# 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, [*]HP Labs, Palo Alto, CA, USA
[#]parthavi@gatech.edu [*]{shruti.sanadhya,abhinav.parate,kyu-han.kim}@hp.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%* in various scenarios, while keeping estimation overhead low.

## 1. INTRODUCTION AND MOTIVATION

As mobile traffic continuous to grow enormously [6], 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 [10]. Despite the significant advance in WiFi networks, enterprise access networks are required to support more bandwidth as well as cellular-like experience indoor.

To address such challenges, cellular service providers are proposing indoor small cell LTE deployments [4, 5] as an additional network of choice for enterprises. For example, SpiderCloud's Small Cell Services Node [9] and Airvana's OneCell [3] 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 [16], but little work has gone into LTE networks. Atom [14] 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 patterns 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,

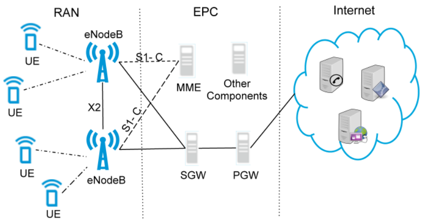


**Figure 1: LTE Evolved Packet Core Network**

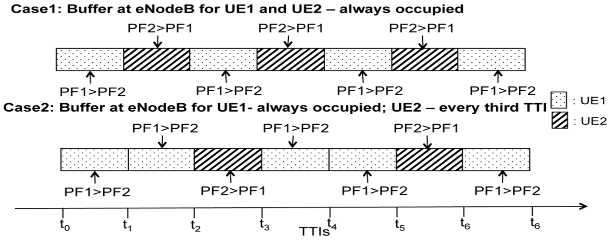

Case1: Buffer at eNodeB for UE1 and UE2 – always occupied
PF2>PF1
PF2>PF1
PF2>PF1
PF1>PF2
PF1>PF2
PF1>PF2
PF1>PF2
Case2: Buffer at eNodeB for UE1- always occupied; UE2 – every third TTI
: UE1
: UE2
PF1>PF2
PF1>PF2
PF2>PF1
PF1>PF2
PF2>PF1
PF1>PF2
PF1>PF2
t0
t1
t2
t3
t4
t5
t6
t6
TTIs

**Figure 2: Scheduling example**

*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 constitute the data plane. In the control plane, different entities in the EPC perform 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. Indoor LTE small cells deployments have a similar architecture. Some enterprise controller solutions [9, 3] 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 subcarriers, 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 [17] allocates UE $\hat{k}_s$ in subframe $s$ if:

$$\hat{k}_s = \arg\max_{k=1\ldots K} PF_k \tag{1}$$

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

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

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

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 (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. In Case 2, UE2 is scheduled whenever eNodeB has data buffered for it and UE1 gets all the other resources. Thus, UE2 gets $1/3^{rd}$ subframes while UE1 gets $2/3^{rd}$.

Note that the diverse mix of applications in enterprise networks more often create Case 2, and it demonstrates that the use of per-UE traffic information on throughput estimation greatly helps in accurately estimating resource share.

## 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).

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:

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

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 higher layer packets are 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:

$$Pulsar_i = SF_i * \min(TBS_i, P_i) \tag{4}$$

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

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

, 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 [16]), 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, the Small Cell Serving Node allows for monitoring the average packet arrival rate $arr_i$ and the average packet size $P_i$ [9]. The Serving Node can also collect the $CQI_i$ from Small Cell eNodeB [7] every TTI. The linear constants $C_1^{hr}$, $C_1^{lr}$, $C_0^{hr}$ and $C_0^{lr}$ can be learnt over time and as shown in section 4.

# 4. EVALUATION

We use ns-3 simulator [8] 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 enter-

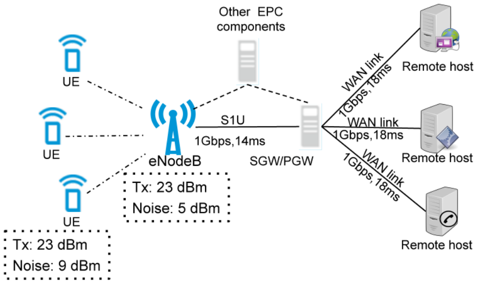


**Figure 3: ns-3 LTE simulation topology**

prise indoor LTE environments [7]. For most of the simulations, WAN delay and bandwidth are taken from [15]. Later in this section, we show that throughput estimation with *Pulsar* is not sensitive to WAN delay and bandwidth.

## 4.1 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 shown in Table 1, with equal probability. These applications represent a diverse set of applications seen in enterprise environments. We model an application as a series of bursts of traffic from a remote host to a UE. In each burst, packets of size $P$ are generated at a constant rate $r$. The duration of a burst $t_{on}$ and the inter burst interval $t_{off}$ are chosen from an exponential probability distribution with averages shown in Table 1. These parameters are computed from real packet traces collected from an enterprise network. 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 user 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 15097 throughput samples.

