2018 IEEE Symposium on Service-Oriented System Engineering

Traffic Differentiation on Internet of Things

Thiago Garrett*, Schahram Dustdar†, Luis C. E. Bona* and Elias P. Duarte Jr.*
*Federal University of Paraná, Brazil
Emails: {tgarrett,bona,elias}@inf.ufpr.br
†TU Wien, Austria
Email: dustdar@infosys.tuwien.ac.at

***Abstract*—The Internet of Things (IoT) is expected to constitute a significant portion of the Internet in the future, both in terms of traffic, and market share. For it to achieve its full potential, innovative solutions are necessary to address several open challenges. In this context we discuss Network Neutrality, which states that all traffic in the Internet must be treated equally, i.e., without traffic differentiation (TD). Unfair traffic management may result in a non-competitive market, affecting selectively the quality of experience of different IoT applications. This scenario might hinder innovation, threatening IoT success. Monitoring TD on the IoT is thus important for a more competitive market. In this paper, we first study the impact of TD on common IoT traffic patterns, such as periodic updates and real-time notifications. We present simulation results, and discuss which types of IoT applications are most affected by TD. We then discuss a solution for monitoring TD on IoT. The solution takes advantage of the IoT to address several open challenges of TD detection. For instance, the large amount of devices results in a prolific environment for making TD-related measurements. The solution can thus employ machine learning for continuously monitoring TD as the numerous IoT devices and applications communicate.**

1. Introduction

The Internet of Things (IoT) is becoming increasingly present in modern life. It consists of a combination of numerous connected devices, sensors and actuators, that gather huge amounts of data and provide different services. Estimates show there will be about 212 billion IoT devices by 2020, and about 45% of Internet traffic will be related to IoT by 2022 [1]. These estimates indicate that IoT will constitute a significant portion of the Internet in the future, both in terms of traffic, and market share. The rapidly growth of IoT will most certainly have a high economic impact in several areas, providing device manufacturers, Internet Service Providers (ISPs), and application developers with new opportunities in the market [1].

There are currently several research projects covering key aspects of IoT [1], such as architectures, availability, reliability, mobility, performance, management, scalability, interoperability, security, and privacy. Innovation is thus essential in order to address the large set of challenges presented by the IoT. New devices, protocols, platforms, and cloud services are examples of different aspects that still need innovative solutions if the IoT is to achieve its full potential, contributing to quality of life and economy growth.

However, innovation, and thus the success of IoT, may be hindered by unfair traffic management practices from ISPs [2]. In this context we discuss Network Neutrality (NN), which is the principle by which all traffic on the Internet must be treated equally. According to NN, an ISP cannot slow down, prioritize or block any specific type of traffic, regardless of its origin, destination and/or content, i.e., traffic differentiation (TD) practices are not allowed [3]. TD may impact selectively the quality of experience (QoE) of different IoT applications, resulting in a non-competitive market, since a difference in QoE may determine the success or failure of a device or application over competitors [4], [5]. For instance, if IoT sensors from one manufacturer have their traffic prioritized, the potential lower loss rates experienced by these sensors might cause much less packet retransmissions. This scenario could result in lower energy consumption, giving the manufacturer a competitive advantage.

On a non-neutral Internet, new innovative devices, applications, or services from small companies may not be able to compete with more established products from larger companies [6]. For instance, device manufacturers and application developers may be compelled to pay extra fees for ISPs in order to have their IoT applications and devices run efficiently or at least on par with competitors [7]. Furthermore, ISPs may employ TD to prioritize traffic from/to specific partner cloud vendors, making their IoT services and platforms more attractive. ISPs may also prioritize their own IoT-related services in order to obtain competitive advantages. Checking for these behaviors in the IoT is thus important to ensure that innovative solutions emerge, addressing the several open challenges and providing a more diverse set of services in the future.

NN has been globally debated for more than a decade, leading several governments to create regulations forbidding ISPs to employ TD [8]. However, regulations alone cannot guarantee ISPs compliance. Moreover, transparency on traffic management practices might contribute to a more competitive market [2]. Therefore, regardless of regulations,

0-7695-6394-5/18/$31.00 ©2018 IEEE
DOI 10.1109/SOSE.2018.00026

monitoring TD on the IoT is important for ensuring a level playing field for the development of new protocols, devices, and applications.

Some solutions for detecting TD on the Internet have been published in the last decade in the scientific literature [9], [10], [11], [12], [13], [14], [15], [16], [17]. These solutions are based on network measurements and statistical inference. However, several issues arise when employing these solutions on IoT. For instance, they focus on traditional Internet traffic, the so-called Human-Type Communication (HTC), and thus might not be effective for detecting TD of IoT traffic. Furthermore, limitations present on IoT devices may turn unfeasible the use of these solutions.

In this paper, we first study how TD may impact IoT traffic. We discuss how TD may be implemented by ISPs to discriminate IoT devices, applications, and services. We then present common traffic patterns generated by IoT applications and how they might be affected by TD. Simulation results of each traffic pattern under different TD scenarios are then presented. We then discuss a solution for monitoring TD on IoT. This proposal takes advantage of the IoT infrastructure to address several open challenges identified in the current state of the art. It is based on continuous passive measurements and machine learning [18]. The main idea is to passively monitor IoT traffic, in order to establish the "default network performance" of different IoT traffic patterns. If the perceived performance of the traffic from an IoT device or application differs from this baseline, TD may have occurred. To the best of our knowledge, there is currently no solution for monitoring the presence of TD on IoT.

The rest of this paper is organized as follows. In Section 2 we discuss how ISPs may discriminate traffic on the IoT. Traffic patterns generated by IoT applications and how TD may impact these patterns are then described in Section 3. We describe our simulations and discuss the results in Section 4. We then discuss a solution for monitoring TD on IoT in Section 5. Related work is presented in Section 6. We conclude the paper in Section 7.

