Performance Evaluation 170 (2025) 102512

Contents lists available at ScienceDirect

Performance Evaluation

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

Mitigating massive access with Quasi-Deterministic Transmission: Experiments and stationary analysis☆

Jacob Bergquist [a], Erol Gelenbe [b,c,*], Mohammed Nasereddin [b], Karl Sigman [a]

[a] Department of IEOR, Columbia University, New York, 10027, N.Y., USA
[b] Institute of Theoretical & Applied Informatics (IITIS-PAN), 5 ul. Baltycka, 44100 Gliwice, Poland
[c] CNRS I3S Université Côte d'Azur, 28 Avenue de Valrose, Nice, 06103 Cedex 2, France

ARTICLE INFO

Keywords:
Internet of Things
Massive Access Problem
Flood attacks
Traffic shaping
Attack Detection (AD)
Quasi-Deterministic Transmission Policy
Queueing theory
Stationary point processes
Cyberattack detection
Measurements

ABSTRACT

The Massive Access Problem arises due to devices that forward packets simultaneously to servers in rapid succession, or by malevolent software in devices that flood network nodes with high-intensity traffic. To protect servers from such events, attack detection (AD) software is installed on servers, and the Quasi-Deterministic Transmission Policy (QDTP) has been proposed to "shape traffic" and protect servers, allowing attack detection to proceed in a timely fashion by delaying some of the incoming packets individually based on their arrival times. QDTP does not cause packet loss, and can be designed so that it does not increase end-to-end packet delay. Starting with measurements taken on an experimental test-bed where the QDPT algorithm is installed on a dedicated processor, which precedes the server itself, we show that QDPT protects the server from attacks by accumulating arriving packets at the input of the QDTP processor, then forwarding them at regular intervals to the server. We compare the behaviour of the server, with and without the use of QDTP, showing the improvement it achieves, provided that its "delay" parameter is correctly selected. We analyze the sample paths associated with QDTP and prove that when its delay parameter is chosen in a specific manner, the end-to-end delay of each packet remains unchanged as compared to an ordinary First-In-First-Out system. An approach based on stationary ergodic processes is developed for the stability conditions. Assuming mutually independent and identically distributed inter-arrival times, service times and QDTP delays, we exhibit the positive recurrent structure of a two-dimensional Markov process and its regeneration points.

1. Introduction

The number of devices in the Internet of Things (IoT) reached $18Bn$ by the end of 2023, and is expected to attain $20Bn$ by the end of 2025 [1]. While this is less than the $30Bn$ devices that was predicted in 2020 [2] for 2023, it is still extremely large. Since the majority of these devices are low-cost simple machine-to-machine devices [3] which communicate via base stations or IoT Gateways, and forward large amounts of data to the Cloud and Edge, these large networked systems can experience a form of congestion known as the "Massive Access Problem" [4], causing untenable delays, possible packet loss, and the slowdown of needed attack detection software, due to the higher-priority packet processing tasks that are carried out by the multi-core servers.

☆ This article is part of a Special issue entitled: 'MASCOTS 2024' published in Performance Evaluation.

* Corresponding author at: Institute of Theoretical & Applied Informatics (IITIS-PAN), 5 ul. Baltycka, 44100 Gliwice, Poland.
E-mail address: seg@iitis.pl (E. Gelenbe).

https://doi.org/10.1016/j.peva.2025.102512

Available online 3 November 2025

The effect of the Massive Access Problem is similar to that of "flood attacks" or Distributed Denial of Service (DDoS) attacks which occur when a large number of packets are sent towards one or more IP addresses, often as a result of a Denial of Service (DoS) or Botnet attack. Thus, many efforts have addressed the Massive Access Problem with congestion-based adaptive routing [5], access class barring [6,7], randomization and scheduling of packets [8], smart machine-to-machine communication [9,10], device clustering [11,12] and other techniques such as Joint Forecasting-Scheduling and Priority based on Average Load [13], and reactive techniques that adapt the receiver's capacity to receive and process incoming traffic [14–18]. Other work has proposed that the transmitters may cooperate to improve channel usage efficiency and QoS [19,20], despite the well-known difficulty of managing access among unsynchronized distributed devices [21], and proactive prediction of IoT traffic patterns [22–24]. However, a scheduling approach may require additional computation, while Machine Learning (ML) to analyze the arrival and service characteristics, and sophisticated scheduling techniques can cause additional computation and communication costs.

Traffic shaping is a simpler approach that can be implemented at the sources of traffic [25]. It is widely used in networks [26] to reduce latency and optimize the bandwidth available to certain packets by delaying some other packets. Typically used at the source or edge, it is defined by the International Telecommunication Union (ITU) [27] as a scheme which *"alters the traffic characteristics of a stream of cells ... to achieve a desired modification of those traffic characteristics, in order to achieve better network efficiency whilst meeting the QoS objectives or to ensure conformance"*. However, the ITU also indicates that many traffic shaping techniques have the *" ... consequence of increasing the mean cell transfer delay"*. Though traffic shaping is mainly accomplished by delaying packets, it is sometimes confused with "traffic policing" which includes preventive packet dropping [28], while traffic shaping can result in more delay for some packets that may cause loss of data in finite buffers. Both approaches have been widely discussed for Asynchronous Transfer Mode (ATM) communications [29] and for the Internet Protocol (IP) [30,31].

Recent work [32] has introduced the Quasi-Deterministic-Transmission-Policy (QDTP) for the shaping of traffic sent by IoT devices, to protect an IoT Gateway from the Massive Access Problem, and from the massive amount of traffic generated by Denial of Service (DoS) or UDP flood attacks. When a cyberattack detection algorithm (AD) is installed in the IoT Gateway or server, to detect a DoS or flood attack rapidly and help mitigate its effect, the AD, which is implemented as application-level software, can be substantially slowed down by the amount of processing used by higher-priority network protocol software and the operating system to handle and store the incoming packets. The AD's slowdown then delays its ability to detect both attacks and the equally important end of an attack. In such circumstances, QDTP, which we detail in Section 3, does not drop packets and can be placed on specific hardware (such as a Raspberry Pi) between the network and the Gateway, to dynamically delay the packets' arrival at the Gateway and protect the Gateway software, including the AD, from being overwhelmed.

If the QDTP delay is set to a value that does not exceed the processing time of the AD, it was shown in [33] that the total end-to-end delay of incoming packets, including the queueing time for the QDTP, the QDTP delay, plus the waiting and processing time at the AD, does not increase as compared to the case where QDPT is not used, both under normal operation and when an attack occurs. Experiments with IoT data [34] have also experimentally demonstrated QDTP's effectiveness to alleviate the Massive Access Problem and improve QoS [35]. This paper builds on prior work presented at the IEEE MASCOTS Conference in 2024 [36], and its extensions [37,38].

In Section 1.1 we briefly recall the QDTP System, which is composed of the QDTP algorithm, and the AD queue at the server. Then, in Section 2, we evaluate the QDTP algorithm in an experimental setting. In particular, we study the system in the case of flood attacks, where the system cannot be in steady-state, since the external interarrival times are far shorter than both the QDTP delays and the service times. We briefly describe the computer and network architecture that is used for the experiments, including a set of Raspberry Pis that emulate IoT devices, and an Ethernet switch that interconnects them with an IoT Gateway server, as well as the server that supports the SNMP communication management software, a software Attack Detector (AD), and processing software for packets that leave the AD. The implementation of QDTP as a software module that resides on its own specific low-end computer (a Raspberry Pi), which we call the"Smart QDTP Forwarder" (SQF), as shown in Fig. 2, is also briefly described. The SQF receives packets from IoT devices or other sources via an Ethernet Local Area Network, shapes the packets' departure instants using QDTP, and forwards the same packets to the IoT Gateway server. We are thus able to evaluate the effect of QDTP in the presence of the congestion caused by a flood attack in two ways:

1. We examine the large queue that builds up at the AD input when high levels of congestion or flood attacks occur without the use of QDTP, and also observe the very small queue that builds up at the server when QDTP is used, so that all the congestion is accumulated at the entrance to the QDTP algorithm itself (i.e. at the SQF).
2. We measure the slow-down that occurs in the AD's packet processing times when an attack occurs without the use of QDTP, and show that with QDTP, this slow-down remains very small under 10% on average. These results, which cannot be observed or predicted using standard probabilistic analysis, allow us to illustrate the value and usefulness of QDTP.

