Pulsar: Improving Throughput Estimation in Enterprise LTE Small Cells

Uma Parthavi Moravapalle[#] Shruti Sanadhya[*] Abhinav Parate[*] Kyu-Han Kim[*]
[#]Georgia Institute of Technology, Atlanta, GA, USA,
[*]Hewlett Packard Labs, Palo Alto, CA, USA
[#]parthavi@gatech.edu [*]{shruti.sanadhya,abhinav.parate,kyu-han.kim}@hpe.com

ABSTRACT

With the great success of LTE(-A) outdoor, LTE-based small cell technology has become popular and is penetrating indoor enterprise environment, co-existing with WiFi networks, to provide better user experience or Quality-of-Experience (QoE). However, accurate estimation of LTE links is challenging and critical to continue providing QoE for many enterprise applications (e.g., video/audio) and services (network selection). While prior work on LTE link throughput estimation depends mostly on a single factor (e.g., link rate), we argue that it needs to consider more factors to improve the estimation to meet increasing demands on QoE. In this paper, we propose a new metric, called *Pulsar* (Per-user LTE ShAre of Resources), that estimates per flow throughput in LTE networks by leveraging both underlying channel information and application traffic characteristics. Our extensive evaluation study through ns-3 shows that *Pulsar* reduces the estimation error *more than 92%*, compared to prior work, in various scenarios, while keeping estimation overhead low.

1. INTRODUCTION AND MOTIVATION

As mobile traffic continues to grow enormously [7], enterprise access networks face unprecedented pressure. Such a pressure is attributed mainly to i) the rise in bring your own devices trends (Forrester research predicted 905 million tablets in use for work and home by 2017 [1]), and ii) the increasing adoption of unified communications(UC) systems in offices instead of wireline solutions [14]. Despite the significant advance in WiFi networks, enterprise access networks are required to support more bandwidth as well as cellular-like experience indoor.

CoNEXT '15 December 01-04, 2015, Heidelberg, Germany
 ISBN 978-1-4503-3412-9/15/12...$15.00
DOI: http://dx.doi.org/10.1145/2716281.2836105

To address such challenges, cellular service providers are proposing indoor small cell LTE deployments [5, 6] as an additional network of choice for enterprises. For example, SpiderCloud's Small Cell Services Node [13] and Airvana's OneCell [4] allow for the integration of mobile operator's small cells, co-existing with WiFi networks, while meeting enterprise specific requirements, such as quality, policies, handovers, etc. In the context of such heterogeneous enterprise access networks, a centralized network solution is needed that intelligently maps devices to the appropriate network type (LTE or WiFi), which can lead to efficient utilization of LTE and WiFi, while improving Quality-of-Experience (QoE) for individual devices. A key requirement in such a solution is to accurately estimate and monitor the network utilization, measured in terms of per user throughput. Underestimation of a user's throughput may result in underestimating the congestion on the network, whereas overestimation may result in under-utilization of the network.

Prior work has looked at throughput estimation largely for WiFi networks [21], but little work has gone into LTE networks. Atom [18] has recently been proposed for LTE throughput estimation to offload traffic to WiFi, but it uses a simplistic throughput utility model based on effective link rate in WiFi and LTE. This model serves well under the assumptions that (i) public cellular networks are dominated by video traffic and (ii) all flows are always backlogged, and hence network share of each user is entirely dependent on the best available physical link rate. However, traffic in enterprise networks consists of a mix of diverse applications, such as Voice over IP (VoIP), video conferencing, web browsing, video streaming for webcasts, email, file transfers, backup, etc. A purely physical rate based estimation does not account for such diverse application demands.

In this context, we propose a new LTE metric, called *Pulsar* (Per-user LTE ShAre of Resources), that accurately estimates per-flow throughput by accounting for application behavior. *Pulsar* is a network-side solution that sits in LTE core network and monitors LTE last-hop throughput, which often dominates the end-to-end experience of an application. In addition, *Pulsar* takes both network state information (e.g., CQI) of LTE links and application traffic pat-

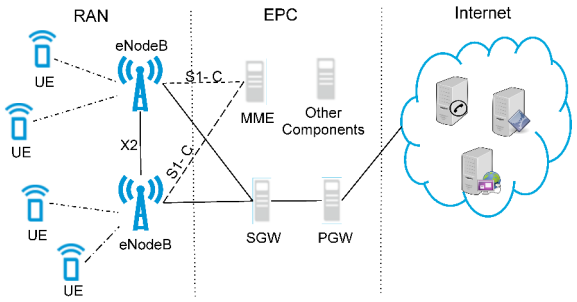


Figure 1: LTE Evolved Packet Core Network

terns to improve the estimation accuracy and maximize network utilization. To the best of our knowledge, *Pulsar* is the first work to take multiple factors into account to improve throughput estimation, and we believe it will help improve network utilization (via intelligent network selection) and manageability.

