INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 5 ISSUE 2 (2019) PAGES 4172 - 4181
RECEIVED : 16.02.2019 PUBLISHED : 14.03.2019

March 14,2019

IMPROVEMENT OF QUEUING MECHANISM BY FEEDBACK FORWARD AQM ALGORITHM FOR COMPRESSION OF INFORMATION MODEL IN WIRELESS ADHOC NETWORK

[1]G.ESAIYARASI, [2] JAYANTHI.N,
[1]PG Scholar, Department of Electronics and Communication Engineering, Mahendra Engineering College,
[2]Assitant professor, Department of Electronics and Communication Engineering, Mahendra Engineering College.

ABSTRACT

Developing feed forward model predictive controller as an active queue management (AQM) scheme is studied in this paper. MPC is an advanced control strategy for AQM. Existing models and AQM design methods usually lack sufficient consideration of heterogonous round-trip times, uncertain endpoint mechanisms, and time-varying network conditions. However, the conventional MPC is usually an implementable form of feedback MPC. In this paper, a feed forward and feedback optimal control law is presented. It is a clean, easily implementable, version of model predictive control that incorporates feed forward. Firstly, we use the nominal fluid model to design the feed forward control input so that the output tracks the given queue length with small error. Furthermore, in order to achieve robust performance and to reject the (unmeasured) disturbance, the feedback component is designed. In particular, a disturbance observer is incorporated into the prediction output in standard feedback MPC. This framework can significantly improve performance in the presence of measurement noise and certain types of model uncertainty. Finally, the simulation results show the effectiveness of FF-AQM algorithm.

INDEX TERMS: Feedback forward Active Queue Management (FF-AQM), Model Predictive Controller (MPC)

1. INTRODUCTION

COMPUTER network needs to serve the diverse requirements of different traffic flows simultaneously and effectively. Apart from high throughput and fairness, low latency is extremely important for current Internet with the emergence of many interactive and transaction-based applications.‘Bufferbloat’, a term coined to refer to the oversized buffer in a lot of network elements, has significantly degraded the quality of service (QoS) of interactive applications in the presence of concurrent TCP traffic. During the past few years, much attention has been paid to offer predictably low queuing latency to endpoint users. Controlled Delay (CoDel) and Proportional Integral controller Enhanced (PIE) are two notable queuing-delay based active queue management (AQM) algorithms that have recently been presented as potential solutions to overcome buffer bloat on access links. They achieve satisfying performance in scenarios with normal network delay. However, they encounter performance degradation or instability as delay increases.

2. RELATED WORK

From the analysis of previous reviews, a (typically distributed) algorithm to allocate network resources among competing traffic sources, congestion control is full of challenges that are important but hard to handle as the network grows, i.e. heterogeneity, stability, and fairness. Current congestion control mainly comprises of endpoint congestion control, queue management algorithm, scheduling algorithm, and routing algorithm for load balancing. A sound solution to network congestion cannot be achieved without the cooperation of these complementary control strategies. Among them, AQM schemes aim at reducing buffer occupancy and therefore the end-to-end delay, appearing to be the most promising approach in resolving the bufferbloat problem. An ideal AQM scheme should achieve stability, robustness, responsiveness, high throughput, low queuing latency, and fairness among flows. It should also be scalable and very simple to implement. However, general fairness involves many difficult issues in finding methods with an acceptable overhead cost that can identify and isolate unresponsive flows or flows that are less responsive than TCP. Further research and discussions are needed to address this issue. Therefore, this paper decouples congestion control from fairness control and focuses on the former. The major reasons why the TCP/AQM system is hard to control are summarized as follows. (1) Endpoint congestion control mechanisms are diverse. Traffics in the Internet may comprise of short-lived flows and unresponsive flows among responsive TCP long-lived flows, and TCP flows may be a mix of various TCP variants. Different transport/application protocols employ different congestion control policies. They react differently to congestion indications from the receiver or the network. Even for endpoints employing the same TCP protocol, totally different behaviors will be observed according to its current phase, congestion window (CWND) size, and round-trip time (RTT).