We discuss the queueing theoretic model of QDTP in Section 3, where the packets are *customers*, and QDTP's delay facility is called a *"café"*, where the customers would prefer to spend most of the end-to-end delay, rather than in a queue. Then, in Section 3.1, we develop a sample-path approach, and prove an important result in Proposition 3.1 regarding the end-to-end packet delay of QDPT. Specifically, we show that the inequality in Theorem 1 of [33] **is in fact an equality**. Thus, we prove that QDPT does not change the end-to-end delay of packets (customers) **at all** as compared to a FIFO queue in front of the AD that does not use QDTP, contrary to most traffic shaping techniques, as stated by the ITU [27] (see above).

A stochastic model with stationary and ergodic input is considered in Section 4, and the stability of QDTP is examined in Section 4.2. Then in Section 4.3, we assume the independence and identical distribution (i.i.d.) of inter-arrival, service, and the QDTP delaying parameters, to prove the Harris recurrence of an underlying two-dimensional Markov process and exhibit the positive recurrent regeneration points, including for cases where the system may never empty. Finally, conclusions and suggestions for future work are presented in Section 5.

1.1. The QDPT system

The QDTP System is comprised of two First-In-First-Out (FIFO) queues in tandem, that contain and forward packets [33]:

- The first queue is formed by packets that arrive from the "external world" to the QDTP algorithm at instants $\{a_n : a_{n+1} \geq a_n,\ n = 1,\ 2,\ \ldots\}$, and leave the QDTP algorithm at the instants $t_n : t_{n+1} \geq t_n,\ n = 1,\ 2,\ \ldots\}$ to join the second queue, where $t_1 = a_1$ and:

$$\begin{aligned} t_{n+1} &= a_{n+1},\ if\ a_{n+1} > t_n + D_n, \\ &= t_n + D_n,\ if\ a_{n+1} > t_n + D_n\ , \end{aligned} \tag{1}$$

  where $D_n \geq 0$, is the delay constant, a real number that can depend on n. We then define $W_1 = 0,\ W_{n+1} = t_{n+1} - a_{n+1},\ n = 1,\ 2,\ \ldots$, so that from (1) we obtain:

$$W_{n+1} = (W_n + D_n - (a_{n+1} - a_n))^+\ , \tag{2}$$

  which is the total delay experienced by the $n+1$-st packet that passes through the QDTP algorithm. Note that the QDPT algorithm can be implemented as software on a specific device, such as a Raspberry Pi, whose input is connected to the outside network (e.g. the Internet or an IoT network), and whose output is connected to the IoT Gateway or server.
- The second queue forms in front of the AD, which in practice is installed on the IoT Gateway or server, and the service (or attack detection time) of the AD for the nth incoming packet at a_n is denoted by $S_n \geq 0$. The packets arrive at the second queue at the instants $\{t_n,\ n = 1,\ 2,\ \ldots\}$, and leave at the instants $\{t_n + V_n\}$, where:

$$V_1 = 0\ and\ V_{n+1} = (V_n + S_n - (t_{n+1} - t_n))^+,\ n = 1,\ 2,\ \ldots\ . \tag{3}$$

Eqs. (2), (3), both have the form of the well-known Lindley's Equation [39], and provide useful insight into how the "free" parameter of the QDTP algorithm, namely D_n should be chosen.

In particular, we notice that when a flood attack takes place, the external arrivals will occur in close succession, i.e., $a_{n+1} - a_n \approx 0$ and $t_{n+1} - t_n \approx D_n$ for long sequences of packets, so that the waiting times at the first queue (the QDTP algorithm) will constantly increase. Thus, in the presence of an attack, the successive interarrival times of the second queue are the $\{D_n\}$, which should obey $D_n \geq S_n$ so that the delay in the second queue V_n remains as small as possible. However, according to Proposition 3.1, if $D_n \leq S_n$, the total end-to-end-delay of each packet $W_n + V_n$ remains unchanged by the QDTP System.

Thus, although setting $D_n = S_n$ seems to be the ideal option, this is – in practice – impossible because D_n cannot be selected on-line to match S_n, since the latter can only be measured and known **after** the nth packet is first delayed using D_n, and then forwarded to the second queue placed in front of the AD.

Thus we propose to set D_n to a fixed constant value D such that the empirically measured probability $P_A(D \leq S_n) > 1 - \epsilon,\ \epsilon > 0$, so that for a large fraction of the packets, we have $D \leq S_n$. This approach will be validated with measurements in the next Section 2.

2. Experimental results

The test-bed that we use to illustrate the QDTP algorithm, and measure the different quantities of interest, is presented in Fig. 1. It shows IoT devices that are emulated by Raspberry Pi 4 Model B Rev 1.2 (RPi1 and RPi2) computers, having 1.5 GHz ARM Cortex-A72 quad-core processors and 2GB LPDDR4 – 3200 SDRAM. They run the Raspbian the GNU/Linux 11 (bullseye) operating system. One Raspberry Pi is programmed to send packets that emulate an intense UDP flood attack in a predetermined manner, while other Raspberry PIs send ordinary UDP packets to the server containing real data about each PI's own temperature. The server itself has an Intel 8-Core i7 – 8705G processor running at 3.1 GHz with 16GB of RAM and a 500GB hard drive; it uns the Linux 5.15.0 – 60– 66–Ubuntu SMP operating system and communicates with each Raspberry Pi.

The QDTP algorithm is installed on a dedicated Raspberry PI, and designated by SQF in Fig. 2. On the other hand, the AD is installed as an application program on the server to examine all incoming packets, remove (or place in a safe buffer for further analysis) all those packets that appear to be part of a potential cyberattack, and forward for further processing only those packets that the AD decides are benign. It processes packets arriving from the QDTP algorithm in FIFO order. The AD that we use is described in [40] and studied in [41]. It is designed with the Random Neural Network [42] with auto-assistive Deep Learning, and trained to distinguish between normal (benign) and attack traffic using the FISTA optimization algorithm [43], using the MHDDoS [44] training dataset that includes DoS attacks and 56 different attack emulators.

In Fig. 2 we see the same architecture, with the important exception that it has been modified to include the QDPT algorithm implemented in software, and installed on a dedicated Raspberry Pi (designated by SQF), which is placed between the Ethernet switch and the Gateway server. Here, packets arrive at the server which houses the AD, after they have been delayed by the QDTP algorithm.

2.1. Measurements on the experimental test-bed

When the QDTP algorithm is not used, Fig. 3 reports measurements of the server's AD processing times per packet when there is no attack (figure above) and when a UDP flood attack does occur (figure below). The data shown here is based on a $10 - \text{sec}$

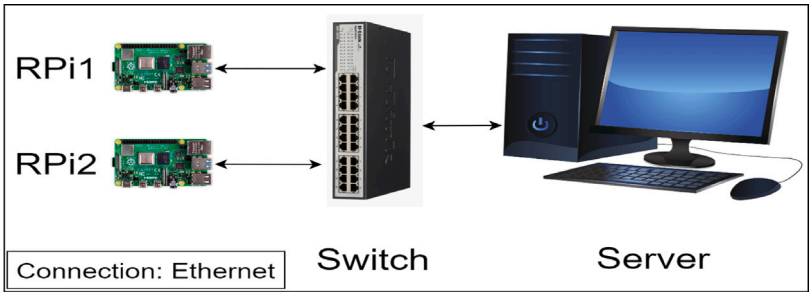

Fig. 1. The experimental test-bed shown in this figure uses Raspberry Pi machines to emulate IoT devices. These are connected via an Ethernet switch to the Gateway server. The Raspberry Pis are programmed to send both normal and flood attack traffic to the server. In this figure, the QDTP algorithm is not included in the system so that the Raspberry Pis communicate directly with the server via an Ethernet switch using the UDP protocol. Once they arrive to the Gateway, the packets will be processed by the AD in FIFO order.