We evaluate *Pulsar* in various application environments via ns-3 simulations and show significant improvement in its throughput estimation, compared to existing approaches (e.g., Atom). Briefly, *Pulsar* shows strong correlation with the actual throughput for various applications; correlation coefficient of 0.99 for *Pulsar* vs 0.19 for Atom. In addition, *Pulsar* reduces the mean square throughput estimation error by 92.34%, compared to Atom.

In the rest of this paper, we explain LTE core networks and radio access networks in Section 2. We then present the details of *Pulsar* in Section 3, followed by its evaluation in 4. Section 5 will discuss a few related issues. Finally, Section 6 presents related work and Section 7 concludes the paper.

2. LTE PRIMER

Network Architecture: The LTE network architecture, shown in Figure 1, is the fourth generation network defined under the 3rd Generation Partnership Project (3GPP) [2]. A mobile device, referred to as User Equipment (UE), connects over licensed spectrum to an LTE base-station eNodeB. The UE and eNodeB form the radio access network (RAN), which connects to the internet through the evolved packet core (EPC). Within the EPC, a Serving Gateway (SGW) manages UE mobility and handovers across multiple eNodeBs or across multiple 3GPP standards (2G/3G). The EPC also includes a Packet Data Network Gateway (PGW) that manages mobility across 3GPP and non-3GPP networks (such as WiFi and WiMAX). PGW also serves as the IP gateway for UEs.

The UE, eNodeB, SGW and PGW form the data plane. The EPC also contains a control plane, composed of different entities performing subscriber management, authentication, billing, etc. Among these, Active Network Discovery and Selection Function (ANDSF) helps UEs discover and connect to non-3GPP networks, such as WiFi. Outdoor LTE macro cells and indoor LTE small cells primarily differ in the eNodeB capabilities, such as transmission power, number of active users supported, etc. Some enterprise controller solutions [13, 4] have been developed to integrate security and policies in LTE small cells, but none have been standardized.

Downlink Scheduling: The eNodeB is responsible for sharing physical resources (in time and frequency domain) across multiple UEs associated with it. Separate channels are used for downlink and uplink transmissions, reducing contention. We focus on downlink transmissions in this work and discuss briefly about uplink in section 5.

Within a single eNodeB, each UE is allocated time-frequency chunks in a 10 ms radio frame, by a scheduler. There are multiple scheduling algorithms, such as Round Robin (RR), Proportional Fair (PF), etc. Among these, PF scheduler is more prevalent as it can offer resource share proportional to each UE's physical link rate and past resource share. A radio frame is split into ten $subframes$, each spanning 1 Transmission Time Interval(TTI) of 1 ms. A subframe contains multiple sub-carriers, which can be allocated to different UEs, but for the purpose of our discussion we consider that one subframe is allocated completely to one UE. We explain this assumption in the next section. Every TTI, the PF scheduler [23] allocates UE $\hat{k}_s$ in subframe s if:

$$\hat{k}_s = \underset{k=1...K}{\arg\max}\, PF_k \tag{1}$$

Here PF_k is computed as:

$$PF_k = \frac{Achievable\ Rate_k}{Past\ average\ throughput_k} = \frac{R_k(s)}{T_k(s)} \tag{2}$$

$R_k(s)$ is computed from the Channel Quality Indicator (CQI) sent to eNodeB by UE k every TTI. The long-term average user throughput $T_k(s)$ is computed as an exponentially weighted moving average of $R_k(s)$ with a weight of $1/t_c$. Here, t_c is the period of fairness.

The PF scheduler provides resources to UEs that have the highest ratio of achievable rate to past achieved throughput. An important aspect of PF scheduler is that it not only favors UEs with good physical channel quality, *but also favors UEs that have been deprived in the past*. While the motivation of such a PF scheduler is to reduce starvation among contending UEs, it also prioritizes UEs with less traffic over those with more traffic. This is the key observation for our throughput estimation metric. We discuss this observation in more detail in the next section.

