Transport Layer Performance Analysis and Optimization for Smart Metering Infrastructure

Tarek Khalifa\textsuperscript{a,}\textsuperscript{*}, Atef Abdrabou\textsuperscript{b}, Khaled Bashir Shaban\textsuperscript{a}, Maazen Alsabaan\textsuperscript{c}, Kshirasagar Naik\textsuperscript{d}

\textsuperscript{a}Computer Science and Engineering Department, Qatar University, Doha, Qatar, P.O. Box: 2713
\textsuperscript{b}Dept. of Electrical Engineering, United Arab Emirates University, Al-Ain, Abu Dhabi, UAE, 15551
\textsuperscript{c}College of Computer and Information Sciences, King Saud University, Saudi Arabia, Riyadh
\textsuperscript{d}Dept. of Electrical & Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada, N2L3G1

Abstract

In a smart power grid, collecting data from a large number of smart meters and sensors over the conventional one-hop transmission control protocol (TCP) communication is prone to a high packet loss rate and degraded throughput due to the ineffectiveness of the TCP congestion control mechanism. The Split and Aggregated TCP (SA-TCP) proposes upgrading intermediate devices (known as regional collectors or RCs) to combine meters’ TCP connections and forward data over a unified TCP connection to the utility server.

This paper provides a comprehensive performance analysis of the SA-TCP scheme. It studies the impact of varying various parameters on the scheme, including the impact of network link capacity, buffering capacity of RCs, propagation delay between the meters and the utility server, and finally the number of RCs utilized as SA-TCP aggregators. The performance results show that by adjusting those parameters, it is possible to tune throughput of a smart metering network to the desired amount while keeping the deployment cost minimal.

Keywords: congestion control, smart meters, smart metering infrastructure, telecommunication traffic

1. Introduction

Integrating the power grid with a communication infrastructure definitely enables a great deal of new and sophisticated services [1, 2, 3]. The collection of data from a large number of metering and sensing devices [4, 5, 6] enables utility companies to manage energy efficiently. This can be achieved by distributing power generation stations optimally, utilizing renewable energy sources, and quickly localizing and self-healing faulty spots. However, from a communication perspective, the large amount of data requires effective congestion control. The fact that a meter’s data rate is low and that the large volume of data comes from a large number of sources renders Transmission Control Protocol (TCP) congestion control ineffective [7]. The lack of an effective congestion control causes high loss rate and degrades throughput for metering data and similarly for any traffic that shares the smart metering communication network.

Our solution, Split- and Aggregated-TCP (SA-TCP) scheme [7] enhances TCP congestion control performance in a smart metering infrastructure (SMI). Instead of having the meters form individual TCP connections with the utility server, they rather connect to intermediate devices (e.g., Regional Collectors or RCs), which we call SA-TCP Aggregators. Those devices forward meters’ data over a unified TCP connection to the utility server. Since the SA-TCP aggregator node will have the data of a large collection of meters, maintaining the full range of congestion control becomes viable, making TCP congestion control effective. As a result, packet loss rate is reduced, and so a better link utilization is achieved.

This paper comes as a continued study of the SA-TCP scheme, which was previously introduced in [7]. A mathematical model for the scheme was detailed in [8]. The model captures the SMI traffic behaviour and draw conclusions about the network performance, including throughput, loss rate and delay. The model takes into account the characteristics of meters and the TCP congestion control mechanism (i.e., the dynamics of the congestions window size). It also takes into account the number of meters and number of SA-TCP aggregators, network setup parameters such as link capacities, buffering capacities and propagation delay. The modelling approach combines a Markovian model for the application layer and the TCP behavior of a single meter and then analyzes the superposition of several meters sharing a network by means of a queuing system. Based on the model, performance analysis and optimization of the SA-TCP architecture are performed and reported in this paper.

The contributions of this work are twofold:
It provides performance analysis of the SA-TCP scheme in comparison with the conventional one-hop TCP protocol. The impact of various design and network parameters on the scheme are considered. Specifically, it shows the impact of SMI link capacity, propagation delay, number of SA-TCP aggregators and their buffering capacities. Such analysis is important for enhancing the SA-TCP scheme to achieve better TCP performance.