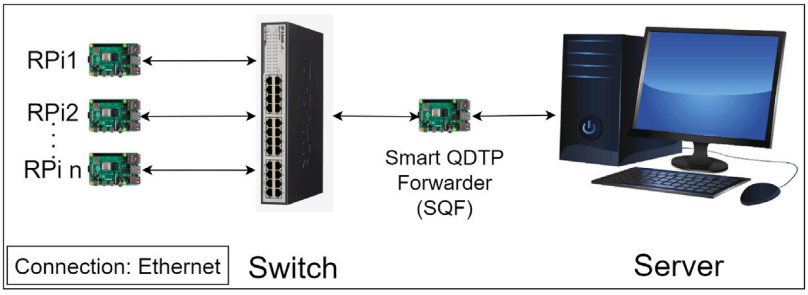

Fig. 2. Here we show the same architecture as in Fig. 1, **except** that it has been modified by placing **an additional Raspberry Pi, designated as the SQF** between the Ethernet switch and the Gateway server. The SQF supports the QDTP algorithm that is implemented in software. In this architecture, the packets sent by the IoT device emulators will traverse the Ethernet switch, then be processed in FIFO order by the QDTP algorithm, and then be forwarded to the Gateway where they are processed in FIFO order by the AD.

flood attack that launches approximately 420,000 packets against the server, i.e. with a traffic intensity of circa 42,000 packets per second, directly through the Ethernet switch against the server. The figure above shows that the average AD processing time per packet when there are **no attacks** is 2.98 ms (ms), while (below) we see that when the server is under a flood attack, the average processing time of the AD algorithm rises significantly to 4.82 ms. When the server is under attack, the AD processing time also has large outliers, as shown in the histogram in the diagram that is below in Fig. 3. Instances of these infrequent but very large outliers of the service time during the attack against the server when the QDTP algorithm is not used, are shown in Fig. 4, and they significantly exceed the average value of the service time, as indicated on a short time scale (above), and on a long time scale (below).

Fig. 5 presents the measurements of the AD processing time per packet in the form of histograms, when we use the QDTP algorithm installed in the SQF of Fig. 2, with the value $D_n = D = 2.7$ ms that is chosen based on the recommendations developed in Section 1.1. The histogram above presents the service time distribution of the AD without a flood attack, while the histogram given below concerns the measurements taken when an attack occurs. When we compare the results in the lower part of Fig. 5 with the ones in the lower part of Fig. 3, we observe that the QDPT algorithm installed on the Raspberry Pi (SQF), with $D = 2.7$ ms, is very effective in limiting the AD's slowdown during an attack. We have also plotted in Fig. 6, with a logarithmic y-axis, the queue length at the input of the AD against time in the x-axis, when the QDTP algorithm and SQF are not used (in red), and the same quantity when the QDTP algorithm and SQF are used with $D = 2.7$ ms (in blue), and we notice that the QDTP algorithm largely eliminates the queue at the AD. Obviously, since none of the packets are lost, the packets form a queue at the input of the SQF, rather than at the input of the AD.

3. The QDTP queueing model

Let us now use the notation and definitions in Section 1.1, and also define the "nominal model" to represent the case where there is no QDTP traffic shaping, and denote by L_n the nth customer's (packet's) waiting time (or queueing delay) for this case, which satisfies Lindley's Equation for the FIFO single server queue:

$$L_{n+1} = (L_n + S_n - A_n)^+, \ n \geq 0, \tag{4}$$

where $A_n = a_{n+1} - a_n$. Also define $T_n = t_{n+1} - t_n$, and recall from the details given in Section 1.1, that $T_n \geq D_n, \ n \geq 0$.

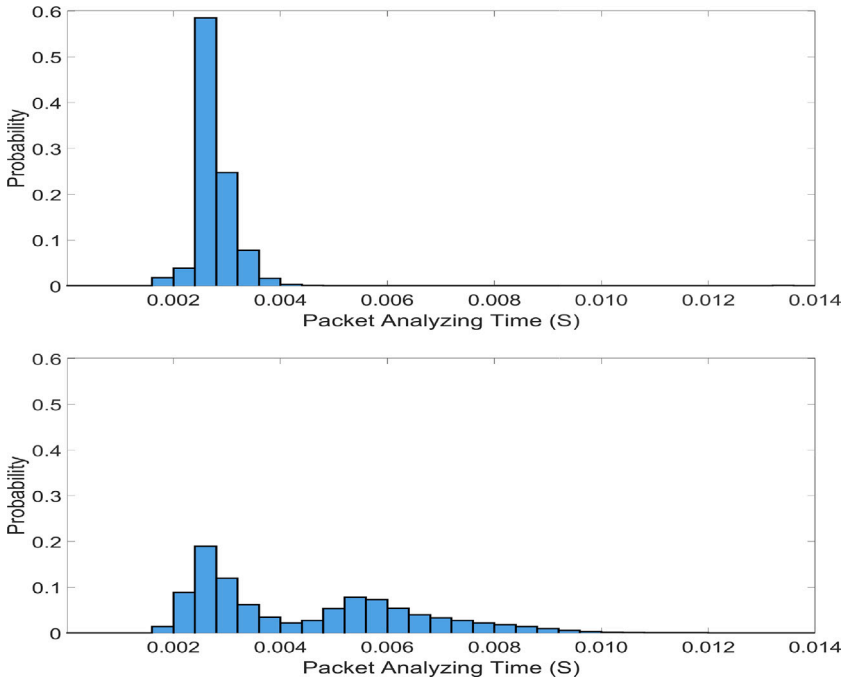

Fig. 3. Without the QDPT algorithm installed, the figure above shows the measured histogram of the AD processing time per packet, measured without an attack; its average value is 2.98 ms (ms), with a variance of $0.0055\ \text{ms}^2$. The figure given below (again without QDTP) shows the measured histogram of the AD processing time when a $10-\text{sec}$ attack occurs with 420,000 packets, and we observe that the average AD packet processing time increases to $4.82\ \text{ms}$ with a substantially higher variance of $0.51\ \text{ms}^2$.

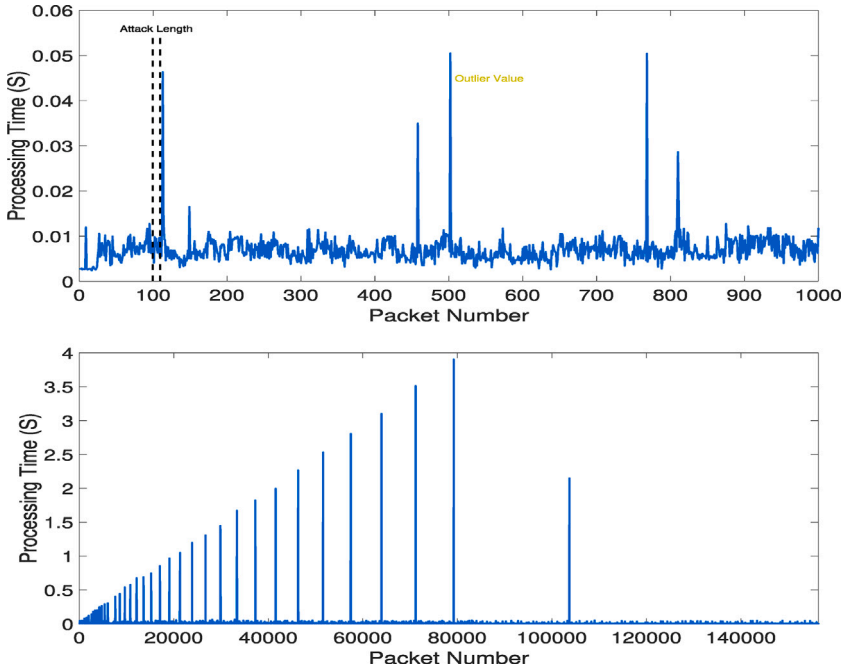

Fig. 4. Without the QDPT algorithm, the figure above shows measurements over time, on a short time scale, while below they are shown over a much longer time scale, for the AD's processing time per packet, measured during and after a flood attack that lasts 60 seconds. We notice that these very large but infrequently occurring service times can exceed the value of the average service time by several orders of magnitude.

