



This is a peer-reviewed, post-print (final draft post-refereeing) version of the following published document, © 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. and is licensed under All Rights Reserved license:

**Rukh, Mah ORCID logoORCID: <https://orcid.org/0000-0001-7660-1150>, Ocampo, Andres F and Elmokashfi, Ahmed (2022) Measuring and Localising Congestion in Mobile Broadband Networks. IEEE Transactions on Network and Service Management, 19 (1). pp. 366-380. doi:10.1109/TNSM.2021.3115722**

Official URL: <https://doi.org/10.1109/TNSM.2021.3115722>

DOI: <http://dx.doi.org/10.1109/TNSM.2021.3115722>

EPrint URI: <https://eprints.glos.ac.uk/id/eprint/11376>

#### **Disclaimer**

The University of Gloucestershire has obtained warranties from all depositors as to their title in the material deposited and as to their right to deposit such material.

The University of Gloucestershire makes no representation or warranties of commercial utility, title, or fitness for a particular purpose or any other warranty, express or implied in respect of any material deposited.

The University of Gloucestershire makes no representation that the use of the materials will not infringe any patent, copyright, trademark or other property or proprietary rights.

The University of Gloucestershire accepts no liability for any infringement of intellectual property rights in any material deposited but will remove such material from public view pending investigation in the event of an allegation of any such infringement.

PLEASE SCROLL DOWN FOR TEXT.

# Measuring and Localising Congestion in Mobile Broadband Networks

Mah Rukh Fida, Andres Ocampo, and Ahmed Elmokashfi  
SimulaMet, Center for Digital Engineering  
Oslo, Norway

**Abstract:** Mobile broadband networks, although increasingly popular, suffer large fluctuations in performance. Download speeds can drop by 50% or more during peak hours. Hence, understanding and dissecting the causes of these fluctuations is central to improving current and future networks. In this paper, we propose a congestion detection and localisation method, Q-TSLP, that combines and extends the two state-of-the-art congestion detection tools: Q-Probe and TSLP. Q-Probe monitors patterns in packet arrivals, while TSLP tracks shifts in RTT to detect bottleneck at different segments of an end-to-end path. QProbe can attribute congestion, at a very coarse level, to either radio or non-radio related. TSLP on the other hand cannot pinpoint radio related congestion. Q-TSLP provides a per-hop congestion attribution thus addressing these limitations. To this end, we build two small scale LTE testbeds and experiment with a series of congestion scenarios. These controlled experiments show that apart from correct congestion localisation to finer granularity, the detection accuracy improves significantly with Q-TSLP, up to 100% in some cases. We then run a three-month long measurement campaign of congestion over two commercial operators in Norway. Overall, we run 17 million tests from a large number of geographically distributed probes. We find that both operators suffer congestion at different parts of the network. Our findings indicate that apart from mobile radio access, a non-trivial fraction of cases is related to congested mobile operator and Internet paths beyond the mobile network core. These findings hint that operators may need significant infrastructure upgrades to cope with potential 5G traffic volumes.

Index Terms: Bottleneck localisation, congestion detection, download speed, latency, mobile broadband network

## I. Introduction

Congestion is a degradation in network performance that occurs when traffic demand exceeds available resources. It is, unfortunately, a recurring problem on the Internet as well as on mobile broadband networks (MBB). Thus, over the past three decades most efforts towards designing and optimizing transport protocols have chiefly focused on improving resource utilization without exacerbating congestion [1].

MBB networks are known to exhibit significant performance swings. A recent report by Opensignal, an independent mobile analytics company, shows that MBB speed can be between 5.8 and 31.2 Mbps slower during peak hours [2]. This report leverages crowd sourced data from 94 million devices from 77 countries. Such fluctuations can render some applications unusable. For instance, Netflix recommends at least a speed of 25 Mbps for watching ultra HD videos.<sup>1</sup> If the speed drops in peak hours to below the recommended value, consumers could essentially be denied access to certain content for extended time periods. This problem will become more central as MBB networks are increasingly marketed as an alternative to fixed broadband services (i.e. fixed wireless access).

Tracking and understanding congestion in MBB networks is thus paramount. MBB networks, however, do not lend themselves easily to such instrumentation for a couple of reasons. First, the time slotted nature of channel access in MBB networks complicates delay measurements by introducing effects like bursty arrivals. Secondly, MBB networks tunnel traffic, above IP layer, between base stations and mobile core. This results in end to end paths that are partially opaque to tracing tools at the IP layer and above, which makes it hard to attribute congestion that is related to the opaque segment.

In this paper, we measure and attribute persistent congestion in MBB networks, by conducting a large scale study of congestion in two commercial operators of Norway. We explore the suitability of a state-of-the-art tool, QProbe [3] that exploits the pattern of inter-packet gaps in the receiving probe packets, in flagging and attributing congestion in MBB networks. For evaluation, we build two small scale LTE testbeds using software defined radios (SDRs), general purpose servers and open source LTE stacks. We find that the existing tool suffers a number of limitations that result in non-trivial inference inaccuracy. Further, we find that by extending the QProbe tool with TSLP [4], we can boost accuracy tangibly. When alone, the TSLP tool fails to identify radio congestion, but in conjunction with Q-Probe it both adds to fine grained localization of the congested segments in WAN and enhances accuracy of identifying co-existing congestion sources. To this end, we train a machine learning classifier to identify which features from the two methods can be combined for flagging congestion in different parts of the network. The

extended method, named as Q-TSLP, achieves higher accuracy, is able to discriminate complex simultaneous bottlenecks and allows for attributing congestion to network segments at a finer granularity.

Employing Q-TSLP, we conduct a large-scale congestion measurement campaign, using around 70 stationary probes, in two commercial networks in Norway.

We find that congestion is a common occurrence and in most cases related to the lack of resources in the radio access network (RAN). However, congestion beyond the RAN is relatively common. Pairing congestion with downlink speed measurements confirms that fluctuations in speed are mainly congestion related. Hence, predictions that the extra spectrum brought by 5G will help smooth speed variation is partially true since it will only address the RAN congestion. Our findings indicate that MBB operators will probably need to invest in upgrading their core networks as well as their connectivity to the rest of the Internet.

We make the following contributions in this paper:

- Using longitudinal measurements from a large number of geographically spread probes, we investigate the stability of downlink and uplink speeds for two mobile operators in Norway. We observed significant variations in speeds between off-peak and peak hours, which motivated us to assess the role of congestion.
- We evaluate the suitability of two state of the art congestion detection tools, namely QProbe and TSLP, in detecting and attributing congestion, which occurs at different locations, in a mobile network. To this end, we built two small scale testbeds that mimic a typical mobile networks. This evaluation underscores the limitations of these tools. These limitations can result in both false negatives and misses in attributing congestion.
- Having identified these limitations, we propose Q-TSLP, a combination of QProbe and TSLP. Our evaluations of Q-TSLP show non-trivial gains in accuracy.
- Finally, we leverage Q-TSLP to detect and describe congestion in two commercial networks in Norway. We conduct measurements from tens of probes for over six months in 2020. Our results flag a small percentage of test instances, mostly below 5%, with congestion. Half of the congestion episodes are related to the last mile in the RAN. Although uncommon, congestion can lead to significant drops in speed, up to 78% in some cases.

## II. Background

This study focuses on localizing bottleneck on the end-to-end path between an LTE user equipment and a server in the Internet. In the following, we provide an overview of the architecture of a typical LTE network, which describes the various network segments where congestion can occur. Additionally, we explain a couple of commonly used scheduling algorithms that LTE basestations (eNodeBs) employ to coordinate the sharing of the wireless medium. Understanding the way scheduling algorithms work is important to detecting RAN congestion.

a) *LTE network components.*: LTE networks comprise a radio access network (RAN), and a core network (CN) (see Fig. 1). The RAN has user equipment (UE) and base station (eNodeB) and connects both with a radio access link. The eNodeB is then connected to CN with a transport link. The CN or the evolved packet core (EPC) comprises mobility management entity (MME), serving gateway (S-GW), packet data gateway (P-GW), policy and charging rules function (PCRF) and home subscriber (HSS) entities. These functions facilitate management of users' connectivity, mobility and accounting. Lastly a P-GW, is responsible of data transfer to/ from an end node residing outside LTE network such as a server in the Internet.

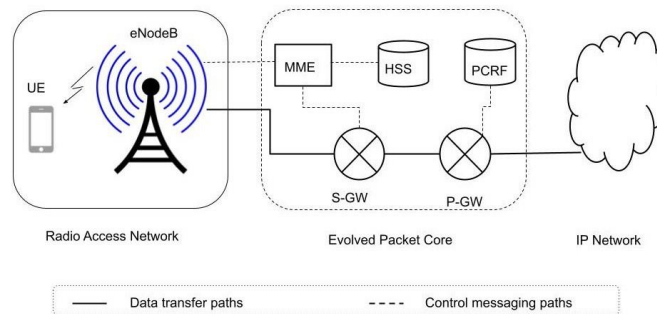


Fig. 1. Basic LTE architecture

b) *Downlink scheduling at LTE base stations:* To enable data transfer, an eNodeB allows UEs to connect to EPC. It manages the important role of uplink and downlink scheduling of data packets. Some basic scheduling methods for downlink data transmission include Round Robin, Proportional Fair Queuing (PFQ) and Best Channel Quality Indication [5]. <sup>1</sup>In Round Robin,

<sup>1</sup> Scheduling is not part of the standards and is determined by vendors.

terminals are assigned resources one after another without taking other factors into consideration. Although this method results in poor performance, fairness is guaranteed since all terminals are equally scheduled. The Best Channel Quality Indication scheduling assigns resource blocks to the user with the best radio link conditions. On the other hand, Proportional Fair scheduling tries to maximize the total throughput, while providing all users at least a minimal level of service. It balances between throughput and fairness among all the UEs. According to [3], 3G and LTE base stations usually employ PFQ. To improve the aggregate network throughput and being fair to users, PFQ allocations of a transmission time interval (TTI) to a UE get spaced apart as the number of users increases. When a UE is scheduled, multiple packets buffered at the base station can be transmitted to it back-to-back, depending on the bitrate associated with the instantaneous link quality and the average allocated rate from the recent past.

### III. Stability of MBB Throughput

In this section, we take a closer look at variability in downlink and uplink throughput in commercial LTE networks. This sheds light on how rampant congestion is in today's networks.