2. Traffic Differentiation on IoT

Traffic differentiation (TD) is a discriminatory traffic management practice, in which some types of traffic are treated differently than others. An ISP may implement this by prioritizing or slowing down specific types of traffic traversing its network. To identify these different types, traffic may be classified based on its characteristics [19], [20], such as origin/destination address, destination port, application protocol, flow behavior, or even the whole payload (deep packet inspection). Traffic is then treated differently based on the classification, through several possible mechanisms. Traffic shaping [21] and traffic policing [22] are common traffic management mechanisms which may be used to implement TD. We further describe these mechanisms in Section 4, when the TD scenarios employed in our simulations are presented.

IoT devices usually generate small amounts of traffic [23], but the aggregate traffic from billions of such devices may motivate ISPs to throttle IoT traffic and/or charge for prioritization in the future. TD may take place at different levels of the IoT architecture, affecting different components as they interact with each other. These components connect to the Internet through several different access networks. Therefore, in this work we make no distinction between wired and wireless networks, since TD may be employed on both, equally affecting IoT traffic.

An IoT system is a combination of connected components with different capabilities and purposes, from low-capacity sensors to high-performance cloud servers. In general, a plethora of sensing devices generate raw data, which is sent to the Cloud for processing and storage. Cloud services may also send data to the edge devices, such as commands and notifications. IoT platforms are often implemented as middleware [24] coordinating the interaction between the heterogeneous edge devices and cloud services. There could also be IoT gateways which aggregate data from several sensing devices, intermediating the communication between them and IoT platforms or other cloud services.

Figure 1 shows a common IoT architecture and shows where TD may take place. Any traffic that traverses the Internet is subject to TD: from/to edge devices or gateways, from/to IoT platforms, as well as from/to cloud services. An ISP may differentiate traffic involving specific device manufacturers (e.g., a brand of sensors or vehicles), applications (e.g., domain-specific or proprietary protocols), or origin/destinations (e.g., premium clients, cloud vendors, IoT platforms).

3. IoT Traffic Patterns and the Impact of TD

Data traffic in the IoT is mostly comprised of the so-called Machine-Type Communication (MTC) [1] – also known as Machine-to-machine (M2M) communication. MTC is characterized by the communication of several devices among each other without the need of human interaction, as opposed to Human-type communication (HTC). In the IoT, sensing devices, gateways, middleware, and cloud services communicate autonomously with each other. Human interaction is also present, such as command inputs (e.g., smart home), or during critical situations (e.g., decision making, security alarms).

MTC traffic is usually comprised of short and sparse transmissions of small packets [25]. It may be real-time or not, with varying intervals between transmissions. HTC, on the other hand, is characterized by a continuous flow of large packets. In access networks, MTC generates traffic that is predominantly in the upload direction (from sensing devices to the Cloud), while HTC generates traffic that is mostly in the download direction (from the Cloud to end-users devices). Examples of HTC traffic include instant messaging, VoIP, video/audio streaming, web pages, and file sharing. We present below, in Subsection 3.1, common IoT traffic patterns. We then discuss, in Subsection 3.2, how TD might impact the different traffic patterns.

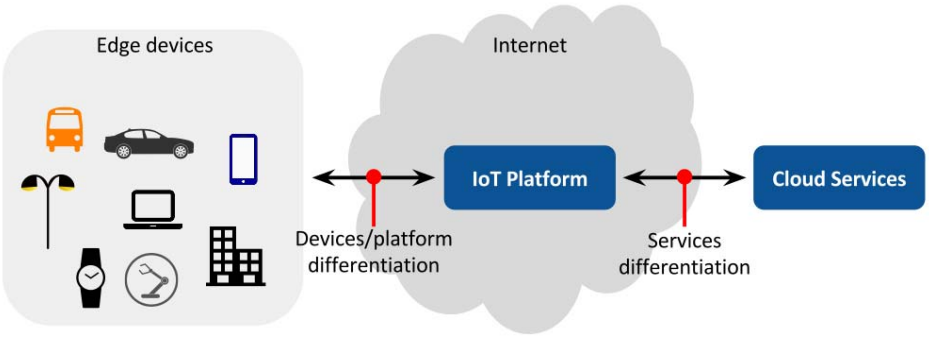


Figure 1. TD on a common IoT architecture.

3.1. IoT Traffic Patterns

Three common MTC traffic patterns have been recently identified [25]: Periodic Update (PU), Event-Driven (ED), and Payload Exchange (PE). According to the authors, these patterns are those observed in the majority of M2M applications. IoT applications are often comprised of a combination of these patterns. For instance, a river monitoring system periodically sends measurements regarding the water level of rivers to a central server (PU). If the water level surpasses a certain threshold, a flood alarm may be issued (ED). In order to deal with this event, pictures may be sent to the server, or a data stream may be initiated to update the measurements in real-time (PE). The system may also operate dams for controlling the water level by sending commands to actuator devices (ED). These three patterns are described below.

Periodic Update (PU): The PU pattern consists in periodically sending update reports to a central entity. Traffic is generated at a regular interval (e.g., each second), usually comprised of small packets of constant size. An example of this pattern is river monitoring, presented above. Other examples include smart meters reading, and remote health monitoring.

Event-Driven (ED): In this pattern, traffic is sporadic, generated only when an event occurs. An event may be detected by sensing devices (e.g., when a threshold is exceeded), or may be issued by servers (e.g., a human inputs commands). Data size may vary depending on the application and the amount of information of each event. This type of traffic is usually real-time, specially when the events refer to situations that must be acted upon quickly. An example of this traffic pattern is the generation of a security alarm in a surveillance system when something suspicious is detected. Other examples include health emergencies, disaster alerts, and notification of new routes.

Payload Exchange (PE): The PE pattern corresponds to the transfer of larger amounts of data. It usually takes place after an event is notified, furthermore after the event has occurred more data is required to be deal with the situation. For instance, in the security alarm example presented above for the ED pattern, it is possible to start a video streaming from surveillance cameras to better assess the situation and act accordingly. Another example is firmware upgrading. After receiving the notification that a new firmware version is available, the corresponding devices may start downloading the new version. This pattern may be real-time or not, depending on the corresponding event.