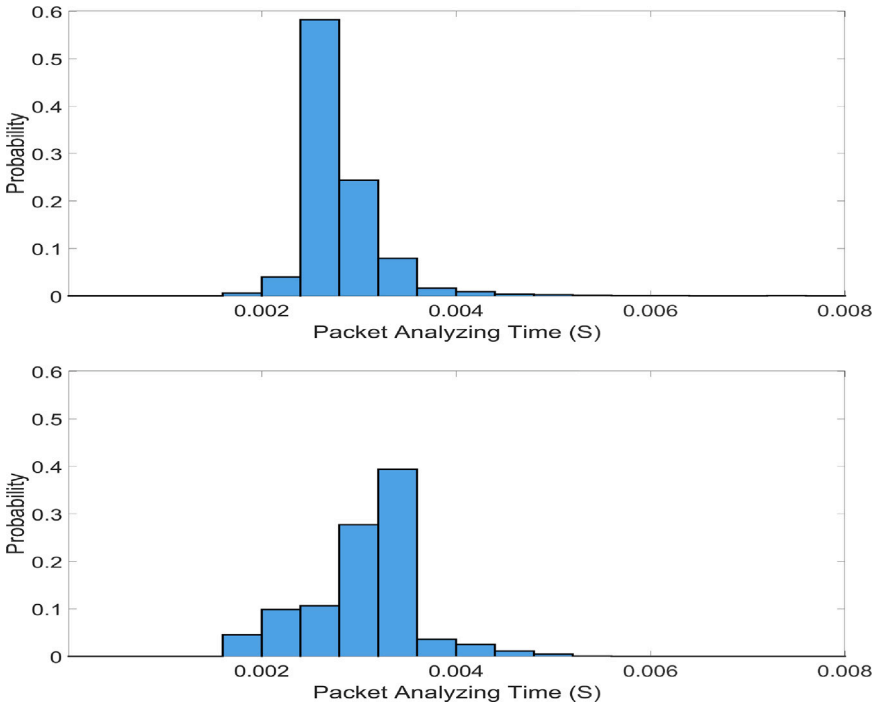

Fig. 5. This figure shows the measured histograms of the AD processing time per packet, when we use the QDTP algorithm that is installed in the SQF of Fig. 2, with the value $D_n = D = 2.7$ ms. The histogram above concerns the service time distribution of the AD without a flood attack, while the histogram given below concerns the case during a flood attack. When a flood attack does not occur (above), the average AD processing time is 2.97 ms with a variance of 0.0041 s^2. When a flood attack does occur (below), the average AD processing time grows by 10% to 3.28 ms with a variance of 0.0023 s^2. By comparison with the curve in the lower part of Fig. 3, this result shows that the QDPT algorithm installed on the Raspberry Pi (SQF), with $D = 2.7$ ms is highly effective in limiting the AD's slowdown during an attack.

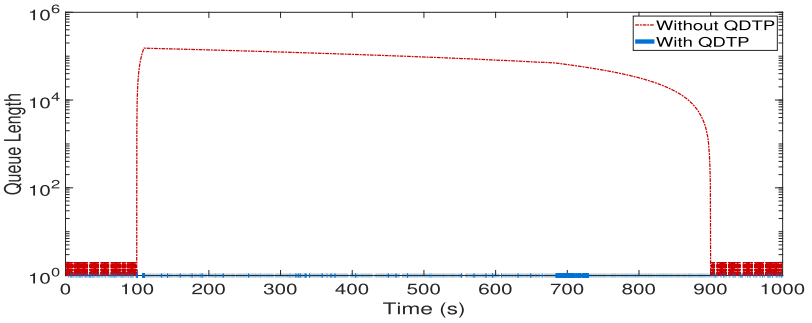

Fig. 6. The figure above shows the effect of a 42,000 per second flood attack on the AD during a 10 s flood attack. The curve in Red corresponds to the logarithmic scale packet queue length at the entrance of the AD if the SQF with QDTP algorithm is *not* used, while the Blue Curve shows the measurements of the same queue length when the SQF with QDTP is installed with $D = 2.7$ ms.

Since the D_n are strictly positive, it follows that $t_{n+1} > t_n,\ n \geq 0$, so that $\{t_n\}$ forms a *simple* point process even if $\{a_n\}$ has batches, i.e., if $a_n = a_{n+1}$ for some values of n. Also note that it is possible that $t_n = a_n$ for some values of n.

We define the total delay of the nth customer in the QDTP model as:

$$Z_n = W_n + V_n, \tag{5}$$

and its *sojourn time* as:

$$R_n = Z_n + S_n = W_n + V_n + S_n. \tag{6}$$

3.1. The sample-path properties of QDTP

Recalling (5), we see that – in principle – it is possible to control the total delay via the pair W_n, V_n, by selecting the values of the $\{D_n\}$ in (1). The following result refines and expands Theorem 1 of [33], which had proved that when $D_n \le S_n$, then $Z_n \le L_n$; instead the present paper **proves the stronger result** that $Z_n = L_n$.

Proposition 3.1. *Assume that $Z_0 = L_0$. We then have:*

(a) If $D_n \le S_n,\ n \ge 0$, then $Z_n = L_n,\ n \ge 0$, i.e. the total delay in QDTP is identical to that in the nominal FIFO $G/G/1$ model.
(b) If $D_n = S_n,\ n \ge 0$, and if $V_0 = 0$, then $Z_n = L_n = W_n,\ n \ge 0$: Every customer enters service immediately when arriving at the service facility; they spend no time delayed in the queue; all delay is spent at the café.
(c) If $D_n < S_n,\ n \ge 0$, then $Z_n = L_n,\ n \ge 0$, but for any $n \ge 1$, if $W_n > 0$ then $V_n > 0$ (equivalently if $V_n = 0$ then $W_n = 0$, i.e., $t_n = a_n$). Any customer who spends time at the café also spends time delayed in the queue; I.e., delay is shared.
(d) If $D_n > S_n,\ n \ge 0$, (and $V_0 = 0$), then: $V_n = 0,\ n \ge 0$, and thus $Z_n = W_n,\ n \ge 0$. All of the delay is spent at the café but $Z_n \ge L_n,\ n \ge 1$ with $Z_n > L_n$ if $L_n > 0$: Total delay, hence sojourn time, is increased for each customer as compared to the nominal model. (But even in this case, in some queueing applications customers might prefer spending all their delay at the café, even if it is at the expense of increasing total delay.)

Proof. For (a) it suffices (since by assumption $Z_0 = L_0$) to prove that if $Z_n = L_n$ for a given $n \ge 0$, then $Z_{n+1} = L_{n+1}$. To this end, assume that $Z_n = L_n$ for some n. Recalling (5), (2) and (3), we have:

$$\begin{aligned} Z_{n+1} &= W_{n+1} + V_{n+1} = [W_n + D_n - A_n]^+ + [V_n + S_n - T_n]^+, \\ &= [W_n + D_n - A_n]^+ + [Z_n + S_n - A_n - W_{n+1}]^+, \\ &= [W_n + D_n - A_n]^+ + [L_n + S_n - A_n - W_{n+1}]^+. \end{aligned} \tag{7}$$

We consider two cases, (A) and (B):

(A) $W_{n+1} = (W_n + D_n - A_n)^+ = W_n + D_n - A_n > 0$. Then starting with the last line of (7), using our assumption that $L_n = Z_n = W_n + V_n$, and noting that $[V_n + S_n - D_n]^+ = V_n + S_n - D_n$ if $D_n \le S_n$ yields

$$\begin{aligned} Z_{n+1} &= W_n + D_n - A_n + [L_n + S_n - A_n - W_{n+1}]^+, \\ &= W_n + D_n - A_n + [V_n + S_n - D_n]^+ \\ &= W_n + D_n - A_n + V_n + S_n - D_n, \\ &= W_n + V_n + S_n - A_n = L_n + S_n - A_n = L_{n+1}. \end{aligned}$$

(B) $W_{n+1} = [W_n + D_n - A_n]^+ = 0$. Then starting with the last line of (7) immediately yields $Z_{n+1} = [L_n + S_n - A_n]^+ = L_{n+1}$.