3. PROPOSED METHOD

A feed forward and feedback optimal control law is presented. It is a clean, easily implementable, version of model predictive control that incorporates feed forward. Firstly, we use the nominal fluid model to design the feed forward control input so that the output tracks the given queue length with small error. Furthermore, in order to achieve robust performance and to reject the (unmeasured) disturbance, the feedback component is designed. In particular, a disturbance observer is incorporated into the prediction output in standard feedback MPC.

Block Diagram:

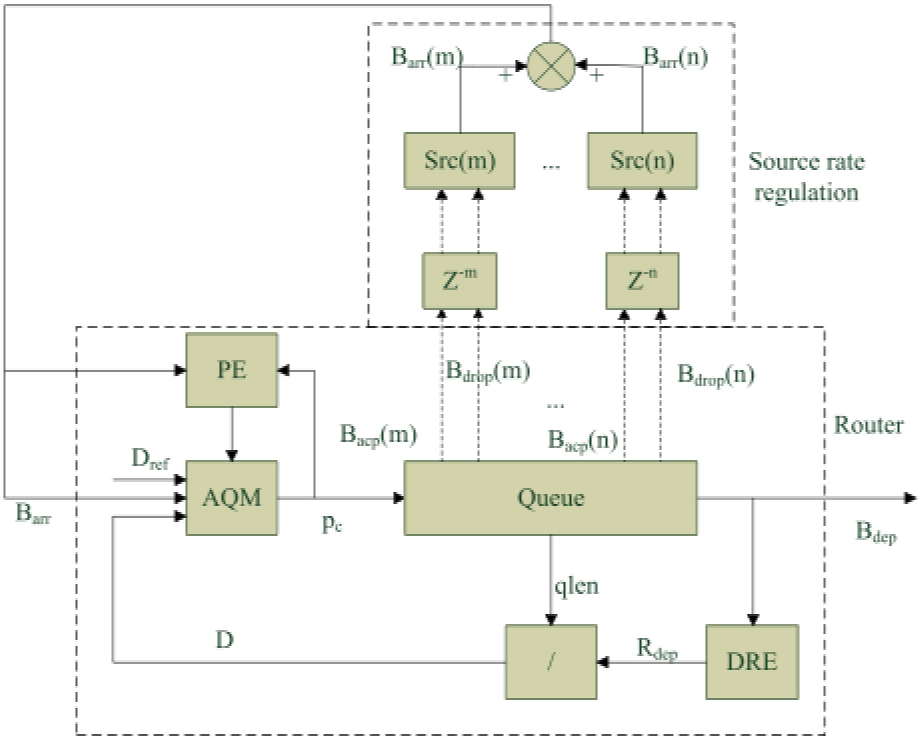


Figure 3.1 proposed Block Diagram:

Different understandings about the essence of network congestion result in different selection of congestion indicators, which directly affects the measurement accuracy of current load. AQM proposals, so far have been queue based, rate-based, load-based, packet-loss-based or a combination of these, and of which the average (usually exponential weighted moving average (EWMA)) or instantaneous samples were used as the congestion indicator. Thus, it recommended tracking the minimum sojourn time experienced by each packet as an effective way to detect the standing queue. For example,During a certain interval, sources with small RTT may send two or three windows of data whereas sources with large RTT may only send partial window of data. Here we argue that themismatch between bandwidth demand and pipe size during sampling interval is the essential cause of standing queue in router. Therefore, we adopt the amount of arrived data during each sampling interval and current queuing latency as congestion indicators.

4. NETWORK SIMULATOR -2

Ns are an object oriented simulator, written in C++, with an OTCl interpreter as a front end. The simulator supports a class hierarchy in C++ (also called the compiled hierarchy in this document), and a similar class hierarchy within the OTCl interpreter (also called the interpreted hierarchy in this document). The two hierarchies are closely related to each other; from the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy. The root of this hierarchy is the class Tcl Object. Users create new simulator objects through the interpreter; these objects are instantiated within the interpreter, and are closely mirrored by a corresponding object in the compiled hierarchy. The interpreted class hierarchy is automatically established through methods defined in the class Tcl Class. User instantiated objects are mirrored through methods defined in the class Tcl Object. There are other hierarchies in the C++ code and OTCl scripts; these other hierarchies are not mirrored in the manner of Tcl Object.

