

Achieving Throughput Fairness in Smart Grid Using SDN-Based Flow Aggregation and Scheduling

Wei Guo, V. Mahendran, and Sridhar Radhakrishnan

School of Computer Science

University of Oklahoma, Norman, Oklahoma 73019

Email: {wguo, mahendran.veeramani, sridhar}@ou.edu

Abstract—Latest research in smart grid communications has advocated the aggregation of multiple traffic flows in order to achieve an improved throughput. While aggregation improves the overall throughput, the individual flows still suffer from unfair throughput performance. As a result, the enablers for time sensitive smart grid services such as load-shedding that require a timely report of data are the most affected.

In this paper, we propose a novel SDN-based framework to provide fairness among smart-meters, through flow aggregation and scheduling. By exploring the SDN's flow-level manageability features, for the first time in this paper, we present an implementation-based architecture to perform effective aggregation-and-scheduling of traffic flows. The proposed framework ensures fairness (among the smart-meters), as well as improve the throughput performance. Our extensive experiment results validate the efficacy of our proposed framework.

I. INTRODUCTION

Data communication is the key enabler in smart grid networks. The deployment of communication paradigm in the power domain yields benefits to all participants in the system such as utility companies, governments, and consumers. Typical smart grid network spans a vast geographical area connecting many devices such as Smart Meters (SMs). The purpose of SMs is to enable continuous monitoring and better utilization of resources at the customer-end users (*i.e.*, the electricity consumers). Some of the benefits include automatic billing, load balancing, remote connect/disconnect [1]. The latest SMs are advanced with processing capabilities, and are integrated with full network transport suite such as TCP or UDP.

An essential requirement of SM-based communication includes the reporting of meter-reading information to the end-server (for performing fault-detection, and load shedding), in a *timely* manner. From the networking context, this requires the communication infrastructure to provide a homogeneous delivery rate to all SMs. There are several factors a communication infrastructure needs to consider in order to enable all SMs to deliver data with a fair throughput performance. One factor includes heterogeneous propagation delay. Typical in smart grid environments, the data from SMs in different geographical region would experience different propagation delays before reaching the end-server.

Latest research encourages the communication of SMs over wireless networks such as cellular Long-Term Evolution (LTE) [1], [2]. The LTE-based communication has its own

benefits such as geographically wide coverage range (in the order of few kms), which is an essential design requirement for the ubiquitously deployed SMs. In addition to large coverage range, LTE also supports high data rates. From the base-station, the SMs' data is transported over wired network to the end-server. Much of the existing studies on throughput frameworks in smart grid communications considers only the wired infrastructure [3]. In order to be future-proof, it is vital to consider both LTE and wired communication substrates in the infrastructure providing the smart grid services.

Even in the wired smart grid networks, the conventional way of enabling end-to-end transport between each SMs and the end-server causes throughput degradation [4], [5]. Because a large number of short end-to-end TCP flows form a chaotic performance in the network causing more retransmissions and connection failures. As the individual TCP flows do not get sufficient time to sense the present congestion condition in the network, they fail to tune their sending rates, thereby severely degrading the delivery throughput performance. To combat such scenario, recent works [3], [6] have advocated the use of aggregate-points in the network.

The aggregator nodes (in the network) function in the form of an application layer on top of the regular TCP transport protocol suite. This application-layer combines several short TCP connections (mice flows) into a single long TCP flow (also known as 'Elephant flow'). The TCP sender of the long flow can effectively sense the congestion in the network, and thereby adapts the sending rate appropriately. In this manner, the associated clients such as SMs can benefit from an improved throughput performance. While such aggregation point based approaches only help in improving the throughput performance, we argue in this paper that such frameworks would not provide fairness to individual SMs, which is an important requirement in smart grid communications.

For illustration, let us consider a typical urban scenario, with end-server located at the city center that is connected to SMs distributed across a vast geographical area covering the outskirts of the city. A group of (spatially close) SMs are connected wirelessly over LTE base stations, called eNodeBs (eNBs). Each region has multiple eNBs deployed, and the number of eNBs are based on the number SMs to be covered in an area (for example, city center sees a denser SMs (so more eNBs) than in the outskirts). Thanks to the advancement in LTE allowing cloudlet-based computing frameworks such as

SMORE [7], we can perform aggregation of SM flows right at the respective eNBs. Subsequently, aggregation can be done at the joining points of multiple long flows in the wired network part (as observed in [3]).

