Proceedings of the 5th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications, Madrid, Spain, February 15-17, 2006 (pp108-113)

Adaptive Rate Control (ARC) with Erlangian Traffics over Resilient Packet Ring (RPR) Network

SURASEE PRAHMKAEW AND CHANINTORN JITTAWIRIYANUKOON

Department of Telecommunications Science
Faculty of Science and Technology
Assumption University, Bangkok 10240, Thailand

Abstract: - In this paper an Adaptive Rate Control (ARC) with Erlangian traffics in Resilient Packet Ring (RPR) network has been proposed. First, the RPR network will be introduced then the necessity of Erlangian distribution for traffic will be discussed. The traffic flow prior to the entrance of RPR network will be controlled by the window size adjustment in the ARC algorithm. The algorithm per se introduces the basic regulation of appropriate window upsize or downsize. Input traffic rate, packets drop, preset size of the window and RTT are taken into account by the ARC algorithm. Simulations are employed to investigate how input Erlangian traffic rate or other parameters mentioned above affect the performance. Results before and after applying ARC are compared. In addition simulations show how the performance can be improved by the ARC algorithm. By varying Erlang means, the results indicate the improvement.

Keywords: - Erlangian input traffic, ARC, window flow control, RPR network, simulation.

1. Introduction

A simulation will provide solutions to any sophisticate models if and only if all input parameters and modeling technique are precise. This will give rise to robust qualitative behavior of the analytical model representing those complex systems. In other words, simulation can create exact solutions but, however, takes more time and requires prudence. The paper is organized as follows. We begin by giving an overview of the practical arrival process, which is the Erlangian traffics in Resilient Packet Ring (RPR) network in section II. The Adaptive Rate Control (ARC) algorithm will be presented in section III. Simulation will be briefly discussed in section IV. Section V will demonstrate results and analysis and finally conclusion and future works will be summarized in section VI.

2. The Erlangian Traffics in RPR network

The concept of high speed dual ring networks has been introduced since the 1990s. Some of the proposed dual ring networks include Metaring [9], CRMA-II [7], ATMR [5], and FULL [6]. Practical dual ring network products were later developed by CISCO using the Spatial Reuse Protocol (SRP) technology [1] and by Nortel using the OPTera Packet Edge technology[12]. These two products were competing for the adoption of their own proprietary implementation of dual ring networks. However, eventually both companies realized that is was mutually more beneficial to users to have only one common accepted standard instead and so the IEEE 802.17 Working Group [2] was formed to develop a common dual ring network standard. It can concluded that the mentioned 2 proprietary standards (from CISCO and Nortel) much reflect the RPR standard. Currently this standard [2] has been approved by the IEEE since 2004, also it is expected that there will be an amendment in the adoption of future specification of RPR networks. Figure 1 shows the RPR node architecture. Note that the Ingress queues represent the entrance of Erlangian traffics to the RPR network while the Transit queues will keep all packets transiting the RPR node as the fluctuation of Ingress traffic has arose. In the paper we allow Ingress queue to transmit an Ingress packet first although another packet arrives from the Transit channel. As the RPR node finishes transmitting the Ingress packet, the Transit packet will be stored in the Transit queue.

Proceedings of the 5th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications, Madrid, Spain, February 15-17, 2006 (pp108-113)

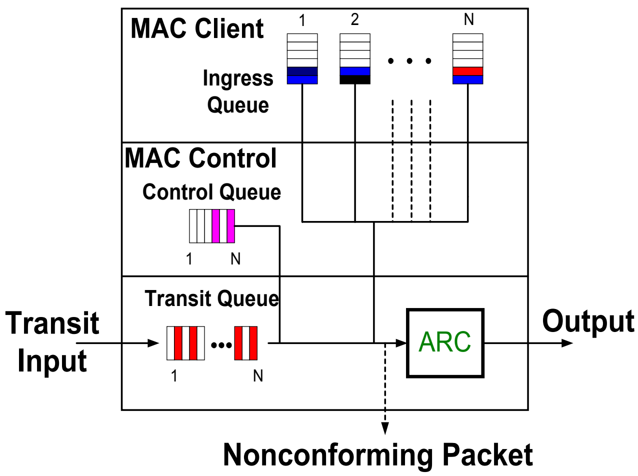


Figure 1. RPR node architecture.