It formulates an optimization framework, which ensures satisfactory performance results in terms of loss rate and end-to-end delay. It also considers minimizing the SA-TCP scheme deployment cost by balancing the number of SA-TCP aggregators and link bandwidth capacity while satisfying performance requirements.

In short, the SA-TCP scheme addresses the performance degradation caused by the TCP congestion control, which cannot adapt to a large number of low rate smart meters. The developed analytical model is a key component in our ability to analyze the performance and effectiveness of the proposed scheme under various design factors. Accordingly, the optimization formulation, which is based on the analytical model, further tunes the performance of SA-TCP as desired.

The rest of this paper is organized as follows. Section 2 reviews the related SMI TCP work. Section 3 presents the SA-TCP system model and summarizes our previous mathematical work developed to capture the SMI traffic behaviour. Section 4 provides a comprehensive performance analysis for SMI traffic under SA-TCP and conventional one-hop TCP congestion control set-ups. Finally, Section 6 concludes the paper.

2. Related Work

The TCP layer issues in smart power grid have been addressed in a number of recent publications, addressing congestion control, delay and security. Allalouf et al. [9] propose hop-by-hop traffic reduction assuming that meters data samples are required at certain intermediate devices, but not at the utility center, and assuming that the intermediate devices are sophisticated enough in terms of hardware and software capabilities to process and analyze data packets at the application layer. Therefore, they propose to route traffic through devices where more reduction can be applied to such data. Kim and Thottan [10] propose a new transmission control protocol that targets mainly delay-sensitive smart grid applications. However, congestion control is not studied. Kim et al. [11] also aim at making TCP faster. Their focus, however, is on the security aspect of TCP. They design a secure transport protocol for a smart grid data collection protocol that avoids the need to support TCP with Transport Layer Security (TLS) [12]. In the same line, Dan et al. [13] develops a secure collection protocol that allows an intermediate concentrator to be an untrustworthy device. As such, the protocol is applicable to SA-TCP allowing the TCP session to be split at a device such as the aggregator while stay secure.

At a first glance, SA-TCP seems to have resemblance with Indirect-TCP (I-TCP) [14] in splitting TCP connections and with [15] in aggregating TCP connections. However, I-TCP, which was introduced to support Internet Protocol (IP) mobility, does not change the number of TCP connections between the end systems. Aggregation of TCP connections in [15] is in the form of sharing TCP state information (e.g., round trip time and congestion window size) among a set of TCP connections that a single mobile device initiates, so the number of connections does not change. The idea of splitting TCP has appeared in other areas too. Amir et al. [16] develop a transport protocol that connects a group of devices over a wide area network, in which if packets get lost, they are retransmitted from an intermediate device. In wireless sensor networks [17], splitting TCP over multiple hops is also common [18, 19]. The purpose of splitting in both cases is to achieve low latency.

3. Overview of SA-TCP Scheme

3.1. SA-TCP System Model

Smart Metering Infrastructure is composed of a large number of smart meters and smart sensors forming regions of local area networks. By means of wireless communication technology (or possibly Power Line Carrier (PLC)), the meters communicate over multiple hops with intermediate devices known as Regional Collectors (RCs) (also called concentrators) [20]. These RCs are installed at pre-selected locations in every region and act as gateways to route the meters’ data packets through a wide area network to the utility server, in which data is collected for processing [21] [22].

The scheme, SA-TCP, enhances TCP performance [7] by introducing the concept of aggregation at the transport layer. The RCs implement the added service of splitting and aggregating TCP connections as depicted in Fig. 1. Hence, we refer to RCs as SA-TCP aggregators. Every set of $n$ meters establish $n$ TCP connections with an SA-TCP aggregator. Thereby, the meters’ data packets are received at the application layer of the SA-TCP aggregator and are forwarded by the aggregation application over a single TCP connection to the utility server. In other words, the TCP connections between the meters and the utility server are no longer one-hop, but rather, two-hop connections. As for the TCP Protocol mechanisms, there is no change in all the end points (i.e., meters, SA-TCP aggregators and the utility server).