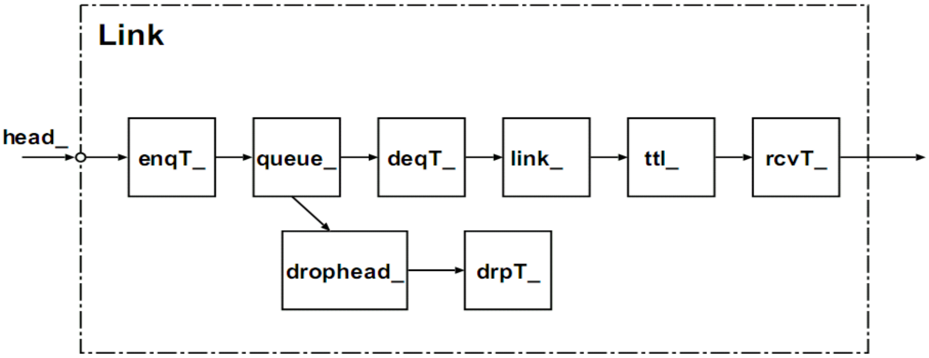


Figure 4.1: Simple link structure.

NS2 Structure

NS2 is an object oriented simulator, written in C++, with a Tcl interpreter as a front-end. The simulator supports a class hierarchy in C++ (also called the compiled hierarchy), and a similar class hierarchy within the Tcl interpreter (also called the interpreted hierarchy).

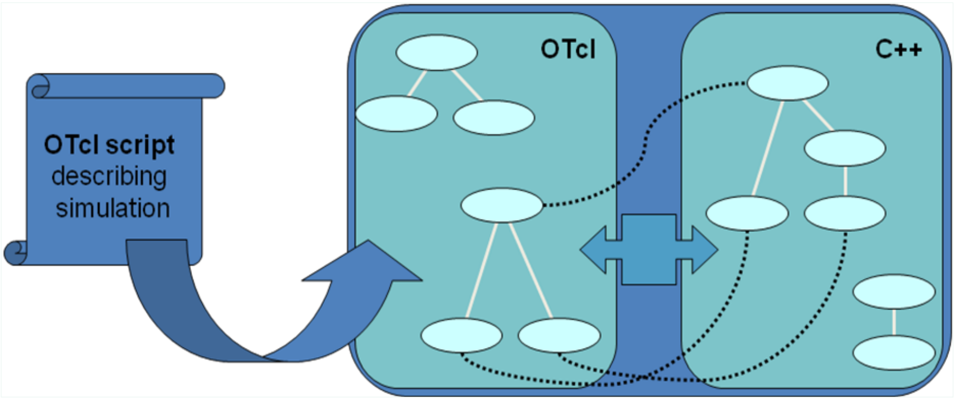


Figure 4.2: NS2 internal schematic diagram

The two hierarchies are closely related to each other; from the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy.

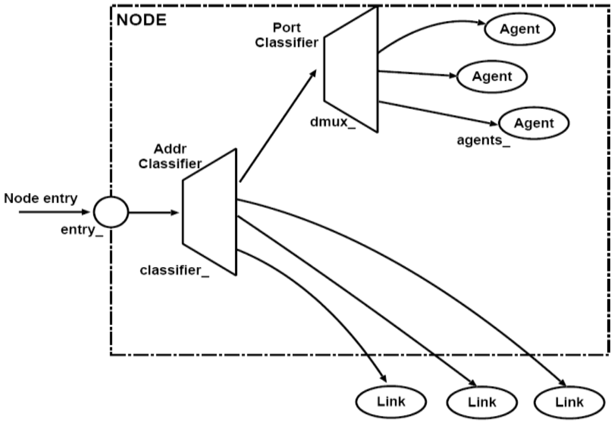


Figure 4.3: Node structure.

