IKN-MAC proposal for IEEE 802.17 RPR
Cyclic-Reservation Access

Draft version 1
Orlando, January 2002

Harmen R. van As, Guenter Remsak, Jon Schuringa
Vienna University of Technology, Austria
Email: harmen.r.van-as@tuwien.ac.at
guenter.remsak@tuwien.ac.at
jon.schuringa@tuwien.ac.at

This document describes the mechanisms of a Medium Access Control Protocol intended for inclusion in the IEEE 802.17 Resilient Packet Ring.

This protocol allows combined greedy and cyclic reservation access performing at the theoretical fair limits and therefore exhibits excellent performance in terms of throughput, end-to-end delay, guarantees of service level agreement, and traffic dynamics.

Other major features are
- the support of multiple service classes,
- the support of heterogeneous link rates,
- no measurements on the links,
- no buffer thresholds
- self-adaptive.

Introduction

Fairness control mechanisms for rings can be classified in global and link fairness mechanisms. Traditional medium access control protocols are based on global fairness, where each station obtains the same throughput, independently whether a node disturbs flows of other nodes or not. Today, advances in microelectronics allow the design of more sophisticated link or bottleneck fairness mechanisms, potentially resulting in much high network throughputs.

Definition of Global fairness: Fairness based on a mechanism that allows nodes to share the same amount of the transmission capacity of the ring, independently whether their traffic interfere or not.
Definition of Link fairness: Fairness based on a mechanism that coordinates ring access of only those nodes that interact during their packet transfer. Thus, all nodes that do not interfere are not throttled in their performance.

In Figure 1, it can be seen that in the case of global fairness the flow from station 5 to station 6 is throttled down to a rate of 0.5 because of the bottleneck on the link between stations 1 an 2. In case of link fairness, this unnecessary throttling does not take place.

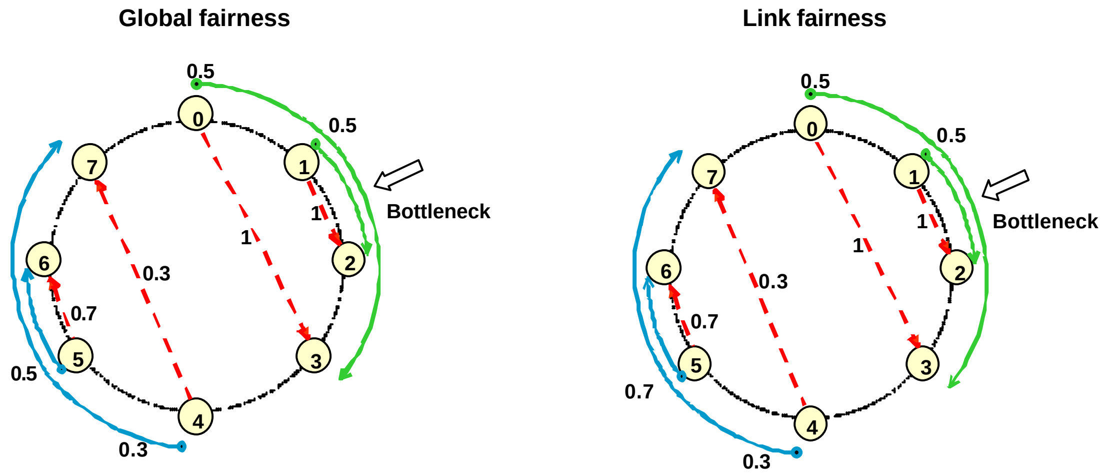
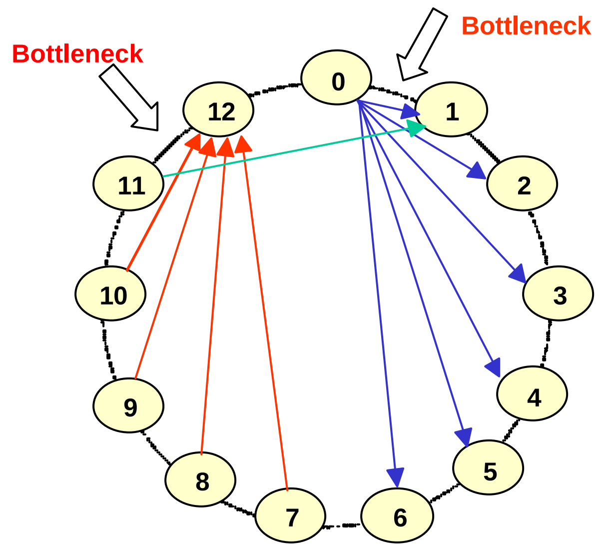