3.2. Impact of TD

Each IoT traffic pattern is sensitive to different network performance metrics. In this work, we consider three metrics that might impact QoE on IoT [5]: end-to-end delay, loss rate and throughput. We analyze below how these performance metrics may affect each pattern, as well as how TD could benefit a prioritized application/device over competitors.

Impact on PU: In the PU pattern, large delays might be misinterpreted by the system as inactivity or faulty behavior. For instance, an IoT device may periodically send information regarding its status (e.g., uptime, battery level) through the Internet to an IoT platform. If this periodical report is largely delayed, specially if delays are perceived repeatedly, the system may wrongly assume that the device is inactive or faulty. Furthermore, high loss rates may significantly increase the amount of data transmitted over long periods of time due to the retransmissions. Throughput, however, may not be an important performance metric for this traffic pattern, since the data rate is small. In conclusion, TD may turn a prioritized device less likely to be considered inactive or faulty by the system, and smaller loss rates may result in lower energy consumption than competitors (less retransmissions).

Impact on ED: It is important that the real-time notifications from this pattern arrive within the required time limits. ED traffic is thus sensitive to end-to-end delay. Packet loss might also affect this pattern, since retransmissions increase the end-to-end delay. Throughput, as with the PU pattern, may not be important since the amount of ED traffic is small, in most cases. For instance, let us consider the following scenario. A person driving a smart car is making use of a guidance application, which should present to the user the fastest route to the desired destination. The smart car is connected to the Internet, enabling such application to constantly check for better routes. If an accident in the

current route occurs, it may cause a traffic jam, significantly affecting the driving time until the destination. In such a situation, the smart car might receive a notification about the accident, causing the driving guidance application to provide a new and faster route to the user. If this notification gets delayed, it might arrive after the user has reached the traffic jam, and from this point it may be impossible to take a detour. Therefore, TD might result in better response times upon the occurrence of events for prioritized devices and applications. In the scenario described above, if cars from a given manufacturer have priority over others, they will perceive smaller delays that may cause a significantly difference in the QoE perceived by users, and this in turn may affect consumer decisions when buying a new car.

Impact on PE: PE traffic is similar to HTC traffic, since it consists of a continuous transfer of larger amounts of data. Throughput is thus relevant, while end-to-end delay is not as important as with the ED pattern. Packet loss may impact throughput, since less data is transferred in the same amount of time. A prioritized application may experience a higher throughput. This may result in an overall better QoE, specially when there is human interaction (video/audio streaming).

4. Simulation Results

In this section we describe several experiments executed with simulation to evaluate the impact of TD on different IoT traffic patterns. We simulate each pattern under three different TD scenarios, totaling 9 simulations. We employed the OMNeT++ [26] simulation framework for implementing and executing these simulations. The duration of each simulation was 1800 seconds, which was set empirically. We observed no significant difference in the results when we executed experiments that took longer than that to complete.

Figure 2 shows an overview of the experiments executed. In each simulation, there are three different sets of traffic sources: cross-traffic, high priority, and low priority. All the traffic ingressing at the network goes through a classifier, which identifies the priority of the traffic (as high or low). A TD mechanism is then employed based on the classification. Each of the three TD scenarios corresponds to a TD mechanism employed. The traffic is then routed to the destination through a single router. The links between the traffic sources and the network have maximum rate of 10 Mbps and a propagation delay of 10 ms, the same delay of the links between the host running the TD mechanism and the router, and between the router and the destinations. The total propagation delay is thus 30 ms, and the maximum output rate of the network is 10 Mbps.

All the traffic generated by the traffic sources traverses the same path in the network, competing thus for the same network resources. However, traffic from one of the sources (called high priority) is prioritized over others (low priority and cross-traffic). The goal is to check how this prioritization affects the end-to-end performance. We implement this prioritization by reserving a small ratio (1%) of the maximum rate (10 Mbps) of the output link to the high priority traffic, i.e., the high priority traffic has at least 100 Kbps of the bandwidth guaranteed.

The high priority and low priority traffic sources generate traffic corresponding to the three IoT traffic patterns described previously in Section 3: Periodic Update (PU), Event-Driven (ED), and Payload Exchange (PE). Cross-traffic simulates background Internet traffic generated by sources other than IoT devices and applications.

The rest of this section is organized as follows. We describe how cross-traffic is generated in our simulations in Subsection 4.1. Then, we describe how we implemented the three IoT traffic patterns in Subsection 4.2. The TD scenarios are described in Subsection 4.3. We present the results in Subsection 4.4, and a discussion in Subsection 4.5.

4.1. Cross-traffic

In our simulations, cross-traffic is generated according to the HTC pattern, since IoT traffic competes with HTC for network resources in the Internet [23]. Therefore, cross-traffic was implemented by generating several continuous flows of packets of variable size and rate. We employed the UDP transport protocol, since it allows better control of the amount of traffic introduced in the network (no congestion control, ACKs, retransmissions, etc.). Each different flow consists of packets with random sizes ranging from 250 to 1000 bytes. Packets are sent at random intervals ranging from 8 to 12 ms, resulting in an average sending rate of 500 Kbps.

In order to evaluate how IoT traffic patterns fare under different conditions of cross-traffic and congestion, cross-traffic is generated in 4 different levels during the 1800 seconds of each simulation. In the first 100 seconds, there is no cross-traffic. From seconds 100 to 500 of the simulation, the cross-traffic rate increases gradually (as new flows are started) up to 10 Mbps. At 1300s the cross-traffic increases by 500 Kbps, and that is repeated at 1600s. Figure 3 shows the cross-traffic sending rate during the 1800 seconds of each simulation.

4.2. IoT Traffic Patterns

We implemented the different traffic patterns based on the MTC traffic model proposed in [25], and according to the IoT traffic characterization presented in [27]. Each pattern differs in terms of several parameters: the number of traffic sources, packet size, total data size, sending rate, and the interval between transmissions.