The resulting communication framework will have a number of aggregation points in tandem, along the path from SMs to the end-server. The individual long TCP flows between aggregation points sense the congestion at the frequency of received ACK messages, and subsequently control the sending rates at the clock of Round-Trip Times (RTTs), i.e., the time between a data sent and its ack received. The RTTs are comprised of two factors of delay, namely link propagation delays, and link queuing delays. To avoid bufferbloat scenarios, the network engineers recommend the use of small queue size at the network routers. Therefore it is practical to assume negligible queueing delay in the network with small queues. In the case of smart grid networks, with the SMs typically distributed in a geographical distance (in the orders of tens to hundreds of kilometers), the propagation delay becomes a predominant factor that causes heterogeneous RTTs, for different aggregation points. This becomes the root cause for the unfairness among SMs across different regions.

To the best of our knowledge, we are the first to propose a framework to enable fairness to SMs (insensitive to their geographical distance). Unlike existing approaches our model comprehensively captures both the wireless and wired scenarios of a typical smart grid. The efficacy of our proposed framework is demonstrated by the fact that fairness is achieved without much loss in throughput performance; typically obtained from an aggregation-only framework.

Fortunately, with the advent of Software-Defined Networks (SDNs) the network becomes more accessible, manageable, and programmable than ever. Using SDN, we address fairness among multiple flows with our novel implementation-based ‘Aggregation-and-Scheduling’ framework. Unlike existing works such as [3] that proposed the conceptual use of aggregation idea in a simulation environments and demonstrated improved performance; we on the other hand take a practical approach of an implementation-based aggregation-and-scheduling framework, with a white-box design that describes intricacies involved in a real working prototype. In summary, our contributions in this work are as follows:

- We propose a novel SDN-based aggregation-cum-scheduling framework to improve fairness and as well as maintain the improved TCP throughput performance found in traditional aggregation-only frameworks.
- Unlike conceptual idea on simulation, we present a white-box design of the proposed framework by highlighting the implementation functionalities equivalent to developing a working prototype.
- We extensively study the throughput performance and fairness with appropriate analytical model validating the experimental results.

The organization of the paper is as follows: In Section II we discuss the state-of-the-art study related to aggregation of flows and their related performance study. The smart grid

communications network model and assumptions are described in Section III. The crux of the work proposing the SDN-based aggregation and scheduling framework is discussed in Section IV, with brief discussion on analytical model of throughput. In Section V, an extensive performance evaluation study of throughput and fairness is performed. Finally, we conclude our work by highlighting some future extensions.

II. RELATED WORKS

On a wired smart grid network, the authors in [4], [3] demonstrate an improved TCP performance for the SMs using aggregator nodes that combine multiple TCP connections. However, a natural extension of studying fairness among tandem aggregator nodes is not investigated. In our earlier work [6], we proposed SDN-based Fog computing nodes for Internet of Things (IoT) applications, and demonstrated an improved TCP performance, by migrating the remote server functionality to the edge switch for immediate response. A backup data transport is enabled by a single TCP from edge switch server to a remote end-host server. However, the concept of multiple edge-servers (aggregators) and their corresponding fairness is not explored. Unlike the aforementioned works, in this paper, we exclusively study the fairness among multiple (tandem) flows over wireless and wired networks in a smart grid scenario.

III. SMART GRID NETWORK MODEL

We consider a typical urban scenario smart grid environment, wherein the SMs are connected through LTE User Equipments (UEs) to the wireless LTE eNBs, and the several eNBs are connected to the end-server (located at the city center) via wired network. We considered the UDP transport from SMs to eNBs, and TCP transport from eNBs to the end-server. The reason for UDP is to fully utilize the wireless bandwidth, and also to have fairness among the first-mile LTE wireless uplinks. In our ns-3 wireless experiments, we have obtained a success rate of receiving 99% packets at wireless links for a wide range of input traffic rates (between 50 packets per second and 500 packets per second), with each packet of 1500 bytes in length. In future, for higher traffic loads, without loss of generality, a reliable UDP protocol can be considered to further improve the success rates. The fairness at the wireless uplinks between multiple UEs and eNBs comes from the following factors: (i) in a smart grid all nodes are static i.e., the UEs, SMs, and eNBs. Static nodes create fixed and same channel conditions across all UEs [8], and (ii) the LTEs default proportional fair MAC scheduler allows a round-robin type of switching among all the backlogged UEs. Our experiments showed accurate fairness among all UEs connected to the same eNBs. The results are not show due to page constraints.