#### A. Measurements and data set

We leverage a country-wide deployment of stationary probes to measure MBB uplink and downlink speeds. Each probe is a single-board computer which connects to one or more MBB networks, via miniPCI cards which supports up to LTE CAT-6 (equipped with 2x2 MIMO antenna), using commercial subscriptions. To measure speed, we use a command line client for testing Internet speed using Ookla's speedtest.net.<sup>2</sup> The test is essentially based on downloading/uploading files of an increasing size from/to Ookla's servers.

The test runs three times a day at 2:00 a.m., 2:00 p.m. and 7:00 p.m. local time to capture different traffic profiles. In the following, we use speedtests to measure two mobile operators, i.e. Op<sub>1</sub> and Op<sub>2</sub> over LTE, from 102 probes and 64 probes respectively.<sup>3</sup> The collected measurements span the whole of 2019.

#### B. Speed variations

Figure 2 shows the measured median speed by each probe for Op<sub>1</sub> and Op<sub>2</sub> at 2:00 a.m., 2:00 p.m. and 7:00 p.m. Both operators use the a set of frequencies (800, 900, 1800, 2100, 2600) MHz, but 1800 MHz dominates. Here we study speeds associated with 1800MHz. The number of probes connected to 1800MHz are 47 for Op<sub>1</sub> and 55 for Op<sub>2</sub>. Downlink speed For both operators, speeds during the day are always lower than at 2:00 a.m. The performance difference between 2:00 p.m. and 7:00 p.m. is negligible for almost all the probes. For Op<sub>1</sub>, 95.7% of the probes exhibited a reduction in downlink speed during the day compared to 2:00 a.m, in 2019. The reduction ranged between 0.36% to 59.7%. For Op<sub>2</sub>, 96% of the probes experienced an up to 69.0% drop in downlink speed.

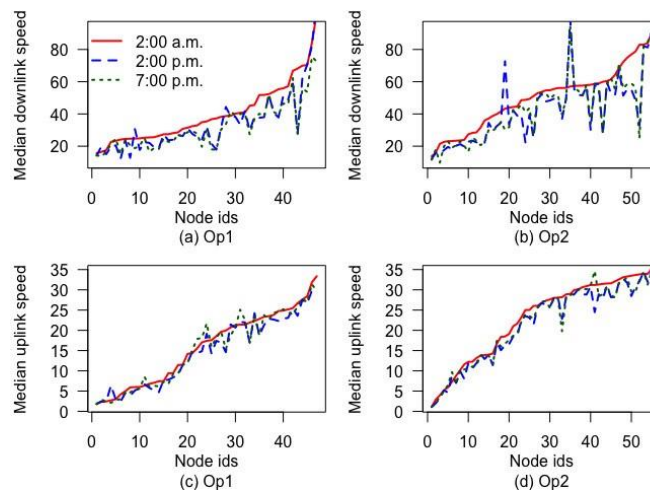


Fig. 2. Variation in speed over 1800 MHz for Op<sub>1</sub> and Op<sub>2</sub>.

<sup>2</sup> <https://github.com/sivel/speedtest-cli>

<sup>3</sup> These two operators are often ranked among the top 10 globally in terms of performance.

Uplink speed We find smaller differences in uplink speed between off-peak and peak hours. Compared to 2:00 a.m., 76% of the probes exhibited a reduction in uplink speed for Op<sub>1</sub>. The reduction varied between 0.15% and 41.8%. The number of probes that showed a decrease in uplink speed, with Op<sub>2</sub>, are 87.2% with the worst recorded drop of 33.7%.

Takeaways Download speeds fluctuate significantly between off-peak and peak hours, which agrees with previous observations [2]. Upload speeds also exhibit diurnal variability albeit to a lower extent. At first glance the omnipresence of speed reduction (see Fig. 2) suggests that stressed links or congestion to be the likely cause. Understanding potential causes of these fluctuations is essential for supporting future use cases like fixed wireless access.

#### IV. Detecting Congestion in a Controlled Setting

There is currently no tool that can both detect congestion in mobile networks and attribute it to the respective hop on the end to end path. The most promising method, QProbe [3], suffers certain limitations. Though light-weight, it provides a coarse-grained attribution and may have nontrivial interactions with RAN configurations. To provide a per hop attribution we extend QProbe with TSLP [4] which is dependent on ICMP responses from the routers. It is expected to provide both a method specific for MBB networks and a per hop attribution, while avoiding reinventing the wheel. There are, however, a few questions that need to be addressed before concluding on the viability of this combination:

- How sensitive is QProbe to RAN configuration?
- How does QProbe perform when there is overlapping RAN and WAN bottlenecks, since it is an end-to-end tool?
- Mobile networks tunnel IP traffic between UEs and the P-GW, effectively opaquing the radio and transport links. Given the opacity of MBB networks, how well can TSLP separate RAN/Transport bottlenecks from those beyond **the transport network?**
- Does extending QProbe with TSLP boost the accuracy or just have a confirmatory effect?

Answering these questions requires carrying out a number of experiments that capture congestion at different segments of the end to end path, which is not feasible in a commercial network. We have thus built a small-scale LTE testbed to circumvent this limitation.

##### A. QProbe and TSLP

Before delving into presenting our testbed and the series of experiments that we have run, we briefly review QProbe and TSLP and the assumptions that underpin their operation.

QProbe QProbe is a light-weight end-to-end tool (i.e. a clientserver tool) for measuring downlink congestion. It provides a coarse classification of congestion that is by separating bottlenecks into radio access link and wide area network (WAN). WAN involves all links between the QProbe server and the base station serving the QProbe client.

QProbe sends a train of small (probe) UDP packets interleaved with several MTU sized (load) UDP packets. The probe packets are sent to the UE (i.e. QProbe client), while the load packets are designed to be dropped by the last hop before the UE (i.e. their TTL is set to UE's TTL minus one).

Each packet in the probe train is transmitted after a fixed interval, which we refer to as *sleep time* in the following. When the arrivals of probe packets at the UE are spaced by longer than the fixed sending interval, it is a sign of congestion in the WAN. This happens because, in presence of congestion, load packets will be queued pushing probe packets further apart. On the other hand, bursty arrivals of probe packets at the UE is a sign of congestion in the radio. Essentially, a congested base station will allocate fewer downlink transmission grants to every UE, which results in buffering all downlink data that arrives between consecutive grants and sending them in the same TTI.

TABLE I QPROBE FEATURES.

Feature	Description
<i>Stretch Ratio (SR)</i>	a measure of how stretched the arriving probe packets are. It is calculated by dividing the width-wise duration of the arriving probe train by the original spacing. An SR higher than one is indicative of a congested WAN.
<i>Burst</i>	the largest number of packets minus 1, in a probe train, arriving in the same TTI. It is expected to be zero, when there is no bottleneck at the radio access link.
<i>minIPG</i>	the minimum inter-packet gap, in a probe train, seen by the UE.

<i>maxIPG</i>	the maximum inter-packet gap, in a probe train, seen by the UE.
---------------	---

The study proposing QProbe, defined two features to detect and discriminate congestion, namely the stretch ratio of probe packets (*SR*) and whether they exhibit a bursty arrival (*Burst*). We extract two extra features that give more insights into the arrival pattern of probe packets. These are *minIPG* and *maxIPG* of measurement probe train. In presence of radio bottleneck, with bursty arrival of some and large gap between other probe packets result in a mixture of smaller and larger than expected inter-packet gaps. We summarize the four features in Table I.

TSLP Time series latency probes (TSLP) was originally designed for detecting congested inter-domain links on the path to a particular destination. We, instead, use it to attribute delay suffered by packets at intermediate links on the end-to-end path. To identify links of a path, we initiate TSLP with traceroute. TSLP next determines, from the probe’s point of view, the routers at the near and far end of each link and sends a series of ICMP packets to them. It then investigates for significant jumps in measured round trip times (RTTs) when compared to the previous measurements. Evaluating routers against own history means that there is no need for defining network-wide congestion detection thresholds. This is desirable, since routers differ with respect to capabilities and served traffic volumes.

TSLP employs a CUMSUM [6] based algorithm for detecting level shifts in RTTs time series. For a measurement duration of length *l* time units, the algorithm detects level shifts that last for at least  $l/2$  time units. As mentioned above, the detection of the level shift is not dependent upon latency crossing a particular threshold. A shift is detected when the difference between current and past RTTs is statistically significant. TSLP employs the Student’s t-test to this end. Figure 3 displays upward level shifts detected by TSLP on RTT history of a link in an end-to-end path, from a client in a MBB network towards a server in the Internet. The red pulses demarcate the detected shifts.

*Applying TSLP to MBB networks.* The previous TSLP studies were limited to attributing latency to near and far ends of a specific link [7], [4]. We rather focus on collecting delay profiles on different segments of the entire path. Running

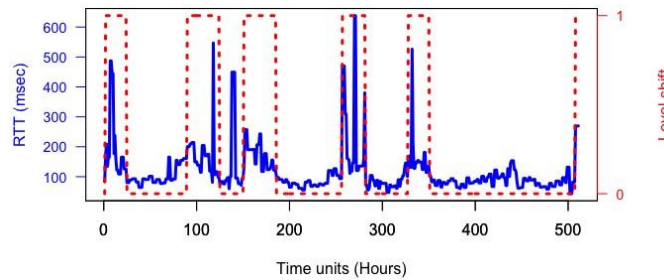


Fig. 3. Level shifts detected by TSLP in RTT profile of a link in end-to-end transmission path.

TABLE II  
TSLP FEATURES.

Feature	Description
<i>FH</i>	Level shift in the FH i.e. between the UE and P-GW.
	Level shift in the operator’s network. Here the near end is the P-GW and the far end is the operator’s egress IP.
<i>IN<sub>dl</sub></i>	Level shift in the portion of the forward path from the operator’s egress to the measurement server.
<i>IN<sub>ul</sub></i>	Level shift in the portion of the return path from the measurement server to the operator’s ingress.
<i>OP<sub>dl</sub></i>	Level shift in the portion of the path between the operator ingress and the P-GW i.e. the last responding hop in the operator’s network.

TSLP in the uplink direction (i.e. from a UE) towards an end point in the Internet allows for instrumenting three path segments: 1) First Hop (FH): it comprises of the radio, transport and parts of the mobile core. Because of tunnelling in the MBB networks the first hop that responds to an ICMP request in the uplink direction is the P-GW.

2) Operator (OP): these are the links within the MBB operator’s network. In other words, it consists of all the links between the P-GW and the operator’s egress and 3) Internet (IN): which consists of all the links between the operator’s egress and the measurement servers. TSLP can also be run in the downlink direction, revealing the same groups of links up to the P-GW. Overall, running TSLP on both the upward and downward links returns a number of features that we summarize in Table II. These features take a binary value with one and zero denoting the presence and absence of level shifts, respectively.