Thus in both cases $Z_{n+1} = L_{n+1}$, and the proof of the first assertion is complete.

For (b): Since we assume that $Z_0 = L_0$, if also $V_0 = 0$, then $0 = V_0 = Z_0 = W_0$ from (a) and so the recursions for $\{L_n\}$ and $\{W_n\}$ both start at 0 and hence yield identical processes $L_n = W_n,\ n \ge 0$. Thus from (a) it follows that $V_n = 0,\ n \ge 0$.

For (c): Suppose that $0 < W_n = W_{n-1} + D_{n-1} - A_{n-1}$. Then

$$\begin{aligned} V_n &= (V_{n-1} + S_{n-1} - A_{n-1} - W_n + W_{n-1})^+ \\ &= [V_{n-1} + S_{n-1} - D_{n-1}]^+. \end{aligned}$$

Obviously, when $S_{n-1} - D_{n-1} > 0,\ n \ge 1$, we will have $V_n = V_{n-1} + S_{n-1} - D_{n-1} > 0$.

For (d): Since $T_n \ge D_n$, an upper bound $V_n \le \overline{V}_n,\ n \ge 0$, is established by using the recursion $\overline{V}_{n+1} = (\overline{V}_n + S_n - D_n)^+,\ n \ge 0$. Thus if $D_n > S_n,\ n \ge 0$, then $S_n - D_n < 0,\ n \ge 0$, and the result $V_n = 0,\ n \ge 0$ follows. Thus $Z_n = W_n,\ n \ge 0$. But again using the assumption that $D_n > S_n,\ n \ge 0$, we obtain (by substituting each S_n for D_n in the recursion for W_n) that $W_n \ge L_n,\ n \ge 0$, and the strict inequality, $D_n > S_n$, implies that $W_n > L_n$ whenever $L_n > 0$. ■

4. A probabilistic framework

Now assume that $\{(A_n, S_n, D_n) : n \ge 0\}$ forms a (general) stationary ergodic sequence of random variables, equivalently that $\{(a_n, (S_n, D_n))\} : n \ge 0\}$, forms a point-stationary ergodic marked point process. Since the random variables are stationary, we let $A = A_0$, $S = S_0$ and $D = D_0$ denote their generic versions. We also recall that any stationary sequence of random variables $\{X_n : n \ge 0\}$ can be extended to a two-sided stationary sequence, as a standard result of probability theory that is fully discussed in Section 4.3, p. 91 and Section 6.2, p. 131 of [45].

If the **arrival rate**, $\lambda = \frac{1}{E(A)}$, is positive and finite, our next objective is to prove **stability conditions** of the QDTP model, under which one can guarantee the existence of a unique limiting distribution and an associated (proper) stationary ergodic version. As we will see over the next several sections, the first condition:

$$0 < E(D) < E(A) < \infty, \tag{8}$$

yields the stability of the first queue (or café), concerning the sequence $\{W_n\}$, while the second condition:

$$0 < E(S) < E(A) < \infty, \tag{9}$$

together with the first, constitute the necessary and sufficient conditions for the joint stability of $\{(W_n, V_n)\}$.

4.1. Stability of $\{W_n\}$

A proof of the following is based on Loynes' Lemma [46], see Pages 131–137, Lemma 6.1 and Theorem 6.1, in [45].

Proposition 4.1. *If the stability condition (8) holds, then there exists a (2-sided; $n \in \mathbb{Z}$ instead of only $n \geq 0$) jointly stationary ergodic version of $\{(W_n, A_n, D_n)\}$ denoted by $\{(W_n^0, A_n^0, D_n^0) : n \in \mathbb{Z}\}$, such that*

$$W_{n+1}^0 = (W_n^0 + D_n^0 - A_n^0)^+, \ n \in \mathrm{Z}. \tag{10}$$

As $n \to \infty$, W_n converges in total variation to the distribution of W_0^0, regardless of the initial conditions $W_0 = x \geq 0$. If $E(D) > E(A)$ then $\{W_n\}$ is unstable, i.e., $P(W_n \to \infty) = 1$.

Proposition 4.1 allows us to construct a stationary ergodic version of the point process $\{t_n\}$ with the same rate as λ as $\{a_n\}$:

Corollary 4.1. *If the stability condition (8) holds, then*

$$t_n^0 = a_n^0 + W_n^0, \tag{11}$$

defines a point-stationary ergodic version of $\{t_n\}$, that is, $T_n^0 = t_{n+1}^0 - t_n^0$ defines a stationary ergodic sequence of interarrival times. Moreover, $E(T^0) = \frac{1}{\lambda}$; and $\{t_n\}$ has rate λ, the same as $\{a_n\}$.

Proof. Defining $t_n^0 = W_n^0 + a_n^0$, so that $T_n^0 = t_{n+1}^0 - t_n^0 = A_n^0 + W_{n+1}^0 - W_n^0$ yields a stationary ergodic sequence of interarrival times, since it is a function of $\{W_n^0\}$, which has already been shown to be a stationary ergodic sequence. Thus, $\{t_n^0\}$ is a point-stationary ergodic version of $\{t_n\}$. The fact that its rate is λ follows immediately from:

$$E(T_n^0) = E(A_n^0) + E(W_{n+1}^0 - W_n^0) = \frac{1}{\lambda} + 0 = \frac{1}{\lambda}. \quad \blacksquare$$

4.2. Stability of QDTP

From Proposition 4.1 and Corollary 4.1, we replace $\{(W_n, A_n, T_n, S_n, D_n)\}$ by a two-sided stationary ergodic joint version, $\{(W_n^0, A_n^0, T_n^0, S_n^0, D_n^0)\}$ in the following total delay recursion, so that it jointly uses stationary ergodic versions of the input:

$$Z_{n+1} = (W_n^0 + D_n^0 - A_n^0)^+ + (V_n + S_n^0 - T_n^0)^+ \ n \geq 0. \tag{12}$$

The first term on the right of. (12), derived from (10), already forms a stationary ergodic sequence. We now deal with the second term. Recalling from Corollary 4.1 that $E(T_n^0) = \frac{1}{\lambda}$, and our stability condition (9), $\lambda < \mu$, we can analogously obtain, using Proposition 4.1 methods, on the second piece, a jointly stationary ergodic pair $\{(W_n^0, V_n^0) : n \in \mathrm{Z}\}$, yielding a stationary ergodic version $\{Z_n^0\}$ of $\{Z_n\}$ satisfying

$$Z_{n+1}^0 = (W_n^0 + D_n^0 - A_n^0)^+ + (V_n^0 + S_n^0 - T_n^0)^+, \ n \in \mathrm{Z}. \tag{13}$$

We can also jointly throw in $\{S_n^0\}$ to obtain a stationary ergodic sojourn time sequence via $R_n^0 = Z_n^0 + S_n^0$. Analogous to Proposition 4.1, we thus obtain:

Theorem 4.1. *For the QDTP model with stationary ergodic input satisfying the stability conditions (8) and (9), there exists a unique stationary ergodic version of total delay and sojourn time. (W_n, V_n) converges in total variation to the joint distribution of (W^0, V^0) regardless of initial conditions, and Z_n converges in total variation to the distribution of $W_0 + V_0$, regardless of initial conditions.*

4.3. Independent and identically distributed inputs: Harris recurrence and regeneration

We now focus on the special case when each of the following two input sequences, $\{A_n\}$ and $\{(S_n, D_n)\}$, are i.i.d. and independent of each other, and we will refer to the model that satisfies these assumptions as the *i.i.d. input case*. Note however, that we can allow the two random variables S_n and D_n to be dependent of each other for each n, and D_n may be chosen to be a function of S_n. Thus, we do not place restrictions on the form that this dependency may take.