3. THROUGHPUT ESTIMATION ON LTE

3.1 Need for Per-UE Traffic Information

As we discussed in the previous section, the resource scheduler (i.e., PF) significantly influences the resource share per UE, which then impacts its throughput. To better explain, let us discuss the resource share distribution achieved by the PF scheduler. We consider a single eNodeB using PF scheduler. We assume that each user is running a single application flow[1] and aim to estimate the throughput of each flow accurately. Note that analyzing multiple flows per user is part

[1]Because of this assumption, we use user, UE, application and flow interchangeably in the rest of the paper.

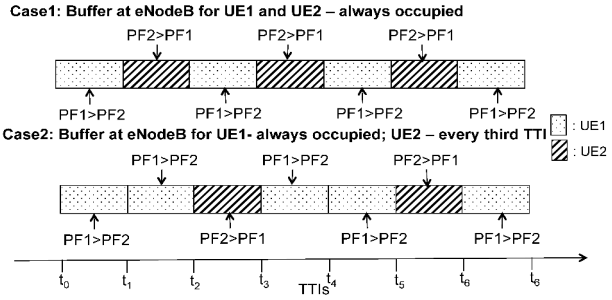


Figure 2: Scheduling example

of future work. We also assume that users are static, which is mostly the case in enterprise environments[20]. Static users do not experience frequency-selective fading effects, i.e. channel conditions are same across all sub-carriers within a subframe for a particular UE. In such scenarios, an entire subframe is assigned to a user in a single TTI.[2]

Consider two UEs – UE1 & UE2 – are associated with an enterprise small cell eNodeB, and are communicating with two remote hosts - A & B, respectively. Assume that both the UEs experience the same channel conditions i.e. CQIs are the same. Consider the following two cases, which refer to the timeline in Figure 2:

Case 1: Both the UEs are always backlogged: Here, eNodeB always has data to be scheduled for UE1 and UE2. At every TTI, the PF scheduler computes PF_1 and PF_2 from equation (2). The relation between PF_1 and PF_2 alternates. Transmissions to UE1 and UE2 are hence scheduled alternately, and both get an equal share of subframes.

Case 2: UE1 is backlogged and UE2 is not: Say, eNodeB has data to send to UE2 once every 3 TTIs. In this case, eNodeB has data buffered for UE2 only at t_2 and t_5. At both these instants, $PF2 > PF1$, as it has not been scheduled in the previous TTI, and transmission to UE2 is scheduled. Thus, UE2 is scheduled whenever eNodeB has data buffered for it and UE1 gets all the other resources. Overall, UE2 gets $1/3^{rd}$ subframes while UE1 gets $2/3^{rd}$.

Note that the diverse mix of applications in enterprise networks more often creates Case 2. For example, UE2 could be in a VoIP audio call while UE1 is downloading a large presentation or watching a live webcast. In such mixed scenarios, per-UE traffic information is essential to accurately estimate the resource share of each UE. This resource share can then be used to accurately estimate the per-user throughput from within the network infrastructure. We next describe a traffic-aware metric for resource share in enterprise LTE small cells.

3.2 Pulsar

Motivated by the above observation, we now present a mathematical model for LTE resource share with mixed application demands. Based on this model, we will define a throughput estimation metric, which we call Per User Lte ShAre of Resources (PULSAR).

[2]In the rest of this section, we refer to subframe as the smallest resource assigned to a UE.

Let us consider that k UEs are associated with an eNodeB. For each UE_i, we define the following terms:

- P_i: Average size of application packets arriving at eNodeB for UE_i
- CQI_i: Channel quality indicator (CQI) for the link between UE_i and eNodeB
- TBS_i: Transport Block Size (TBS) for UE_i, i.e. the number of bytes that can be sent in one TTI in current channel conditions. This is computed from CQI_i.
- arr_i: Number of packets arriving at eNodeB for UE_i in one TTI
- n_i: Number of resources required to send one packet to UE_i, given by $n_i = \lceil P_i/TBS_i \rceil$. This accounts for fragmentation at the physical layer.
- X_i : Number of resources demanded by UE_i per TTI, i.e.: $X_i = arr_i * n_i$

If we consider that every UE is backlogged, i.e., it has data to send all the time, the network is shared evenly and each UE gets $1/k$ resources, as in Case 1 above. However, if all UEs are not backlogged, some UEs may not have packets to receive in their fair share of resources. In such a mixed scenario, we can classify UEs into two sets:

- Low rate UE set LR, if $X_i <= 1/k$
- High rate UE set HR, if $X_i > 1/k$

Now recall that the PF scheduler allocates resources based on the ratio of achievable rate to past rate. The UEs belonging to LR have an empty downlink queue at the eNodeB most of the time, leading to a low past rate. Such low rate UEs will get the resources whenever they have data buffered at the eNodeB. The backlogged flow for high rate UEs get a fair share of the remaining resources, in effect getting more than the even share of $1/k$. Based on this classification we define a subframe share SF_i, which is computed as:

$$SF_i = \begin{cases} X_i & \text{, if } UE_i \in LR \\ \min(\ X_i, \frac{1}{|HR|}(1 - \sum_{LR} X_i)) & \text{, if } UE_i \in HR \end{cases} \quad (3)$$

Given that TBS_i is the number of bytes that a UE can send in one TTI, the maximum resource share for UE_i can be computed as $SF_i * TBS_i$. Recall that $n_i \geq 1$ by definition, as it assumes that P_i is at least as large as TBS_i. However, for some applications P_i may be smaller than TBS_i. In such cases, even though the entire subframe is allocated to UE, the data transferred in it is less than the TBS_i. The resource share for UE_i in this case is $SF_i * P_i$.

Based on the above definition, we present our resource share estimation metric, $Pulsar$, that is defined as:

$$Pulsar_i = SF_i * \min(TBS_i, P_i) \quad (4)$$

We estimate throughput as a linear function of $Pulsar$.

$$Throughput_i = \begin{cases} C_1^{hr} * Pulsar_i + C_0^{hr} & \text{, if } UE_i \in HR \\ C_1^{lr} * Pulsar_i + C_0^{lr} & \text{, if } UE_i \in LR \end{cases} \quad (5)$$

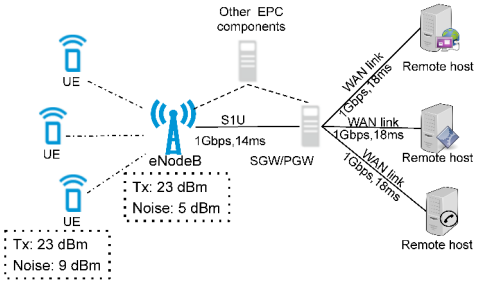


Figure 3: ns-3 LTE simulation topology

, where $C_1^{hr}, C_1^{lr}, C_0^{hr}$ and C_0^{lr} are constants. While we use a linear function as in previous work (e.g., Witt [21]), our estimation differs in that it is computed separately for UEs in HR and LR. This prevents the high variance in throughput estimation of UEs in HR from affecting the UEs in LR.

3.3 Deployment Model

Pulsar requires per-UE traffic and channel information for the throughput estimation. We envision that *Pulsar* can be deployed inside the enterprise small cell network and provide throughput estimation information to network services like network selection. For example, Small Cell Services Node (SCSN) [13] allows for monitoring the average packet arrival rate arr_i and the average packet size P_i. The SCSN can also collect the CQI_i from small cell eNodeB [8] every TTI. The linear constants C_1^{hr}, C_1^{lr}, C_0^{hr} and C_0^{lr} can be learnt over time and are not sensitive to the mix of training set, as shown in section 4.

4. EVALUATION

We use ns-3 simulator [10] to evaluate Pulsar. The simulation topology is shown in Figure 3. It consists of an LTE/EPC network and many remote hosts, connected to the SGW/PGW with direct wide area network (WAN) links. Multiple UEs are connected to a single eNodeB and each UE communicates with a different remote host. The parameters used in simulations (shown in Figure 3) represent real enterprise indoor LTE environments [8]. For most of the simulations, WAN delay and bandwidth are taken from [19]. Later in this section, we also show that throughput estimation with *Pulsar* is not sensitive to WAN delay and bandwidth.

4.1 Application Profile

In this simulation scenario, we consider six application profiles which represent a diverse set of applications seen in enterprise environments. We model the downstream traffic of each application as follows:

- Web Browsing: A web browsing session typically involves the browser downloading a main html file, parsing it for embedded objects like Javascript, CSS, images etc. and downloading each of these embedded objects. In our simulations, the UEs use HTTP/1.1 persistent mode to download these objects. We generate the main object size, number of embedded objects, embedded object sizes and the parsing time, randomly according to the traffic model in [3].
- VoIP: We assume that the VoIP session uses G.711 audio codec, in which a remote host/VoIP server, sends constant bit rate traffic, at the rate of 156Kbps [11]. The audio payload size is set to 160 Bytes.
- Video Conferencing: We model video conferencing downlink traffic as a constant bit rate traffic (rate = 800Kbps), assuming that the UE uses H.264/RT Video codec with a resolution of 640x360 or 640x480 [11].
- Video Streaming: We assume that a UE is streaming a HD 720p/1080p video from a remote host. Related literature has studied the nature of YouTube streaming traffic. Particularly [22] shows that, in steady state during a video streaming session, the server throttles the download rate of a video at 1.25 times the encoding rate. Therefore, we generate a constant bit rate traffic to model video streaming, with bitrates 1.25 times a random encoding rate, between 1500Kbps and 6000 Kbps [9].
- Desktop Sharing: We model this application as a series of bursts of traffic to the UE. In each burst, packets are generated at a constant rate of 981 Kbps [12]. The ratio between the duration of a burst and the inter-burst interval is set to 1:4.
- FTP Download: The remote host generates application packets at a high rate of 20 Mbps and sends them to the client.