## B. Controlled Experiments

We have built custom small-scale LTE testbeds that are based on two open source LTE implementations srsLTE [8] and OpenAirInterface (OAI) [9]. Both srsLTE and OAI implement basic LTE components. These include the EPC components, an eNodeB and a UE part. We use the two implementations because they provide different downlink scheduling algorithms.

srsLTE uses a Round Robin, with equal priorities assigned to UEs, whereas OAI deploys Proportional Fair scheduling. This will help in controlling for the impact of scheduling algorithm on radio congestion observability.

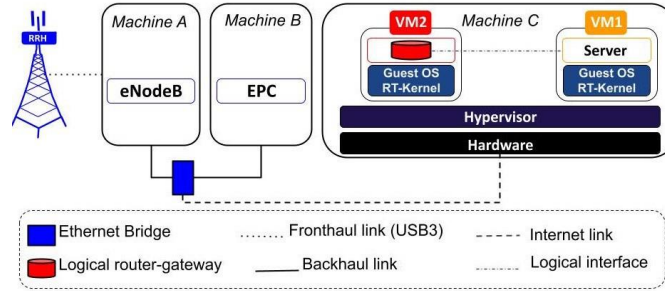


Fig. 4. Experimental setup used for srsLTE and OAI

The two testbeds have an identical layout, which we depict in Figure 4. Each testbed consists of two physical machines running Linux, one for the eNodeB and the other for the EPC. We use a USRP B210, a software defined radio, as our eNodeB radio. Further, we use two Samsung Galaxy S6 phones as UEs, both connected to the same eNodeB. Second UE is to stress the testbeds with additional traffic for some of the controlled experiments, we therefore use small bandwidths for both the testbeds. For OAI the smallest we can use is 5 MHz, but for srsLTE we can use even smaller so we use 3 MHz. Besides the mobile network, the testbed includes a third physical machine to emulate connectivity beyond the mobile core. This machine runs two Linux containers, one acting as a measurement server and the other as an intermediate router between the EPC and the measurement server. We consider this router as the egress/ingress of the mobile network.

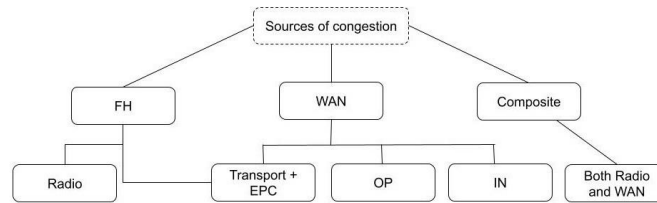


Fig. 5. Locations of emulated congestion.

**Emulating congestion** We emulate congestion at various parts of the testbed illustrated in Figure 5. We overload the radio access link by continuously sending data over an iperf session from a separate server connected to the EPC towards a second UE. To load the WAN links (i.e. all links beyond the radio access), we leverage the network emulator tool, *netem* [10]. Since congestion on a link increases end-to-end delay, we emulate the effect of congestion by adding extra delay at various locations via *netem*. These locations include the corefacing interface of the eNodeB, P-GW, router and the end server, which mimic impact of congestion at transport, EPC, Operator’s network (OP) and the Internet (IN), respectively. Further, we create composite cases of congestion where we load both the radio and a part of the WAN. The median baseline end-to-end delay is 25 ms, we increase it to between 50 and 250 ms to represent load. The latency in each emulation follows a normal distribution with a standard deviation that increases with the mean latency value and ranges from 10 to 100 ms. Note that the extent of increase in latency that matters to a service quality of experience is not in the scope of the paper. Rather we investigate if the congestion detection tools correctly identify periods of a statistically significant increase in the delay.

**Experiments** We run a total of 6500 experiments on srsLTE and 5560 on OAI. Each experiment involves a QProbe and a TSLP test. About 500 of these tests correspond to the case of no congestion and form our baseline.

**Supervised Learning** We use machine learning to evaluate the accuracy of QProbe and TSLP. The use of machine learning allows us to pick arbitrary threshold values for various features. Having emulated congestion at various locations, we label each test as congestion or not as well as the location of congestion if any. For each measurement test the features, mentioned in Table I and

Table II are collected. We then train and test a classifier with 10-fold cross-validation to check whether the measured features are adequate for discriminating congestion at different locations. Note that the QProbe paper used a similar approach, fitting a Decision Tree to detect and attribute congestion.

Besides Decision Tree, we also evaluate a number of classifiers including K-Nearest Neighbours, Logistic Regression, Random Forest and Support Vector Machine, all with 10fold cross-validation. The Random Forest classifier yields the best results followed by the Decision Tree and Support Vector Machine. For example, the accuracy of Random Forest is 84%, when separating RAN and WAN congestion using QProbe features, compared to a 76% accuracy for the secondbest classifier. We, therefore, pick the Random Forest as our classification method of choice in the remainder of this paper.

QProbe We evaluate the efficacy of QProbe in discriminating three cases of congestion: 1) radio congestion from no congestion, 2) non-overlapping radio and WAN congestion, and 3) simultaneous radio and WAN congestion.

We run QProbe by sending five packet trains, each consisting of 25 probe packets and a varying number of load packets. The *sleep time* between every two packets in a train or interpacket gap is set to 1 ms. Each of the five packet trains are configured with different number of load packets, namely 15, 30, 45, 60 and 100. Each train runs as a different experiment, but to ensure similar conditions the 5 trains are run back-toback.

QProbe performs reasonably well when separating radio congestion from no congestion. However, the accuracy of the Random Forest classifier varies depending on the testbed and number of load packets. It ranges from 83% to 87.6% for srsLTE and from 73.4% to 80% for OAI. Figure 6(a) shows that the model performs marginally better when trained and tested on data from the srsLTE testbed. Further, setting the number of load packets to 45 or 60 results in a slightly better classification for data from both testbeds. Figure 7 depicts the best and worst classifiers Receiver Operating Characteristic (ROC) curves for both testbeds. The plots also show the measured area under the ROC curve (AUC). The best and worst classifiers for both testbeds exhibit an AUC over 0.8, which indicates that the model performs excellent discrimination and neither suffers from excessive false positives nor from excessive false negatives [11].

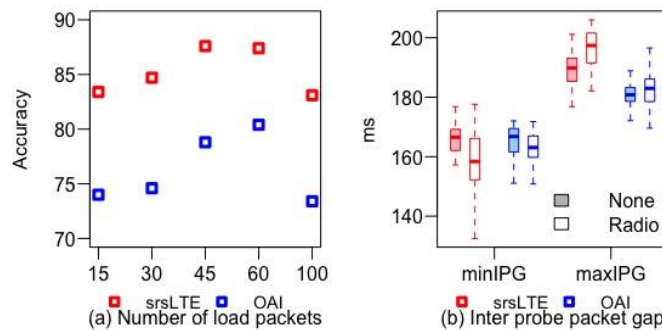


Fig. 6. The Accuracy as a function of the number of load packets (left) and the distribution of inter packet gap in the presence and absence of radio congestion (right).

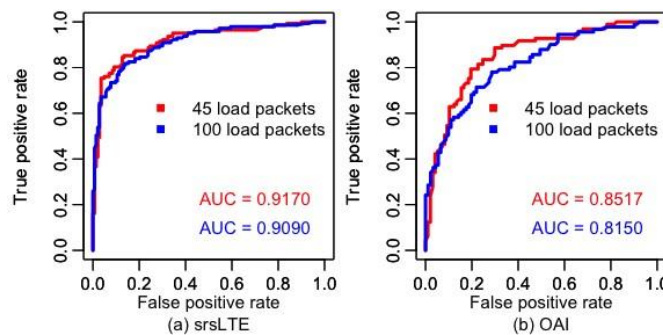


Fig. 7. ROC curves to distinguish Radio congestion events from absence of congestion.

The excellent performance of QProbe across the board, when discriminating radio congestion from no congestion, confirms its suitability under different scheduling regimes. Recall that QProbe is designed with a PFQ scheduler in mind, but srsLTE employs a Round Robin scheduler. To understand why the classifier performs best when trained and tested on the srsLTE testbed, we compare the patterns in probe packet inter-arrivals for both testbeds, because this is the main metric that QProbe uses for distinguishing radio congestion. More specifically, we examine the *minIPG* and *maxIPG* features, when sending trains with hundred load packets. Figure 6 (b) shows that differences in *minIPG* and *maxIPG* are more pronounced for srsLTE. We

believe that this is caused by the fact that the two implementations use different numbers of physical resource blocks (PRBs). srsLTE in our setup uses 15

PRBs while OAI uses 25 PRBs. 15 PRB means that we have a 3 MHz bandwidth and that the maximum bitrate is around 7.5 Mbps for srsLTE. On the other hand, for OAI we have a 5 MHz bandwidth, and the maximum bitrate is 12.5 Mbps. In other words, the smaller number of PRBs makes it easier to stress the radio. For radio congestion, this results in more bursts (i.e. smaller *minIPG*) for srsLTE than for OAI.

Next we assess performance of QProbe in separating nonoverlapping as well as co-existent radio and WAN congestion. Figure 8 shows that QProbe performs reasonably well in distinguishing non-overlapping radio from WAN congestion, which is essentially what it was designed for. The classification accuracy hovers around 84% and the corresponding AUC score is over 0.91. QProbe, however, struggles with discriminating composite congestion episodes. The classification accuracy decreases to 71% and 64% for srsLTE and OAI, respectively.

Can TSLP help? The tests with QProbe show that there is much room for improvement in distinguishing isolated and composite congestion events of radio and WAN part of the end-to-end path. We also know from before that QProbe cannot narrow down congestion to the corresponding WAN segment. We believe that with the application of TSLP on all the links of the path the latter issue can be resolved.

To investigate this, we train Random Forest classifiers using the TSLP features that are listed in Table II. We generate two classifiers, first with target class depicting bottleneck segment on the end-to-end path (i.e. FH, OP and IN) and second to distinguish isolated congestion events from the composite ones.

We find that TSLP is quite precise in attributing singular congestion events to the respective network segment. Table III presents the confusion matrix for the TSLP classifier, which we train on data from the srsLTE testbed. The overall accuracy of TSLP is 95.3% and 99% for srsLTE and OAI, respectively. However, like QProbe, TSLP also struggles with composite bottlenecks. Note that by composite here, we refer to simultaneous bottlenecks at radio (the FH in TSLP case) and any part in the WAN. With composite bottlenecks, the classification accuracy drops to 78% and 79% for srsLTE and OAI, respectively.