Thus, in the *i.i.d. input case*, the Lindley equation for the delay W_n, i.e., $W_{n+1} = (W_n + D_n - A_n)^+$, implies that $\{W_n : n \geq 0\}$ is a Markov chain.

Since $T_n = t_{n+1} - t_n = A_n + W_{n+1} - W_n$ we can re-write the recursion for $\{V_n\}$ by using the Markov chain $\{W_n\}$ to drive it:

$$V_{n+1} = (V_n + S_n - T_n)^+, \tag{14}$$

$$= \Big(V_n + S_n - A_n - (W_{n+1} - W_n)\Big)^+, \tag{15}$$

$$= \Big(V_n + S_n - A_n - ((W_n + D_n - A_n)^+ - W_n)\Big)^+. \tag{16}$$

Focusing on (16), and recalling the i.i.d. assumptions, it follows that for $M_n \stackrel{\text{def}}{=} (W_n, V_n)$,

$$\{M_n : n \geq 0\}, \text{ forms a Markov chain on } \mathbb{R}_+^2. \tag{17}$$

We will next show that the Markov chain $\{M_n\}$ is Harris ergodic. For basics on Harris recurrence and ergodicity, we refer to Chapter VII, Section 3, including Proposition 3.13, p. 205, of [47]. A key feature of Harris recurrent Markov chains is that **they always form regenerative processes**. Therefore in Proposition 4.3 we explicitly find two different kinds of regeneration points, **Type I** and **Type II**, which are exhaustive and cover all the ground. Type 1 visits are visits to the empty state, while Type 2 are more elaborate.

Proposition 4.2. *For the i.i.d. input case that satisfies the stability conditions (8) and (9), the Markov chain $M_n = (W_n, V_n)$ is Harris ergodic.*

Proof. From Theorem 4.1, $\{M_n\}$ is ergodic and converges in total variation to a limiting stationary probability distribution π, regardless of initial conditions on M_0. Thus for $A \subset \mathbb{R}_+^2$, if $\pi(A) > 0$, then regardless of initial conditions, by ergodicity, $\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^n I\{M_i \in A\} = \pi(A) > 0$, with probability one. Therefore, A is visited infinitely often. Thus π serves as a recurrence measure, while $\{M_n\}$ is positive Harris recurrent by definition. ■

To proceed further, we need an important Lemma:

Lemma 4.1. *When the stability conditions (8) and (9) hold, then either of the following conditions must be true:*

$$\textbf{Type 1}: \quad P(A > \max\{S, D\}) > 0, \tag{18}$$

or

$$\textbf{Type 2}: \quad P(D > S) > 0. \tag{19}$$

Note: *A natural sufficient condition for obtaining (18) is that the interarrival time distribution has unbounded support, i.e., $P(A > x) > 0,\ x \geq 0$.*

Proof. If (18) does not hold, then (19) must hold, for if it did not, then $P(D \leq S) = 1$ implying that $S = \max\{S, D\}$, and thus (18) is equivalent to $P(A > S) > 0$ which indeed holds from the stability condition (9); we get a contradiction. ■

Proposition 4.3. *Assume the stability conditions, (8) and (9). We then obtain the following result:*

1. **Type I Regeneration***: If (18) also holds, then the successive times when $M_n = (0, 0)$ can be chosen as positive recurrent regeneration points. In particular, total delay, $Z_n = W_n + V_n$, forms a positive recurrent regenerative process, with visits to state 0.*
2. **Type II Regeneration***: If (18) does not hold, then (19) does hold (by Lemma 4.1) and in this case positive recurrent regeneration points can be found for $\{M_n\}$ of the form (in distribution upon regeneration) $(X, 0)$ where the construction of the random variable X is given explicitly below in Algorithm Algorithm 4.1.*

Proof (*Type I Regeneration*)**.** Since the recursion for $\{W_n\}$ describes a stable $GI/GI/1$ queue, $P_\pi(W_0 = 0) > 0$. Thus there exists a $B > 0$ such that $P_\pi(W_0 = 0,\ V_0 \leq B) > 0$. By Harris recurrence, the event $\{W_n = 0, V_n \leq B\}$ thus occurs infinitely often and does so a positive proportion of time. For a fixed sufficiently small $\delta > 0$, the assumed (18) implies $p = P(A_n > \max\{S_n, D_n\} + \delta) > 0$. If we define $k = \lceil B/\delta \rceil$ (the smallest integer $\geq B/\delta$), and define the event $F_n^k = \{A_{n+i} > \max\{S_{n+i}, D_{n+i}\} + \delta,\ 0 \leq i \leq k-1\}$, and whenever the event $\{W_n = 0, V_n \leq B\}$ occurs, the event F_n^k is independent of it and will occur with probability $p^k = P(F_n^k) > 0$.

Using (14), suppose that for some n, both events $\{W_n = 0, V_n \leq B\}$, and F_n^k occur. Then since $W_{n+1} = (W_n + D_n - A_n)^+$, we conclude that $W_{n+i} = 0,\ 0 \leq i \leq k$, implying that:

$$V_{n+1} = \Big(\ V_n + S_n - A_n - (W_{n+1} - W_n)\ \Big)^+ = (V_n + S_n - A_n)^+ \leq \Big(B - \delta\Big)^+.$$

We can continue in step-by-step fashion to obtain:

$$V_{n+2} \leq (B - 2\delta)^+, \ldots, V_{n+k} \leq (B - k\delta)^+ = 0,$$

so that we have $W_{n+k} = V_{n+k} = 0$. Since, by the Borel–Cantelli Lemma, the event $\{W_n = 0,\ F_n^k\}$ occurs infinitely often with a positive proportion of times $\geq p^k P_\pi(W_n = 0, V_n \leq B) > 0$, the regenerative cycle length distribution is aperiodic: given that $M_n = 0$, there is a positive probability $P(A_n > \max\{S_n, D_n\})$, that $M_{n+1} = 0$ as well. Thus, the proof of Type I regeneration is complete.

Proof Type II Regeneration. First, note that since $T_n \geq D_n,\ n \geq 0$, we have $V_{n+1} = (V_n + S_n - T_n)^+ \leq (V_n + S_n - D_n)^+,\ n \geq 0$. We thus define a new upper bound process $\{\hat{V}_n\}$ by using the recursion

$$\hat{V}_{n+1} = (\hat{V}_n + S_n - D_n)^+,\ n \geq 0, \tag{20}$$

for which it follows that

$$V_n \le \hat{V}_n,\ n \ge 0,\ \text{if } V_0 = \hat{V}_0. \tag{21}$$

Now choose $B > 0$ sufficiently large so that $P_\pi(W_0 = 0,\ V_0 \le B) > 0$ which implies the event $\{W_n = 0,\ V_n \le B\}$ will happen infinitely often. Choose a $\delta > 0$ such that $P(D > S + \delta) > 0$. Define $k = \lceil B/\delta \rceil$, and $F_n^k = \{\{D_{n+i} > S_{n+i} + \delta\},\ 0 \le i \le k-1\}$. Now suppose that for some n, both the events $\{W_n = 0,\ V_n \le B\}$ and F_n^k occur. Then similar to the proof of Proposition 4.2 (we use (20) and (21) and set $\hat{V}_n = V_n$), we have $\hat{V}_{n+k} = 0$ and hence $V_{n+k} = 0$.

Meanwhile, the random variable $X = W_{n+k}$ was constructed from only i.i.d. $\{(D_{n+i}, A_{n+i}) : 0 \le i \le k-1\}$, *conditional on* F_n^k, and is independent of all else; that is how M_n regenerates; next we give a more explicit algorithm for the construction of such as X. ■

Algorithm 4.1.

1. *Let* $\{(S_i, D_i) : 0 \le i \le k-1\}$ *denote* k *i.i.d. pairs conditional on each pair satisfying* $F_0^k = \{D_i > S_i + \delta\},\ 0 \le i \le k-1$.
2. *Also, let* $\{A_i : 0 \le i \le k-1\}$ *be i.i.d.*
3. *Use as input* $\{A_i, D_i : 0 \le i \le k-1\}$ *(starting with* $W_0 = 0$*) in the recursion* $W_{n+1} = (W_n + D_n - A_n)^+,\ 0 \le n \le k-1$.
4. *Set* $X = W_k$. *Then when a regeneration occurs for* $\{M_n\}$ *at a time* $n + k$, *it is distributed as* $(X, 0)$.