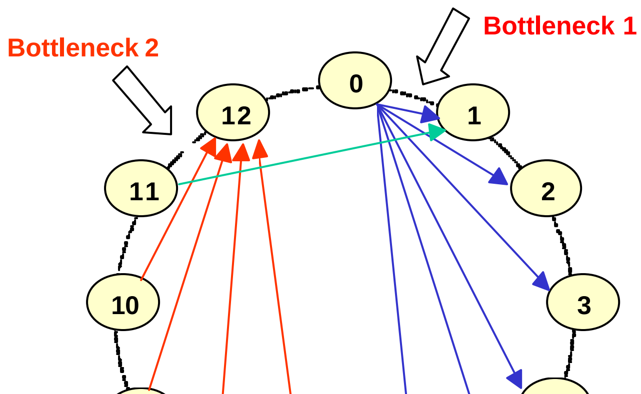
Figure 1: Global and link fairness on a single ringlet



Fairness Definition 1

Source	Sink	Rate	Fair
0	1	0.01	0.01
0	2	0.01	0.01
0	3	0.01	0.01
0	4	0.01	0.01
0	5	0.01	0.01
0	6	0.01	0.01
7	12	0.1	0.071
8	12	0.1	0.071
9	12	0.1	0.071
10	12	0.1	0.071
11	1	1	0.71

Figure 2: Fair station throughputs in case of fairness definition 1 (proportional throttling)



Fairness Definition 2

Source	Sink	Rate	Fair
0	1	0.01	0.01
0	2	0.01	0.01
0	3	0.01	0.01
0	4	0.01	0.01
0	5	0.01	0.01
0	6	0.01	0.01
7	12	0.1	0.1

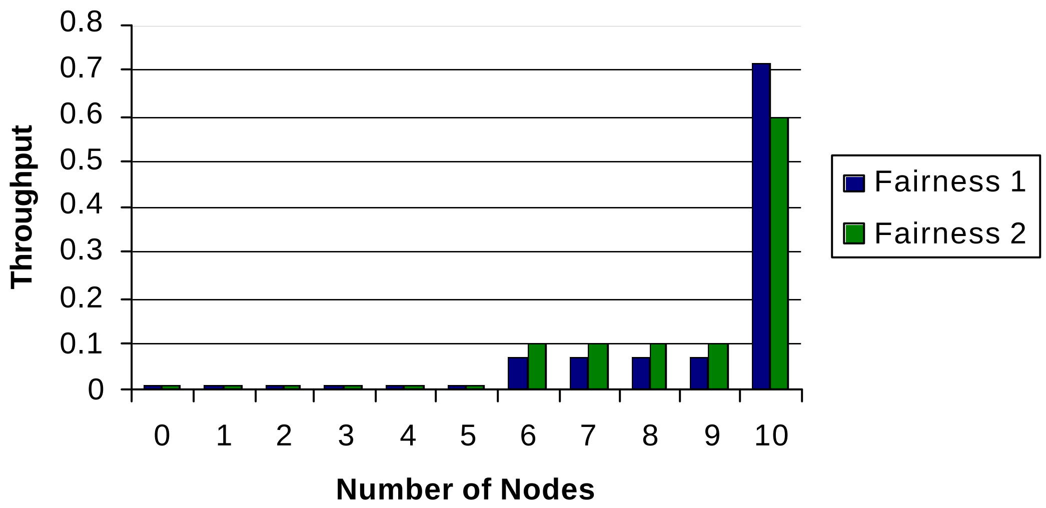


Figure 4: Fair station throughputs for each of the definitions

Furthermore, one distinguishes between reactive and proactive control mechanisms. In reactive control, a node detecting a congestion on its outgoing link typically sends a backpressure control packet in the opposite direction to its upstream nodes enforcing them to stop transmission or enforcing to reduce their rate. In proactive fairness control, a control packet circulates around the ringlet to coordinate the individual source-destination flows of each node. For the content of the control packet several variations are possible. One possibility is that each station i (i = 1,…, N) measures the number of bytes of each flow f_{ij} from source i to destination j on its outgoing link i (i = 1, …, N) during the cycle time T_c of the control packet. When the control packet arrives at station i, it calculates the fair rate r_i over its outgoing link i and writes the result into the data field of link i in the control packet. Since each station does this measurement, all stations are cyclically updated with all the current fair link rates r_i on the ring. For a dual ring, there is on control packet on each ringlet. Control packets can either circulate in the same direction of the data flow or in the opposite direction. In the latter case, one ringlet is used for the data flow and the other ringlet for its control.