Unless otherwise specified, we set the payload size of the application to 1400 Bytes.

4.2 Dataset Generation

Each simulation scenario consists of 10 UEs associated to an eNodeB. Each UE randomly chooses whether to communicate with a remote host or not. If a UE chooses to communicate, it randomly picks one of the six applications described in 4.1, with equal probability. These applications represent a diverse set of applications seen in enterprise environments. Each application starts at either 0^{th}, 15^{th}, 30^{th} or 45^{th} seconds into the simulation, with equal probability, and generates bursty traffic for a randomly chosen multiple of 15s.

The transport layer protocol used by the application is also randomly chosen among: TCP New Reno, TCP Westwood and UDP. The CQI value for a UE is fixed for the duration of the simulation and is randomly chosen between 2 (very poor) and 15 (very good). At every 15s, we track the average number of packets arriving in one TTI (arr_i) and the average packet size (P_i) in SGW/PGW. We also measure the actual throughput and resource shares for each application. We generate a large dataset by simulating the described scenario 1000 times (each lasting 60s), with different random seeds. The generated dataset has 16399 throughput samples.

4.3 Correlation analysis

We first measure the Pearson correlation of observed throughput with *Pulsar* as well as other metrics. It is a measure of linear dependence of two variables, defined as: $\rho_{X,Y} = \frac{COV(X,Y)}{\sigma_X \sigma_Y}$; $-1 \leq \rho_{X,Y} \leq 1$; where X and Y are random variables, COV is the covariance and σ is the standard deviation. The values +1, -1 and 0 represent total positive, total negative and no correlation, respectively. To compute Pearson correlations, we consider the following metrics, in addition to *Pulsar*:

(a) *Single-User* (TBS_i): Assuming there is only one user connected to eNodeB, resource share for this user is 1 and the throughput is directly proportional to TBS_i.

(b) *Sender-Rate* (arr_i*P_i): Assuming that all packets sent by the remote host are received at the UE without any delay, throughput is directly proportional to the sender rate.

(c) *Atom* (as defined in [18]): Resources are shared equally among all active users, irrespective of the application load.

Metric	Single-User	Sender-Rate	Atom	Pulsar
Correlation	0.2568	0.5810	0.2373	0.9988

Table 1: Pearson correlation with observed throughput

We compute the Pearson Correlation $\rho_{X,Y}$, with X as the observed throughput and Y as one of the metrics above. We can clearly observe in Table 1 that *Pulsar* has a correlation value close to 1 and outperforms other metrics. This justifies the assumption that throughput is directly proportional to it. *Sender-rate* has high correlation for low rate senders and low correlation for high rate senders. We can see that *Sender-rate* correlates better than *Single-user* and *Atom* since low rate senders form 65.5% of all senders in our dataset.

Application	ρ_{pulsar}	ρ_{atom}	ρ_{sr}
Web Browsing	0.9999	0.0276	0.9999
VoIP	0.9999	-0.0379	0.9999
Video Conferencing	0.9976	0.0379	0.9801
Video Streaming	0.9985	0.4726	0.7288
Desktop Sharing	0.9903	-0.0107	0.9903
FTP Download	0.9980	0.6102	0.3403

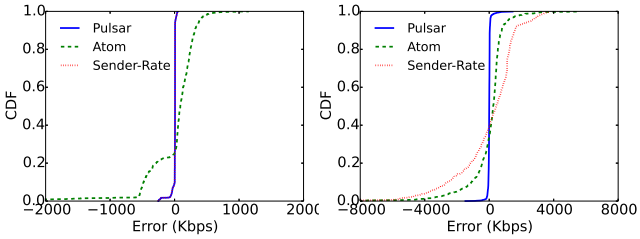
Table 2: Correlation for individual applications

We next compute the Pearson correlation of individual application throughput with $Pulsar$ (ρ_{pulsar}), *Sender-Rate* (ρ_{sr}), and with $Atom$ (ρ_{atom}) (shown in Table 2). For all applications, ρ_{pulsar} is close to 1, while ρ_{atom} is low, and in some cases negative. ρ_{sr} is high for Web Browsing, VoIP, Video Conferencing and Desktop Sharing. This is because, these applications have a low data rate, and the UEs are almost always classified as low rate and have a throughput equal to the sender rate. However, for high data rate applications like FTP Download and Video Streaming, the correlation is low. We also compute the correlation between SF_i computed by *Pulsar* in equation (3) and the actual observed subframe share. We observe that for all applications, the correlation is over 0.9, contributing to high ρ_{pulsar}.

