control method which is known to provide stability in many cases. We use a rotating global control packet to collect and deliver the fairness control message for every node, in order to achieve the global coordination, fairness and efficiency over the entire RPR ring. Our detailed analysis and simulation results provide evidence for the satisfactory performance of the proposed scheme.

REFERENCES

[1] RPR Alliance, "An Introduction to Resilient Packet Ring Technology," October 2001. http://www.rpralliance.org/.

[2] C. N. Hawkins, J. Green, M. Sharma, and K. Vasani, "Resilient Packet Rings for Metro Networks," August 2001. http://www.rpralliance.org/.

[3] F. Davik, M. Yilmaz, S. Gjessing and N. Uzun, "IEEE 802.17 resilient packet ring tutorial," *IEEE Communications Magazine*, vol. 42, no. 3, March 2004, pp. 112-118.

[4] Andrew S. Tanenbaum, *Computer Networks* (Third Edition). Prentice Hall, Inc.1996.

[5] I. Cidon and Y. Ofek, "MetaRing – A Full-Duplex ring with Fairness and Spatial Reuse," *IEEE/ACM Transaction on Communication*, vol. 41, no. 1, January 1993, pp. 110 -120.

[6] D. Tsiang and G. Suwala, "The Cisco SRP MAC Layer Protocol," Internet RFC 2892, August 2000.

[7] V. Gambiroza, Y. Liu, P. Yuan, and E. Knightly, "High Performance Fair Bandwidth Allocation for Resilient Packet Rings," in *Proceedings of the 15th ITC Specialist Seminar on Traffic Engineering and Traffic Management* ,Wurzburg, Germany, July 22-24 2002.

[8] V. Gambiroza, P. Yuan, L. Balzano, Y. Liu, S. Sheafor, and E. Knightly, "Design, analysis, and implementation of DVSR: a fair high-performance protocol for packet rings," *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, January 2004, pp. 85-102.

[9] A. Mekkittikul, "Alladin Proposal for IEEE Standard 802.17," Draft 1.0, November2001. http://grouper.ieee.org/groups /802/17/documents/drafts/zz_draft_04.zip.

[10] J. Kao et al, "Gandalf Proposal for IEEE Standard 802.17," Draft 0.4, Noverrnber2001. http://grouper.ieee.org/groups /802/17/documents/drafts/gandalf_04Plus.pdf.

[11] J. Kao et al, "Darwin Proposal for IEEE Standard 802.17," Draft 1.0, Jan.2002.http://www.ieee802.org/17/documents /presentations/jan2002/ib_fairdraft_02.pdf.

[12] E. Knightly, L. Balzano, V. Gambiroza, Y. Liu, S. Shaefor, P. Yuan, and H. Zhang, "Achieving High Performance with Darwin's Fairness Algorithm," IEEE 802.17 Resilient Packet Ring Standard Meeting, St. Louis, MO, March 2002.

[13] Liansheng Tan, Hanwu Wang and Moshe Zukerman, "Adapative Bandwidth Allocation for Metropolitan and Wide-Area Networks," *IEEE Communications Letters*, vol. 9, no. 6, June 2005.