In this proposal, however, control and data packets flow in the same direction. This has the advantage, that in case of multiple parallel ringlets, there is a clear and simple association between data and control packets belonging to a ringlet. In addition, we use no measurement data but in stead the current traffic load waiting in each station to be transmitted. Due to this, the proactive control is based on the latest

Station architecture

Structure and architecture is as being proposed in several submitted drafts. To allow variable packets to access the transmission medium the buffer insertion technique is used. These transmit buffers are however not used for scheduling purposes. The rule is clear, the transmit buffer must be emptied, before a packet can be send on the medium. We consider the shared medium as a flexible end-to-end transmission pipe, where once a packet is on the medium it is forward with minimum addition delay.

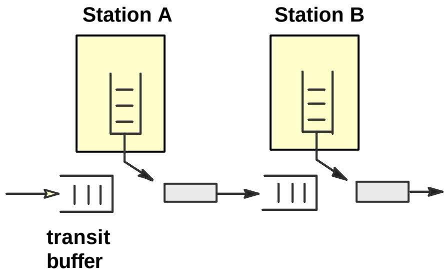
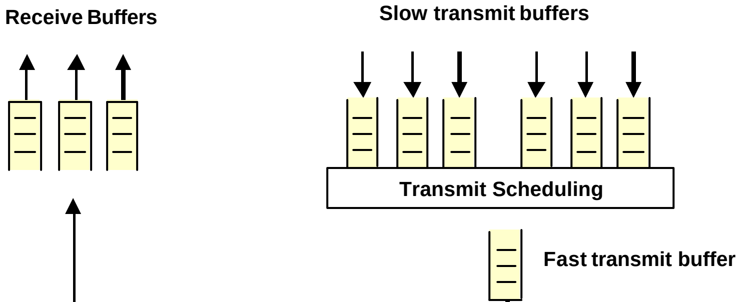


Figure 5: Insertion buffer technique for packet collision avoidance

We prefer to use the cut-through mode, thus permitting the shortest MAC end-to-end delays. We like to allow also store-and-forward. Furthermore, we prefer to allow for the possibility of having different traffic classes on the ring, for instance three classes, each with its own parallel transit buffer. In order to prevent head-of-the-line blocking, we propose to provide a separate queue for each destination. The general node structure is given in Figure 6.



Cyclic Reservation Fairness Control-

Cyclic reservation control is based on the simple concept that a control packet circulates around a ringlet and advertises the amount of traffic that is waiting in each station. Data and its control flow in the same direction. The cyclic time is kept constant in that each station holds the control packet if required. For this, each station has a Cycle Timer, which is set as the control packet is sent out. The control packet contains an entry for each source-destination flow.

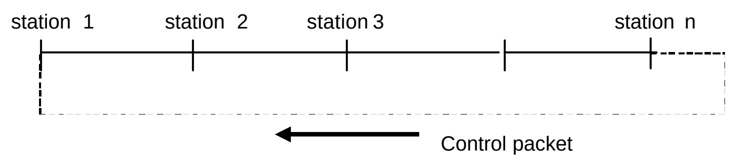


Figure 7: Fairness cycle (data and control on same ringlet)

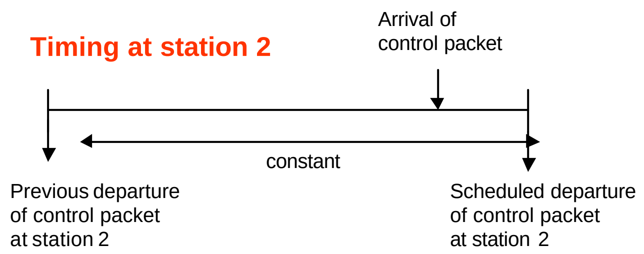


Figure 8: Constant fairness cycle

Traffic classes

We consider three types of source-destination flows:

- Provisioned constant bit rate (CBR) traffic,
- High priority traffic, characterised by a guaranteed bit rate and a variable additional bit rate,
- Low priority traffic, characterised by a variable bit rate.

We assume that all stations know

- the bit rates of all transmission links (heterogeneous links)
- the constant bandwidth of all source-destination flows (CBR)

Medium access

We distinguish between:

- Greedy access for flows passing underutilized links,
- Reserved access for flows passing bottleneck links.