4.4. Some mathematical examples

In Proposition 4.3, the stability conditions imply $P(A > S) > 0$ and $P(A > D) > 0$ but are not strong enough to imply $P(A > \max\{S, D\}) > 0$, when S and D are dependent. Individually, each of $\{V_n\}$ and $\{W_n\}$ will empty infinitely often, a positive proportion of times; but in general, they do not do so at the same time n; hence the need to derive more involved regeneration points in such a case. We illustrate here with a counterexample. Choose $P(A = 2.6) = 1$ and set $(S, D) = (2, 3)$ w.p. 0.5, and $(S, D) = (3, 2)$ w.p. 0.5. Then $P(A > S) = P(S = 2) = 0.5$, and $P(A > D) = P(D = 2) = 0.5$. But $P(A > \max\{S, D\}) = P(A > 3) = 0$.

To see that $M_n \ne (0, 0)$ for $n > 0$, we will show that W_n and V_n move/alternate in opposite directions. Suppose $W_{n+1} - W_n \le 0$ for some n which can happen only when $(S_n, D_n) = (3, 2)$. Then $T_n = 2.6 + W_{n+1} - W_n \le 2.6$ and thus $V_{n+1} = (V_n + 3 - T_n)^+ \ge (V_n + .4)^+ = V_n + .4$; hence $V_{n+1} - V_n \ge 0.4$. Thus if $W_{n+1} - W_n \le 0$, then $V_{n+1} - V_n > 0$, and if $V_{n+1} - V_n \le 0$, then $W_{n+1} - W_n > 0$; $M_n \ne (0, 0)$ for $n > 0$.

To explicitly characterize the regeneration points of Type II, we choose any $b > 0$ such that $P_\pi(W_0 = 0, V_0 \le b) > 0$, then find the smallest such b. Supposing that the event $\{W_n = 0, V_n \le b\}$ occurs, one can then condition on alternating $\{(S_{n+i}, D_{n+i}) : 0 \le i \le m-1\} = \{(2, 3), (3, 2), (2, 3), \dots, (3, 2)\}$, for any length m, which occurs with positive probability $(1/2)^m$. Thus, $W_{n+1} = 0.4, W_{n+2} = 0, W_{n+3} = 0.4, \dots$, alternating between 0.4 and 0. When $T_{n+i} = 3$ for even i and $T_{n+i} = 2.2$ for odd i, then V_{n+i} goes down by 1 and up by 0.8 until we have $V_{n+i} = 0$ for some i. If $V_{n+i} = 0$, we must have $W_{n+i} = 0.4$ (since $M_n \ne (0, 0)$ for $n > 0$), and hence $P_\pi(W_0 = 0, V_0 = 0.4) > 0$ holds. Thus, as regeneration points we can take those consecutive times n such that $M_n = (0.4, 0)$.

Another example of this phenomenon is the classic FIFO $GI/GI/c$ queue with $c \ge 2$, which can be stable, but where an arrival may never find it to be empty. Indeed, a necessary condition for it to be empty when an arrival occurs is $P(A > S) > 0$. Indeed, when $c = 2$, if one takes $A_n = 1.5,\ n \ge 0,\ S_n = 2,\ n \ge 0$, then $\rho = \lambda/\mu = 4/3 < 2$, so stability holds, and all arriving customers for $n > 0$ will find one server free, but the other server will be busy. Nonetheless, for any stable ($\rho < c$) FIFO $GI/GI/c$ queue, regeneration points can be found as shown in Chapter 7, Section 2, Page 344 in [47]. For another classic example, see [48].

5. Conclusions

This paper briefly surveys the Massive Access Problem, which is caused by the proliferation of IoT devices and the congestion that they cause, and by the congestion caused by frequent cyberattacks against IoT networks and Gateways. We also survey solutions that have been proposed in the literature to these problems. Then, in Section 1.1 we recall the Quasi-Deterministic Transmission Policy (QDTP) for traffic shaping at the entrance of IoT Gateways to mitigate the MAP, by delaying in a bounded manner the arrival times of incoming packets, and develop the basic equations that characterize the resulting system. We also address the choice of the key delay parameter of the algorithm, and suggest a simple heuristic for this choice.

Since the QDTP algorithm may be installed on a special low-cost processor such as a Raspberry Pi to protect the Gateway servers that is subject to cyberattacks, the server will typically support an AD algorithm and software to detect potential attacks by processing the incoming packet sequence. We first present experiments which show that the AD processing time itself may be significantly increased (or even stalled) by an incoming packet flood due to the higher priority operating system software which is handling the large number of incoming packets. We also study the effect of the QDTP algorithm's critical delaying parameter through several measurements, when it is set to a deterministic value D, which is slightly smaller than the AD's normal average processing time per packet, to guarantee that there is no congestion directly in front of the AD, allowing it to operate in a timely manner. This choice also allows the SQF, (that hosts the QDTP algorithm) to forward the incoming packets to the AD smoothly, and empty the large packet queue that builds in front of the SQF at a fixed rate of D^{-1}, when an attack occurs. We also demonstrate the practical value of QDTP through experiments on the test-bed to, show that without QDTP and the same flood attack, a huge queue forms at the Gateway server, and a significant slowdown occurs in the AD's useful operations.

Then, assuming that the characteristic delay of QDTP does not exceed the AD service time, we prove that the end-to-end-delay of QDTP is exactly identical to that of a First-in-First-Out conventional server, showing that QDTP is useful in modifying the arrival process of packets into a Gateway server in a manner that reduces significantly the server congestion without modifying the end-to-end delay of the packets. We also analyze the QDTP queueing system by assuming a stationary stochastic process that characterizes its interarrival, QDTP delay and service times, and obtain the relevant stability conditions. This is followed by an analysis based on "independent and identically distributed" assumptions that analyze the conditions for stability and recurrence and lead to a Harris recurrent Markov chain.

In future work, we will investigate adaptive algorithms for updating D_n as a function of prior values of the AD service times $S_k,\ k = 1, \ldots, n-1$ and of the arrival rate of packets to the system. We will also investigate additional algorithms that can accompany the QDTP policy, such as optimum packet dropping during flood attacks, and congestion control to minimize the latency and loss of benign traffic.

CRediT authorship contribution statement

Jacob Bergquist: Writing – original draft, Investigation, Formal analysis. **Erol Gelenbe:** Writing – review & editing, Writing – original draft, Validation, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Mohammed Nasereddin:** Validation. **Karl Sigman:** Writing – original draft, Project administration, Methodology, Investigation, Formal analysis.

Declaration of competing interest

The authors have no conflicts of interests to declare.

Acknowledgment

The research in this paper has been supported in part by the European Commission's H2020 DOSS Project under Grant Agreement No. 101120270.