The aggregation application can implement various scheduling policies depending on the nature of traffic and desired performance. For example, a priority-based or time-based
scheduling can be applied to enable urgent data (e.g., alerts) to be delivered sufficiently fast. The work of this paper, however, does not address such policies; rather the focus is on the performance of TCP as a result of its congestion control mechanism.

Our model assumes that the RCs are reliable devices. Given the importance of such devices, other publications, such as [23], address the idea of deploying redundant regional collectors for added reliability. In case of the failure of an RC, the work of Leu et al [24] can be employed to guide traffic through another RC. Their work develops a mechanism that switches to a better path whenever the first path fails or communication quality worsens.

Our assumptions about the smart metering architecture and traffic are in agreement with [25, 26, 27] and are summarized as follows:

- Reliable delivery of every report sent by a meter or sensor is required. Every report carries a unique piece of information that is related to a certain meter and a certain time duration, e.g., for billing purposes.
- Data aggregation is not applicable. Applications such as real-time pricing and demand side management necessitate delivery of raw data [28]. Because the utility provider is interested in each measurement, rather than statistical summaries of the data; hence, data aggregation techniques, such as mean, median, maximum and minimum [29], cannot be employed here.
- Data sources such as smart sensors and meters are stationary nodes distributed throughout the power grid. Collection of data takes place at a centralized location, namely, the utility server.
- TCP connections are long-lived, so there is no connection setup overhead. The sources are set up to submit their reports continuously at a pre-configured schedule as well as in response to triggered events.
- End-to-end delay $d$ is calculated from the view point of the receiver; that is, the time it takes until a data packet is successfully received.
- Meters are configured with a routing protocol that enables them to route data to the SA-TCP aggregator [30] [31]. The SA-TCP aggregator takes care of routing the data to the utility server.
- Data packets vary in size from tens of bytes to a few hundreds of bytes, depending on the information carried and security system employed [32].

3.2. Mathematical Model and Results

This section gives a summary of the math model of SA-TCP. A detailed explanation of the model is found in [8]. The math model is meant to capture the SMI traffic behaviour in terms of the network operating point metrics, namely, average meter offered load ($\lambda$), packet loss rate ($P$) and end-to-end delay ($d$). Those metrics are important for the analysis of the smart metering system. The model allows us to reproduce the actual behavior of the meters’ traffic in SMI conveniently yet accurately and fast (e.g., a few seconds of running time). Additionally, the analytical model allows us to embed it into an optimization formulation. Alternatively simulations would require high processing power machines to test for such large numbers of meters and would take weeks to finish an experiment that examines a single parameter over a given range of values.

Figure 2 illustrates the two stages of the SA-TCP scheme. In the first stage, smart meters act as traffic sources. They produce data according to the application behaviour and TCP congestion window mechanism. The traffic is routed from the meters to the SA-TCP aggregator of that region, where the traffic is buffered and forwarded next to the utility server over a unified TCP connection. Thus, in the second stage, SA-TCP aggregators are modelled as traffic sources producing data packets according to the TCP congestion window mechanism. Data traffic while passing through a WAN competes for bandwidth and naturally experience delays and losses due to networks bottlenecks.

The traffic generation mechanism for meters and SA-TCP aggregators is modeled by continuous-time Markov chains (Figures 3 and 4), and the network is modeled by a queuing system [33]. We define the network operating point by the triple ($\lambda, P, d$). To determine the network operating point, we apply the fixed-point method shown in Algorithm 1 in each stage, which reflects the steady-state situation of the traffic sources’ model and network model interaction. The fixed point solution is the point ($\lambda, P, d$) that satisfies both: Markov equations (which finds the average traffic load) and queuing model equations (which determine loss rate and delay).