We also evaluate ρ_{pulsar} for individual transport protocols considered in our simulations and observe that the correlation ρ_{pulsar} is over 0.99 for all the protocols.

Metric	C_1^{lr}	C_0^{lr} (Kbps)	C_1^{hr}	C_0^{hr} (Kbps)
Pulsar	0.97	9.09	0.97	0.88
Atom	0.07	144.68	1.20	445.95
Sender-Rate	0.97	9.09	0.12	1783.87

Table 3: Linear model coefficients



(a) low rate **(b) high rate**

Figure 4: CDF of throughput estimation error

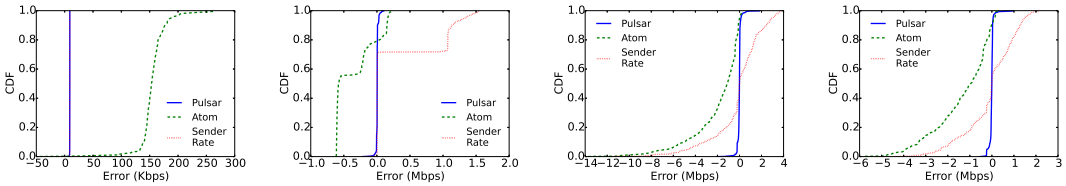
4.4 Throughput Estimation

In this section, we try to fit a linear model, using ordinary least squares regression, to estimate throughput from *Sender-Rate*, *Atom* or *Pulsar*. We chose a linear model as *Pulsar* shows high Pearson correlation with the ground-truth throughput[3]. As described in section 3, separate models are computed for high rate senders (C_1^{hr}, C_0^{hr}) and low rate senders (C_1^{lr}, C_0^{lr}). 20% of the dataset is used for training the linear estimator, and the rest is used for testing. Table 3 shows these values for *Atom*, *Sender-Rate* and *Pulsar*.

Figure 4 shows the CDF of estimation errors for high rate and low rate applications with *Pulsar*, *Sender-Rate* and *Atom*. We compute estimation error as difference in estimated throughput and observed throughput. For low rate applications, 10th and 90th percentile errors for *Atom* are -481.26 Kbps and 313.50 Kbps, respectively, where as for *Pulsar* and *Sender-Rate*, they are -2.29 Kbps and 7.78 Kbps. Note that, for low rate applications, estimations with *Pulsar* and *Sender-Rate* are the same. *Atom* is not an accurate metric for low rate applications because its estimation errors are high. Estimation errors that are over a few Kbps are unacceptable for low rate applications, since their average datarate is low.

For high rate applications, 10th and 90th percentile errors for *Pulsar* are -50.38 Kbps and 36.34 Kbps, respectively, where as for *Sender-Rate*, they are -2584.39 Kbps and 1599.92 Kbps, respectively and for *Atom*, they are -1096.8936 Kbps and 780.75 Kbps. We can see that the throughput estimation with *Pulsar* is accurate and outperforms both *Atom* and *Sender-Rate*. On the testing dataset, compared to *Atom*, $Pulsar$'s root mean square (RMS) estimation error is lower by 90.88% for low rate applications and by 87.80% for high rate applications. Also, for high rate applications, compared to *Sender-Rate*, *Pulsar*'s RMS esti-

[3]While we chose a linear model for *Pulsar* based on high correlation, the same was chosen for *Atom*, based on description in [18].



(a) Web Browsing **(b) Video Conferencing** **(c) FTP Download** **(d) Video Streaming**

Figure 5: CDF of throughput estimation error for different applications

mation error is 92.83% lower.

