Defeating Architectures for Low-Latency Services: The Case of L4S

Marius Letourneau
LIST3N
University of Technology of Troyes
Troyes, France
Email: marius.letourneau@utt.fr

Guillaume Doyen
OCIF - IRISA
IMT Atlantique
Rennes, France
Email: guillaume.doyen@imt-atlantique.fr

Rémi Cogranne
LIST3N
University of Technology of Troyes
Troyes, France
Email: remi.cogranne@utt.fr

***Abstract*—For a couple years, new services with low-latency requirements appear to be a major challenge for the future Internet. Many optimizations, all targeting the latency reduction have been proposed, and among them, re-architecting the network packet processing has been particularly considered. In this effort, the L4S proposal aims at allowing both classic and low-latency traffic to cohabit within a single node architecture. However, although this architecture sounds promising for latency improvement, it also lacks an in-depth study of security issues. More specifically, it is possible for an attacker to manipulate end-to-end protocols to defeat the low-latency feature of L4S nodes. In this paper, we analyze the set of weaknesses in L4S that could be exploited by an attacker to perform a set of malicious actions whose purpose is to defeat the low-latency.**

I. Introduction

Over the past few years, Internet actors are designing new architectures and protocols to enable new services that requires low-latency such as cloud robotics, cloud gaming or telemedicine. From a security perspective, these emerging solutions are likely to offer a wider attack surface to malicious users. Especially, although traffic bandwidth exhaustion has been at the core of numerous Denial of Service attacks [1] for the last two decades, the latency in itself becomes now a target for an attack as a resource to exhaust.

Latency reduction has to be performed both in endpoints, with scalable congestion control and in the network through queuing delay reduction. To that aim, Low Latency, Low Loss and Scalable Throughput (L4S) [2], [3] is a novel architecture that enables the coexistence between classic flows and low-latency flows on cable networks. However, to make L4S architecture able to ensure the expected properties and Quality of Service (QoS), it must prove its robustness regarding malformed flows or malicious users. For instance, in [13], it has been experimentally proved that the sharing behavior of the L4S architecture is sensitive to traffic bursts, leading to unfairness among flows depending on congestion control algorithms implemented by the endpoints. In this context, it appears possible to an attacker to manipulate endpoints transport protocols [6] to defeat the low-latency guarantee that L4S must provide.

The open questions which raise consequently concern (1) the possible attacks a malicious user can implement against L4S, (2) their impact over the different components and neighboring flows crossing a node under attack and (3) the means that can be designed and implemented to detect and mitigate them while still respecting the expected low-latency feature.

II. Low Latency, Low Loss and Scalable Throughput (L4S)

Low Latency, Low Loss and Scalable Throughput (L4S) [2], [3] is an architecture under standardization at the IETF that focuses on reducing queuing delay for flows with a low-latency requirement. Coexistence and fairness between low-latency flows and classic flows are strong prerequisites in the design of L4S. This is realized leveraging Explicit Congestion Notification (ECN) [4] and a Dual Queue Coupled Active Queue Management (DQC AQM) [5].

The DQC AQM is itself composed of three elements: an ECN classifier, a coupling mechanism and a scheduler. The ECN classifier differentiates classic flows and L4S flows at the IP level by checking ECN flags. The coupling mechanism between L4S queue and classic queue ensures a better reactivity to congestion for low-latency flows without injuring classic flows neither risking starvation. The scheduler guarantes flow fairness, priorization of L4S flow and may absorb small packet bursts.

III. Threats Altering Latency in L4S Architecture

In this section, we present vulnerabilites already identified regarding L4S architecture that a malicious user can exploit to impact the low-latency requirement of future services. We term *undesirable flows* when referring to malformed flows, unresponsive flows and misbehaving flows, which includes both legitimate misbehaving flows and attack flows.

A. Unresponsive Flows

As mention by the IETF in [5], L4S can natively handle traffics that are unresponsive, less-responsive and/or temporarily unresponsive to congestion as long as their proportion is reasonable, however it can be an issue when it starts to lead the Dual Queue Coupled AQM into a queue-building behavior. This kind of traffic can be introduced in both classic and L4S queue and can lead to overloading queues or to congestion signal saturation.

B. Malformed flows

Malformed flows are usually legitimate but undesirable from a L4S perspective. A bursty behavior may occur when the network stack of regular operating systems within endpoints waits for the sending buffer to be fulfilled before actually send data over the network, which result in an on/off pattern that injurs L4S performances [13].

C. Misbehaving Flows

1) Protocol manipulation: A protocol manipulation is the ability of some of the participants to subvert the protocol without the knowledge of the others. Most of these attacks are TCP-centered.

We can first mention acknowledgement manipulation attacks, that we term *hacked ACK*. They manipulate an endpoint of a TCP communication to make the victim saturates the network (more specifically an edge router shared by targeted victims). The optimistic acknowledgment (opt-ack) attack [6], [7], [11] is a well-studied example which consists in misleading a sender to send more packets. The receiver send acknowledgements before it actually receives packets, leading the sender to behave as if the network was in good condition enough to send even more packets.