The PU pattern consists of sending a constant sized packet (500 bytes) every 5 seconds, employing the TCP protocol. In the experiments we employed 50 high priority sources and 50 low priority sources. Figure 4 shows the sending rate of the aggregate PU traffic of each priority class during the simulations.

The ED pattern consists of sending short bursts at random times. The number of packets of a burst is selected randomly and varies from 1 to 900 packets, each packet

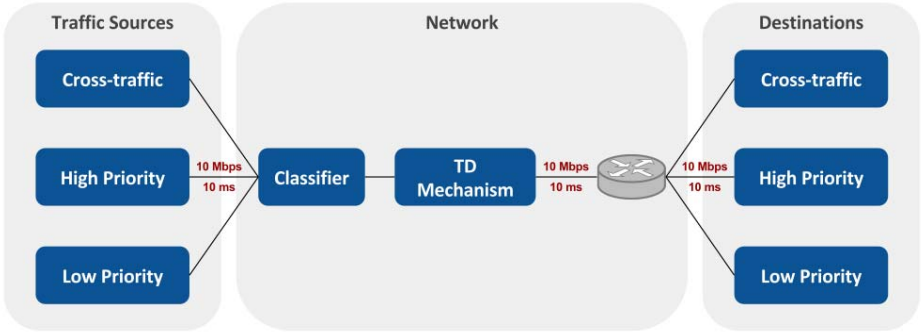


Figure 2. Simulation: the main modules.

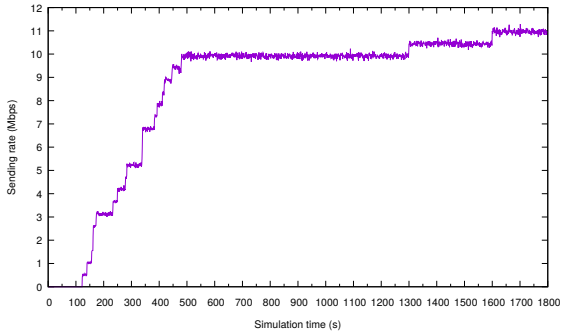


Figure 3. Cross-traffic sending rate.

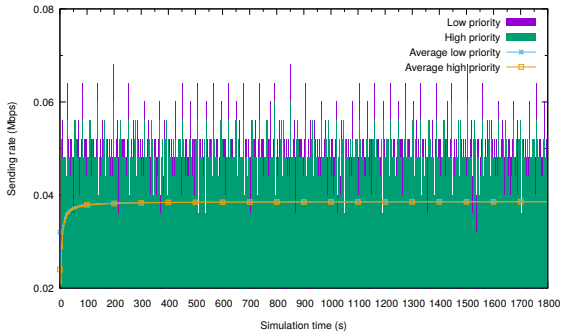


Figure 4. PU pattern sending rate.

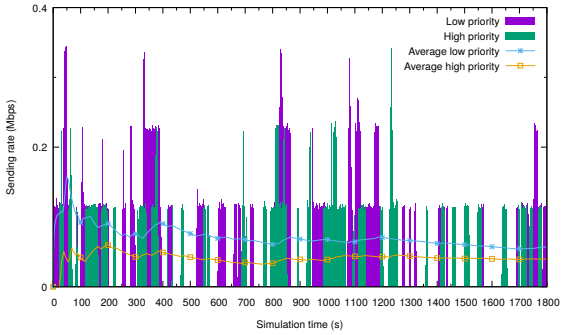


Figure 5. ED pattern sending rate.

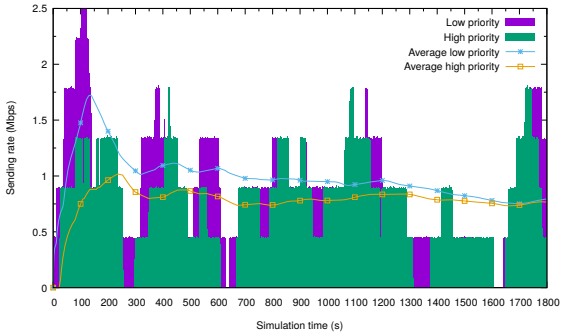


Figure 6. PE pattern sending rate.

size ranges from 800 to 1200 bytes. This pattern represents the notification of an event from sensing devices to a cloud server, or vice versa. In our simulations, we employed 50 high priority sources and 50 low priority sources generating ED traffic. Each source generates an event at a random time instant. Figure 5 shows the sending rate of the aggregate ED traffic of each priority class during our simulations.

We implemented the PE pattern as continuous flows of UDP traffic, similar to the HTC traffic employed as cross-traffic. We employed 30 sources of each priority, thus a total of 60 sources generate PE traffic. Since this pattern usually takes place after the notification of an event, we start PE transfers in the same way as the ED pattern, i.e., each source initiates a transfer at a random time instant. Each transfer size varies randomly from 4 to 8 MB. Figure 6 shows the sending rate of the aggregate PE traffic of each priority class in the simulations.

4.3. TD scenarios

We implemented three different TD scenarios, namely Neutral, Shaping, and Policing. The Neutral scenario employs no TD. The Shaping scenario is based on traffic shaping [21], while the Policing scenario is based on traffic policing [22].

In the Neutral scenario, no TD is performed, thus all traffic is treated equally. A single packet queue is employed. Packets are dequeued and forwarded in the order they arrive, i.e., according to the *First In, First Out* (FIFO) policy. The queue has maximum size equal to 100. When the queue is full, all arriving packets are dropped, employing the *Drop-tail* (DT) approach.

In the Shaping scenario, the reserved rate (100 Kbps) is enforced by queuing high priority packets in a separate queue and forwarding them first. Two DT queues are employed, one for each priority. All high priority packets that fall under the reserved rate are queued in the high priority queue. High priority packets that exceed this rate are queued together with the rest of the traffic, in the low priority queue. Packets from the high priority queue are always forwarded first. Both DT queues have maximum size equal to 100.