Figure 5 shows the CDF of estimation errors for Web Browsing, Video Conferencing, FTP Download and Video Streaming. The CDF for VoIP and Desktop Sharing are also similar. While the estimation error stays close to zero for *Pulsar* and *Sender-Rate* for low rate applications like Web Browsing and VoIP, it is high for *Atom*. Specifically, for low rate applications like VoIP, Video Conferencing and Desktop Sharing, *Atom* overestimates the throughput. For high rate applications like FTP download and Video Streaming, both *Atom* and *Sender-Rate* have a significantly high estimation error and they underestimate the throughput. *Pulsar* achieves 81%-96% reduction in the RMS estimation error, compared to *Atom*, for all applications. Also, *Pulsar* achieves 86%-97% reduction in the RMS estimation error, compared to *Sender-Rate*, for all applications other than Web Browsing. Throughput estimation for Web Browsing is the same for both *Pulsar* and *Sender-Rate*, because UEs using this application are always classified as a low rate UEs in the dataset.

4.4.1 Sensitivity analysis

Statistical estimation techniques can be sensitive to the type of data used for training purposes. In this section, we present the sensitivity of *Pulsar* to varying parameters. Due to lack of space, we only present results for high rate senders. Results for low rate senders look similar.

In the simulations so far, WAN has not been the bottleneck for any flow. However, it could affect throughput for TCP applications. For sensitivity analysis, we measure the estimation error of $Pulsar$ with varying WAN delay (Figure 6a) and bandwidth (Figure 6b). For each value of delay/bandwidth, we generate a testing dataset of about 1500 samples. We observe that estimation error for $Pulsar$ changes minimally with such variations, unlike $Atom$ and *Sender-Rate*. *Pulsar* indirectly accounts for the WAN by monitoring the packet arrival rate arr_i, thus, making it robust to changes in WAN.

The number of samples used for training the linear estimator could affect the estimation error. Figure 6c shows the variation of estimation error with the percentage of the dataset used for estimating linear coefficients. The rest of the dataset was used for testing. There is no significant variation even when only 20% of the data was used for training.

5. DISCUSSION

In this section we discuss some related issues and future extensions of *Pulsar*.

Static users: One of the assumptions in modeling *Pulsar* is that entire subframe is assigned to a single UE. When users are moving, different sub-carriers within a subframe may experience different fading effects, and they may be assigned to different users. In enterprise environments, users do move across multiple indoor locations, e.g. work desk, conference room, cafeteria, but they stay static while in these locations[20]. This supports our assumption that UEs are not in motion. In future, we plan to extend the model to include the small portion of highly mobile users.

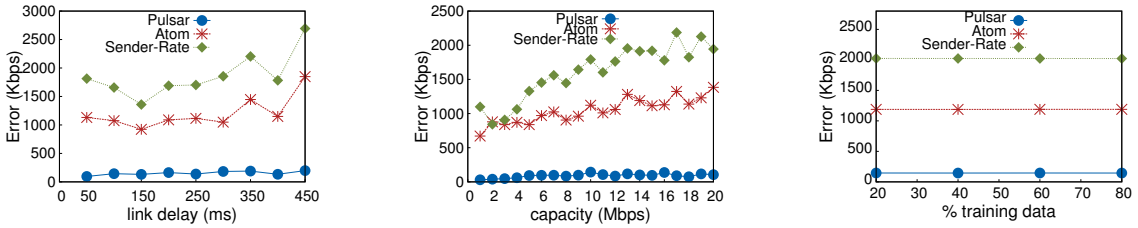
Interference: An enterprise may have multiple small cells which can cause inter-cell interference, affecting individual user throughput. Efficient channel assignment techniques can be used to reduce such interference. Additionally, macro LTE cells can interfere with small cell deployments. If the channel quality of a UE changes drastically due to interference, *Pulsar* might classify a UE differently from the PF scheduler, which has a longer history for the UE. But in a few TTIs, both of them will converge, as $T_k(s)$ (equation (2)) is an exponentially weighted moving average. We plan to investigate this further in future work.

Multiple flows per UE: We consider one flow per UE in our analysis, for simplicity. Users might be running more than one application at a given time, creating multiple flows. In such case, *Pulsar* estimates the effective throughput, which is a weighted average of all individual flows, as it does not distinguish per-flow packets. Such an estimate is still useful for network management and selection solutions to measure user's overall experience and network usage. We plan to evaluate such scenarios in future work.

Simulation traffic: While VoIP and video conferencing are bi-directional applications, for simplicity, we simulate only uni-directional flows for these applications in section 4. The uplink traffic of these applications would not affect the existing analysis as downlink and uplink transmissions occur over separate dedicated channels in LTE.