The schematic diagram of the described network is shown in Fig. 1. We have numbered different eNBs from 1 to N from left to right (i.e., from outskirts region towards city center to the end-server). We consider multiple SMs close in a colony are connected over wired Ethernet links to a LTE UE, which is

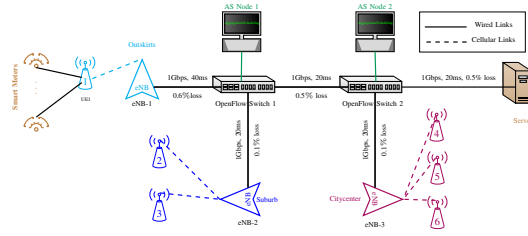


Fig. 1. Schematic diagram of OpenFlow network with Aggregator-cum-Scheduling (AS) nodes. For brevity, SDN controller is not shown. Also the smart-meters in suburb, and citycenter regions are not shown.

in turn connected wirelessly to an eNB. To capture a scenario of denser SMs in city-center, and relatively sparse SMs in the outskirts, we use one UE at the edge of the network and increase the number of UEs towards the server. This naturally captures denser SMs towards the path to the server node. The UDP packets received (from multiple SMs) at eNB are sent via a single TCP sender in the wired-part of the network. As the wireless LTE ensures fairness (through appropriate MAC scheduling and static nodes), we need to provide fairness in the wired-part of the smart grid.

We consider TCP Reno version throughout this paper. The TCP reno version is a widely used variant of TCP in the Internet. As we focus on long TCPs, we consider the performance effects comprising the congestion avoidance phase of TCP. The effects of TCP receiver window and network link bandwidth is considered to be sufficient enough to not affect the throughput performance. Without loss of generality, we assume packet loss as the indication of congestion, and we consider a simple identical and independent probability distribution for the packet loss.

As shown in Fig. 1, we have two eNBs connected to first switch, where these two long TCP flows are aggregated into one composite long TCP flow through the Aggregation/Scheduler (AS) node 1. eNB-3 joins switch 2 wherein it is mixed with the incoming aggregated flow at AS node 2. This represents a typical scenario of many flows getting aggregated towards the proximity of the server, eventually making the server's first hop as the bottleneck link. In the next section, we describe the proposed aggregation/scheduling framework and its implementation.

IV. PROPOSED SDN-BASED AGGREGATOR/SCHEDULER FRAMEWORK

Figure 2 depicts the whitebox functional description of the Aggregator/Scheduler (AS) node and the associated SDN framework. For brevity, we depict the model with one AS node. The AS node implementation design is conceived with following goal:

- **Transparent Aggregation and Scheduling:** To enable the framework to be market-ready, the aggregation and scheduling of flows need to be performed in a seamless manner, without explicit modifications at the end-hosts, namely the UEs and the servers.

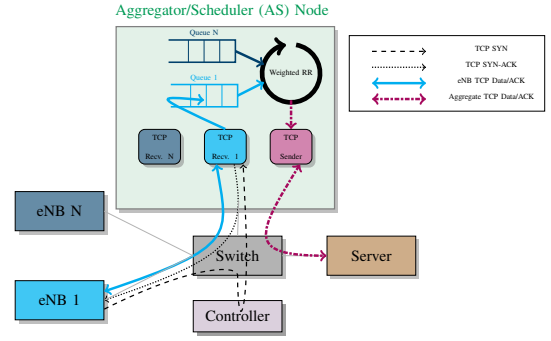


Fig. 2. Internal functionality of the proposed framework with single aggregator/scheduler node. For brevity, only the connection-sequence related to eNB-1 is shown.

To achieve aforementioned design goal, we believe the SDN [9] would be an ideal choice based on the offered features such as: (i) centralized controller logic, and decoupled data and control planes for better manageability, and (ii) flow-based programmable control logic for enabling flexible network services. As seen in the Fig 2, the (SDN) controller primarily connects with all the SDN-based OpenFlow switches in the network. The controller manages the control plane of the network to administer flow-level management by writing appropriate flow-entries at the switches.