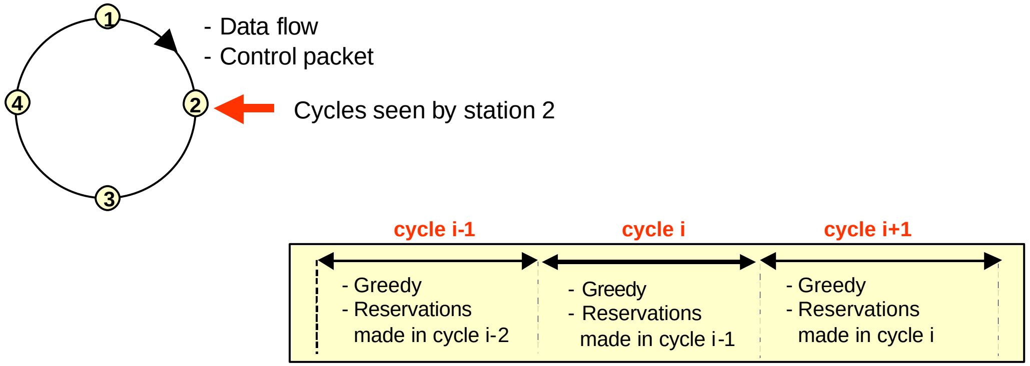
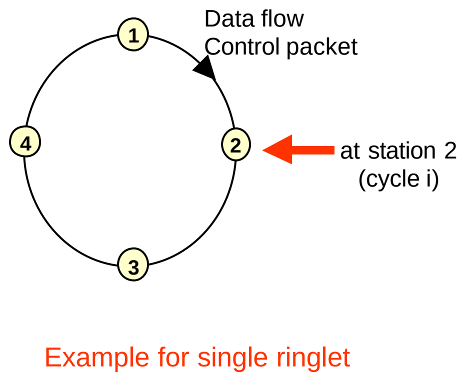


Figure 8: Greedy and reservation access

As shown in Figure 8, each station sees its own cycle starting with the arrival of the control packet carrying all traffic demand of all stations. The node first empties its transmit buffers, transmit then packets from the demand reservations of the previous cycle, and finally transmits in greedy mode over underutilized links.

For the further discussion we use the following notations:

- F_{ij} : constant bit rate traffic demand from source i to destination j,
- L_{ij}: low-priority traffic demand from source i to destination j,
- H_{ij}: high-priority traffic demand from source i to destination j, consisting of G_{ij} and V_{ij} ,
- G_{ij}: guaranteed part of the high-priority traffic demand,
- V_{ij}: variable additional part of the high-priority traffic demand,
- B_i : Occupancy of the transit buffers in station i.



Flow	High	Low	
1 -> 2	H12	L12	cycle i
1 -> 3	H13	L13	
1 -> 4	H14	L14	
2 -> 3	H23	L23	cycle i -1
2 -> 4	H24	L24	
2 -> 1	H21	L21	
3 -> 4	H34	L34	
3 -> 1	H31	L31	
3 -> 2	H32	L32	
4 -> 1	H41	L41	
4 -> 2	H42	L42	
4 -> 3	H43	L43	

Figure 9: Example of the control packet content for a single ring

Actions performed by each station

Upon arrival of the control packet, the following actions are performed:

1) Extract the relevant information from the control packet.
2) Scheduling of the fair rates for the high and low traffic classes.
3) Write the new demand into the control packet.
4) Transmit the control packet to next station at the scheduled time.
5) Transmit the reserved traffic according to calculated fair flow rates.
6) Transmit greedy traffic up to fair flow rates.

Information extraction

As is shown in Figure 10 for station 2, the information required are entries both from the previous and the current control packet.