5. SYSTEM MODULES

5.1 NETWORK CREATION:

In this module, a wireless network is created. All the nodes are randomly deployed in the network area. Our network is a censored network, nodes are assigned with mobility (movement).Source and destination nodes are defined. Data transferred from source node to destination node. Since we are working in wireless network, nodes mobility is set ie., node move from one position to another Adversary Model. The goal of the adversary is to prevent the sender(s) from communicating with all, or a subset of the intended receivers.

5.2 FEED-FORWARD MODULES

Feed-forward neural network called (FFNN-AQM) to control the network congestion efficiently by stabilizing the queue length. It learns the traffic pattern of the nonlinear and dynamic network and predicts the future value of current queue length. The parameters of neurons adjusted depending on the time-varying environment to stabilize the queue length. The NS2 network simulator is used to analyze the performance of FFNN-AQM along with existing techniques. The simulation experiment results demonstrate that FFNN-AQM is stable and achieve faster convergence with small settling time in varying network conditions.

5.3 FEEDBACK MECHANISM

The implementation of a feedback mechanism described in the paper Feedback-based Solution for Avoiding Attacks on Mobile Ad-Hoc Networks[1]. This implementation should be used for simulation of this mechanism in virtual ad-hoc networks via the well-known simulation software NS2. The implementation of the feedback mechanism itself should be done on the routing layer. The mechanism itself works directly on the routing layer. It discovers malicious behavior on the routing layer. It sends additional messages that directly affect the behavior on the routing layer, e.g. no routing over untrusted or malicious nodes. The implementation can be done by modification of existing routing implementation.

To implement the feedback mechanism, the following issues are realized. The header structure of the common packet is changed. The header hdr_cmn is the main header structure utilized for each packet in NS2. The common header is extended with the following information:

bool recv fb req;

bool one hop fb req;

bool two hop fb req;

int fb type;

int fb dst;

int fb path[PATH LEN];

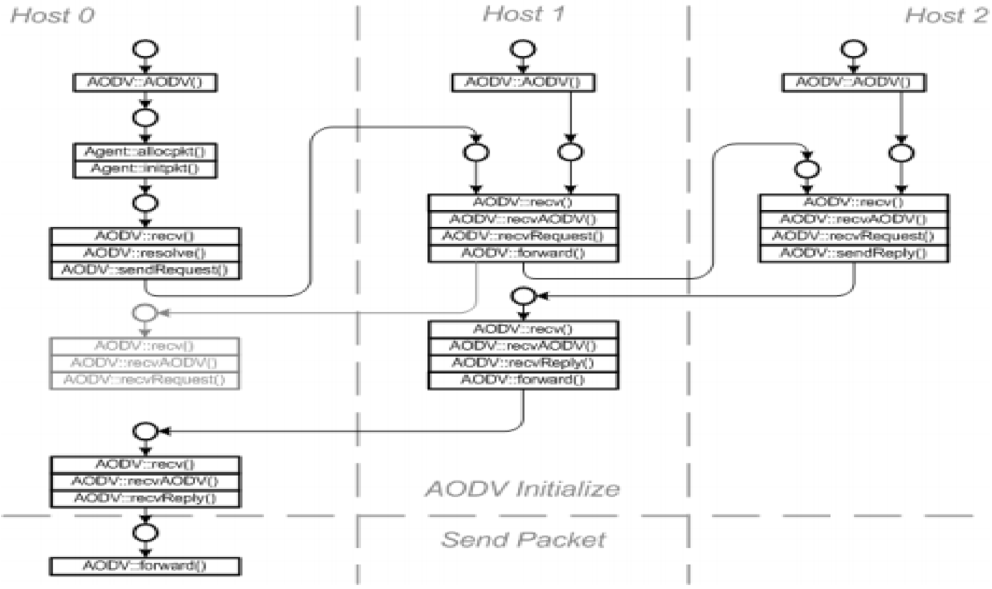


Figure 5.1 Concept of Packet transmission

The first three members are Boolean values utilized to encode the requested feedback into the packet. Each packet needs to carry information for its required feedback. The integer member’s fb_type and fb_dst are used for feedback packets to encode the type of the feedback and its destination. The integer fb_path array fb path is used to save the feedback route.