A switch upon receiving the first packet of the TCP flow (either from the eNB or from the preceding AS node) will forward it to the controller, for appropriate decision making. In our scenario, the first TCP packet TCP-SYN is forwarded by the switch to the controller. The controller with the available topology information, will compute an appropriate flow entry to direct this flow to the near-by AS node (an end-host). Upon writing this flow-entry, all the future packets of this flow will be diverted (by the switch) to the AS node. To enable the TCP receiver 1 inside the AS node to further receive and process the incoming segments, an additional flow-entry is programmed at the switch (by the controller). This additional entry enables the switch to modify/rewrite the destination-field of the TCP header for the eNB-1's flow with AS node information. A similar entry, is made for return traffic carrying ACK packets from TCP receiver in AS node. Therefore, the eNB is abstracted from this rerouting logic to the AS node.

Each TCP receiver is associated with an independent queue for buffering the incoming packets. Without loss of generality, we consider this buffer to be sufficient enough to store incoming packets. To enforce fairness, the scheduler should know the information necessary for distinguishing the incoming composite flows. Though the independent queues serve the purpose of distinguishing different flows (at the link-level); in a tandem aggregator scenario each of these flows could be a composite flow aggregated at the preceding AS nodes. Therefore, the scheduler should be made aware of the appropriate scheduling information necessary for ensuring fairness. Thanks to the SDN's centralized controller logic that receives first packet of all the participating flows in the network. The controller

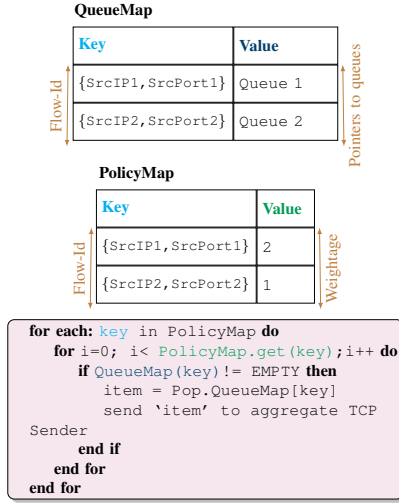


Fig. 3. Weighted round robin scheduling and aggregation implementation at the AS node

therefore maintains the global knowledge of each flows, and their corresponding first aggregation points (*i.e.*, AS nodes).

Each AS node maintains two ‘key-value’ lookup data-structures namely, QueueMap and PolicyMap as shown in Fig. 3. The QueueMap is used by the scheduler to locate the respective queues of the different flows. The PolicyMap is the important data-structure that stores the weightage for each (aggregated) flows indicating the number of actual flows it is composed of or carrying with it. As explained earlier, the controller provides the weightage information for the AS nodes to maintain in their PolicyMaps. The underlying logic for enabling fairness is made possible with a simple Weighted Round Robin Scheduling at the AS nodes, and with the global information support from SDN. The AS nodes aggregate traffic received from each participating eNBs (or preceeding AS nodes), and the TCP senders of the respective AS nodes react only to the associated TCP receiver. Each TCP sender therefore adjusts its sending rate which is homogeneous to all the packets, thereby solving the different RTT scenario observed in a typical non-aggregated network scenario. Therefore the simple yet effective implementation of aggregation-cum-scheduling can provide almost perfect fairness for the participating TCP flows. The underlying aggregation and weighted round robin scheduling logic of presented as ‘pseudo-code’ in Fig. 3.

We now study the throughput performance through analytical investigation and measure the fairness by using the popular Jain’s Fairness Index [10]. All our experiments are performed in a Mininet network emulator environment [11]; which is a realistic virtual network running real kernel and switch functionalities. Therefore, with minimal changes our frameworks can be deployable on a real working prototype. The TCP Reno’s congestion window evolution for the long TCP flow is analytically modeled by authors in [12], and studied the statistical performance of the evolution around mean. Fig. 4 shows the TCP Reno congestion window evolution

in the Congestion Avoidance (CA) phase. Let the congestion window size W be represented in packets with values ranging from 1 to w_m . The value of W evolves at each RTT and is controlled by the loss of packets in the network. The TCP behavior in CA phase is captured in Fig. 4. Upon a packet loss event the congestion window is halved, and on the other hand, a no loss event increases the congestion window size by 1 packet. For analytical tractability, without loss of generality, we assume a Bernoulli link loss model wherein the loss of packets is independent and identically distributed with a probability p . Therefore the loss probability $p(w)$ can be computed as follows: $p(w) = (1 - (1 - p)^w)$.