3.2.1. Network Model:

The queuing model M/M/1/B is used as the network model to calculate the packet loss rate and average queuing delay. A queue is characterized by three parameters: the queue size, input traffic rate, and service rate.
In Stage 1 (Fig. 2), the queue represents buffering at the SA-TCP aggregator device as data is being forwarded. Therefore, the stage 1 queue is characterized by size \((B_1)\), input traffic rate that is coming from \(n\) meters collectively \((\lambda_1 = \sum_{i=1}^{n} \lambda_i)\), and service rate \((\mu_1)\), which corresponds to \((\text{data transfer rate})\) that an SA-TCP aggregator can offer.

In stage 2, the queue represents a WAN bottleneck. The queue parameters become buffer size \(B_2\), SA-TCP aggregators’ traffic \(\gamma_t\), and service rate \(\mu_2\).

The loss rate and expected queuing delay are given by:

\[
P(\rho, B) = \frac{\rho^B (1 - \rho)}{1 - \rho^{B+1}} \tag{1}
\]

\[
ET_q(\mu, \rho, B) = \frac{1}{\mu} \left( \frac{1 - \rho}{1 - \rho^B} - (B + 1) \rho^B + B \rho^{B+1} \right) \tag{2}
\]

\[
Ed(\mu, \rho, B) = 2T_{\text{p}} + ET_q(\mu, \rho, B) \tag{3}
\]

\[
\rho_m = \frac{\lambda_1}{\mu_1}, \quad \mu_1 = \frac{\gamma_t}{N_A}, \quad \rho_a = \frac{\gamma_t}{\mu_2}, \tag{4}
\]

\[
P_m = P(\rho_m), \quad P_a = P(\rho_a) \tag{5}
\]

\[
d_1 = Ed(\mu_1, \rho_m, B_1), \quad d_2 = Ed(\mu_2, \rho_a, B_2) \tag{6}
\]

where \(\rho_m\) and \(\rho_a\) are the queue utilization factors in the first and second stages of the SMI model, respectively. \(ET_q(\mu, \rho, B)\) and \(Ed(\mu, \rho, B)\) are the expected values of the queuing delay and end-to-end delay, respectively. \(T_p\) is one-way propagation delay.

The input traffic rates, \(\lambda_1\) and \(\gamma_t\), are derived from the Markovian models of meters and SA-TCP aggregators, respectively (discussed next). The service rate, \(\mu_1\), is equal to an SA-TCP aggregator’s offered load \((\gamma_j)\). The service rate, \(\mu_2\), is computed as the link capacity \((C)\) over the maximum segment size \((\text{MSS})\) in bytes \((i.e., \mu_2 = \frac{C}{\text{MSS}} \text{ segment/sec})\) [33].

3.2.2. Meter Markov Chain Model

The Markov model determines the average number of segments generated by a meter. The model captures the application and TCP behaviour as shown in the state diagram of Fig. 3. The numbered states correspond to the size of the TCP congestion window, and the transition rates among states correspond to the rate of success or failure of delivering segments. The application alternates between off and on states with rates \(\alpha = \frac{1}{T_{\text{off}}}\) and \(\beta = \frac{1}{T_{\text{on}}}\), respectively. The durations \(T_{\text{off}}\) and \(T_{\text{on}}\) represent the time for a meter to be idle and active, respectively. If the segment of State 1 is delivered successfully, the window size grows to two segments (State 2). If segments of a given window size are lost, transition is made to State 0 and then reset to State 1 for retransmission. The transition rates are given in the figure. \(q_w = (1 - P_m)^w\) is the probability of success of delivering all the segments of window size \(w\) successfully. \(P_m\) is the probability of a packet loss. The transition rate \(\tau = \frac{1}{T_p}\) approximates acknowledgement timeout as \(5T_r\), assuming that the TCP protocol estimates timeout as the average round trip time \((\text{RTT})\) plus 4 times its standard deviation.

We solve the Markov chain model for the stationary probabilities of the states; The probabilities \(\pi_1\) and \(\pi_2\) for States 1 and 2 correspond to the transmission of 1 and 2 segments, respectively. Thus, the average traffic generated by a meter is calculated as \(\lambda_1 = \delta \pi_1 + 2 \delta \pi_2\).