5.4 TRAFFIC LOAD BY QUEUING MECHANISM

One of the major component in a QoS-enabled network is active queuemanagement (AQM). Over the last decade numerous AQM schemes have been proposed in the literature. However, much recent work has focused on improving AQM performance through alternate approaches. This study focuses on an unbiased comparative evaluation of the various proposals. The evaluation methodology adopted is the following: we first define the relationship between the terminologies used in this paper, briefly introduce the queue, delay, and loss characteristics – a subset of network characteristics that can be used to describe the behavior of network entities, and give their mathematical description. A method that would be a successful case study based on the NS-2 simulation technique for describing network characteristics and simulation-based comparisons of AQM schemes, which will help understand how they differ from in terms of per node queuing information and per-flow end-to-end behavior. To understand this attribute and behavior is important for the proper design of queue disciplines, for the provisioning of queues and link capacity, and for choosing parameters in simulation.

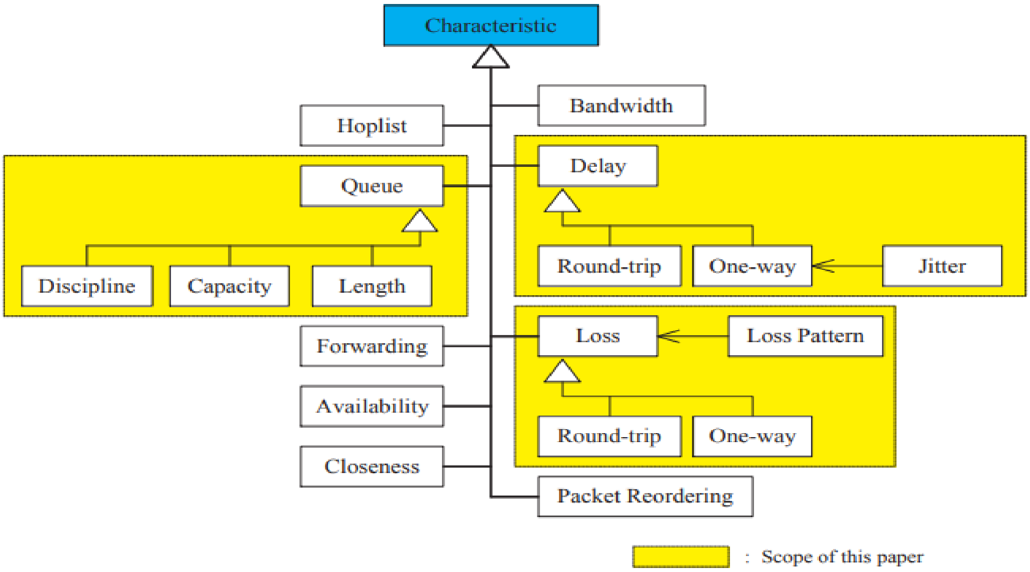
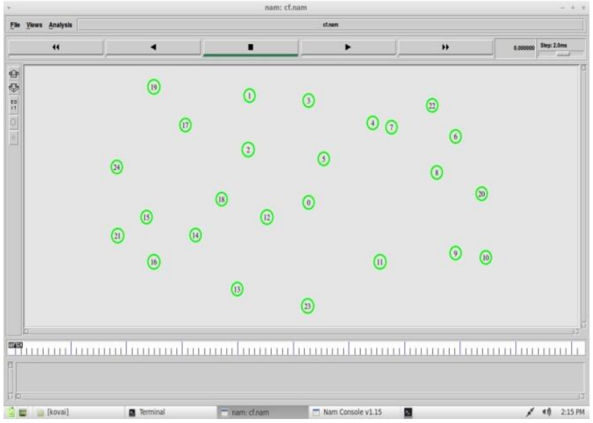
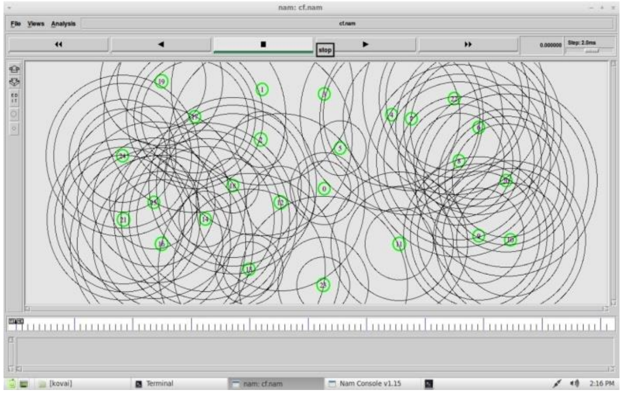