In the Policing scenario, a single DT queue with maximum size of 100 is employed. Low priority packets that exceed the maximum output rate of the network (10 Mbps) are dropped. High priority packets exceeding the reserved rate are reclassified as low priority, thus becoming subject to the same dropping conditions as the low priority packets.

4.4. Results

We present below the results for each traffic pattern, under the different TD scenarios. We focus on the most relevant metrics for each pattern, which were discussed in Section 3.

4.4.1. PU results. As discussed previously, packet loss in the PU pattern may result in a significant increase in the amount of data transmitted over long periods of time. We then evaluated then the number of Retransmission Timeouts (RTOs), computed exactly as the TCP protocol does. Figure 7 shows the Cumulative Distribution Function (CDF) of the number of RTOs, for each priority class and under each TD scenario. Each CDF corresponds to the portion of the time during which the corresponding number of RTOs occurred.

In the Neutral scenario (Figure 7a), the CDFs for both priorities were very similar. 15092 high priority packets and 15086 low priority packets were sent. High priority traffic suffered 2418 RTOs in total, while the low priority traffic suffered 2429 RTOs.

In the Shaping scenario (Figure 7b), the high priority traffic suffered no RTOs. 17341 and 14916 packets were sent by the high and low priority sources, respectively. A total of 2626 RTOs were suffered by the low priority traffic.

In the Policing scenario (Figure 7c), the high priority traffic presented RTOs in only about 10% of the simulation, while the low priority traffic in about 25%. High priority traffic consisted of 17133 packets, while the low priority consisted of 16866 packets. A total of 123 RTOs were suffered by the high priority traffic, and 531 RTOs by the low priority traffic. In this scenario, there were less RTOs than in the Neutral scenario, for both priorities. However, even with this improvement, the Policing scenario introduced a difference between the priorities, which didn't exist in the Neutral scenario.

4.4.2. ED results. The end-to-end delay is an important metric to evaluate the ED pattern, since it consists of real-time event notifications. Figure 8 shows the average end-to-end delay, in milliseconds, experienced by packets from of each priority class, under each TD scenario.

As the amount of cross-traffic increased, the average end-to-end delay also increased in a similar way for both priorities in the Neutral scenario (Figure 8a). In the other two scenarios (Figures 8b and 8c), however, the average end-to-end delay increased significantly more for the low priority traffic.

In the Neutral scenario, 759 low-priority traffic packets suffered end-to-end delays larger than 1 second, while for the high priority traffic 1037 packets suffered similar delays. In the Shaping scenario, 815 low priority traffic packets had end-to-end delays larger than 1 second, while only 297 high priority packets suffered similar delays. In the Policing scenario, 475 low priority packets suffered delays larger than 1 second, while 239 high priority packets suffered similar delays.

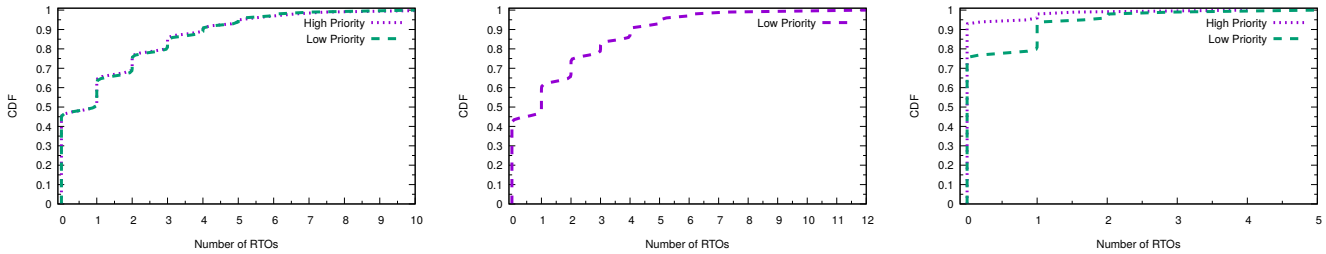
4.4.3. PE results. PE traffic consists of transferring larger amounts of data than the other IoT patterns. Therefore, as discussed previously, throughput is important, since faster transfers may result in better QoE. We evaluated the throughput achieved by traffic of both priority classes under each TD scenario. At first the reserved rate (100 Kbps) had no significant impact on the throughput. We thus ran our simulations again, employing a larger reserved rate for the high priority traffic. We set the reserved rate to 10% of the link bandwidth, i.e., 1 Mbps. The goal was to check if a larger reserved rate would result in a significant difference in throughput between the two traffic priorities.

Figure 9 shows the throughput for each priority, under each TD scenario. It is possible to observe that the average throughput for high priority traffic increased in the Shaping and Policing scenarios, in comparison with the Neutral scenario. The difference is most noticeable after 1300 seconds of simulation, when cross-traffic reaches a higher level.

4.5. Discussion

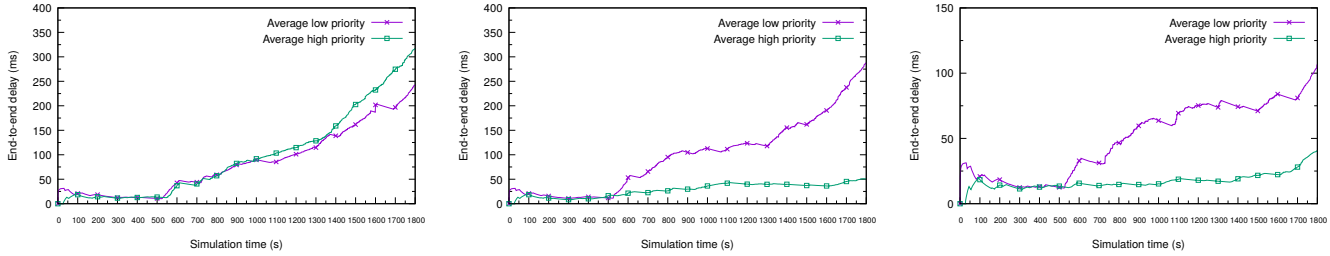
Based on the obtained results, we argue that TD impacts the ED pattern the most, due to the real-time nature of this type of traffic. The difference observed on the end-to-end delay between high and low priorities in non-neutral scenarios shows that even a small reserved rate (1%) may be enough to create a significant difference on the QoE perceived by an end-user, which might result in unfair competition. We argue that the end-to-end delay may be a good metric for detecting differentiation of real-time IoT traffic.