The total traffic generated by all the meters in a region, therefore, is as follows.

\[
\lambda_t = \sum_{i=1}^{n} \lambda_i \tag{7}
\]

where \(n\) is the number of meters in a region.

3.2.3. SA-TCP Aggregator Markov Chain Model

The model is depicted in Fig. 4 capturing the behaviour of TCP Reno [34] congestion control mechanism, with a maximum congestion window size \(W_M\) of 16 and initial slow start threshold \(w_s\) of 8. This design is inspired by [35]. The states are numbered to represent the congestion window size, indicating the number of segments to send in an RTT. Numbers with dashes in the right-most column states represent the fast retransmit phase but do not correspond to actual transmissions.
The transition rates are shown in Fig. 4. The probability of successfully delivering all segments of a given window size, $q_w$, is calculated as $(1 - P_a)^w$, where $P_a$ is the probability of a packet loss in Stage 2. The probability of $i$ packet losses from a window of size $w$ is calculated as $P_w^i = {w \choose i} P_a^i (1 - P_a)^{w-i}$.

We solve the Markov chain model balance equations for the stationary probabilities of all states, $\pi_s$. The traffic load served by an aggregator ($j$) is then calculated as $\lambda_j = \lambda_1 + \Delta$, where $\lambda_2$ is calculated using the Markov chain model $f_M$ (Equation (7)). Now, we check whether both models $f_Q$ and $f_M$ produce the same values of the offered loads $\lambda_1$ and $\lambda_2$. If so (i.e., the relative error is small), then this point would be the fixed point solution ($\lambda^*_1, P^*$, and $RTT^*$). However, if not, the search linearly modifies $\lambda_1$ by a small amount $\Delta$ and iterates through the procedure again.

The SA-TCP scheme analytical model has been validated by the network simulator ns-2 in [8]. Comparing the ns-2 results with the model in terms of traffic offered load (i.e., throughput), packet loss rate, and end-to-end delay has showed a close match that is at worst 10% deviation. Thus, the analytical model can be used to analyze and optimize SA-TCP as done in the next section.

4. Performance Analysis

This section provides performance analysis for a smart metering infrastructure in terms of throughput, loss rate and delay. The varying parameters are link capacity, buffering capacity, propagation delay, and number of SA-TCP aggregators. The analysis are based on our mathematical model to study what affects the SA-TCP scheme design and how this scheme compares to the conventional one-hop TCP scheme, Reno standard [34].

Figure 2 depicts the meter-to-utility server architecture with RC devices acting as SA-TCP aggregators. It shows the network and SA-TCP scheme parameters. The link capacity, $C$, in bps is the bandwidth available on the

The fixed-point offered load $\lambda^*_1$ is expected to be close to the bottleneck service rate $\mu$ because the TCP protocol tries to utilize the available capacity to the maximum. Therefore, as the flow chart shows, the search for the fixed point solution starts initially with assuming $\lambda_1 = \mu$. We calculate packet loss rate and packet delay $P$ and $RTT$ at this point $\lambda_1$ using the network model $f_Q$ (Equations (5) and (6)). Next, using the obtained ($P$ and $RTT$), the offered load $\lambda_2$ is calculated using the Markov chain model $f_M$ (Equation (7)). Now, we check whether both models $f_Q$ and $f_M$ produce the same values of the offered loads $\lambda_1$ and $\lambda_2$. If so (i.e., the relative error is small), then this point would be the fixed point solution ($\lambda^*_1, P^*$, and $RTT^*$). However, if not, the search linearly modifies $\lambda_1$ by a small amount $\Delta$ and iterates through the procedure again.

The SA-TCP scheme analytical model has been validated by the network simulator ns-2 in [8]. Comparing the ns-2 results with the model in terms of traffic offered load (i.e., throughput), packet loss rate, and end-to-end delay has showed a close match that is at worst 10% deviation. Thus, the analytical model can be used to analyze and optimize SA-TCP as done in the next section.
ters still transmit data at the same throughput (Fig. 6a). The rate (Fig. 6b) has been significantly reduced while the performance of an SMI system since the packet loss rate nor high queuing delay.