3. Adaptive Rate Control (ARC) Algorithm

In the section ARC will be discussed. The algorithm will not consume much time to calculate for the adjustment. Although an application for a fast routing network is concerned, it is guaranteed that ARC would not cause any burden to the IP network as such [8]. Key idea of window size adjustment (upsize or downsize) is to find a criterion, which is well-fit to the expected value of system output. Another method called sliding window as part of the flow control algorithm can be found in [4]. Fundamental input parameters to the system are then opted for window size computation. These parameters can be number of packet drop, estimated RTT (Round Trip Time) delay, window size in the previous interval of time and the basic arrival rate of input traffic.

The idea of this ARC algorithm somehow works like an ON/OFF control valve for all arrival traffics. In the beginning, traffic arrives in front of ON/OFF valve with no competitor (assuming that it is an preemptive condition, that is, no initial packet drop found or no initial packets found at time t=0) then the valve is ON allowing this traffic goes through by presetting the flow rate of the valve to be moderate. The flow rate of the valve remains unchanged until whenever the packet drop hits the unacceptable records. This may be caused by a fluctuation of input traffic. If the fluctuation of the traffic reaches then all dropped packets will be held and kept in a storing room until retransmission occurs. The packet drop soon will be taken into account for the flow rate re-calculation by the ARC algorithm. The result may simply improve the transmission by adjusting the ON valve to higher rate of flow. The action takes step by step regarding to the impact of result from the calculation. The adjustment takes no longer action if it is highest rate possible. That means it is now reaching the maximum size of the window flow control (max_winsize). Although max_winsize is set, somehow the fluctuation of traffic may continue, assuming that packet drops now are beyond the capacity of the storing room then the counting for packet drop (non-conforming frame) starts from this point of time onwards. As the fluctuation of input traffic may change to fewer, the ON valve will release the contention by adjusting to new lower rate of the flow. It is apparent the window size (ON/OFF control valve) will shrink (downsize) or expand (upsize) back and forth several times during the functioning of the ARC algorithm. The window size is ranging between min_winsize up to max_winsize. While min_winsize in our ARC algorthm is set to one, max_winsize will reflect the channel capacity (bandwidth) of RPR network per se. By well-adjusting the window size in ARC algorithm, it will be able to reduce packet drop compared vis-à-vis non-ARC application. That will be directly beneficial to the quality of service (QoS). For example, if the arrival Erlang traffic rate (λ_a) as an input rate is lower than the packet drop rate (λ_p) then ARC will set the window size to be min_size (=1) initially. The application of ARC in the system is illustrated in figure 2 where the algorithm is shown in figure 3.

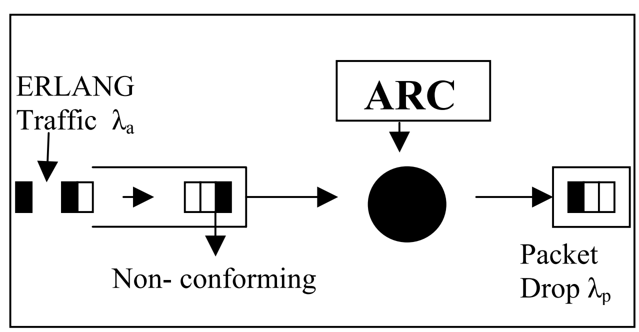


Figure 2. Flow control with ARC function.

```
/*********** ARC Algorithm ************/

PROCEDURE
/****** Window Size Calculation ******/

 Current allocation rate (Ai) ;
 Current win_size (Wi) ;
 Bandwidth (BW) ;
DO WHILE Transmission is Ongoing ;
 {
    IF  Packet Drop λp ≥ Ai  THEN  {
    Calculate new allocation rate (An) ;
    Calculate new win_size (Wn) ;
       Ai <= An ;
       Wi <= Wn ;   }
    ELSE {
       Ai <= ABW  ;
       Wi <= Wn ;  }
  }  END_DO  ;
/*** Calculate new allocation rate (An) ***/
An <= BW * frame_size/( λp)^1/2 ;
/*** Calculate current win_ size (Wn) ***/
IF  Packet Drop exists  THEN
{
 Wn <= Wn ++ ;
 IF Wn  >  Win_max THEN Wn <= Win_max  ;
}
ELSE {
Wn <= Wn – –;
 IF Wn  ≤  0  THEN Wn = 1  ; }

/*** Calculate available BW (ABW) ***/
ABW <= Max_BW – Used_BW;

 /******* END OF ARC Algorithm *******/
```

Figure 3. ARC Algorithm.