6. RELATED WORK

Patro et al.[21] proposed a metric to estimate TCP throughput in WiFi network by observing channel interference, contention and physical link rate. In contrast, our work fo-



(a) Variations with delay **(b) Variations with bandwidth** **(c) Variation with training set size**

Figure 6: Sensitivity of estimation error to different factors

cusses on estimating throughput in LTE networks, which have very different characteristics. Other works have focussed on building QoE metrics for various applications. Authors in [16] proposed a QoE metric for Skype application whereas Prometheus[15] looked at QoE for video and VoIP applications. These QoE metrics cannot be generalized to the diverse set of applications encountered in an enterprise environment. Recently, Delphi[17] and ATOM[18] looked at the problem of selecting the best network interfaces for user devices where they argue that the network-choice decision should be made using metrics that capture network utilization. In our evaluation, we have made a detailed comparison with the metric used by ATOM. Delphi does not specify any metric to estimate the network utilization.

7. CONCLUSION

We argue that an accurate estimation of LTE network usage is important to provide better QoE to small cell users. To this end, we proposed a new metric - *Pulsar*, that computes per user network resource share from channel conditions and application demand. We extensively evaluated *Pulsar* with the ns-3 simulator and showed that, compared to prior work, *Pulsar* reduces estimation error by over 92%. Validation of our metric in real testbed is part of future work.

8. REFERENCES

[1] 2013 Mobile Workforce Adoption Trends. https://www.vmware.com/files/pdf/Forrester_2013_Mobile_Workforce_Adoption_Trends_Feb2013.pdf.

[2] 3rd generation partnership project. www.3gpp.org/.

[3] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Feasibility Study for OFDM for UTRAN enhancement; (Release 6). ftp://www.3gpp.org/tsg_ran/TSG_RAN/TSGR_24/Docs/PDF/RP-040221.pdf.

[4] Airvana OneCell System. http://www.airvana.com/products/enterprise/onecell/.

[5] Alcatel Lucent: In-Building Solutions. https://www.alcatel-lucent.com/solutions/in-building.

[6] Cisco Universal Small Cell Solution. http://www.cisco.com/c/en/us/solutions/service-provider/small-cell-solutions/index.html.

[7] Cisco Visual Networking Index, February 2015. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf.

[8] ip.access Small Cells. http://www.ipaccess.com/en/smallcells/.

[9] Live encoder settings, bitrates, and resolutions. https://support.google.com/youtube/answer/2853702?hl=en.

[10] ns-3. www.nsnam.org.

[11] Plan network requirements for Skype for Business 2015. https://technet.microsoft.com/en-us/library/gg425841.aspx.

[12] Remote Desktop Protocol Performance. http://download.microsoft.com/download/4/d/9/4d9ae285-3431-4335-a86e-969e7a146d1b/RDP_Performance_WhitePaper.docx.

[13] SpiderCloud: Scalable Small Cells for Reliable Indoor Coverage. http://www.spidercloud.com/small-cell-ran.

[14] What are the tangible benefits of adopting UC? http://www.voss-solutions.com/news/blog/2014/Tangible-Benefits-of-Adopting-UC/.

[15] AGGARWAL, V., HALEPOVIC, E., PANG, J., ET AL. Prometheus: Toward quality-of-experience estimation for mobile apps from passive network measurements. In *Proc of the HotMobile* (2014).

[16] CHEN, K.-T., HUANG, C.-Y., ET AL. Quantifying skype user satisfaction. In *Proc. of the SIGCOMM* (2006).

[17] DENG, S., SIVARAMAN, A., AND BALAKRISHNAN, H. All your network are belong to us: A transport framework for mobile network selection. In *Proc. of the HotMobile* (2014).

[18] MAHINDRA, R., VISWANATHAN, H., SUNDARESAN, K., ET AL. A practical traffic management system for integrated lte-wifi networks. In *Proc. of the 20th MobiCom* (2014).

[19] NGUYEN, B., BANERJEE, A., GOPALAKRISHNAN, V., ET AL. Towards understanding tcp performance on lte/epc mobile networks. In *Proc. of the AllThingsCellular* (2014).

[20] OWEN, N., SPARLING B., P., ET AL. Sedentary behavior: Emerging evidence for a new health risk. In *Mayo Clinic Proceedings*.

[21] PATRO, A., GOVINDAN, S., AND BANERJEE, S. Observing home wireless experience through wifi aps. In *Proc. of the MobiCom* (2013).

[22] RAO, A., LEGOUT, A., ET AL. Network characteristics of video streaming traffic. In *Proc. of the seventh ACM CoNEXT* (2011).

[23] SESIA, S., TOUFIK, I., AND BAKER, M. *LTE, The UMTS Long Term Evolution: From Theory to Practice*. Wiley Publishing, 2009.