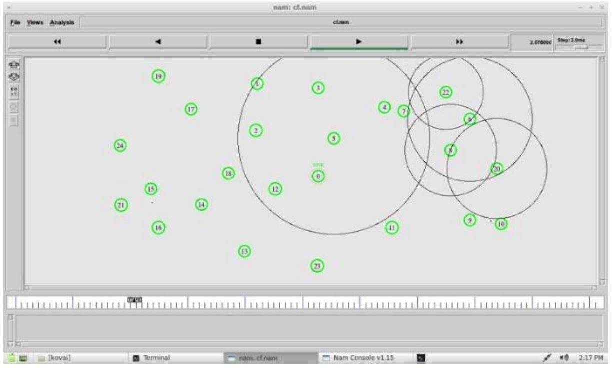
Figure 5.2 Characteristics of networ

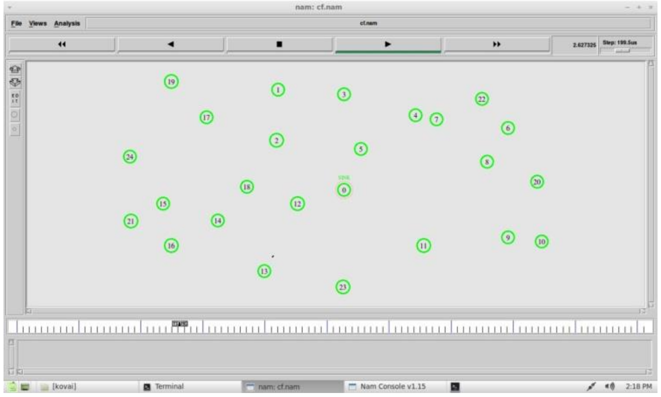


Node construction



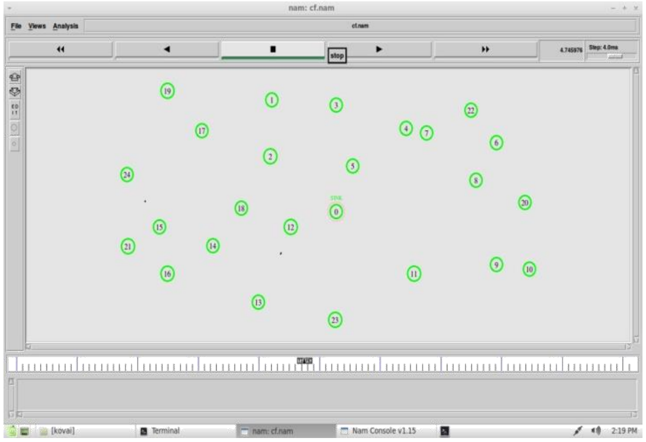
Node activation

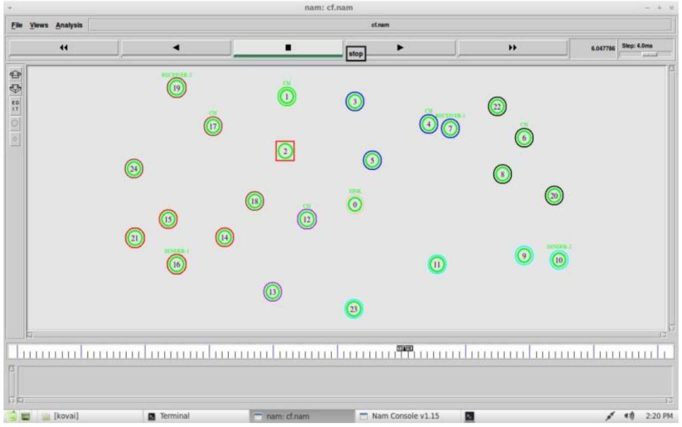




Node Communication

Sink Creation





Sink Creation

Cluster Creation

A queuing system in networks can be described as packets arriving for service, waiting for service if it is not immediate, and if having waited for service, leaving the system after being served. In most cases, four basic characteristics of queuing processes provide an adequate description of a queuing system in networks: (1) Arrival pattern of packets, (2) Service pattern of schedulers, (3) Queue discipline, and (4) system capacity.To model network behavior, analytical models require information about the properties of the queues.The queuing delay and good put performance shown below.Initially the cluster is created and the alternative path is generated from sender to receiver.