As observed in Fig. 4 the Markov chain is approximately aperiodic and irreducible for practical values of maximum congestion window size; therefore, the chain has a unique steady-state distribution, say π . By Birkhoff’s ergodic theorem, the sample mean of congestion window with scale (or sample size N), represented as $\bar{W}^N$ almost-surely converges to mean of the steady-state distribution π , when N grows to ∞ (N is time in our context). Mathematically expressed as follows [12]:

$$\bar{W}^N = \frac{1}{N} \sum_{i=1}^N W_i \xrightarrow[N \rightarrow \infty]{a.s.} \bar{W}^\infty = \sum_{w=1}^{w_m} \pi_w \times w = E[W]. \quad (1)$$

Where $E[W]$ is the average congestion window size. In other words, from Eq. 1 the scaled mean throughput $\bar{W}^N$ and its difference in value with $\bar{W}^\infty$ tends to 0 as N grows to infinity; and this behavior of difference converging to zero is termed as ‘rare event’ which can be characterized by large deviations theory.

From the classical results of the large deviations theory, an irreducible and aperiodic Markov chain, with finite state space, holds a large deviations spectrum as given below [12]:

$$\lim_{\epsilon \rightarrow 0} \lim_{N \rightarrow \infty} \frac{1}{N} \log \Pr(\bar{W}^N \in [\alpha - \epsilon, \alpha + \epsilon]) = f(\alpha) \quad (2)$$

where, $f(\alpha)$ is called the large deviations spectrum. In our context, the large-deviations spectrum $f(\alpha)$ can be computed [13] as the Legendre Fenchel transform of spectral radius’(ρ) logarithm of a matrix $(\mathbf{R}(q))_{ij} = \exp(qj)\mathbf{T}_{ij}$. Where $\mathbf{T}_{ij}$ is the transition matrix underlying the Markov chain of states of TCP congestion window as shown in Fig. 4.

$$f(\alpha) = \inf_{q \in \mathbb{R}} (\log \rho(\mathbf{R}(q)) - \alpha q) \quad (3)$$

An essential property of large-deviations spectrum is that it is concave and satisfies the following property that if $\alpha = \bar{W}^\infty$ then $f(\alpha) = 0$, and for all other values $f(\alpha) < 0$. Therefore, the mean congestion window size $E[W]$ can be computed as the value of α at $f(\alpha) = 0$. Subsequently, the average throughput $E[T]$ in (bps) is computed as follows:

$$E[T] = \frac{E[W]}{RTT} \times MSS. \quad (4)$$

where RTT is the round-trip time (in seconds), and MSS is the Maximum TCP Segment Size (in bits). It is worthwhile to note that for the proposed aggregation-cum-scheduled framework, the measured RTTs were almost steady to a fixed value

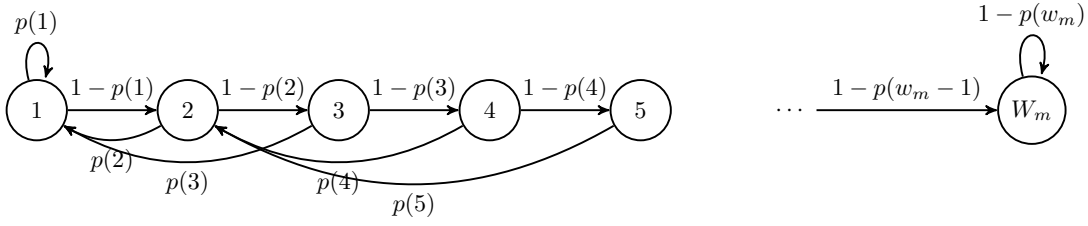


Fig. 4. Markov Chain Model Long TCP Congestion Window Size Evolution in CA phase. Where $p(i)$ is the loss probability with congestion window size of ' i ' packets.

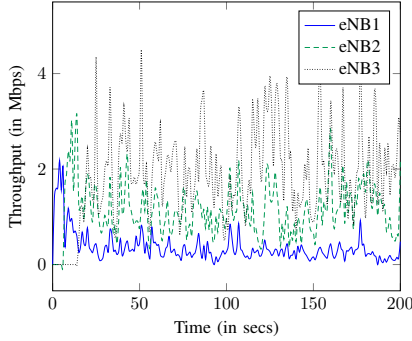


Fig. 5. eNBs TCP throughput with aggregation and without scheduling.

in our experiments, which also a significant factor towards ensuring fairness.