TABLE III  
CONFUSION MATRIX FOR IDENTIFYING CONGESTION SOURCES USING  
TSLP (SRSLTE).

		Predicted		FH		OP	IN	Error
				<i>Radio and Transport</i>	<i>EPC</i>			
Actual	FH	<i>Radio and Transport</i>	432	4	0	0	0.9%	
	<i>EPC</i>	21	390	15	10	10%		
		OP	0	32	404	0	7.3%	
		IN	0	7	0	429	1.6%	

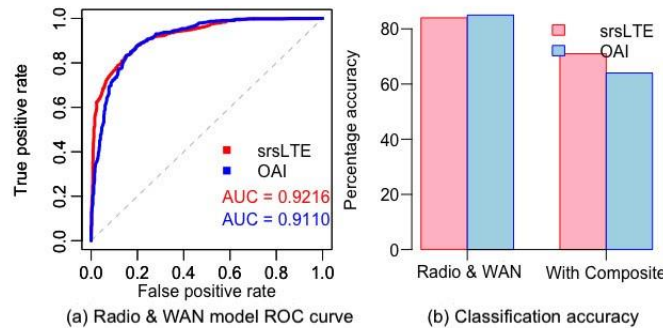


Fig. 8. The AUC values and accuracy when distinguishing Radio from WAN as well as composite congestion events.

Feature importance To find the most deciding features for both the QProbe and TSLP classifiers, we leverage the Mean Decrease in Accuracy (MDA) [12]. It measures the reduction in a model’s accuracy after randomly permuting the values of a feature. We list the top two features per classifier in Table IV. For the QProbe classifier we learn that *minIPG* and *SR* are the most important features. It is slightly different from the original QProbe study which takes *Burst* and *SR* to be the deciding features. Unlike [3] our analysis shows that in case of radio congestion, *minIPG* decreases but does not always result in an inter-packet gap as low as  $< 1$ ms. From TSLP, its *FH<sub>ut</sub>* and *OP<sub>dl</sub>* features appear to play the most important role in attributing bottlenecks to the respective network hop.

TABLE IV  
TOP TWO FEATURES WITH MDA.

Method	Classification Type	Feature 1	Feature 2
QProbe	None and Radio	minIPG (33)	maxIPG (14)
QProbe	Radio and WAN	SR (90)	minIPG (42)
QProbe	Radio, WAN and Composite	SR (110)	minIPG (72)
TSLP	Path segments	$FH_{ul}$ (41)	$OP_{dl}$ (41)

Takeaways QProbe can distinguish fairly well between no and Radio congestion cases. The comparable performance on both testbeds indicates its suitability to both the Proportional Fair and Round Robin scheduling methods. Though QProbe can distinguish well between isolated Radio and WAN congestion, its classification accuracy drops by more than 15-25% when congestion co-exist at both parts. Furthermore, it cannot narrow down the source of congestion in WAN. TSLP on the other hand, has the potential to almost accurately distinguish source of congestion among three primary segments of the end-to-end path. However, like QProbe, its accuracy drops by 17-20% when distinguishing Composite bottlenecks.

## V. Q-TSLP

Our controlled experiments highlight the strengths and weaknesses of QProbe and TSLP. They generally appear to complement each other. However, both methods struggle with complex bottlenecks that simultaneously impact the radio and WAN. Instead of reinventing the wheel, we believe that by extending QProbe with TSLP we can improve the overall performance and attribution granularity. To this end, we propose Q-TSLP.

### A. Q-TSLP working

Running Q-TSLP is a three-step approach:

- 1) The UE sends a QProbe to initiate a request to the measurement server. The measurement server transmits one or more probe train(s) to the UE. Both ends record the respective QProbe metrics.
- 2) The UE performs a traceroute to the measurement server, records the IP addresses of the responding intermediate hops and sends a sequence of ICMP ping messages to each responding IP hop including the measurement server. The received RTT from the responding hops are then used to derive latency in each of the three segments including (i) UE to the first accessible hop under in the MBB network, (ii) first to the last accessible hop in the MBB network and (iii) the Internet hops beyond that.
- 3) The measurement server performs a traceroute to the UE, records all responding IP hops, sends a sequence of ICMP ping messages to each of these hops and derives latency on the IN and the OP segments in the downlink direction.

End-to-end congestion detection Q-TSLP captures features of both QProbe and TSLP (i.e. the features listed in Table I and Table II).

To gauge the potential of Q-TSLP, we look at the use-cases with room for improving accuracy. These include distinguishing (i) Radio congestion instances from absence of congestion, (ii) Radio from WAN congestion (iii) Composite congestion events and (iv) attributing congestion in the opaque part of the network (i.e. the FH) to radio, transport and EPC.

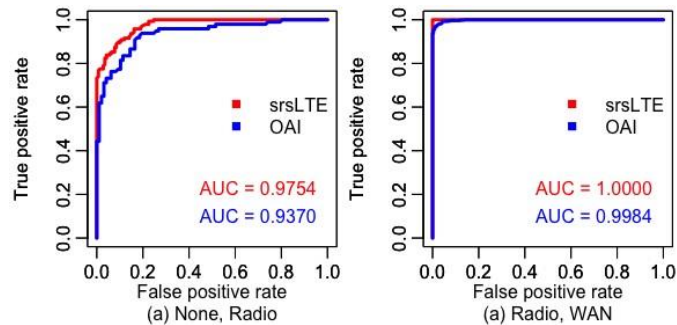


Fig. 9. Improvement in AUC score and accuracy when classifying no bottleneck and Radio bottleneck instances by using Q-TSLP.

We train a Random Forest model using the features of QTSLP. We find that Q-TSLP indeed boosts the classification accuracy. Figure 9 shows the ROC curve along with the respective AUC values for Q-TSLP, when discriminating radio congestion from no congestion (left) and when separating radio and WAN congestion (right). The overall accuracy for the radio and no congestion case increases from 87% to 93% for srsLTE and from 80% to 89% for OAI.

Q-TSLP also improves the accuracy of identifying composite congestion events. Table V shows that by reducing the individual class errors Q-TSLP improves accuracy by up to 40% over QProbe and up to 15% over that of TSLP. The results in the table also confirm that Q-TSLP can correctly attribute congestion in the opaque part of the network (i.e. see the last combination with a congestion at radio, transport and EPC). Table VI lists the top two features across all the Q-TSLP classifiers. The value beside every feature depicts its MDA value where a higher value indicates more importance. The relative importance of the features confirms that the accuracy boost is a direct product of extending QProbe method with TSLP on the links of end-to-end path.

TABLE V  
CLASSIFICATION ERROR AND ACCURACY.

<i>Radio, WAN and Composite</i>				
Error	Radio	WAN	Composite	Accuracy
srsLTE				
QProbe	0.22	0.30	0.32	71%
Q-TSLP	0.00	0.11	0.20	88%
OAI				
QProbe	0.30	0.31	0.42	64%
Q-TSLP	0.00	0.13	0.13	90%
<i>Radio, OP/IN and Composite</i>				
Error	Radio	OP/IN	Composite	Accuracy
srsLTE				
TSLP	0.00	0.38	0.24	78%
Q-TSLP	0.00	0.07	0.21	90%
OAI				
TSLP	0.00	0.00	0.61	79%
Q-TSLP	0.00	0.10	0.18	90%
<i>Radio, Transport and EPC</i>				
Error	Radio	Transport	EPC	Accuracy
srsLTE	0.006	0.006	0.11	95%
OAI	0.00	0.07	0.0	97%

TABLE VI  
TOP TWO FEATURES OF QTSLP WITH MDA.

Classifiers	Feature 1	Feature 2
None and Radio	$minIPG(29)$	$FH_{ul}(20)$
Radio and WAN	$FH_{ul}(125)$	$IN_{ul}(53)$
) Radio, WAN & Composite	$FH_{ul}(94)$	$SR(44)$
Radio,OP/IN & Composite	$IN_{ul}(39)$	$minIPG(28)$
Radio, Transport and EPC	$FH_{ul}(113)$	$OP_{dl}(41)$

### B. Q-TSLP in practice

The high accuracy of Q-TSLP begs the question of how it can be leveraged to detect congestion in an uncontrolled environment. This is primarily because collecting ground truth can be infeasible. The models trained on controlled experiments are not directly applicable in the wild, because of the differences in traffic profiles and network characteristics. This means that we perhaps need to build a per network operator model. This is confirmed by the fact that our models are not immediately transferable between the two testbeds. For example, the accuracy drops from 100% to 94%, when using the radio-WAN

congestion classifier that is trained on srsLTE for discriminating between measurements from the OAI testbed. Doing the opposite results in 11% drop in accuracy.

The task of building a per operator model can be impractical since it requires collecting fine grained ground truth and updating it as the traffic profile changes. Another alternative is to understand how the most important features, in our model, change in presence of congestion. This information can help both flagging and attributing congestion.

The *minIPG* is the most important feature for detecting *radio congestion*, since it captures the fact that packets are made to wait longer at the basestation, and when scheduled the queued packets are transmitted back-to-back. We measure a recurrent drop in the values of *minIPG* for all the cases where we emulate radio congestion (see Figure 10) (a). The second most important feature in this respect is *FH<sub>ul</sub>*, which captures RTT hikes in the opaque part of the path. The right panel of Figure 10 shows a heatmap, where x-axis depicts the fraction of events where *FH<sub>ul</sub>* is 0 (i.e. no upward level shift is detected) and 1 when congestion is detected. Here the y-axis depicts the ground-truth i.e. tests with no congestion and when radio is overloaded. Hence, a drop in *minIPG* in combination with *FH<sub>ul</sub>* is indicative of a congested radio link.

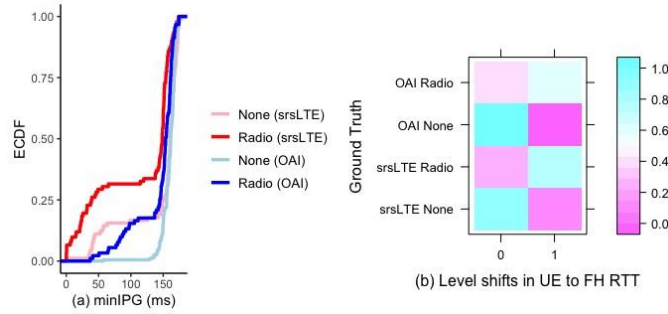


Fig. 10. With Radio congestion, *minIPG* tends to decrease and *FH<sub>ul</sub>* captures comparatively more upward level shifts in RTT.