The fact that the amount of delay is larger in this scenario is explained by the store-and-forward point (i.e., queuing delay at the SA-TCP aggregator) which is needed to enqueue packets when congestion occurs in the network. However, increasing bandwidth in the case of SA-TCP is more effective in reducing latency. The results show that unless an effective congestion control mechanism is in place, bandwidth cannot be utilized efficiently. Increasing a link capacity does not resolve congestion since the available bandwidth changes randomly [36, 37]. Therefore, increasing link capacities should be utilized in the support of more meters and more applications, rather than solving congestion in a network [7].

4.1. Impact of Link Capacity

Figure 6 shows the impact of the WAN shared link bandwidth C on the SMI performance. The bandwidth is tested for the range 1 Mbps to 3 Mbps. While Section 4.3 studies the trend of performance as the buffering capacity B1 changes from 20 to 60 packets, this section assumes that B1 is set to a size of 40 packets, which is a middle point in this range that does not cause high packet loss rate nor high queuing delay.

It is evident that the SA-TCP scheme greatly improves the performance of an SMI system since the packet loss rate (Fig. 6b) has been significantly reduced while the meters still transmit data at the same throughput (Fig. 6a). These results are in comparison with the one-hop TCP experiment conducted with a link capacity of 2 Mbps. As explained above, the high packet loss rate of the one-hop TCP scheme, reaching up to 0.8 and 0.9, shows that congestion control is ineffective since the data sources do not reduce the transmission speed when congestion occurs. On the other hand, in SA-TCP, the aggregators are able to adjust the TCP congestion window to control the amount of traffic injected into the network and to enqueue the packets when congestion occurs. Consequently, packet loss rate is significantly reduced more than a half, depending on the number of aggregators.

As expected, increasing the link capacity reduces packet loss rate and packet delay and allows a traffic source to increase its offered traffic load. It is worth noting, however, that with SA-TCP, the impact of changing the link capacity value is higher than that of the conventional one-hop TCP. For example, Fig. 6c shows that increasing C decreases delay faster than the case with one-hop TCP.

The effect of the buffering capacity of SA-TCP aggregators is shown in Fig. 8. In the conventional one-hop TCP, none of the performance metrics changes since there is no aggregator in the architecture. Meters keep transmitting at a high traffic load despite the high loss rate that results. With aggregators, the situation shows that SA-TCP

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of meters, N_M</td>
<td>400,000</td>
</tr>
<tr>
<td>Number of Aggregators, N_A</td>
<td>0 to 1000</td>
</tr>
<tr>
<td>Meter’s T_on / T_off</td>
<td>100 msec / 1 minute</td>
</tr>
<tr>
<td>Packet size</td>
<td>200 bytes</td>
</tr>
<tr>
<td>Link bandwidth, C</td>
<td>1 Mbps to 3 Mbps</td>
</tr>
<tr>
<td>End-to-end propagation delay, T_p</td>
<td>80 to 260 msec</td>
</tr>
<tr>
<td>Aggregator buffer capacity, B_1</td>
<td>20 to 60 packets, DropTail</td>
</tr>
<tr>
<td>Bottleneck buffer, B_2</td>
<td>100 packets, DropTail</td>
</tr>
</tbody>
</table>

WAN network, and B_2 is the buffering capacity in packets. Propagation delay is the time duration it takes for a signal to propagate from a meter to the utility server excluding queuing delays. The buffering capacity considered is B_1 (in segments), which is an SA-TCP aggregator’s. Finally, N_A represents the number of SA-TCP aggregators (i.e., RCs). It is worth noting that the typical bandwidth values for SMIs vary greatly from one country to another and from one utility company to another depending on the communication technology employed, ranging from a few hundred kilobytes as in narrow-band PLC to Mbps as in WiMAX [22]. The values experimented in this section refer to a bottleneck capacity as a result of sharing a link by different smart grid applications. The total number of meters in all the experiments below is assumed to be 400,000. As the number of regional collectors changes, it is assumed that the meters are distributed equally among the regions. The SA-TCP scheme is tested with a number of aggregators N_A equal to 14, 16 and 18 to demonstrate the difference that N_A makes to performance as the other network parameters are tested. For the study of the impact of a larger range of N_A, Section 4.4 presents interesting results pertaining to N_A changing from 0 to 1000.