## 4.2 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 $\sigma$ 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* ($arr_i$*$TBS_i$): 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 [14]): Resources are shared equally among all active users, irrespective of the application load.

| Application | $r$ | $P$(bytes) | $t_{on}(s)$ | $t_{off}(s)$ |
|---|---|---|---|---|
| VoIP | 75Kbps | 150 | 15s | 0s |
| Video Conference | 300Kbps | 1024 | 0.135s | 0.1s |
| Desktop Sharing | 1.7 Mbps | 1300 | 2s | 0.1s |
| Whiteboard | 20 Kbps | 200 | 0.25s | 0.1s |
| Video Streaming | 2 Mbps | 1400 | 15s | 0s |
| Bulk Download | 20 Mbps | 1400 | 15s | 0s |

**Table 1: Applications used in the simulation**

We can clearly observe in Table 2 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 64.59% of all senders.

| Metric | Single-User | Sender-Rate | Atom | Pulsar |
|---|---|---|---|---|
| Correlation | 0.2156 | 0.7365 | 0.1962 | 0.9993 |

**Table 2: Pearson correlation with observed throughput**

| Application | $\rho_{pulsar}$ | $\rho_{atom}$ |
|---|---|---|
| VoIP | 0.9867 | -0.0651 |
| Video Conference | 0.9977 | -0.0172 |
| Screen Share | 0.9965 | 0.1716 |
| Whiteboard | 0.9714 | -0.0159 |
| Video Streaming | 0.9985 | 0.2061 |
| Bulk Download | 0.9989 | 0.6407 |

**Table 3: Correlation for individual applications**

We next compute the Pearson correlation of individual application throughput with $Pulsar$ ($\rho_{pulsar}$), and with $Atom$ ($\rho_{atom}$) (shown in Table 3). For all applications, $\rho_{pulsar}$ is close to 1, while $\rho_{atom}$ is low, and in some cases negative. 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 0.99, contributing to high $\rho_{pulsar}$.

| Protocol | $\rho_{pulsar}$ | $\rho_{atom}$ |
|---|---|---|
| UDP | 0.9988 | 0.2199 |
| TCP New Reno | 0.9998 | 0.1816 |
| TCP Westwood | 0.9998 | 0.1846 |

**Table 4: Correlation for different transport protocols**

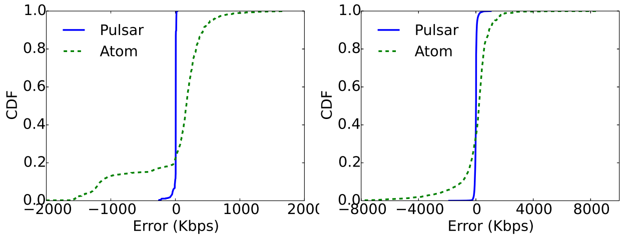
We also evaluate $\rho_{pulsar}$ and $\rho_{atom}$ for individual transport protocols considered in our simulations (Table 4). We can see that *Pulsar* correlates well with throughput compared to *Atom*, because the resource share computation is accurate across different protocols (over 0.99).

## 4.3 Throughput Estimation

In this section, we try to fit a linear model, using ordinary least squares regression, to estimate throughput from

| Metric | $C_1^{lr}$ | $C_0^{lr}$ (Kbps) | $C_1^{hr}$ | $C_0^{hr}$ (Kbps) |
|---|---|---|---|---|
| *Pulsar* | 0.98 | 5.61 | 0.98 | -1.12 |
| *Atom* | 0.11 | 112.66 | 1.22 | 279.29 |

**Table 5: Linear model coefficients**



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

**Figure 4: CDF of throughput estimation error**

*Atom* or *Pulsar* [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 5 shows these values for both *Atom* and *Pulsar*.

Figure 4 shows the CDF of estimation errors for high rate and low rate applications with *Pulsar* 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 -1159.66 Kbps and 429.29 Kbps, respectively, where as for *Pulsar*, they are $-4.43$ Kbps and 13.30 Kbps. Note that estimation errors 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 *Atom* are -969.07 Kbps and 917.34 Kbps, respectively, where as for *Pulsar*, they are -74.71 Kbps and 56.51 Kbps. We can see that the throughput estimation with *Pulsar* is accurate and outperforms prior approaches. On the testing dataset, $Pulsar$ reduces the root mean square (RMS) estimation error by 94.67% for low rate applications and by 92.34% for high rate applications.