CONCLUSION

AQM algorithms are required to be compatible with different network traffic scenarios. In order to be independent of RTT heterogeneity and uncertain endpoint protocols, an novel information compression model for AQM design is proposed in this paper. An adaptive AQM scheme (including a customized parameter identification algorithm) based on this model is designed to deal with the time-varying network conditions. Simulation results show ICM-AQM scales well to scenarios with different RTT and traffic load without the need to tune or reconfigure initial parameters. It provides predictably low queuing latency and high goodput to endpoint users regardless of time-varying network conditions. This method improve performance by reducing delay and loss of packet and analyze node capacity and forward packet between sources to destination to avoid occurrence of traffic.Feedback mechanism is used to achieve robust performance.

REFERENCES

[1] N. Kuhn, D. Ros, A. B. Bagayoko, C. Kulatunga, G. Fairhurst, and N. Khademi, “Operating ranges, tunability and performance of CoDel and PIE,” Computer Communications, vol. 103, pp. 74–82, 2017.

[2] G. Abbas, S. Manzoor, and M. Hussain, “A stateless fairness-driven active queue management scheme for efficient and fair bandwidth allocation in congested Internet routers,” Telecommunication Systems, vol. 6, pp. 1–18, 2017.

[3] G. Abbas, Z. Halim, and Z. H. Abbas, “Fairness-Driven Queue Management: A Survey and Taxonomy,” IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 324–367, 2016.

[4] L. Xue, S. Kumar, C. Cui, P. Kondikoppa, C. H. Chiu, and S. J. Park, “Towards fair and low latency next generation high speed networks: AFCD queuing,” Journal of Network& Computer Applications, vol. 70, pp. 183– 193, 2016.

[5] L. Khoshnevisan and F. R. Salmasi, “A robust and high-performance queue management controller for large round trip time networks,” International Journal of Systems Science, vol. 47, no. 7, pp. 1–12, 2016.

[6] G. White,“Active queue management in DOCSIS 3.1 networks,” Communications Magazine IEEE, vol. 53, no. 3, pp. 126–132, 2015.

[7] E. G. Fairhurst, “IETF Recommendations Regarding Active Queue Management,” Information on Rfc, vol. RFC7567, 2015.

[8] Q. Xu and J. Sun, “A simple active queue management based on the prediction of the packet arrival rate,” Journal of Network & Computer Applications, vol. 42, no. 4, pp. 12–20, 2014.

[9] R. Adams, “Active Queue Management: A Survey,” IEEE Communications Surveys& Tutorials, vol. 15, no. 3, pp. 1425–1476, 2013.

[10] R. Pan, P. Natarajan, C. Piglione, M. S. Prabhu, V. Subramanian, F. Baker, and B. Versteeg, “PIE: A lightweight control scheme to address the bufferbloat problem,” in 2013 IEEE 14th International Conference on High Performance Switching and Routing (HPSR), IEEE, 2013, pp. 148–155.

[11] W. J. Chang, P. H. Chen, and C. T. Yang, “Robust Fuzzy Congestion Control of TCP/AQM Router via Perturbed Takagi-Sugeno Fuzzy Models,” International Journal of Fuzzy Systems, vol. 15, no. 2, pp. 203–213, 2013.

[12] K. Nichols and V. Jacobson, “Controlling Queue Delay,” Queue, vol. 55, no. 7, pp. 42–50, 2012.

[13] S. Harbarth and D. Pittet, “Open Research Issues in Internet Congestion Control,” Qjm An International Journal of Medicine, vol. rfc 6077, no. 8, pp. 519–521, 2011.