Congestion can be created in intermediate nodes by several manners. When it comes to manipulate ECN, we term this as *hacked ECN*. One can conceal the congestion notification by not informing the partner of the communication [9], [6], [10], an attacker can also generate false congestion notification in order to steal more bandwidth.

2) Low-Rate DoS: The general model for low-rate DoS (LDoS) attacks is described in [12]. The idea is to send periodic bursts of packets that are synchronized with the victim's Retransmission Timeout (RTO) in order to overflow router's queue and eventually latency increase. LDoS attacks are more difficult to detect in comparison with regular DoS or DDoS attacks and can be sustained as long as the periodic generation of burst is appropriately synchronized.

D. Countermeasures

The IETF has identified these issues [3], [5], and overall, the proposed countermeasures handle unresponsive flows by renouncing to some performances (sacrificing L4S delay, L4S throughput or introducing L4S drop). Traffic shaping and traffic policing (or queue protection) are also considered for malformed flows. However, classical technics for traffic shaping are not always applicable, as it may lead to the bufferbloat problem. TCP Pacing is a solution that can be required for endpoints to respect before sending anything on the network, combined with fair-queuing within the endpoints's network scheduler which can drastically reduce traffic burstiness.

IV. Ongoing Work and Perspectives

Our first objective consists in understanding what vulnerabilites can be exploited by leveraging protocol manipulation attacks and undesirable flows to target latency in the L4S architecture.

As TCP Pacing and fair-queuing are neither strongly deployed nor mandatory, an attacker can deliberately inject bursty traffic without being suspected. In addition, he can trigger and amplify effects presented in Section III by jointly generating an unresponsive ECN-capable traffic: a protocol-compliant flow that respect expected signaling but that does not reduce its sending rate accordingly. A traffic adopting this behavior can steal bandwidth to others by taking advantage of other's reduction in their sending rate. Introducing packet bursts, in a low-rate DoS manner, when congestion notification are received, can also maximize the damage done while being difficult to detect since standing for a legitimate flow. To assess these attack scenarii, we are currently experimenting them in a testbed in which the L4S Linux reference implementation is deployed.

Besides, we plan to develop a consistent mathematical model whose purpose is to link together the congestion signaling probability, the congestion window evolution and the damage anticipated regarding bandwidth stealing and latency increase in order to facilitates the detection solution we plan to design. We then forecast to focus on detection considering statistical models to implement into a dedicated micro-service to detect undesirable flows targeting L4S architecture. As a mitigation solution, we consider the forwarding of problematic flows into scrubbing facilities to block the attack and prevent any disturbances in low-latency services.

References

[1] S. Cook, *DDoS attack statistics and facts for 2018-2021*. Kent, United Kingdom: Comparitech, 2021.

[2] B. Briscoe and K. De Schepper and O. Tilmans and G. White and A. S. Ahmed and O. Albisser, *Low Latency Low Loss Scalable Throughput (L4S)*. TSVWG, IETF, 2020.

[3] B. Briscoe and K. De Schepper and M. Bagnulo and G. White, *Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture*. Internet Engineering Task Force, 2021.

[4] S. Floyd and K. K. Ramakrishnan and D. L. .Black, *The Addition of Explicit Congestion Notification (ECN) to IP*. Internet Engineering Task Force, 2001.

[5] K. De Schepper and B. Briscoe and G. White, *DualQ Coupled AQMs for Low Latency, Low Loss and Scalable Throughput (L4S)*. Internet Engineering Task Force, 2021.

[6] N. Kothari and R. Mahajan and T. Millstein and R. Govindan and M. Musuvathi, *Finding Protocol Manipulation Attacks*. SIGCOMM Comput. Commun. Rev., 2011.

[7] R. Sherwood and B. Bhattacharjee and R. Braud, *Misbehaving TCP Receivers Can Cause Internet-Wide Congestion Collapse*. ACM Conference on Computer and Communications Security, 2005.

[8] E. Blanton and V. Paxson and M. Allman, *TCP Congestion Control*. Internet Engineering Task Force, 2009.

[9] D. Ely and N. Spring and D. Wetherall and S. Savage and T. Anderson, *Robust congestion signaling*. ICNP, 2001.

[10] A. Laraba and J. François and I. Chrisment and S. R. Chowdhury and R. Boutaba, *Defeating Protocol Abuse with P4: Application to Explicit Congestion Notification*. IFIP Networking, 2020.

[11] A. Laraba and J. François and S. R. Chowdhury and I. Chrisment and R. Boutaba, *Mitigating TCP Protocol Misuse With Programmable Data Planes*. IEEE Transactions on Network and Service Management, 2021.

[12] W. Zhijun and L. Wenjing and L. Liang and Y. Meng, *Low-Rate DoS Attacks, Detection, Defense, and Challenges: A Survey*. IEEE Access, 2020.

[13] D. B. Oljira and K. J. Grinnemo and A. Brunstrom and J. Taheri, *Validating the Sharing Behavior and Latency Characteristics of the L4S Architecture*. SIGCOMM Comput. Commun. Rev., 2020.