Figure 5 shows the CDF of estimation errors for VoIP, Video Conference, Bulk Download and Video Streaming. The CDF for Screen Share and Whiteboard are similar. While the estimation error stays close to zero for *Pulsar* for all applications, it is much higher for *Atom*. For Bulk Download, which is always classified as high rate, *Atom* significantly underestimates the throughput. On the other hand, for VoIP and whiteboard applications, which are always classified as low rate, *Atom* overestimates the throughput. Specifically, *Pulsar* reduces the RMS estimation error by 93.9%, 40.22%, 99.19%, 95.96%. 99.25%, and 99.17% for VoIP, Video Conference, Screen Share, Whiteboard, Bulk Download and Video Streaming applications, respectively.

We also compare the estimation error for $Pulsar$ and $Atom$ for different transport layer protocols. While the estimation error is close to zero for *Pulsar*, *Atom* underestimates throughput for over 45% of data samples. Overall *Pulsar* improves the estimation error by 98.58%, 99.03% and 99.11% for UDP, TCP New Reno and TCP Westwood, respectively. Due to lack of space we do not present the CDFs here.

### *4.3.1 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 low rate senders. Results for high 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 $Atom$. *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

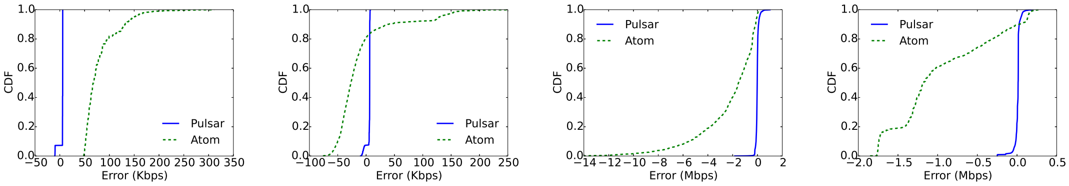
In this section we discuss some related issues and future extensions of *Pulsar*.

**Static users**: One of the assumptions in modeling *Pulsar* is that entire subframe is assigned to a single UE. When users are moving, different subcarriers 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, cafetaria, but they stay static while in these locations. 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.

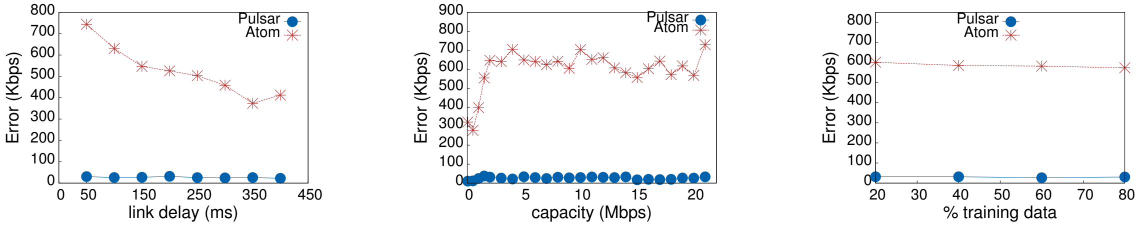
**Multiple flows per UE**: We consider one flow per UE in our analysis, for simplicity. Users might be running more than one application at a given time, creating multiple flows. In such case, *Pulsar* estimates the effective throughput, which is a weighted average of all individual flows, as it does not

[3] While we chose a linear model for *Pulsar* based on high correlation, the same was chosen for *Atom*, based on assumptions in [14].



**(a) VoIP** **(b) Video Conference** **(c) Bulk Download** **(d) Video Streaming**

**Figure 5: CDF of throughput estimation error for different applications**



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

**Figure 6: Sensitivity of estimation error to different factors**

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.[16] proposed a metric to estimate TCP throughput in WiFi network by observing channel interference, contention and physical link rate. In contrast, our work focusses on estimating throughput in LTE networks, which have very different characteristics. Other works have focussed on building QoE metrics for various applications. Authors in [12] proposed a QoE metric for Skype application whereas Prometheus[11] 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[13] and ATOM[14] 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 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] Airvana OneCell System. `http://www.airvana.com/products/enterprise/onecell/`.

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

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

[6] 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`.

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

[8] ns-3. `www.nsnam.org`.

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

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

[11] 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).

[12] Chen, K.-T., Huang, C.-Y., et al. Quantifying skype user satisfaction. In *Proc. of the SIGCOMM* (2006).

[13] 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).

[14] 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).

[15] Nguyen, B., Banerjee, A., Gopalakrishnan, V., et al. Towards understanding tcp performance on lte/epc mobile networks. In *Proc. of the AllThingsCellular* (2014).

[16] Patro, A., Govindan, S., and Banerjee, S. Observing home wireless experience through wifi aps. In *Proc. of the MobiCom* (2013).

[17] Sesia, S., Toufik, I., and Baker, M. *LTE, The UMTS Long Term Evolution: From Theory to Practice*. Wiley Publishing, 2009.