Computer Networks 93 (2015) 54–65

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

QRM: A queue rate management for fairness and TCP flooding protection in mission-critical networks[☆]

Maurizio Casoni, Carlo Augusto Grazia*, Martin Klapez, Natale Patriciello

Department of Engineering Enzo Ferrari, University of Modena and Reggio Emilia, via Pietro Vivarelli 10, Modena 41125, Italy

ARTICLE INFO

Article history:
Received 30 April 2015
Revised 12 October 2015
Accepted 17 October 2015
Available online 24 October 2015

Keywords:
AQM
Congestion control
Middleware
Mission-critical networks
Queueing delay
Satellite

ABSTRACT

When statistical multiplexing is used to provide connectivity to a number of client hosts through a high-delay link, the original TCP as well as TCP variants designed to improve performance on those links often provide poor performance and sub-optimal QoS properties. Centralised and collaborative resource management tools like **C^{2}ML** have been proposed to guarantee intra-protocol fairness, inter-protocol friendliness, low queues utilisation and optimal throughput along with the reliable delivery of packets. However, such tools offer only very limited security guarantees. Both *good citizenship* and *security from flooding attacks* are fundamental conditions for the provision of fairness, especially in mission-critical networks. For example, perpetrators of a man-provoked disaster may want to perform a resource exhaustion attack on the network supporting disaster recovery operations, so as to cut out legitimate users from the communications and increase the emergency impact. In this paper we present Queue Rate Management (**QRM**), an Active Queue Management scheme able to provide protection from traffic overflow attacks in scenarios where access to the shared link is controlled by a tool that assigns to client hosts a bandwidth upper bound. The proposed algorithm checks whether a node is exceeding its allowed rate, and consequently decides whether to keep or drop packets coming from that host. We mathematically prove that with **QRM** the gateway queue size can never exceed the Bandwidth-Delay Product of the channel. Furthermore, we employ the ns-3 network simulator to compare **QRM** with CoDel, RED and GREEN, showing how **QRM** provides better performance in terms of both throughput and QoS guarantees in the aforementioned scenarios.

1. Introduction

When a high-delay link is shared among a number of client hosts, both the original TCP and other TCP variants designed to improve performance on those links are unable to effectively make optimal use of the channel. In these scenarios, in fact, the major limit is not represented by a particular congestion control algorithm, nor by the link performance, but instead by the transport protocol fundamentals. TCP infers the channel conditions indirectly, by loss detection and/or ACK-based timing information, and then reacts accordingly. When client hosts are *wirelessly* connected to the gateway that in turn provides them with access to the high-delay link, their collective performance may further degrade as a consequence of the nature of wireless access; the end-to-end congestion control of each node, which is independent from adjacent hosts, may not be able to ensure a high Quality of Experience (QoE).

[☆] This work was also supported by the European Commission under PPDR-TC Grant No. 313015, a collaborative project part of the FP7-SECURITY. The authors would like to thank all partners within PPDR-TC for their cooperation and valuable contribution.

* Corresponding author. Tel.: +39 3494351546.
E-mail addresses: maurizio.casoni@unimore.it (M. Casoni), carloaugusto.grazia@unimore.it, graziacarloaugusto@gmail.com (C.A. Grazia), martin.klapez@unimore.it (M. Klapez), natale.patriciello@unimore.it (N. Patriciello).

http://dx.doi.org/10.1016/j.comnet.2015.10.010
1389-1286/© 2015 Elsevier B.V. All rights reserved.