4. Simulation

Simulation model in the experiment can be shown in figure 4. From this figure it is clearly seen that the input traffic (both from Ingress and Transit channels) rate will vary between 622 Mbps and 2.4 Gbps. In case of ARC, all traffics will be handled immediately by ARC algorithm before releasing them to the RPR node. In case of no ARC, traffics will proceed straightforward to the RPR node.

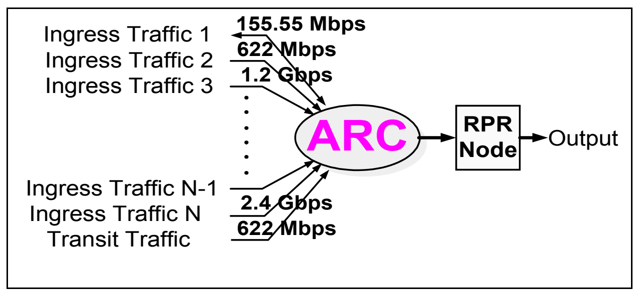


Figure 4. Simulation model.

4.1 Input Traffic

The input traffic can be categorized into data, voice, video, image and graphics [3]. This paper will focus on three categories, mainly data, voice and video. Voice sources are generally burst type in practice while data and video sources are either continuous or burst type, depending on the compression and coding techniques used. We also found that the exponential distribution was not always an appropriate candidate for representing the practical situation with regard to service times and interarrival times. It is obviously observed that to allow a more general service distribution (such as Erlangian distribution) would have destroyed the Markovian property and then It will be much more complicate than the Markovian distribution. However what we can find the solution for Erlangian distribution is to decompose both service time distribution and interarrival time distribution into a collection of structured exponential distributions.

4.2 The Queue E*n*/E*n*/1 Model

Here the system with the Erlangian distribution (n stages) arrival process and the Erlangian distribution (n stages) service time is considered. The roles of interarrival time are different from those of the previous publications [10]. The En/En/1 model operates as whenever an arrival has just occurred, then one immediately introduces a new arrival into an n-stage Erlangian facility. When the arrival is generated from source he must then pass through n exponential stages each with parameter nλ. It is clear that the probability density function (pdf) of the time spent in the arriving facility will be given by equation 1.

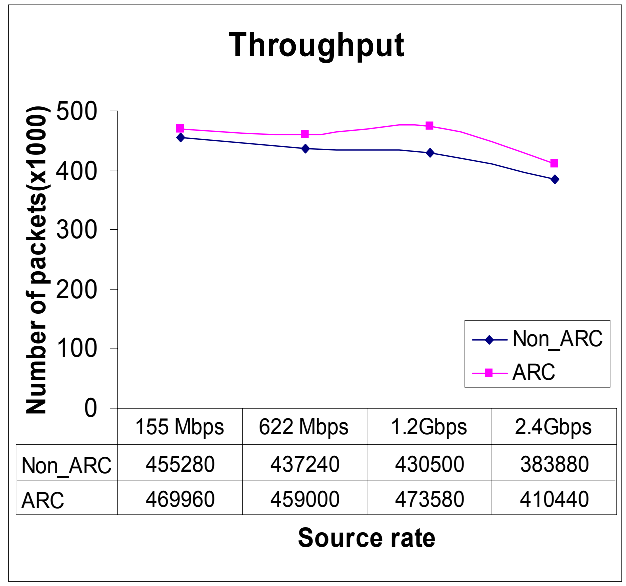
$$A(t) = \frac{n\lambda(n\lambda t)^{n-1}e^{-n\lambda t}}{(n-1)!} \qquad t\geq 0 \qquad (1)$$

It is also possible to specify the number of n stages remaining in the service facility for each job in service, it behooves us to represent each job in the queue prior to the service, as possessing n stages of service. Then service times will be given by equation 2.

$$g(x) = \frac{\mu^n x^{n-1} e^{-\mu x}}{(n-1)!} \qquad x\geq 0 \qquad (2)$$

5. Results and Analysis

In this section, we use simulations [11] to study the performance of RPR. All simulation results are obtained with our publicly available *EZSIM* implementation of RPR. We consider 622 Mbps links (OC-12), 200 KB buffer size, 1 kByte packet size, and 0.1 msec link propagation delay between each pair of RPR nodes [13]. Input Erlangian traffic rates are ranging between 622 Mbps and 2.4 Gbps (which is apparently over capacity of the optical link OC-12) in order to prove the benefit of ARC algorithm. Erlangian distribution stage 2 is employed throughout the experiments. The comparisons between before and after applying ARC algorithm are shown as follows.