Cycle i-1

Flow	High	Low
1 -> 2	H12	L12
1 -> 3	H13	L13
1 -> 4	H14	L14
2 -> 3	H23	L23
2 -> 4	H24	L24
2 -> 1	H21	L21
3 -> 4	H34	L34
3 -> 1	H31	L31
3 -> 2	H32	L32
4 -> 1	H41	L41
4 -> 2	H42	L42
4 -> 3	H43	L43

Old table in staton 2

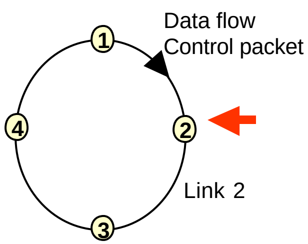
Cycle i

Flow	High	Low
1 -> 2	H12	L12
1 -> 3	H13	L13
1 -> 4	H14	L14
2 -> 3	H23	L23
2 -> 4	H24	L24
2 -> 1	H21	L21
3 -> 4	H34	L34
3 -> 1	H31	L31
3 -> 2	H32	L32
4 -> 1	H41	L41
4 -> 2	H42	L42
4 -> 3	H43	L43

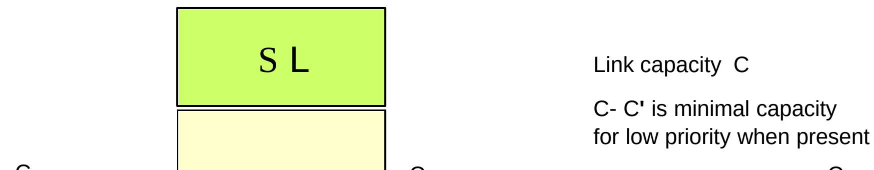
New table in station 2

Figure 10: Control packet information that is used by station 2 for fair access scheduling

Scheduling of the fair rates



S L : all low-traffic flows
S V : all non-guaranteed high-traffic flows
S G : all guaranteed high-traffic flows
S F : all CBR traffic flows
$V_i = H_i - G_i$: variable part of high-priority traffic flow



New demand

Transmission of control packet

Transmission of reserved traffic

Transmission of greedy traffic

Appendix

Given:

- Number of nodes N
- Requested rate from node i to node j $r_{i,j}$

Calculated:

- Flow on link I f_i
  Sum of all requested rates passing link I
- Number of demands passing link i nd_i
- Remaining capacity on link i rc_i
  Link capacity minus the sum of all allowed rates passing link i
- Allowed rate from node i to node j $ar_{i,j}$
  Rate calculated by the algorithms

Fairness Definition 1

Flow rates on bottleneck are proportionally reduced by the total amount of offered traffic for that bottleneck link

Algorithm

Set: $rc_i=1$;

Step 1: for all links: calculate flow on link i: f_i **Step 2:** if (rc_i/f_i <1) // condition for a bottleneck

take always the highest overloaded bottleneck: min(rc_i/f_i)

bottleneck link: indicated by index b

else $ar_{i,j}= ar_{i,j}+ r_{i,j}$; stop;

Step 3: for all flows passing this bottleneck set: $ar_{i,j}=rc_b/f_b$´$r_{i,j}$ and $r_{i,j}=0$

Step 4: calculate remaining capacities rc_i of all links; goto **Step 1**;

Fairness Definition 2

Flow rates on bottleneck are proportionally reduced by the total number of connections on bottleneck link

Algorithm

Set: $rc_i=1$;

Step 1: for all links: calculate flow on link i: f_i **Step 2:** if (rc_i/f_i <1) // condition for a bottleneck

take always the highest overloaded bottleneck: min(rc_i/nd_i)

bottleneck link: indicated by index b

else $ar_{i,j}= ar_{i,j}+ r_{i,j}$; stop;

Step 3: for all flows passing this bottleneck:

if ($rc_b/nd_b> r_{i,j}$)

$ar_{i,j}= r_{i,j}$; $nd_b= nd_b-1$; $r_{i,j}=0$;

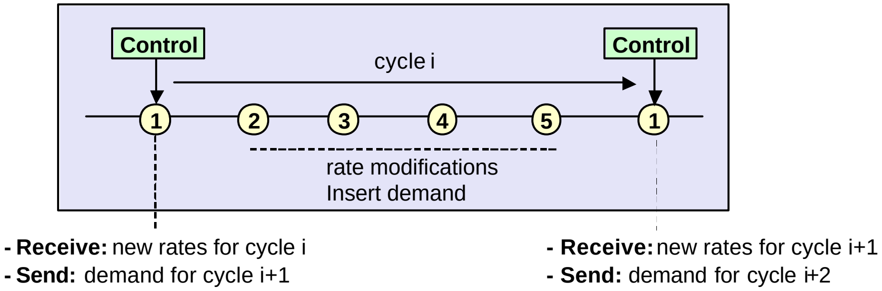


Figure 12: IKNv1: Each intermediate station insert its demand and modifies the demands of other stions in case of a bottleneck

