International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 21 Issue 2 – APRIL 2016.

HYBRID CONGESTION CONTROL FOR WIRELESS NETWORKS

Logeshwaran. B[#1], Kishore Shreyas. S[#2], Vignesh. M[#3], Mohd Avesh Gour[#4], Mr. S.P. Maniraj[*5]

[#1]*Department Of Cse, Srm University, Ramapuram*
[#2]*Department Of Cse, Srm University, Ramapuram*
[#3]*Department Of Cse, Srm University, Ramapuram*
[#4]*Department Of Cse, Srm University, Ramapuram*
[*5]*Assistant professor, Department Of Cse, Srm University, Ramapuram*

asklogeshwaran@live.com
shreyaskishore23@hotmail.com
vignesh_m25@yahoo.in
avesgour@gmail.com
spmaniraj@gmail.com

***Abstract*--These days congestion and its negative effects (e.g., Head-of-line blocking) threaten the performance of any network, and so the entire system. Congestion control (CC) is crucial to ensure an efficient utilization of the network during congestion situations. As one major trend is to reduce the effective wiring in any network to reduce cost and power consumption, the network will operate very close to its capacity. Thus, congestion control becomes essential. Existing CC techniques can be divided into two general approaches. One is to throttle traffic injection at the sources that contribute to congestion, and the other is to isolate the congested traffic in specially designated resources. However, both approaches have different, but non-overlapping weaknesses: injection throttling techniques have a slow reaction against congestion, while isolating traffic in special resources may lead the system to run out of those resources. In this paper we propose a new Hybrid Congestion Control technique, that combines injection throttling and congested-flow isolation to minimize their respective drawbacks and maximize overall system performance. This new strategy is suitable for current commercial switch architectures, where it could be implemented without requiring significant complexity.**

I. INTRODUCTION

As a result of a strict end-to-end congestion control, the current Internet suffers from two issues: Congestion collapse from undelivered packets, and unfair allocations of bandwidth between competing traffic flows.

The first issue — congestion collapse from undelivered packets — arises when packets that are dropped before reaching their destination which ultimately continually consume bandwidth.

The second issue —unfair bandwidth allocation to competing network flows—arises in the Internet for a variety of reasons, one of which is the existence of applications that do not respond properly to congestion. Adaptive applications (e.g., TCP-based applications) that respond to congestion by rapidly reducing their transmission rates are likely to receive unfairly small bandwidth allocations when competing with unresponsive applications. The Internet protocols themselves can also introduce unfairness. Thus we propose this Hybrid congestion Control technique for wireless networks which uses the techniques of traffic isolation and injection throttling to overcome their individual disadvantages.

II. PROBLEM DEFINITION

HIGH-SPEED, wireless networks are nowadays essential components for different types of wireless systems, from networks-on-chip to Massively Parallel interconnection network. In particular, in wireless systems, the performance offered by the wireless network must mandatorily meet the high requirements of the applications and the users, otherwise the network would become the system bottleneck and the processing nodes would be idle while waiting for new data to arrive, thereby wasting both computational power and bandwidth. Thus, every aspect of the networks of wireless systems (topology, routing, etc.) should be designed bearing in mind the very high requirements of these networks.

In the existing system, only the System is capable of preventing congestion collapse from undelivered packets

Router Does Not Support in the existing System and Data packets causes frequent congestion collapse.

ICTCP incast congestion control, is implemented primarily through algorithms operating at end systems. Unfortunately, ICTCP incast congestion control also illustrates some of the shortcomings the end-to-end argument. End-to-end congestion control algorithms alone, however, are unable to prevent the congestion collapse and unfairness created by applications that are unresponsive to network congestion. The Internet's excellent scalability and robustness result in part from the end-to-end nature of Internet congestion control.