	155 Mbps	622 Mbps	1.2Gbps	2.4Gbps
Non_ARC	455280	437240	430500	383880
ARC	469960	459000	473580	410440

Figure 5. Throughput.

It is apparent that ARC algorithm will help improve throughput, which reflects the performance of the system as shown in figure 5. Particularly at high rate of congestion (1.2 Gbps-2.4 Gbps), ARC will outperform compared to non-ARC case. However, the ARC will have to pay the price for higher mean queue length (MQL) as shown in figure 6. Max_MQL for ARC is about 2 packets while non-ARC can accommodate only 0.5 packets. Although it is a matter of four times higher but the quantity of two shows insignificant to buffer requirement. Also ARC implementer has to be aware of the higher utilization factor at RPR node. This is due to higher fluctuation of traffics from source managed by ARC. However, the increment of utilization factor is only about 6 % in order to gain higher throughput. The result graph of utilization factor is shown in figure 7. The interesting result is that the ARC can help

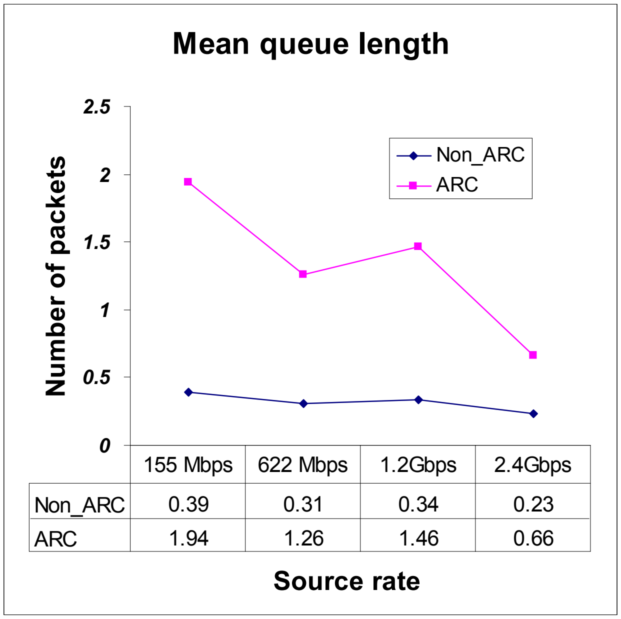


Figure 6. Mean queue length.

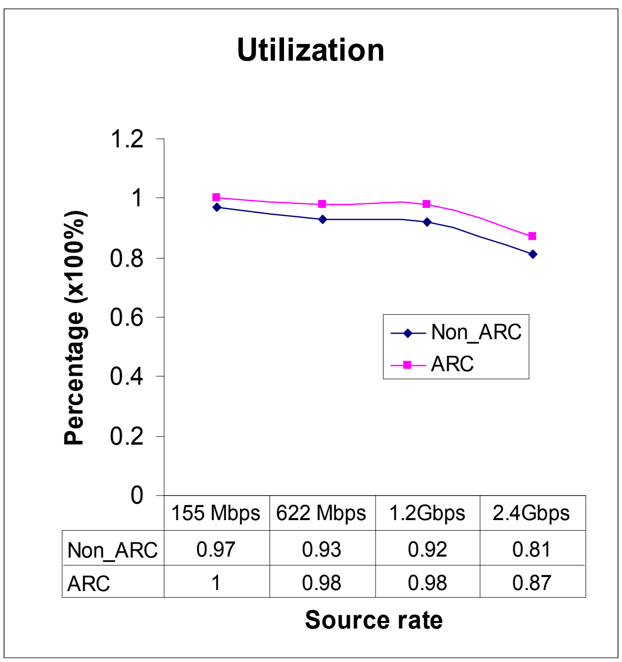


Figure 7. Utilization.