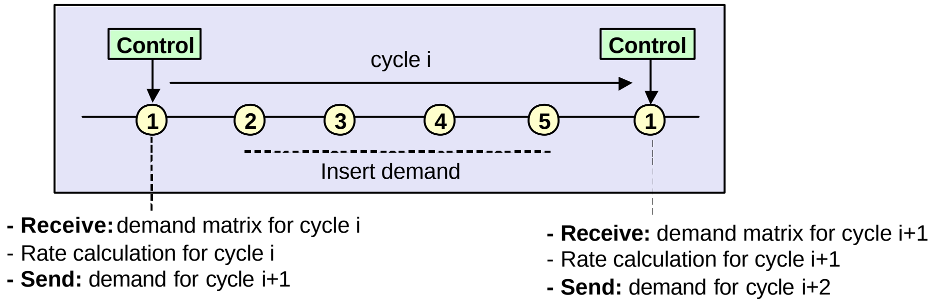
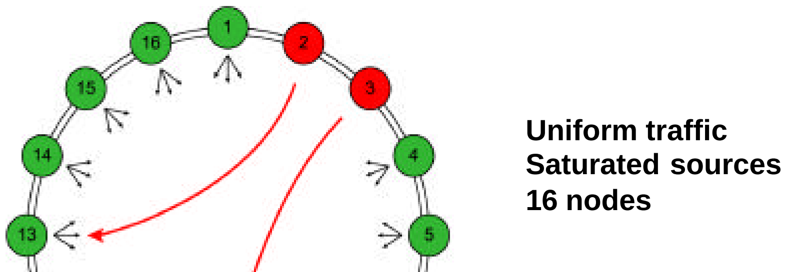


Figure 13: IKNv2: Each intermediate station just insert its demand



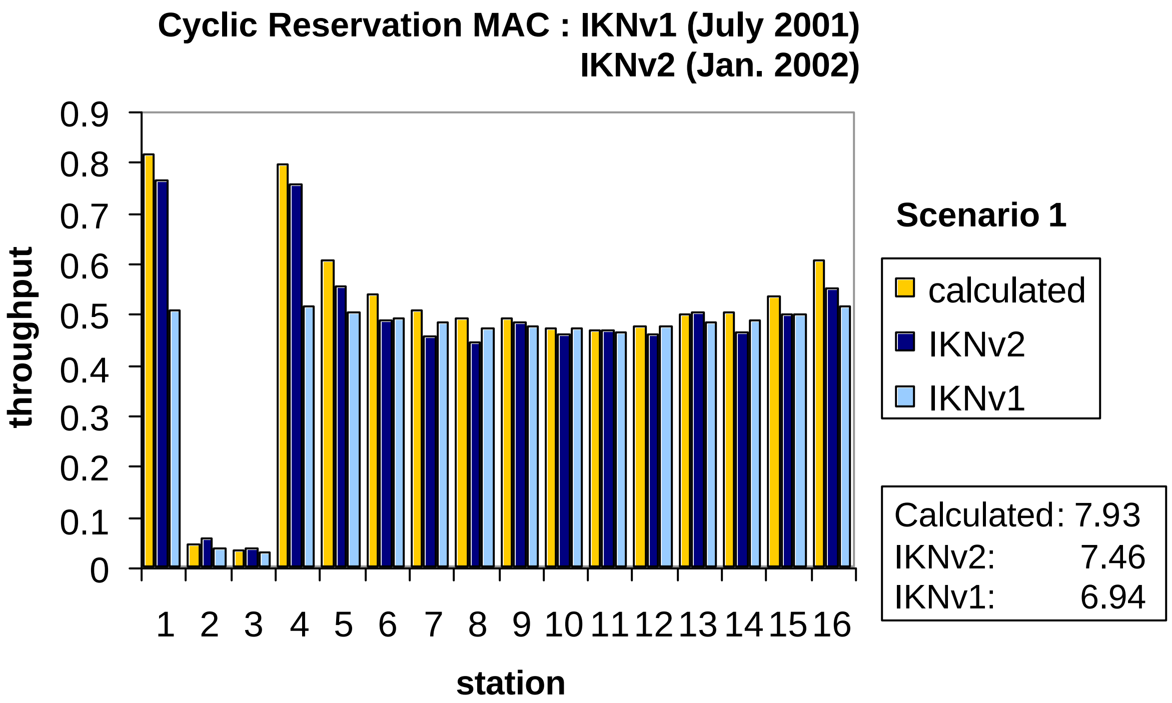


Figure 15: Throughput comparison