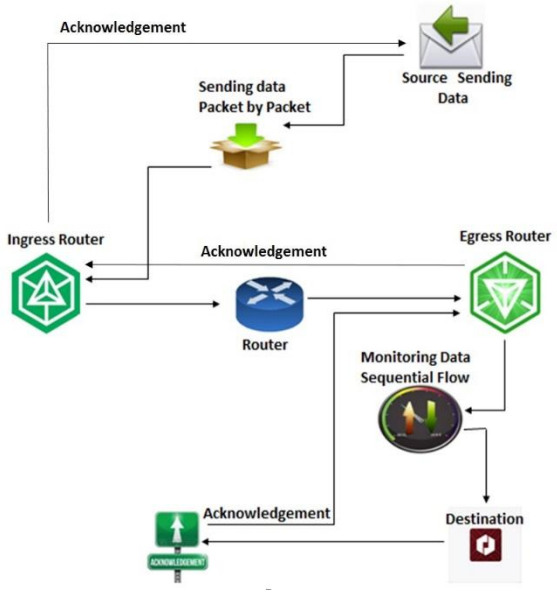
III. PROPOSED SYSTEM

SYSTEM ARCHITECTURE:

This system uses two types of routers namely Ingress router and Egress router. An Ingress router is an edge router through which data flows into the network, whereas the Egress router is an edge router through which data flows out of the network towards the nodes.

The advantage of using this router is to avoid congestion flow and isolate the data packets that flow through the same route. The Ingress and Egress router are connected via a single or multiple routers which connects them from source to destination. The data packets are acknowledged from time to time. The inbound packets are acknowledged once a handful of packets are received. This method avoids congestion as few packets are acknowledged together which reduces the traffic flow from destination to source which in turn reduces the load on the network.

HYBRID CONGESTION CONTROL ARCHITECTURE:



This architecture mainly has two concepts, which are a) monitoring data sequential flow and b) control timer. The sequential flow concept is the one that manages to maintain the order of the packets or data that are being transferred. Control timer is the one that maintains the timer for each data packet to be transferred from either source to destination or destination to source. It also includes other techniques such as injection throttling and isolation of packets where in the existing system both the techniques are used separately whereas in the proposed techniques both are used together which controls the congestion efficiently. The architecture clearly explains the sender sending a packet into he network with the help of the ingress router .The forward feedback which is sent as packets gets into the ingress router where this router is a edge router which helps the packets to get into the network and proceed into the general router. After proceeding the packets into the general router. The packets are further forwarded into the egress router which is also another edge router where the packets are outside the network. The egress router. After the packets gets outside the egress router the sequential and monitoring of data flow is taken care easily the packets are arranged in order to reach the destination. The destination successfully getting the packets sends the acknowledgement packet back through the egress router and at last reaching the sender the confirmation of the packets getting received. NBP (network border patrol) prevents congestion collapse through a combination of per flow rate monitoring at egress router and per flow rate control at ingress router. Rate control allows and an ingress router to police the rate at which each flows packets enters network. The general router maintains at each output queue a value which representing the sending rate it desires from all users traversing the link. The desired rate is updated every period when a control timer expires.

MONITROTING DATA AND SEQUENTIAL FLOW

Efficient error detection is of fundamental importance in dependable computing systems Among the numerous approaches for error detection control of checking is a well-established cost efficient technique Various types of control over monitors based on watchdog processors have been devised to ensure the integrity of control over provides a survey of the basic approaches Within control over monitoring the instruction stream is partitioned into basic blocks A basic block consists of a sequence of consecutive instructions with an unique entry point and an unique exit point In most approaches each block is assigned an identifier that is determined by some of its properties like start address and block size as in or a signature of the instruction sequence within the block. A watchdog processor not only observes the correct signature or size respectively of the block but most importantly it checks the sequence in which blocks are executed For this purpose information describing allowed control ow paths must be provided In most approaches this information is embedded in the instruction stream there are however various strategies to minimize the resulting memory and performance overhead.

IV. LEAKY BUCKET ALGORITHM

The leaky bucket algorithm is a method of temporarily storing a variable number of requests and organizing them into a set-rate output of packets in an asynchronous transfer mode (ATM) network.

The leaky bucket is used to implement traffic policing and traffic shaping in Ethernet and cellular data networks. The algorithm can also be used to control metered-bandwidth. Internet connections to prevent going over the allotted bandwidth for a month, thereby avoiding extra charges.

The algorithm works similarly to the way an actual leaky bucket holds water: The leaky bucket takes data and collects it up to a maximum capacity. Data in the bucket is only released from the bucket at a set rate and size of packet. When the bucket runs out of data, the leaking stops. If incoming data would overfill the bucket, then the packet is considered to be non-conformant and is not added to the bucket. Data is added to the bucket as space becomes available for conforming packets.