Regarding the PU pattern, TD may have a meaningful impact depending on the application. In cases where energy consumption is important, for example, the larger amount of



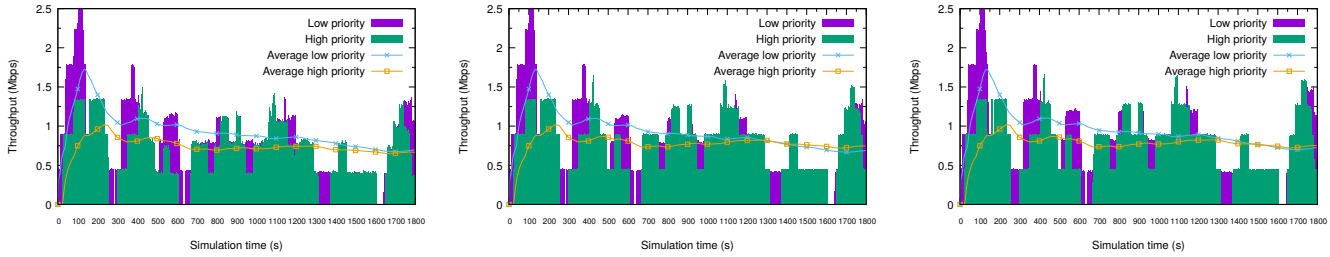
(a) Neutral scenario. (b) Shaping scenario. (c) Policing scenario.

Figure 7. CDF of the number of RTOs for the PU pattern.



(a) Neutral scenario. (b) Shaping scenario. (c) Policing scenario.

Figure 8. Average end-to-end delay for the ED pattern.



(a) Neutral scenario. (b) Shaping scenario. (c) Policing scenario.

Figure 9. PE pattern throughput.

packet retransmissions may be a concern. The impact on the PE pattern may also depend on the application. For example, for real-time streaming an affected throughput may result in significant QoE degradation.

5. Proposed Solution

Most existing solutions for detecting TD require a large enough number of vantage points for making their measurements. We argue that IoT naturally solves this issue, since it readily provides numerous vantage points. However, when employing existing solutions on IoT, two main issues arise: (i) those strategies are designed for HTC; and (ii) they are mostly based on comparing two or more traffic flows originating from a given source.

As discussed in Section 3, IoT devices and applications are affected by TD in a way that is different from HTC applications. Therefore, the most adequate metrics and methods for TD detection on IoT devices and applications necessarily vary from those from HTC. Furthermore, IoT devices often have limited capabilities. Thus it is not feasible to assume that such devices will be able to issue active probes, or generate any additional traffic flows. Therefore, the current state of the art might not be viable for detecting TD on IoT. In addition to these issues on IoT, we identified other general open challenges not addressed by current solutions, described below: dynamic TD practices, ISP evasion, and solution adoption.

Dynamic TD occurs, for instance, when an ISP employs TD on some specific periods of the day, or when the ISP constantly changes the TD mechanisms over time. Detecting this type of behavior is still an open challenge. Continuously monitoring the presence of TD is a possible direction to address this challenge.

Most existing solutions generate their own traffic in

order to make measurements and infer TD. However, the artificial traffic generated by such solutions might be identified by ISPs [28], which could then evade the TD inference, by prioritizing the measurement traffic, for example.

In order to achieve meaningful results, some solutions require that a large number of end-users report measurements for several different applications, and from multiple vantage points. Therefore, it is important to create incentives which may increase the adoption of the solution by a large number of users. Another challenge is to allow any arbitrary application to monitor how its traffic is performing compared to others, without having to implement TD detection on its own. This would enable not only end-users, but also applications and services to benefit from TD inference and to contribute to increase its accuracy. Taking advantage of pre-existent infrastructures and/or real traffic monitored passively also allows measurements to be made without the need to control a large number of end-hosts or rely on a large number of end-users.

Our goal is to create a TD detection solution for IoT that addresses all the challenges above. In a previous work [2], we proposed a model for continuously monitoring TD in distributed systems. This model takes advantage of the technologies and infrastructure of current and future distributed systems. The idea is to continuously monitor the communication of a plethora of devices (e.g. using crowdsensing), checking the presence of TD in real-time. Measurements are passively obtained as devices communicate, and if the presence of TD is detected, active measurements or other actions may be performed – in a hybrid active/passive approach.

However, we identified **four key challenges** for implementing our model on IoT: (i) determining which metrics to employ for traffic from different IoT devices and applications; (ii) determining which sets of measurements are comparable to each other; (iii) determining how to infer the presence of TD on IoT based on the obtained measurements; and (iv) determining how to continuously monitor TD on IoT in real-time.

We discuss strategies to build effective strategies for online TD monitoring in the IoT, based on continuous passive measurements and Machine Learning (ML) [18]. We argue that ML provides powerful tools for addressing the key challenges presented above. The main idea is to passively monitor IoT traffic, in order to establish the "default network performance" of different IoT traffic patterns. If the perceived performance of the traffic from an IoT device or application differs from this baseline, TD may have occurred. Our proposal is based thus on a ML classifier, or ensemble of classifiers, that receives a set of metrics corresponding to an IoT communication, and outputs the class of such set: *neutral* or *non-neutral*. We envision a TD detection service, which can be used by any IoT platform or cloud service.

The rest of this section is organized as follows. Subsection 5.1 gives an overview of the TD detection solution. Subsection 5.2 describes the four key challenges and how our proposal address them.