Detecting *WAN congestion* relies heavily on the *SR* and all the level-shift features. The left panel of Figure 11 compares the value of *SR* when the WAN is congested to the baseline. In case of no congestion, *SR* is around 1 but it tends to increase significantly when there is congestion in the WAN part. The right panel shows the percentage of cases where at least a single upward level shift is detected when there is WAN congestion (blue) and when there is no congestion (red). It clearly shows that the likelihood of detecting at least a single level shift is significantly higher under congestion (i.e. 95% compared to 10% for srsLTE and 90% compared to 24% for OAI). Hence, an *SR* value greater than one that is accompanied with a level shift are indicative of a congested WAN.

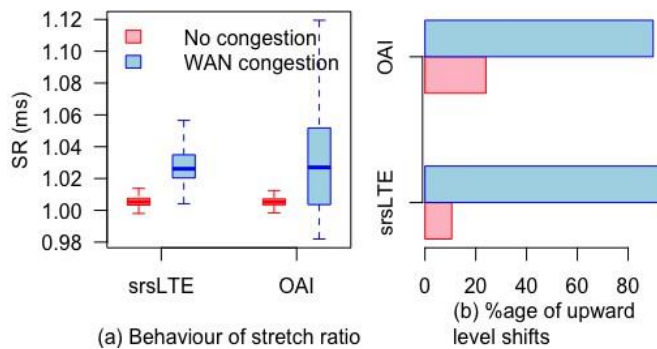


Fig. 11. Variation in values of the features when the WAN is congested.

Finally, we need to identify the combination of features that help flag and attribute congestion that occurs at different segments of the WAN.

Figure 12 shows fraction of time each of the *FH<sub>ul</sub>*, *OP<sub>ul</sub>*, *IN<sub>ul</sub>*, *OP<sub>dl</sub>* and *IN<sub>dl</sub>* records a level shift in RTT for different congestion profiles. As expected, under normal conditions these features rarely report level shifts, i.e. in the case of ‘None’ when the path is free from congestion. If congestion originates in the radio, the transport or within the EPC, *FH<sub>ul</sub>* reports a level shift in RTT close to 100% of the time. If the source of congestion is at operator’s link(s), outside EPC, then most of the time both *OP<sub>ul</sub>* and *OP<sub>dl</sub>* records an upward shift in RTT. When congestion is caused by links in the Internet, both the *IN<sub>ul</sub>* and *IN<sub>dl</sub>* capture it. We further learn from the heatmap in Figure 12 and Table VI that TSLP in the uplink direction, on its own, has the potential to identify the bottleneck part in the communication link. Here, it is worth noting that due to single links in *OP* and *IN* segments in our simplified testbed setup, the delay introduced by *netem* in EPC is also captured by *OP<sub>ul</sub>*, similarly congestion at the router

(i.e., gateway) is captured by *IN*. For the commercial network setup, we expect *OP* and *IN* segments to contain mutually exclusive links, which will avoid false positives in *OP* and *IN* segments.

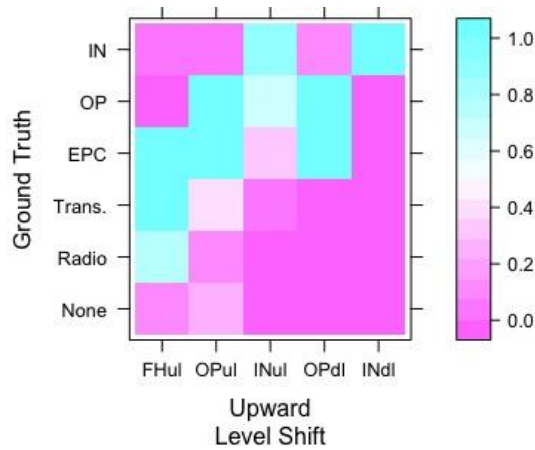


Fig. 12. Fraction of time the TSLP features observed upward level-shift in delay, with different congestion sources..

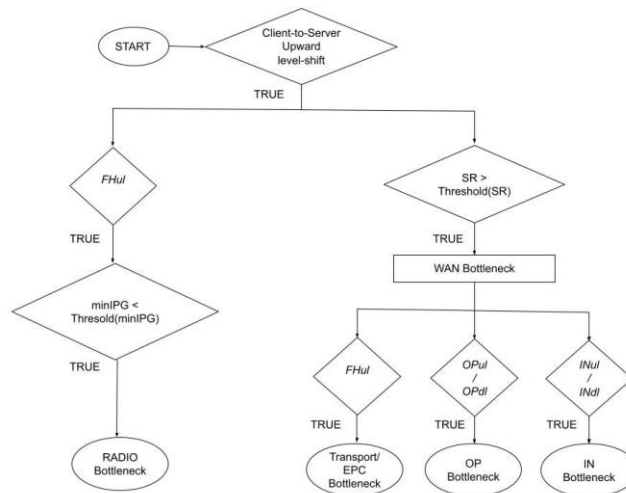


Fig. 13. Flowchart of Bottleneck Localization by Q-TSLP

The flow-chart in Figure 13, depicts the hierarchy of conditions which enables Q-TSLP inferring the bottleneck profile. *Client-to-Server Upward level-shift* denotes the union of the  $FH_{ul}$ ,  $OP_{ul}$  and  $IN_{ul}$  boolean parameters. The thresholds for  $SR$  and  $minIPG$  are heuristically decided based on experience with the network e.g. in Figure 10 we see, that according to our controlled testbeds, around 25% of the experiments have  $minIPG$  below the most recurring  $minIPG$  of around 150ms, so we set threshold to be at most 25% of the all the  $minIPG$ s of the conducted measurements. Note, for  $SR$  to infer congestion in WAN, its decision threshold must be a value greater than 1. Hence,  $SR$ 's thresholds can be adapted to the desired sensitivity. For instance, a value much greater than 1 is suitable for flagging severe congestion.

Applying the flow-chart to the experiments with 1ms *sleep time* from the two testbeds, congestion is detected with an accuracy of 87% and 82% for OAI and srsLTE testbeds, respectively. With supervised learning it is 94% and 93% respectively. The flow-chart distinguishes Radio and WAN bottlenecks with 79% and 82% of accuracy, whereas with supervised learning it has an accuracy of above 97% for the two testbeds. Lastly the source segment within WAN is identified with 97% and 90% of the accuracy, via the flow chart, for the two testbeds. With supervised learning accuracies in the last scenario improves up to 100%.

Takeaways Q-TSLP combines QProbe and TSLP. It not only improves the congestion detection accuracy, but also makes up for the shortcomings of QProbe and TSLP when used in isolation. Other than attributing congestion to its respective segment of *FH*, *OP* and *IN*, Q-TSLP can locate congestion with the *granularity of Radio* and the segment comprising both the *Transport and Core*, in *FH* part of the network.

Additionally, it discriminates composite bottlenecks with up to 40% additional accuracy. Lastly, we show that, by observing how the most important features change in presence of congestion, Q-TSLP can flag and localize congestion even without training a machine learning model.

## VI. Congestion in Commercial Networks

We now proceed to applying Q-TSLP along with the insights we have gained from the controlled experiments to detecting congestion in commercial networks.

### A. Measurement setup and methodology

We utilize the same measurement infrastructure as we presented in Sec. III-A, to investigate for persistent congestion in two commercial MBB networks, which we refer to as Op<sub>1</sub> and Op<sub>2</sub>. Overall, we employ 92 measurement nodes within which 91 are connected to Op<sub>1</sub> and 82 to Op<sub>2</sub>, over LTE. In addition, we set up two well-provisioned measurement servers that are topologically independent and connected to the Internet via distinct upstream providers. This is to control for effects specific to each server's upstream provider. We run Q-TSLP periodically from the nodes and for each test we collect a set of UE metadata to contextualize our results. These include signal quality (i.e. RSRP, RSRQ and RSSI), serving cell identifier and current frequency band. We have conducted our measurements during February, April and June-December 2020. In February and April, we ran Q-TSLP every hour, whereas in June onwards we increased the frequency to three times per hour. Each run tests congestion towards both servers. However, we did not run Q-TSLP between 09:00-10:00 a.m. and between 09:00-10:00 p.m. as the measurement nodes are not available to us during these hours due to daily maintenance activities and other ongoing measurements. Overall we have conducted around 5.2 million tests. In February each test consisted of 5 probe trains with 100 load packets each. From April onwards, each test consisted of 5 probe trains but with 15, 30, 45, 60 and 100 probe packets respectively. Note that in every test the 5 different probe trains are considered different experiments and we run them one after another to have comparable conditions.

To control for nodes in poor coverage, we remove all nodes with RSRP below -109 dBm for at least 25% of the time. This leaves us with 87 and 71 connections to Op<sub>1</sub> and Op<sub>2</sub>, respectively. In the remainder of this section, we focus on measurements from June unless otherwise stated.

#### Q-TSLP configurations

The Q-TSLP configurations for the measurements in the wild are kept similar to those of the controlled experiments. However, in the downlink direction, traceroutes do not reach the P-GW and usually stop at the ingress of the MBB operator. This is due to the strict stateful firewalling and ICMP filtering that MBB networks often follow. From the analysis of the controlled experiments, we learn that though downlink RTT level-shift features adds to the accuracy of QTSLP, the uplink features are more important and can do well alone. We therefore decide to only consider the RTT levelshift features of Q-TSLP in the uplink direction. We then determine whether each responding IP hop is a part of the measured MBB network or if it lies beyond this, i.e. between the MBB operator and the server. To this end, we perform IP to autonomous system (AS) mapping by using Team Cymru's IP to AS mapping service<sup>4</sup> as well as Whois records that are made available by regional Internet registries. Equipped with this mapping, we split each measured topology, similar to controlled experiments, into three segments: (i) First Hop (FH): between the UE and the first responding IP hop in the uplink direction, which is usually the P-GW. This hop involves the radio, transport and part of the mobile core, (ii) Operator (OP): This involves all hops between the first responding and last responding IP (inclusive) within the MBB network (iii) Internet (IN): the part of the path beyond the operator's network all the way to the measurement server.

We configure Q-TSLP to detect upward level-shifts in latency, incurred by each segment, lasting for at least two hours at a 90% confidence level. The short duration of these regimes may lead to false positives. To avoid this, we only retain upward level-shifts where the average RTT is higher than the maximum RTT of the low-level regimes.

Speed tests Besides the congestion tests, we run a speed test towards Ookala's servers.