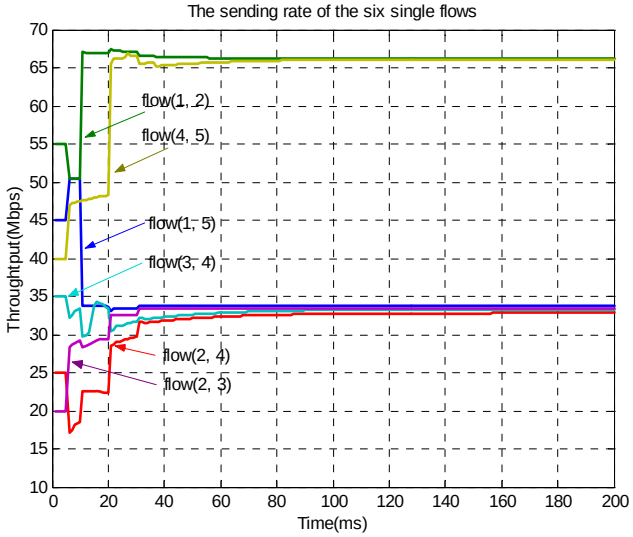

Figure 6. The rate allocation for the six flows

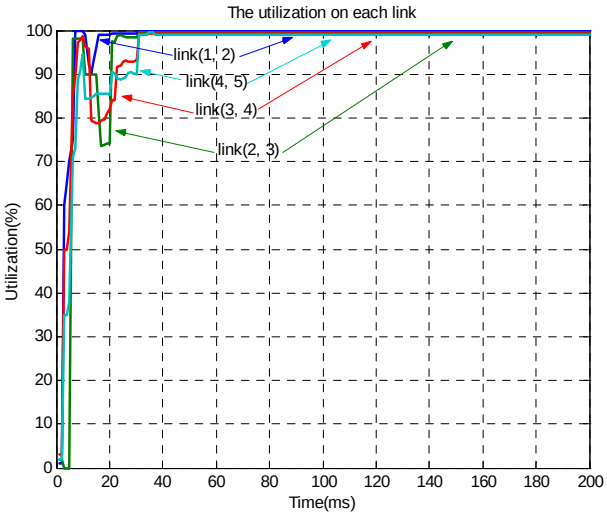

Figure 7. The utilization on the four links

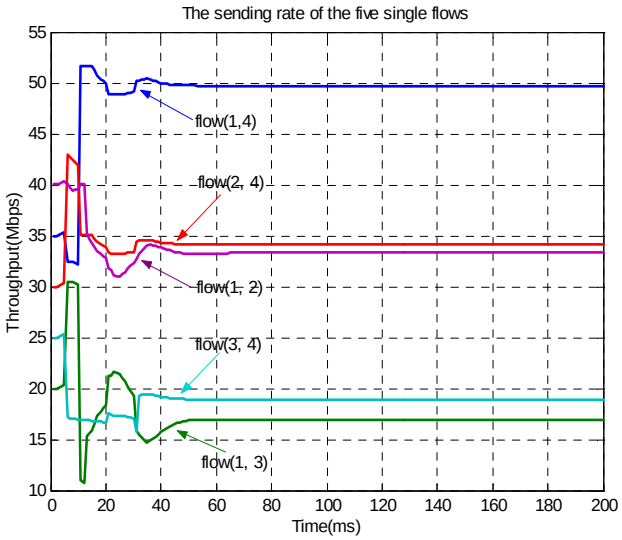

Figure 8. The rate allocation for the five flows

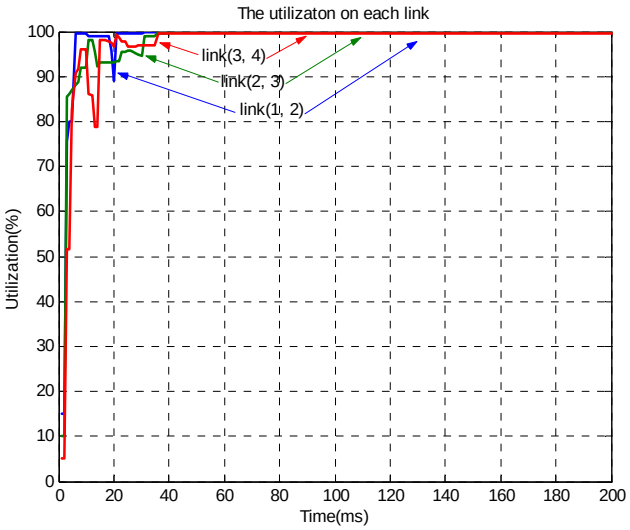

Figure 9. The utilization on the three links