5.1. TD Detection Service

Figure 10 shows our proposed TD detection solution applied to the same IoT architecture presented previously in Figure 1. The IoT platform collects measurements and confounds related both to edge devices and cloud services. These measurements are continuously fed to the TD detection service, which infers whether TD is going on or not. Any other service should be able to employ the TD detection services and obtain the results. The more data is fed to the TD detection service, the more accurate it becomes, since measurements from different sources may be aggregated and used to process new inputs.

Measurements might be, for example, the loss rate experienced when receiving data from an IoT device, or the delay between a request from a device and its response. Furthermore, confounds might be, for example, the geographic location of the edge device, to which network it is connected, time of the day, or traffic characteristics, such as sending rate and packets size, which might help identify the traffic pattern, as discussed in Section 3.

5.2. Key Challenges

The first key challenge refers to the metrics and measurements to be employed. Current solutions for detecting TD employ only a few metrics, such as loss rate, delay, and throughput. They make active or passive measurements of the traffic from different applications, and/or generate an artificial baseline traffic. However, IoT devices and applications follow different patterns of traffic, which may not be evaluated using the same metrics. Furthermore, TD affects IoT traffic in ways that are different from HTC. Large-scale IoT traffic characterization must be done to identify the most adequate metrics and methods for evaluating IoT traffic.

The second key challenge refers to the confounds that must be taken into consideration when comparing different traffic flows. For instance, traffic flows generated in different periods of the day, with different patterns, or from different geographic locations may not be comparable to each other. Most current solutions compare two or more simultaneous traffic flows between a same pair of hosts, in an attempt to avoid several of such confounds. This approach may not be feasible in a real IoT environment. ML classifiers, on the other hand, are able to consider several metrics and other features at once, allowing them to take confounds into account, i.e., only employing the knowledge resulting from data comparable to the new sample.

The third key challenge corresponds to the inference mechanisms, i.e., how to decide, with reasonable confidence level, if a specific traffic flow was affected by TD or not. Current solutions are based on statistical inference, comparing two or more measurement distributions. If there was a significant difference over different sets of measurements, a relative discrimination between the corresponding traffic flows is detected. In our proposal, the classifiers detect TD without the need of two or more traffic flows at once for

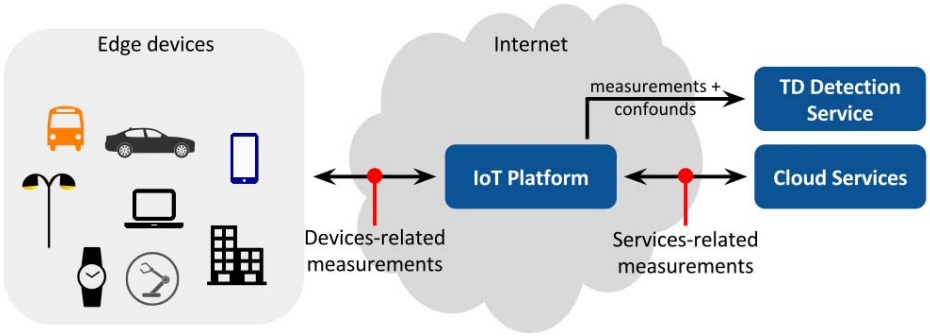


Figure 10. Proposed TD detection service on a common IoT architecture.

comparison. TD inference is achieved by "comparing" metrics from only one sample with the previously accumulated knowledge.

The fourth key challenge refers to the final goal of our proposal, which is an online TD monitoring for IoT. The presence of TD and how it affects the traffic may change over time or depend on network conditions. An ISP might employ TD only on periods of the day during which the network is under heavy load, for example, or change the TD mechanism to employ depending on location. Existing solutions are not designed to detect such dynamic behavior, since they usually consist of one-shot analysis, thus can only detect TD being employed at the time of their execution. We propose to continuously monitor IoT communication, employing data stream classification [29] methods for detecting TD in real-time.

6. Related Work

Some solutions for detecting TD of HTC traffic on the Internet have been published in the last decade [9], [10], [11], [12], [13], [14], [15], [16], [17]. These solutions are based on network measurements and statistical inference, and focus on HTC traffic. In general, they take measurements from one or several end-hosts, employing different types of traffic and probes. The measurements obtained are then analyzed to determine whether there was a significant difference over different sets of samples. Robust statistical models are necessary to distinguish between TD and performance variations caused by other phenomena.

There are also works which study the impact of MTC traffic to cellular networks and how it competes with HTC for network resources. In [23], the authors perform a large-scale measurement in a tier-1 cellular network. The goal was to characterize the MTC traffic and identify its impact on the network, as well as how it competes with HTC. In [30], the authors propose a framework for evaluating the performance of a cellular network when there are both MTC and HTC traffic. In [31], the authors present the requirements and challenges introduced by MTC to cellular networks, which should support both MTC and HTC.

7. Conclusion

Prioritization of traffic from devices or services of specific manufacturers or providers may result in unfair competition, hindering innovation and thus the success of IoT. In this paper, we investigated TD on IoT in the context of NN. We described common IoT traffic patterns and discussed how TD may impact those patterns. We presented simulation results showing how different TD scenarios affected each traffic pattern. We concluded that even a small reserved rate may introduce a significant difference between different traffic priorities. Furthermore, the ED pattern was the most affected by TD, since it caused a significant difference on end-to-end delay depending on priority. This difference might greatly influence the QoE perceived by end-users, given the real-time nature of the ED pattern. We then discuss a solution for monitoring TD on IoT, which takes into consideration the specific characteristics of IoT traffic. Previous solutions are targeted at HTC traffic. The solution is based on continuous passive measurements and ML classifiers, taking advantage of the multitude of data made available by the large amount of IoT devices.

Future work includes further investigating IoT traffic and how to use this knowledge to build a solution to effectively detect TD. Creating ML classifiers requires a significant amount of training data and domain-specific knowledge. Therefore, a deep understanding of IoT traffic characteristics and requirements, as well as a large-scale IoT traffic characterization are necessary.