As we have described in Sec. III, the download and upload speed measurements run three times every day at 2:00 a.m., 2:00 p.m. and 7:00 p.m. These hours correspond to different network usage profiles (i.e. off-peak, mid-day peak and evening peak), which results in non-trivial differences in performance (see Fig. 2). We use speed measurements that coincide with Q-TSLP measurements in time, to correlate speed and the detected congestion episodes.

### B. Radio Congestion

To detect radio congestion, we identify all tests where the *minIPG* is small and there is a level shift in the *FH<sub>ul</sub>*. We consider the *minIPG* as small, if it is less than the first quartile of the distribution of all received probe trains for a particular number of load packets.

#### The role of the number of load packets

To assess the effect of the number of load packets on QTSLP's performance, we compare downlink speed tests that coincide with cases that we flag as congestion with tests that do not. Figure 14 depicts this comparison. We measure a clear drop in speed across the board, where there is evidence of radio congestion. The trend is fairly robust to the number of load packets. It

---

<sup>4</sup> <https://www.team-cymru.com/IP-ASN-mapping.html>

is comparatively less robust for the lower and higher values though. In the remainder of this section, we stick to tests with 45 load packets since it gives the best performance for both operators.

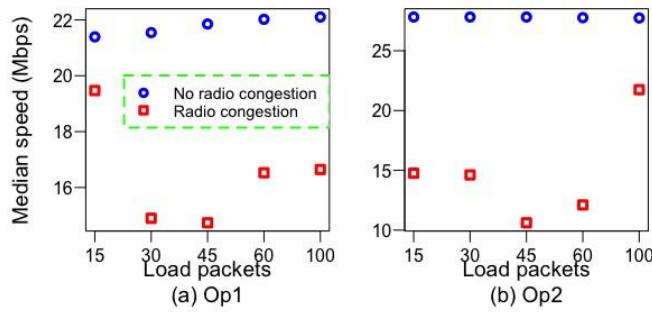


Fig. 14. Median downlink speed at measurement instances inferred as having radio congestion and no radio congestion.

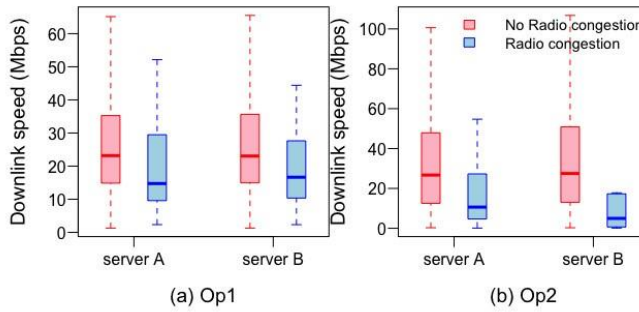


Fig. 15. Impact of Radio congestion on downlink speed.

Figure 15 compares the distribution of downlink speeds for both operators as a function of whether we detect radio congestion towards any of the two servers or not. We observe that the median downlink speed of Op<sub>1</sub> drops by 36% and 27.7% when detecting congestion towards server A and B, respectively. For Op<sub>2</sub> the drop is higher at 60% and 78% depending on the server. Although the identified Radio congestion has a substantial impact on the downlink speed, the

1  
1

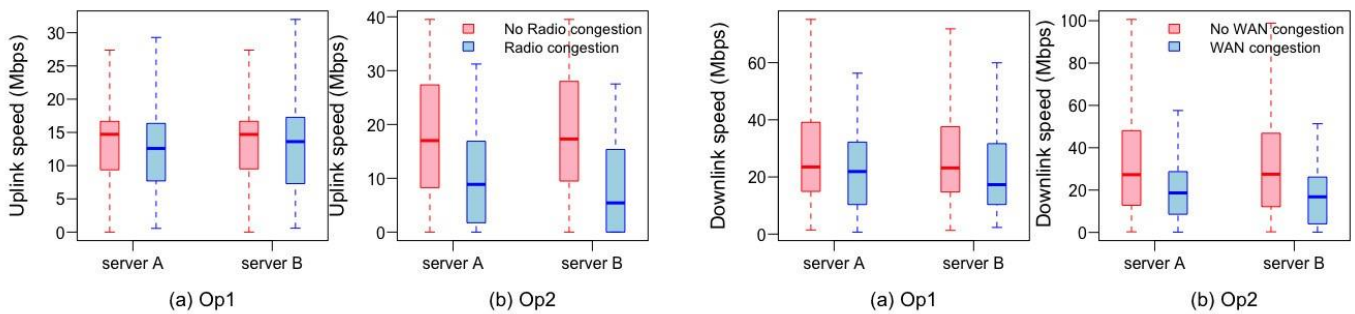


Fig. 16. Impact of Radio congestion on uplink speed.

percentage of speed tests that coincide with a congested radio is small; that is 2% and 0.75% for Op<sub>1</sub> and Op<sub>2</sub>, respectively. For all the Q-TSLP experiments, we find 1.1-1.9% of the runs for Op<sub>1</sub> and 0.47-0.76% of the runs in Op<sub>2</sub>, are impacted by radio congestion.

In a recent study V. Adarsh et al. [13] demonstrated that throughput decreases significantly, by up to 24x, with network congestion. For commercial networks, we have therefore used the correlation of congestion and downlink speed to see how well Q-TSLP points out links suffering from radio congestion. To confirm the impact congestion has on speed, we next test with uplink speed. Again, the events pointed out by Q-TSLP, to be affected by radio congestion, are associated with poor uplink speed. In Figure 16 (a) we find that, although the median difference in uplink speed for the events without and with radio

congestion is small for Op<sub>1</sub>, the first quantile uplink speed drops by 18-23% with radio congestion. For Op<sub>2</sub>, the median uplink speed drops by up to 68.6%.

### C. WAN congestion

The features of Q-TSLP that can detect congestion in the WAN comprises stretch ratio i.e.  $SR$ , and the set of uplink or downlink RTT level shift trackers mentioned in Table II. To detect WAN congestion, we use  $SR$  threshold to be  $> \max(1, 3\text{rd quantile of } SR)$ .  $SR$  value greater than 1 depicts that the gap between the first and last probe packet in the probe train at the receiving end is larger than the sending duration, meaning that it observed delay during transit. We have found that in case of some nodes most of their  $SR$  values are larger than 1, even when there is no congestion in their communication path. To avoid incorrect congestion inferences, we set  $SR$  threshold with a lower-bound of 3rd quantile of  $SR$  of the total measurements instances. Using it in conjunction with up-ward level shifts, further reduces the false positive inferences of the congestion events. To further reduce the wrongly identified WAN congestion events, we do not consider an event, to have **suffered bottleneck, if no upward level shift is recorded by any of the corresponding  $FH_{ul}$ ,  $OP_{ul}$  or  $IN_{ul}$  features.**

Figure 17 illustrates the drop in downlink speed when there is congestion in the WAN. The drop is lower than for the radio case. The median decrease in speed is 6.7-25% for Op<sub>1</sub> and 32-39% for Op<sub>2</sub>. Furthermore, a slightly higher number of downlink tests coincides with WAN congestion than in the radio case (i.e. 3.13% and 0.97% for Op<sub>1</sub> and Op<sub>2</sub>,

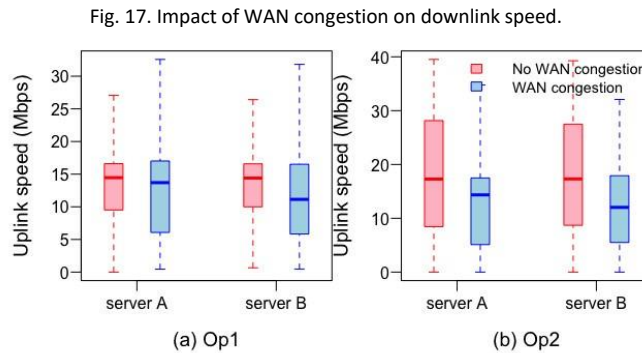


Fig. 18. Impact of WAN congestion on uplink speed.

respectively). For all the Q-TSLP experiments, we find 2.13.1% of the runs in Op<sub>1</sub> and 0.8-0.97% of the runs in Op<sub>2</sub> are impacted by WAN congestion.

Like radio congestion, we find that uplink speed is again lower for the events marked by Q-TSLP as impacted by the congestion in WAN. The median drop in uplink speed for Op<sub>1</sub> is up to 22% and for Op<sub>2</sub> it falls by 30%. These analyses confirm the congestion detection accuracy of Q-TSLP in the wild.

### D. Dissecting congestion

Attributing congestion For Op<sub>1</sub> and Op<sub>2</sub>, Q-TSLP identifies around 3.2-5% and 1.26-1.74% of the measurement runs to be suffering from congestion. We attribute all identified congestion episodes to the respective network segments. Recall that Q-TSLP allows us to group network links into: 1) radio, transport and core (FH), 2) mobile operator network (OP) and 3) the rest of the path towards the server (IN). Figure 19 depicts the distribution of the percentage of congestion events per segment per node.

The location of congestion varies widely across nodes. However, the radio congestion is one of the most prevailing. Depending on the operator, radio congestion comprises around 50% of the congestion instances suffered by the median node. The transport and core part of the MBB network seems to be the least stressed parts of the network.

There is, however, a non-trivial fraction of cases that are related to the operator's network and to the path between the operator's network and our measurement servers. Depending on the operator, these comprises between 33% and 80% of the congestion instances suffered by the median node. These findings are in agreement with the study of J. Huang et al. [14] who found that RTTs between the LTE mobile core to the Internet server are usually higher and exhibit more variance than RTTs between the RAN and core. Note that the upstream of our servers are at most one autonomous system hop away from the two operators.

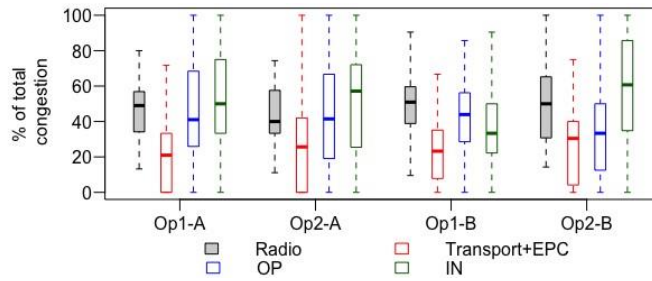


Fig. 19. Percentage of the total congestion, observed at different segments of MBB transmission path.