V. PERFORMANCE EVALUATION

The LTE-based network is studied in ns-3 [14] simulation environment, and wired-part of the network is studied in Mininet network environment. We integrated ns-3 simulation to Mininet environment in order to get a unified model of the smart grid network.

Figure 5 shows the individual eNB's throughput performance in the aggregation-only setting. In other words, weighted round robin is not considered. The unfairness among flows are prevalent. This is due to the fact, that total throughput is determined by the 'last ASnode2 and the server' pair, as this forms the bottleneck link. Due to the high throughput in preceding links, the packets get buffered in the queues and are always available to the last AS node to consume. However, different flows from the preceding links experience different RTT the way they feed the queue is heterogeneous thereby propagating the unfairness at the server. It is also noted that different flows complete their transfer at different points in time due to substantial difference in throughput.

Figure 6 shows the improved fairness conditions in the system utilizing our proposed aggregation-and-scheduling framework, which is evident from the eNBs throughput being proportional to the number of UEs connected. To understand the accuracy of fairness, we have used Jain's Fairness Index to evaluate the fairness of the three respective network settings, namely no-aggregation, aggregation-only, and aggrega-

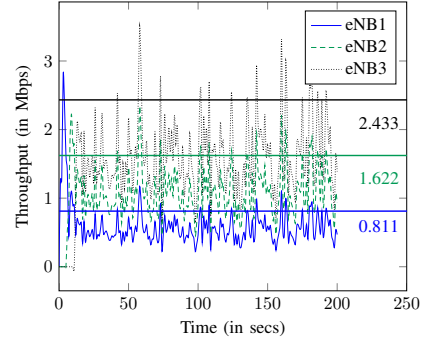


Fig. 6. eNB TCP throughput in Proposed Aggregation and Scheduling Framework. The horizontal lines and the respective values show the analytical average throughput, as computed from Eq. 4.

tion/scheduling. The Jain's Fairness Index [10] is computed as follows:

$$\text{Fairness index} = \frac{(\sum_{i=1}^n E[T_i])^2}{n \times \sum_{i=1}^n E[T_i]^2} \quad (5)$$

where $E[T_i]$ throughput of TCP flow i and n is the total number of flows. Figure 7 shows the fairness of the system for three different network settings. It is more evident that our proposed framework outperformed both no-aggregation, and aggregation-only setups by exhibiting perfect fairness for all the three heterogeneous RTT eNBs in the network. With the practical operating scenario including heterogeneous RTT flows, and composite multi-level aggregated flows due to tandem aggregators, our proposed framework handled fairness more effectively than other traditional systems.

Figure 8 and Fig. 9 show the received throughput of independent UEs clearly demonstrating the fairness of the proposed framework. To understand the throughput deviations of all three types of frameworks, Fig. 10 shows the total throughput distribution for the three different policies. In addition to achieving fairness, the throughput of proposed aggregation with scheduling framework is still close to the aggregation-only framework.

VI. CONCLUSION

We have proposed a novel SDN-based TCP Aggregation/scheduling smart grid framework that achieved a better throughput performance and fairness. We also have proposed the white-box model of implementation with a detailed

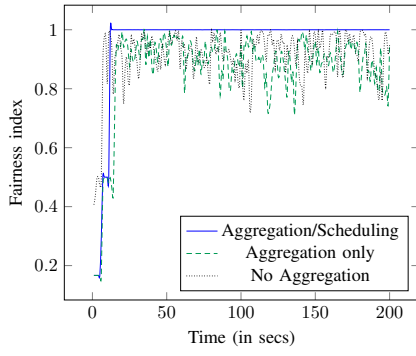


Fig. 7. Throughput fairness index of 3 different frameworks, computed using Eq. 5.

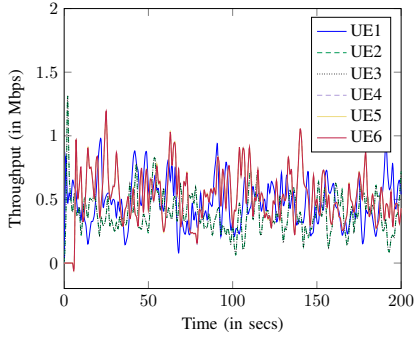


Fig. 8. Individual UEs throughput in without aggregation and scheduling.