Acknowledgments

This work is the result of the mobility period of Thiago Garrett at TU Wien, made possible with the Erasmus Mundus SMART^2 support (Project Reference: 552042-EM-1-2014-1-FR-ERA MUNDUS-EMA2) coordinated by CENTRALESUPELEC.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.

[2] T. Garrett, S. Dustdar, L. C. E. Bona, and E. P. Duarte Jr., "Ensuring Network Neutrality for Future Distributed Systems," in *Int. Conf. Distributed Computing Systems (ICDCS)*, June 2017, pp. 1780–1786.

[3] J. Crowcroft, "Net Neutrality: The Technical Side of the Debate: a White Paper," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, 2007.

[4] K. U. R. Laghari and K. Connelly, "Toward total quality of experience: A QoE model in a communication ecosystem," *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 58–65, April 2012.

[5] D.-H. Shin, "Conceptualizing and measuring quality of experience of the Internet of Things: Exploring how quality is perceived by users," *Information & Management*, February 2017.

[6] B. van Schewick and D. Farber, "Point/Counterpoint: Network Neutrality Nuances," *Commun. ACM*, vol. 52, no. 2, pp. 31–37, February 2009.

[7] R. T. B. Ma, "Pay or Perish: The Economics of Premium Peering," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 2, pp. 353–366, Feb 2017.

[8] H. Habibi Gharakheili, A. Vishwanath, and V. Sivaraman, "Perspectives on Net Neutrality and Internet Fast-Lanes," *SIGCOMM Comput. Commun. Rev.*, vol. 46, no. 1, pp. 64–69, January 2016.

[9] Y. Zhang, Z. M. Mao, and M. Zhang, "Detecting Traffic Differentiation in Backbone ISPs with NetPolice," in *Internet Measurement Conf.*, ser. IMC '09. ACM, 2009, pp. 103–115.

[10] M. B. Tariq, M. Motiwala, N. Feamster, and M. Ammar, "Detecting Network Neutrality Violations with Causal Inference," in *Int. Conf.Emerging Networking Experiments and Technologies*. ACM, 2009, pp. 289–300.

[11] G. Lu, Y. Chen, S. Birrer, F. E. Bustamante, and X. Li, "POPI: A User-Level Tool for Inferring Router Packet Forwarding Priority," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 1–14, February 2010.

[12] P. Kanuparthy and C. Dovrolis, "DiffProbe: Detecting ISP Service Discrimination," in *IEEE INFOCOM*, March 2010, pp. 1–9.

[13] M. Dischinger, M. Marcon, S. Guha, K. P. Gummadi, R. Mahajan, and S. Saroiu, "Glasnost: Enabling End Users to Detect Traffic Differentiation," in *USENIX Conf. Networked Systems Design and Implementation (NSDI)*, 2010, pp. 405–418.

[14] U. Weinsberg, A. Soule, and L. Massoulie, "Inferring traffic shaping and policy parameters using end host measurements," in *IEEE INFOCOM*, April 2011, pp. 151–155.

[15] Z. Zhang, O. Mara, and K. Argyraki, "Network Neutrality Inference," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 63–74, October 2014.

[16] R. Ravaioli, G. Urvoy-Keller, and C. Barakat, "Towards a General Solution for Detecting Traffic Differentiation at the Internet Access," in *Int. Teletraffic Congress (ITC)*, September 2015, pp. 1–9.

[17] A. Molavi Kakhki, A. Razaghpanah, A. Li, H. Koo, R. Golani, D. Choffnes, P. Gill, and A. Mislove, "Identifying Traffic Differentiation in Mobile Networks," in *Internet Measurement Conf.* ACM, 2015, pp. 239–251.

[18] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[19] A. Dainotti, A. Pescape, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE Network*, vol. 26, no. 1, pp. 35–40, January 2012.

[20] C. V. Wright, F. Monrose, and G. M. Masson, "On inferring application protocol behaviors in encrypted network traffic," *J. Mach. Learn. Res.*, vol. 7, no. Dec, pp. 2745–2769, 2006.

[21] P. Kanuparthy and C. Dovrolis, "ShaperProbe: End-to-end Detection of ISP Traffic Shaping Using Active Methods," in *Internet Measurement Conf.* ACM, 2011, pp. 473–482.

[22] T. Flach, P. Papageorge, A. Terzis, L. Pedrosa, Y. Cheng, T. Karim, E. Katz-Bassett, and R. Govindan, "An Internet-Wide Analysis of Traffic Policing," in *SIGCOMM*. ACM, 2016, pp. 468–482.

[23] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "A First Look at Cellular Machine-to-machine Traffic: Large Scale Measurement and Characterization," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 1, pp. 65–76, June 2012.

[24] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A Survey," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 70–95, Feb 2016.

[25] N. Nikaein, M. Laner, K. Zhou, P. Svoboda, D. Drajic, M. Popovic, and S. Krco, "Simple Traffic Modeling Framework for Machine Type Communication," in *Int. Symp. Wireless Communication Systems*, Aug 2013, pp. 1–5.

[26] A. Varga and R. Hornig, "An Overview of the OMNeT++ Simulation Environment," in *Int. Conf. on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools '08. ICST, 2008, pp. 60:1–60:10.

[27] A. Sivanathan, D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Characterizing and Classifying IoT Traffic in Smart Cities and Campuses," in *IEEE Conf.Computer Communications Workshops (INFOCOM WKSHPS)*, May 2017.

[28] A. Maltinsky, R. Giladi, and Y. Shavitt, "On Network Neutrality Measurements," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 4, pp. 56:1–56:22, May 2017.

[29] H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet, "A survey on ensemble learning for data stream classification," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 23:1–23:36, March 2017.

[30] M. Centenaro and L. Vangelista, "A study on M2M traffic and its impact on cellular networks," in *IEEE World Forum on Internet of Things (WF-IoT)*, Dec 2015, pp. 154–159.

[31] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward Massive Machine Type Cellular Communications," *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 120–128, February 2017.