To narrow down potential causes for the majority of congestion cases (i.e. radio congestion), we correlate the top and bottom 25% nodes, in terms of radio congestion instances with population density. To this end, we map each node to a 1 Km x 1 Km geographic area and find the corresponding population information [15]. Figure 20 shows that nodes with excessive radio congestion are likelier to be located at densely populated areas. An obvious solution to deal with congestion, in highly populated areas, is dense deployment of small cells. An alternate solution, as pointed out by H. Deng et al. [16], is to better utilize the channel bandwidth (along with carrier aggregation) of the existing infrastructure, which otherwise go wasted because of the, still prevailing, old cell selection method used by the operators; i.e. always choosing the first scanned cell with good signal quality, rather than the one that provides a higher speed.

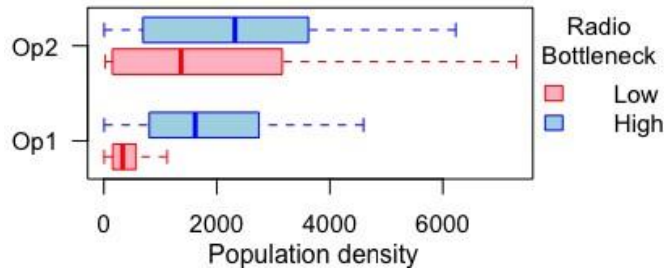


Fig. 20. Population density in 1 Km square grid of nodes with low and high Radio congestion occurrences.

Does congestion explain variations in downlink speed? Figure 21 shows that the number of congestion events gradually increases as the day progresses and decreases after 21:00 o'clock. As expected the load is low during nighttime. The increase and decrease in congestion events at peak and off-peak hours represent the diurnal pattern of congestion within the network. Table VII shows that the fraction of congestion events identified by Q-TSLP at 2:00, 14:00 and 19:00 negatively correlates with speed at the corresponding periods. These observations indicate that congestion is most likely the main reason behind the recorded degradation in downlink speed.

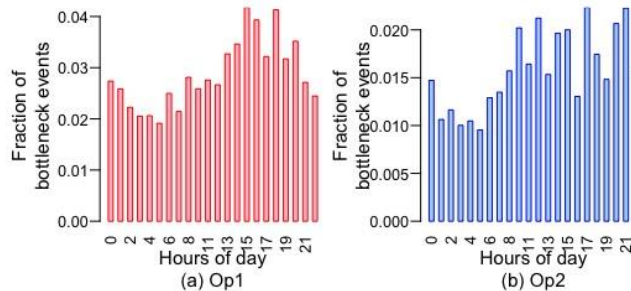


Fig. 21. Fraction of congestion events of across different hours (server A).

Duration of congestion episodes Figure 22 shows a histogram of the duration of congestion episodes. We record relatively short congestion events and with a clear difference between the two operators. Recall that we measure every 20 minutes, if a congestion event appears only once an hour, we consider it non-persistent. Such an episode could have lasted anytime between a few seconds to 20 minutes. The percentage of non-persistent episodes is over 85% and 91% for Op<sub>1</sub> and Op<sub>2</sub>, respectively. The remaining episodes are persistent lasting up to one hour and in some cases longer.

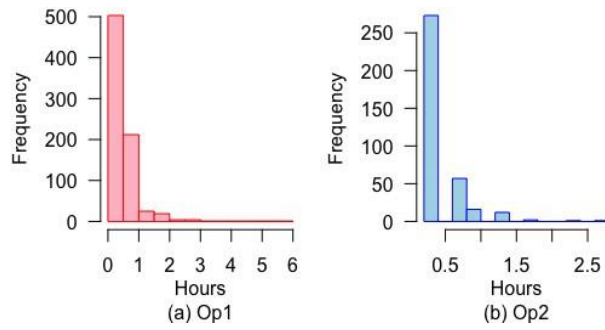


Fig. 22. Duration of the congestion events (server A).

TABLE VII  
CORRELATION OF SPEED AND CONGESTION.

Hour	Frac. of Bottleneck		Median Speed	
	<i>Op1</i>	<i>Op2</i>	<i>Op1</i>	<i>Op2</i>
2:00 a.m.	2.2%	1.1%	24.3	32.1
2:00 p.m.	3.5%	2.0%	20.0	25.9
7:00 p.m.	3.2%	1.5%	19.0	25.7

### Covid-19 and congestion

The Covid-19 outbreak in start of 2020 forced many countries to impose lockdowns that restricted the movement of people. Consequently, many people had to work and study from home, which led to a significant surge in the Internet traffic [17].

For Norway these restriction started in mid March 2020 (see Figure 26), where tele-working was imposed. Figure 26 shows that a significant drop in speed over the 1800 MHz and at 2600 MHz frequency bands for *Op1* and at 2600 MHz for *Op2*. The speed for all the bands improved in June and on wards, except in Dec 2020.

Unlike speed, Q-TSLP does not detect increase in congestion in April 2020. The reason is that Q-TSLP's decisions are based on status of network performance for the month in question. This also shows that even during the months of high demand, the networks retained their stability in performance for a particular node location.



Fig. 23. Covid-19 related restrictions started in mid March 2020, in the country.

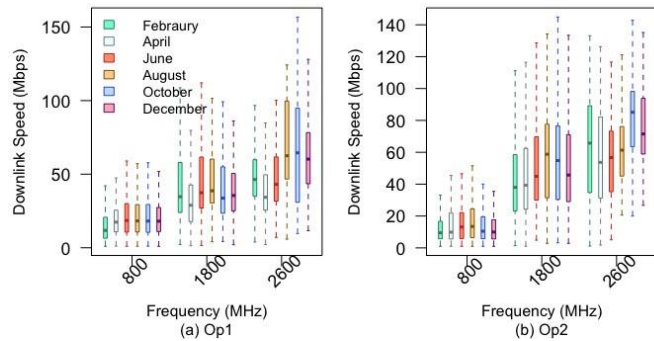
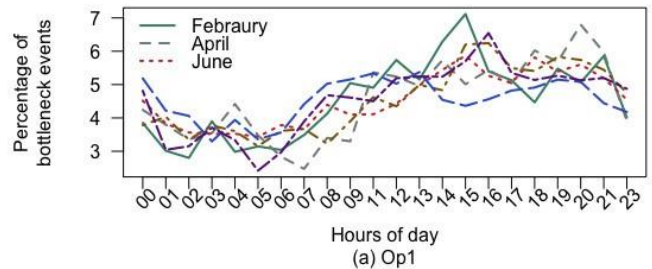


Fig. 24. Download speed in 2020.

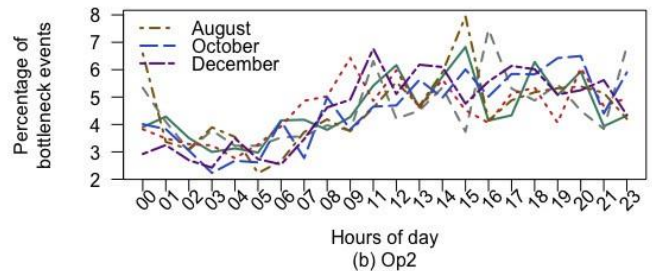


Fig. 26. Bottleneck.

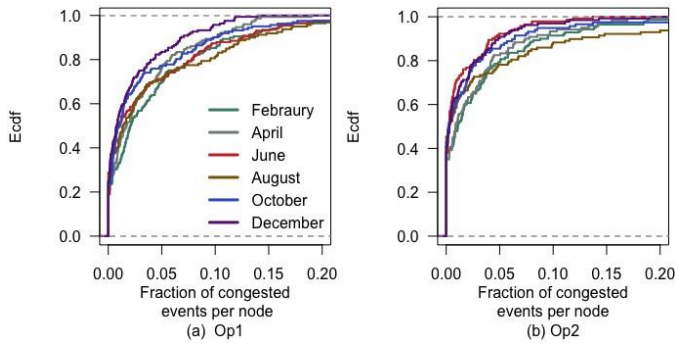


Fig. 25. Fraction of congested events per node, across different months in 2020.

Lastly Figure 26, depicts that despite different periods in 2020, the daily distribution of bottleneck events are similar in different months i.e., higher bottleneck events during peak hours when the demand is high.

Takeaways. Congestion is a common occurrence in today's mobile networks. It follows a diurnal pattern with peaks at midday and evening. While congestion in radio and last mile is common, WAN congestion remains significant. For the last mile, the detected congestion seems more related to scarce radio resources. While the deployment of 5G may ease the resource scarcity problem in the radio, operators need to upgrade their networks beyond core to address the WAN congestion. Interestingly, we find that although speeds dropped during the first wave of the Covid-19 pandemic, the Norwegian mobile networks were stable without a tangible increase in the occurrence of congestion.

## VII. Related Work

**Bottleneck detection on inter-domain paths** To localize congestion on end-to-end paths with focus on points of interconnections between ISPs, S. Sundaresan et al. [18] used network diagnostic test based on throughput measurements from MLab. Over short periods of time bulk TCP data transfers were done to measure bottleneck link by saturating it. S. Sundaresa et al. however, pointed out certain assumptions to be made to detect inter-domain congestion via the measurements: there is no congestion internal to an AS but only at interconnections; the two endpoints of the measurements are directly connected ASes; and there is only one physical link connecting them.

The studies [4], [7] measure congestion on inter-domain links attached to a local network without direct access to them. This is done using the TSLP technique [4], that frequently repeats RTT measurements from a vantage point to the near and far routers of an inter-domain link. RTTs act as a function of the routers' queue length on the forward and reverse paths: as queue lengths increase, so does RTT. When RTTs increase to the far router but not to the near router, it is inferred that a queue between these two routers induced the delay.

To identify the congested inter-domain and intra-domain links in a network, N. Hui et al. [19] developed a low-overhead active measurement tool called Pathneck. It uses a recursive packet train for probing choke points towards a destination with around 70-80% accuracy. The probe train contains a number of load packets with measurement packets of increasing TTL transmitted before, and measurement packets of decreasing TTL transmitted after the load packets. At a hop when the two measurement packets of a certain TTL at either side of load packets expire, they trigger ICMP error packets back to the sender, which then calculates packet train length for the hop. Hops with large packet train lengths are identified as choke points with their downstream links as choke links. The method cannot probe across a firewall and cannot observe high speed links.

**Radio congestion detection** J. Luciani et al. [20], defined a packet disruption technique to assess congestion in a radio cell. The work aims to push notifications and deliver content to mobile users within a radio cell when it is not-congested. The process assumes the mobile users to have a client application installed on their devices, and before a server delivers content to the devices belonging to same radio cell, a train of UDP packets with inter-packet gap information inside their payloads are sent from the server to a sampled set of clients from within the radio cell. Each sampled client assigns a congestion score to the communication link, based on the inter-packet gaps of the received packets and from the information within the incoming payloads, and reports it back to the server. Server then calculates congestion score of the radio cell by aggregating the congestion scores from its sampled clients and avoids content delivery to the radio cell if it has a high congestion score.