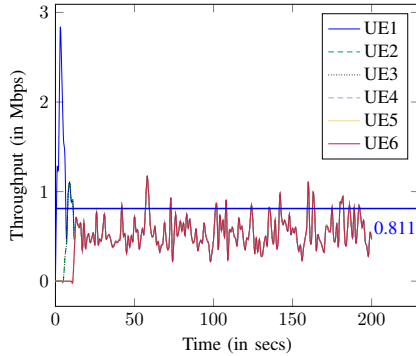


Fig. 9. Individual UEs throughput in with proposed aggregation and scheduling. The horizontal line shows the analytical throughput average value.

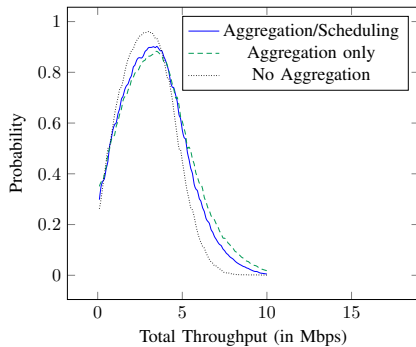


Fig. 10. Throughput probability distribution for three policies, no aggregation, aggregation without policy, and proposed aggregation with policy

functional design. The throughput performance is extensively studied and compared with the respective traditional no-aggregation, aggregation with no scheduling, and our proposed aggregation and scheduling frameworks. Our proposed framework demonstrated fairness to significant level of accuracy through out the course of the experiment, while maintaining a high throughput.

In future, we plan to study the WiFi-based SM communications with TCP over wireless links, to achieve throughput fairness.

REFERENCES

- [1] T. Khalifa, K. Naik, and A. Nayak, "A survey of communication protocols for automatic meter reading applications," *IEEE Communications Surveys Tutorials*, vol. 13, no. 2, pp. 168–182, 2011.
- [2] T. H. Chuang, M. H. Tsai, and C. Y. Chuang, "Group-based uplink scheduling for machine-type communications in lte-advanced networks," in *WAINA '15: Proceedings of the Advanced Information Networking and Applications Workshops*, 2015, pp. 652–657.
- [3] T. Khalifa, A. Abdrabou, K. Naik, M. Alsabaan, A. Nayak, and N. Goel, "Split- and aggregated-transmission control protocol (SA-TCP) for smart power grid," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 381–391, 2014.
- [4] —, "Design and analysis of split- and aggregated-transport control protocol (SA-TCP) for smart metering infrastructure," in *SmartGrid-Comm '12: Proceedings of the International Conference on Smart Grid Communications*, 2012, pp. 139–144.
- [5] T. Khalifa, K. Naik, M. Alsabaan, and A. Nayak, "A transport control protocol suite for smart metering infrastructure," in *ICEDSA '11: Proceedings of the International Conference on Electronic Devices, Systems and Applications*, 2011, pp. 5–10.
- [6] Y. Xu, V. Mahendran, and S. Radhakrishnan, "Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery," in *COMSNETS WACI' 16: IEEE COMSNETS Workshop on Wild and Crazy Ideas on the Interplay Between IoT and Big Data*, 2016.
- [7] J. Cho, B. Nguyen, A. Banerjee, R. Ricci, J. Van der Merwe, and K. Webb, "SMORE: Software-defined networking mobile offloading architecture," in *AllThingsCellular '14: Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges*, 2014, pp. 21–26.
- [8] U. Parthavi Moravapalle, S. Sanadhya, A. Parate, and K.-H. Kim, "Pulsar: Improving throughput estimation in enterprise LTE small cells," in *CoNEXT '15: Proceedings of the ACM International Conference on Emerging Networking Experiments and Technologies*. New York, NY, USA: ACM, 2015.
- [9] "SDN architecture," https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-521_SDN_Architecture_issue_1.1.pdf.
- [10] R. Jain, D.-M. Chiu, and W. R. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Eastern Research Laboratory, Digital Equipment Corporation, 1984.
- [11] "Mininet," mininet.org.
- [12] P. Loiseau, P. Goncalves, J. Barral, and P. V.-B. Primet, "Modeling TCP throughput: An elaborated large-deviations-based model and its empirical validation," *Performance Evaluation*, vol. 67, no. 11, pp. 1030–1043, 2010.
- [13] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*. Jones and Bartlett, Boston, 1993.
- [14] "ns-3 network simulator," <https://www.nsnam.org>.