The leaky bucket algorithm can also detect both gradually increasing and dramatic memory error increases by comparing how the average and peak data rates exceed set acceptable background amounts

V. TCP CONGESTION CONTROL ALGORITHM

TCP short for Transmission Control Protocol is the most dominant protocol used in computer networking and on the Internet. Characteristically, TCP encompasses various features, one of them is congestion control. In TCP, when a connection is established between a sender and a receiver, an appropriate window size must be selected. Actually, there exist two windows: the receiver window and the sender congestion window. The number of bytes that can be transmitted by the sender is the minimum of these two windows. Hence, in case the receiver window size is 16 KB, and the sender congestion window is 8 KB, then the transmission would occur at 8 KB. In contrast, if the receiver window size is 8 KB and the sender congestion window is 16 KB, then the transmission would occur at 8 KB. The sender calculates its congestion window size by inspecting the property of the medium network such as delays, traffic, and bandwidth; whereas, the receiver calculates its window size based on its buffer size. When a TCP connection is established, the sender initializes the congestion window to the size of the maximum segment available on the connection. It then sends one single maximum segment n. If this segment is acknowledged by the receiver before times out, the sender adds another segment doubling the size of its congestion window and sends the two segments 2n. As each of these segments is acknowledged by the receiver, the congestion window is increased by one maximum segment doubling its previous size.

VI. RATE ALLOCATION ALGORITHM

We consider rate allocation algorithm for resolving fundamental problem of bandwidth allocation among flows in a packet-switched network. The classical max-min rate allocation has been widely regarded as a fair rate allocation policy. But, for a flow with a minimum rate requirement and a peak rate constraint, the classical max-min policy no longer suffices to determine rate allocation since it is not capable of supporting either the minimum rate or the peak rate constraint from a flow. We generalize the theory of the classical max-min rate allocation with the support of both the minimum rate and peak rate constraints for each flow. Additionally, to achieve generalized max-min rate allocation in a fully distributed packet network.

The challenge of bandwidth sharing and rate allocation in a lambda network is how to efficiently and fairly share the capacity of each source and sink among active sessions. Of course, the allocation algorithm should also be stable. We describe the main bandwidth sharing objectives as follows. First, the rate allocation (bandwidth sharing) algorithm should efficiently utilize of the capacity of each source and sink while maintaining feasibility.

ADVANTAGES OF RATE ALLOCATION ALGORITHM:

- Adaptability to network congestions
- Smooth adaptation to network dynamism
- The ability to handle measurements inaccuracies
- Supporting multicast

VII. CONCLUSION

In this paper, we have presented the design, implementation, and evaluation of ICTCP to improve TCP performance for TCP incast in data-center network s. In contrast to previous approaches that used a fine-tuned timer for faster retransmission, we focus on a receiver-based congestion control algorithm to prevent packet loss. ICTCP adaptively adjusts the TCP receive window based on the ratio of the difference of achieved and expected per-connection throughputs over expected throughput, as well as the last-hop available bandwidth to the receiver. Our experimental results demonstrate that ICTCP is effective in avoiding congestion by achieving almost zero timeouts for TCP incast, and it provides high performance and fairness among competing flows.

REFERENCE

[1] K. Yoshiro, "Threshold-based exhaustive round-robin for the CICQ switch with virtual cross point queues," in Proc. IEEE Int. Conf. Commun., 2007, pp. 6325–6329.

[2] G. Pfister, M. Gusat, W. Denzel, D. Craddock, N. Ni, W. Rooney, T. Engbersen, R. Luijten, R. Krishnamurthy, and J. Duato,"Solving hot spot contention using InfiniBand architecture congestion control," in Proc. Int. Workshop High Perform. Interconnects Distrib. Comput., 2005, pp. 943–948.

[3] J. Duato, I. Johnson, J. Flich, F. Naven, P. J. Garc_ıa, and T.Nachiondo, "A new scalable and cost-effective congestion management strategy for lossless multistage interconnection networks," in Proc. 11th Int. Symp. High Perform. Comput. Archit., 2005, pp. 108–119.

[4] P. J. Garc_ıa, J. Flich, J. Duato, I. Johnson, F. J. Quiles, and F. Naven,"Efficient, scalable congestion management for interconnection networks," IEEE Micro, vol. 26, no. 5, pp. 52–66, Sep./Oct. 2006.

[5] G. Mora, P. J. Garc_ıa, J. Flich, and J. Duato, "RECN-IQ: A cost-effective input-queued switch architecture with congestion management, "in Proc. Int. Conf. Parallel Process., 2007, p. 74

[6] https://netbeans.org/downloads/

[7] https://www.java.com/