References

[1] L. Sujay-Vailshery, Number of IoT Connections Worldwide 2022–2033, STATISTA, [Online]. Available. URL https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.
[2] Cisco, Cisco Annual Internet Report (201820132023) White Paper. [Online]. Available. URL https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.
[3] F. Ghavimi, H.-H. Chen, M2M communications in 3GPP LTE/LTE-A networks: Architectures, service requirements, challenges, and applications, IEEE Commun. Surv. Tutor. 17 (2) (2015) 525–549.
[4] A. Zanella, M. Zorzi, A.F. dos Santos, P. Popovski, N. Pratas, C. Stefanovic, A. Dekorsy, C. Bockelmann, B. Busropan, T.A.H.J. Norp, M2M massive wireless access: Challenges, research issues, and ways forward, in: 2013 IEEE Globecom Workshops (GC Wkshps), 2013, pp. 151–156.
[5] E. Gelenbe, E. Ngai, Adaptive random re-routing for differentiated QoS in sensor networks, Comput. J. 53 (7) (2010) 1052–1061.
[6] S.-Y. Lien, T.-H. Liau, C.-Y. Kao, K.-C. Chen, Cooperative access class barring for machine-to-machine communications, IEEE Trans. Wirel. Commun. 11 (1) (2012) 27–32.
[7] T.-M. Lin, C.-H. Lee, J.-P. Cheng, W.-T. Chen, PRADA: Prioritized random access with dynamic access barring for MTC in 3GPP LTE-a networks, IEEE Trans. Veh. Technol. 63 (5) (2014) 2467–2472.
[8] M. Nakip, E. Gelenbe, Randomization of data generation times improves performance of predictive IoT networks, in: 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, la, USA, 2021, pp. 350–355.
[9] A. Aijaz, A.H. Aghvami, Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective, IEEE Internet Things J. 2 (2) (2015) 103–112.
[10] A. Aijaz, S. Ping, M.R. Akhavan, A.-H. Aghvami, CRB-mac: A receiver-based MAC protocol for cognitive radio equipped smart grid sensor networks, IEEE Sensors J. 14 (12) (2014) 4325–4333.
[11] I. Park, D. Kim, D. Har, MAC achieving low latency and energy efficiency in hierarchical M2M networks with clustered nodes, IEEE Sensors J. 15 (3) (2015) 1657–1661.
[12] L. Liang, L. Xu, B. Cao, Y. Jia, A cluster-based congestion-mitigating access scheme for massive M2M communications in Internet of Things, IEEE Internet Things J. 5 (3) (2018) 2200–2211.
[13] V. Rodoplu, M. Nakip, R. Qorbanian, D.T. Eliiyi, Multi-channel joint forecasting-scheduling for the internet of things, IEEE Access 8 (2020) 217324–217354.
[14] S.-Y. Lien, T.-H. Liau, C.-Y. Kao, K.-C. Chen, Cooperative access class barring for machine-to-machine communications, IEEE Trans. Wirel. Commun. 11 (1) (2012) 27–32.
[15] Y.-C. Pang, S.-L. Chao, G.-Y. Lin, H.-Y. Wei, Network access for M2M/H2H hybrid systems: A game theoretic approach, IEEE Commun. Lett. 18 (5) (2014) 845–848.
[16] H. Jin, W.T. Toor, B.C. Jung, J.-B. Seo, Recursive pseudo-Bayesian access class barring for M2M communications in LTE systems, IEEE Trans. Veh. Technol. 66 (9) (2017) 8595–8599.
[17] J. Liu, et al., A novel congestion reduction scheme for massive machine-to-machine communication, IEEE Access 5 (2017) 18765–18777.
[18] L. Tello-Oquendo, Performance analysis and optimal access class barring parameter configuration in LTE-A networks with massive M2M traffic, IEEE Trans. Veh. Technol. 67 (4) (2018) 3505–3520.
[19] J. Du, et al., Contract design for traffic offloading and resource allocation in heterogeneous ultra-dense networks, IEEE J. Sel. Areas Commun. 35 (11) (2017) 2457–2467.
[20] N. Li, et al., Cooperative wireless edges with composite resource allocation in hierarchical networks, in: 2020 IEEE International Conference on E-Health Networking, Application and Services, HEALTHCOM, 2021, pp. 1–6.
[21] E. Gelenbe, K.C. Sevcik, Analysis of update synchronization for multiple copy databases, IEEE Trans. Comput. C-28 (10) (1979) 737–747.

[22] V. Petkov, K. Obraczka, Collision-free medium access based on traffic forecasting, in: 2012 IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEEXplore, 2012, pp. 1–9.
[23] Y. Edalat, J.-S. Ahn, K. Obraczka, Smart experts for network state estimation, IEEE Trans. Netw. Serv. Manag. 13 (3) (2016) 622–635.
[24] D. Raca, et al., On leveraging machine and deep learning for throughput prediction in cellular networks: Design, performance, and challenges, IEEE Commun. Mag. 58 (3) (2020) 11–17.
[25] D.O. Awduche, MPLS and traffic engineering in IP networks, IEEE Commun. Mag. 37 (12) (1999) 42–47.
[26] IETF, An architecture for differentiated services: ETF RFC 2475, 1988.
[27] ITU, Traffic control and congestion control in B-ISDN: ITU-T Recommendation I.371.
[28] Cisco, Comparing Traffic Policing and Traffic Shaping for Bandwidth Limiting: Cisco Tech. Notes, Document ID: 19645, August 2005.
[29] H. Brandt, ATM, John Wiley and Sons, Chichester, England, 2001.
[30] C. Barakat, E. Altman, W. Dabbous, On TCP performance in a heterogeneous network: A survey, IEEE Commun. Mag. 38 (1) (2000) 40–46.
[31] J. Helzer, L. Xu, Congestion control for multi-media streaming with self-limiting sources, in: 13th IEEE International Conference on Network Protocols ICNP'05, Nov. 6-9, Boston, MA., 2005, [Online]. Available. URL http://csr.bu.edu/icnp2005/posters/helzer.pdf.
[32] E. Gelenbe, M. Nakip, D. Marek, T. Czachorski, Diffusion analysis improves scalability of IoT networks to mitigate the massive access problem, in: 2021 MASCOTS International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems, November 3-5., EEEX Press, Houston, Texas USA, 2021, pp. 182–189.
[33] E. Gelenbe, K. Sigman, IoT traffic shaping and the massive access problem, in: IEEE International Conference on Communications, 2022, pp. 2732–2737, [Online]. Available. URL https://zenodo.org/records/6349552.
[34] IoT traffic generation pattern dataset, 2021, [Online]. Available. URL https://www.kaggle.com/tubitak1001118e277/iot-traffic-generation-patterns.
[35] E. Gelenbe, M. Nakip, T. Czachórski, Improving massive access to IoT gateways, Perform. Eval. 157–158 (2022) 102308, [Online]. Available. URL https://www.sciencedirect.com/science/article/pii/S0166531622000219.
[36] J. Bergquist, E. Gelenbe, K. Sigman, On an adaptive-quasi-deterministic transmission policy queueing model, in: IEEE Computer Society, MASCOTS 2024, 2024, pp. 1–7, [Online]. Available. URL https://zenodo.org/records/13860521.
[37] E. Gelenbe, M. Nasereddin, Adaptive attack mitigation for iov flood attacks, IEEE Internet of Things Journal 12 (5) (2025) 4701–4714.
[38] E. Gelenbe, M. Nasereddin, Data driven optimum cyberattack mitigation, in: IEEE DSAA'25: Data Science and Advanced Analytics Conference, Birmingham, UK, 2025, pp. 1–8.
[39] E. Gelenbe, I. Mitrani, Analysis and Synthesis of Computer Systems, Second Ed., World Scientific, Imperial College Press, 2010, p. 324.
[40] O. Brun, et al., Deep learning with dense random neural networks for detecting attacks against IoT-connected home environments, in: Security in Computer and Information Sciences: First International ISCIS Security Workshop 2018, Euro-CYBERSEC 2018, London, UK, February 26-27, 2018, Revised Selected Papers, CCIS, vol. 821, Springer International Publishing, 2018, pp. 79–89.
[41] E. Gelenbe, M. Nakip, Traffic based sequential learning during botnet attacks to identify compromised IoT devices, IEEE Access 10 (2022) 126536–126549.
[42] E. Gelenbe, Random neural networks with negative and positive signals and product form solution, Neural Comput. 1 (4) (1989) 502–510.
[43] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (2009) 183–202.
[44] MHDDoS - DDoS Attack Script With 56 Methods, 2022, [Online]. Available. URL https://github.com/MatrixTM/MHDDoS. (Accessed 22 February 2023).
[45] K. Sigman, Stationary Marked Point Processes: An Intuitive Approach, Chapman and Hall, CRC Press, London, UK, 1995.
[46] R. Loynes, The stability of queues with nonindependent inter-arrival and service times, Proc. Camb. Philos. Soc. 58 (1962) 497–520.
[47] S. Asmussen, Applied Probability and Queues, second ed., Springer, 2003.
[48] K. Sigman, Regeneration in tandem queues with multi-server stations, J. Appl. Probab. 25 (1988) 391–403.