A recent study by A. Balasingam et al. [21] proposed BurstTracker to detect if LTE downlink is a bottleneck. It exploits the working of downlink schedulers on eNodeBs, that allocates resources to users in a way revealing if user's downlink queue runs empty during a download. It uses traces of LTE downlink control channel with QXDM [22] or MobileInsight [23] to understand

the behavior of resource blocks assignment to the user in a transmission time interval. It estimates *Throughput time* [24] by identifying start of a burst (active milliseconds) when more than 90% of resource blocks are allocated to the user and end of burst when a non-used millisecond follows an active millisecond with less than 40% of resources allocated to the user. A key issue with the method is the reliance on MobileInsight [23], which needs a device to be rooted to collect the trace log, and for QXDM the mobile device must be connected over USB to a host computer.

Instead of packet-level analysis, V. Adarsh et al. [13] present a passive approach to evaluate load in a radio cell. The method passively collects un-encrypted broadcast messages, from eNodeB, at a UE and analyzes the number of connection reject messages, the back-off time in reject messages and cell barring signals. Higher back-off timer values with higher rates of connection request denials indicate overload in a cell. The authors found that these parameters can indicate level of load in a cell and when high, the web and video streaming quality of experience deteriorates significantly. The method is passive but detects overload, which is a severe bottleneck situation, leading to depletion of resources that are critical to operate a network. End-to-end bottleneck detection Y. Xie et al. [25] recently proposed PBE-CC which detects bottleneck switching between the radio link and rest of the path and provides feedback to source to adjust its transmission rate accordingly. A mobile user snoops on cell towers and assesses cellular capacity from the control messages. The assessment is done with idle number of physical resource blocks a base station has, number of physical resource blocks allocated to the user, the competing number of mobile users and wireless physical bit rate that is derived from MCS index. When a server sends a message to the mobile device, it acknowledges back with calculated capacity and, if higher, the server increases its transmission rate. In case bottleneck happens at the non-cellular link, then packet loss and delay above certain threshold is used to detect it. The delay threshold comprises of the network's propagation delay, its jitter delay plus three times 8 ms (which is maximum delay of (re) transmitting a transport block on air interface). In the later case the server is informed to reduce its transmission speed. The process is primarily suggested to optimally use varying capacity of a cellular link and rest of the end-to-end path.

QProbe [3], on the other hand, is a lightweight active probing method to locate bottleneck link on an end-to-end path. It decides whether the bottleneck lies on the cellular last mile or on the WAN beyond. Wired path is assumed to be following FIFO queuing mechanism and cellular base station to be using Proportional fair (PF) scheduling. It uses back-to-back probe packets arrival to detect congestion at last mile and inter-packet gap between first and last received probe packets to estimate bottleneck at WAN (see section IV-A). Unlike the existing work, this paper is the first to offer both a reliable detection and attribution of congestion. We also perform a comprehensive campaign that validates our assumption about congestion detection.

## VIII. Conclusion

We have investigated the performance of two existing measurement state-of-the-art approaches for detecting congestion in mobile networks, QProbe and TSLP. To this end, we conduct a set of controlled experiments using two smallscale LTE testbeds. We find that combining these methods significantly improves detection accuracy. We then propose using a combination of them, which we call Q-TSLP, for improving both the accuracy of congestion detection and attribution.

Equipped with these findings, we conduct a three-month long study of congestion in two commercial networks. We find that persistent congestion is prevalent, which causes significant variations in download speed between peak and off-peak hours. While most congestion is related to the radio and transport network, a non-trivial fraction of congestion has causes related to the mobile core network and the Internet beyond the operator's network. This indicates that operators may need to consider upgrading their core networks to cope with expected 5G traffic volumes.

From the perspective of Q-TSLP, we intend to investigate its performance with other commonly used downlink packet scheduling algorithms such as the maximum channel quality indicator based scheduler. Secondly it is important to understand the relationship between user's quality of experience e.g. web browsing or streaming video and the detected congestion episodes.

## References

- [1] M. Polese, F. Chiariotti, E. Bonetto, F. Rigotto, A. Zanella, and M. Zorzi, "A survey on recent advances in transport layer protocols," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 4, pp. 3584–3608, 2019.
- [2] F. R. Kevin Fitchard and I. Fogg, "The 5g opportunity: How 5g will solve the congestion problems of today's 4g networks," [https://www.opensignal.com/sites/opensignal-com/files/data/reports/global/data-2019-02/the 5g opportunity report february 2019 0.pdf](https://www.opensignal.com/sites/opensignal-com/files/data/reports/global/data-2019-02/the%205g%20opportunity%20report%20february%202019%200.pdf), 2019.
- [3] N. T. Baranasuriya, V. Navda, V. N. Padmanabhan, and S. Gilbert, "Qprobe: locating the bottleneck in cellular communication," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT 2015, Heidelberg, Germany, December 1-4, 2015*, 2015, pp. 1–7.
- [4] M. J. Luckie, A. Dhamdhere, D. D. Clark, B. Huffaker, and kc claffy, "Challenges in inferring internet interdomain congestion," in *Proceedings of the 2014 Internet Measurement Conference, IMC 2014, Vancouver, BC, Canada, November 5-7, 2014*, 2014, pp. 15–22.
- [5] M. Kayali, Z. Shmeiss, H. Safa, and W. El-Hajj, "Downlink scheduling in lte: Challenges, improvement, and analysis," 06 2017, pp. 323–328.
- [6] W. A. Taylor, "Change-point analysis: A powerful new tool for detecting changes." Taylor Enterprises, Inc, 2000.
- [7] A. Dhamdhere, D. D. Clark, A. Gamero-Garrido, M. J. Luckie, R. K. P.

- Mok, G. Akiwate, K. Gogia, V. Bajpai, A. C. Snoeren, and kc claffy, "Inferring persistent interdomain congestion," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, SIGCOMM 2018, Budapest, Hungary, August 20-25, 2018*, 2018, pp. 1–15.
- [8] I. Gomez-Miguel, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "srsite: An open-source platform for LTE evolution and experimentation," *CoRR*, vol. abs/1602.04629, 2016. [Online]. Available: <http://arxiv.org/abs/1602.04629>
- [9] "The openairinterface initiative," <https://www.openairinterface.org/>, accessed: 2020-02-26.
- [10] O. W. Freedom, "Netem (network emulator)," <https://openwrt.org/docs/guide-user/network/traffic-shaping/sch-netem>, accessed: 2020-02-01.
- [11] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, 2nd ed. John Wiley and Sons, 2000.
- [12] D. R. Cutler, T. C. Edwards Jr, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler, "Random forests for classification in ecology," *Ecology*, vol. 88, no. 11, pp. 2783–2792, 2007.
- [13] V. Adarsh, M. Nekrasov, U. Paul, and E. M. Belding, "Estimation of congestion from cellular walled gardens using passive measurements," *IEEE Transactions on Mobile Computing*, no. 01, pp. 1–1, 2021.
- [14] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck, "An in-depth study of LTE: effect of network protocol and application behavior on performance," in *ACM SIGCOMM 2013 Conference, SIGCOMM'13, Hong Kong, China, August 12-16, 2013*, D. M. Chiu, J. Wang, P. Barford, and S. Seshan, Eds. ACM, 2013, pp. 363–374.
- [15] "Statistics norway: Population," <https://www.ssb.no/en/befolkning/statistikker/folkemengde>, accessed: 2020-05-01.
- [16] H. Deng, Q. Li, J. Huang, and C. Peng, "Icellspeed: Increasing cellular data speed with device-assisted cell selection," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '20. New York, NY, USA: Association for Computing Machinery, 2020.
- [17] M. Branscombe, "The network impact of the global covid-19 pandemic," <https://thenewstack.io/the-network-impact-of-the-global-covid19-pandemic>, April 2020, accessed: 21-08-2020.
- [18] S. Sundaresan, X. Deng, Y. Feng, D. Lee, and A. Dhamdhere, "Challenges in inferring internet congestion using throughput measurements," in *Proceedings of the 2017 Internet Measurement Conference, IMC 2017, London, United Kingdom, November 1-3, 2017*, 2017, pp. 43–56.
- [19] N. Hu, E. L. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "Locating internet bottlenecks: algorithms, measurements, and implications," in *Proceedings of the ACM SIGCOMM 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 30 - September 3, 2004, Portland, Oregon, USA*, R. Yavatkar, E. W. Zegura, and J. Rexford, Eds. ACM, 2004, pp. 41–54. [Online]. Available: <https://doi.org/10.1145/1015467.1015474>
- [20] J. Luciani, T. A. Lobo, and P. Balakrishnan, "Congestion detection in mobile networks and delivery of content in non-congested conditions," October 2017, uS Patent 9,807,010 B2. [Online]. Available: <https://patents.google.com/patent/US9807010B2/en>
- [21] A. Balasingam, M. Bansal, R. Misra, K. Nagaraj, R. Tandra, S. Katti, and A. Schulman, "Detecting if LTE is the bottleneck with bursttracker," in *The 25th Annual International Conference on Mobile Computing and Networking, MobiCom 2019, Los Cabos, Mexico, October 21-25, 2019*, 2019, pp. 1–15.
- [22] Qualcomm, "Qualcomm extensible diagnostic monitor," <https://www.qualcomm.com/documents/qxdm-professional-qualcommextensible-diagnostic-monitor>, accessed: 2020-03-12.
- [23] Y. Li, C. Peng, Z. Yuan, H. Deng, J. Li, and T. Wang, "Mobileinsight: Analyzing cellular network information on smartphones," *GetMobile*, vol. 21, no. 1, pp. 39–42, 2017.
- [24] 3rd Generation Partnership Project (3GPP), "Telecommunication management; key performance indicators (kpi) for evolved universal terrestrial radio access network (e-utran): Definitions," <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2025>, 2015.
- [25] Y. Xie, F. Yi, and K. Jamieson, "Pbe-cc: Congestion control via endpoint-centric, physical-layer bandwidth measurements," in *SIGCOMM '20: Proceedings of the 2020 Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication, Virtual Event, USA, August 10-14, 2020*, H. Schulzrinne and V. Misra, Eds. ACM, 2020, pp. 451–464. [Online]. Available: <https://doi.org/10.1145/3387514.3405880>