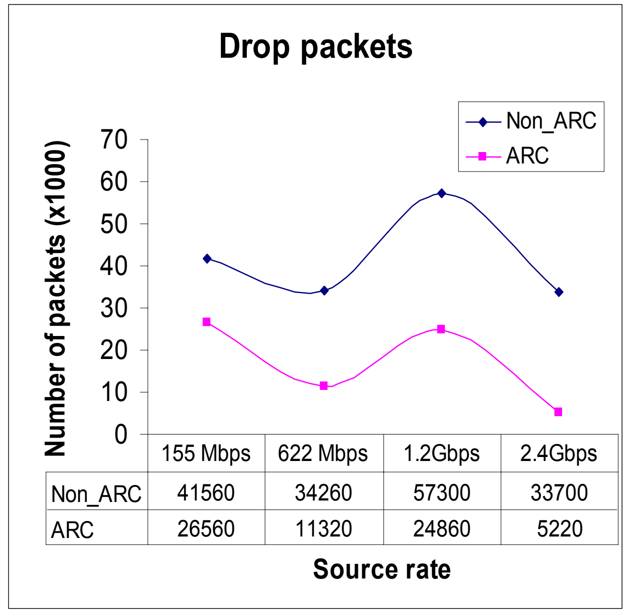


Figure 8. Number of packets drop.

reduce number of packets drop more than half. It is clearly seen that ARC will help reduce 50% of the packets drop at all input traffic rates as shown in figure 8.To ease the problem of packets drop will help boost the system performance in return.

6.Conclusion and Future Works

ARC will outperform especially in case of congestion (as the input arrival rate of the traffic 1.2 Gbps – 2.4 Gbps is much higher than RPR link capacity, 622 Mbps). Simulations confirm that ARC will improve the performance compared to non-ARC one. It does not have to be either costly or complicated but simply adjusts the suitable window size regarding to the rate of input traffic and the packets drop. On the other hand, ARC also gives better performance compared to non-ARC in the case of non-congestion (at 155Mbps). There are many variations and the number of features available keeps asking about our future works. Predictably we have argued against this, so we would conduct some experiments on the expansion of maximum window size. Different link capacities of RPR nodes and ARC extra-ordinary processing time will be further investigated. Also we plan to apply ARC scheme to the extent of low-speed wireless communication. To experiment the case with wider size of the windows we need a lot more modifications than the current simulation. In fact we are on that boundary then our experiences tell it will be hard pressed to provide a decent running platform for our future works. Last but not least an approximation approach will also be another further step of the research.

7.References

[1] D. Tsiang and G. Suwala, The CISCO SRP MAC Layer Protocol, Internet RFC 2892, 2000.

[2] IEEE 802.17, Standards for Resilient Packet Ring (RPR) Networks, 2004.

[3] Pitiporn P. et. al, Performance Comparison of ATM Policy Mechanisms with Telecommunications Traffic, Proceeding of The 6th World Multiconference on Systematics, Cybernetics and Informatics, pp. 373-378, 2002.

[4] Ghanthiya L. et. al, Performance Evaluation for TCP/IP over Satellite Communication with Sliding Window Algorithms, Proceedings of the 4th International Conference on Intelligent Technologies (InTech'03), pp.17-24, 2003.

[5] C. Bach, Fairness Issues of the ATMR MAC Protocol in the Gbit/s Range, Proceedings of the European Conference on Networks and Optical Communications '96, ATM Networks and LANs, IOS Press, pp.30-37, 1996.

[6] T. Saito et. al, QoS Guarantees for High Speed Variable Length Packet LANs, Proceedings of the IEEE PACRIM Conference on Communications, Computers and Signal Processing, Victoria, Canada, vol 2, pp. 770-774, 1997.

[7] C. Bach and A. Grebe, Performance Comparison of Media Access Protocols in the Gigabit/sec Range, 5th IFIP Conference on High Performance Networking, HPN'94, Grenoble,pp.47-61, 1994.

[8] V. Srinivasan and G. Varghese, Faster IP Lookups using Controlled Prefix Expansion, *Proc. ACM SIGCOMM*, pp. 1-10, 1998.

[9] I. Cidon, L. Georgiadis, R. Guerin and Y. Shavit, Improved Fairness Algorithms for Rings with Spatial Reuse, ACM Transactions on Networking,Vol. 5, No. 2, pp.190-204,1997.

[10] S. Prahmkaew et. al, "A Study of Adaptive Rate Algorithm in ATM Network Flow Control," Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'04),pp.668-673, 2004.

[11] B. Khoshnevis, Discrete Systems Simulations, McGraw-Hill, 1994.

.

[12] OPTera Packet Edge System General Specification, vol 1, 2000.

[13] 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*, 11(6), December 2003.