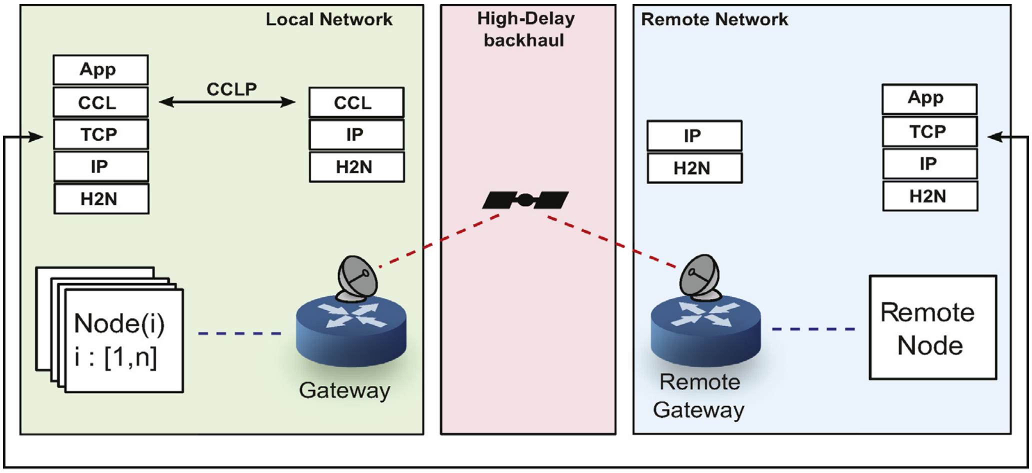

Fig. 1. C^2ML scheme.

These kinds of network topologies are often employed for emergency management in post-disaster situations, where deployable networks must be backhauled by satellite channels when terrestrial infrastructures are destroyed. In these cases, shifting to a top-down, centralised and collaborative management of resources from the gateway has the potential to guarantee overall better performance for hosts [1–5]. To reach this result we previously proposed C^2ML [6], which aimed at improving QoE for users in such scenarios. Fig. 1 shows its architecture. Conceptually, it inserts a Congestion Control Layer (C^2L) between the Application and the Transport Layers, that uses Congestion Control Layer Protocol (C^2LP) as the default standard to communicate. Basically, to the gateway that provides clients with external connectivity, C^2ML gives the authority of instructing these client hosts on the amount of bandwidth they are allowed to use. This is trivially calculated dividing the bottleneck link capacity by the number of clients to serve, in order to guarantee a fair utilisation of the former. The readers interested to the C^2ML protocol can find further details in our previous work [6].

Disasters may be natural or man-provoked. In the latter case, C^2ML does not provide adequate security guarantees; more specifically, it does not protect legitimate users from Resource Exhaustion Attacks nor flooding techniques aimed at ruining end-users QoE, because it assumes that all users would respect the bandwidth upper bound assigned to them. Making a client host to behave maliciously by purposely forcing it to disobey the C^2ML rules, thus sending data at a rate that exceeds the allowed bandwidth, is an operation that does not require extensive technical abilities and that can easily result in denial of service for all the other hosts. For an attacker, it would thus be possible to overload the backhaul link by sending a lot of traffic to a remote host. This would surely exhaust the channel resources, effectively cutting out legitimate hosts from the service. Moreover, a client host that for some reason does not update its rate after a gateway order may lead to impaired performance for all the other clients. The paper presents a per-node Queue Rate Management (QRM), an Active Queue Management (AQM) scheme that aims to fill this gap; coupled with any cooperative solution like C^2ML, it detects an attacker flow on the basis of incoming packets, and reacts accordingly by enqueuing or dropping them. We show how, if compared to other AQM schemes such as CoDel, RED, and GREEN, QRM offers better performance and QoS guarantees when coupled with the aforementioned systems.

The discussion is organised as follows: Section 2 presents related work; Section 3 describes the scenarios that may benefit from centralised and collaborative solutions; Section 4 presents QRM; Section 5 details a formal analysis of QRM, and Section 6 presents simulation results. The conclusions are drawn in Section 7.

2. Related work

AQM techniques complement the end-to-end congestion control performed at the transport layer, either explicitly (e.g. through Explicit Congestion Notification) or implicitly (e.g. through packet drops) [7]. The simplest buffer management algorithm, Drop Tail, simply drops whichever packet tries to enter an already full queue, thus it only acts when congestion is already occurred. In time, the main goal of AQM schemes has evolved so as to perform congestion avoidance [8,9]; a widespread example is RED [10,11], which keeps the average queue size small by anticipating congestion with a linear dropping function or, like a RED variant called Non Linear RED [12], with a quadratic dropping function. However, it does not always provide fair queuing; protocols that do not adjust their transmission rate according to congestion, such as UDP, or protocols that are aggressive in adjusting it to congestion events, such as TCP variants designed for satellite links [13–17], end up using more bandwidth than other TCP flows. Furthermore, RED needs a manual configuration related to the characteristics of the link in order to behave at its best.

FRED [18] introduces per-flow state information to RED, and it effectively succeeds in improving fairness among flows; however, the drawback is the high quantity of information that has to be maintained on the gateway. CHOKe [19] has been designed to improve the fairness of RED without