The discussion below, however, is focused on qualitative interpretation of the results to understand the general trend of performance as the network setup and scenarios change. Table 1 summarizes the SMI network setup and the variable parameters to be analyzed.

4.2. Effect of Propagation Delay

Figure 7 shows the relation between propagation delay and the performance metrics, offered load, loss rate, and delay. As expected, as propagation delay increases, traffic load decreases (Fig. 7a) because it takes longer to increase the congestion window. Since the traffic load is less, the packet loss rate is smaller (Fig. 7b). As for delay, it naturally increases because propagation delay is part of the end-to-end delay. From the figures, we observe that the number of SA-TCP aggregators also impacts the performance results. In this scenario, increasing the number of aggregators causes the delay to increase but the packet loss rate to decrease. For example, an addition of a single aggregator saves around 0.04 on packet loss rate but increases delay by 0.02 seconds.

4.3. Effect of Varying SA-TCP Aggregator Buffer Capacity

The effect of the buffering capacity of SA-TCP aggregators is shown in Fig. 8. In the conventional one-hop TCP, none of the performance metrics changes since there is no aggregator in the architecture. Meters keep transmitting at a high traffic load despite the high loss rate that results. With aggregators, the situation shows that SA-TCP
improves the response to a network congestion by reducing packet loss rate to less than its half. Fig. 8a clearly indicates that SA-TCP improves meter throughput, especially with a higher number of SA-TCP aggregators.

The impact of buffering capacity is shown clearly in terms of packet loss rate and delay (Figures 8b and 8c); the bigger the buffer size, the higher the delay and the smaller the packet loss rate. It is noticed that the meter throughput’s increase (Fig. 8a) is not significant. This is related to packet loss rate and packet delay. Because packet loss rate decreases, the TCP congestion window has a higher chance to grow above the initial value of one, so a meter would be able to transmit more packets. On the other hand, the increased latency caused by queuing slows down the growth of the congestion window and thus limits the increase of throughput. The ripples that appear at the end of the curve $N_A = 18$ are minor calculation errors in the mathematical model, whose values are in the power of $10^{-3}$ range. When zoomed out a little, the general trend of throughput appears to be towards an increase although being a slow increase.

### 4.4. Effect of Varying Number of SA-TCP Aggregators

As shown in Fig. 9, the number of SA-TCP aggregators impacts SMI performance. Increasing the number of SA-TCP aggregators reduces the loss rate in the regions of meters (i.e., LANs). Subsequently, the meter throughput increases (Fig. 9a). However, this increase does not imply that the more aggregators, the better. In fact, when $N_A$ increases, loss in the second stage of the SMI increases (low throughput). This, in turn, makes TCP congestion control mechanism ineffective again. It is evident from Figures 9b and 9c that the total packet loss rate decreases and the total delay increases as $N_A$ increases. Packet loss rate decreases because the added aggregators are able to enqueue packets whenever the network gets congested. With enough pool of data at the SA-TCP aggregators, they are able to enjoy the full range of the congestion window dynamics so as to grow the size or shrink it as the level of congestion changes. Delay increases here in response to congestion since packets tend to wait longer in the queue, rather than being dropped. However, after a certain point ($N_A = 50$), the packet loss rate starts to increase while the packet delay decreases. This behaviour indicates that the TCP congestion control mechanism becomes ineffective as a direct result of having a high number of SA-TCP aggregators, which is similar to the case of having a high number of meters.

### 5. Tuning SA-TCP Architecture

Several input variables impact the performance of the SA-TCP scheme as described in the math model. In what
follows, three optimization problems are formulated. The first problem minimizes the packet loss rate. The second one minimizes the number of SA-TCP aggregators, while the third minimizes the deployment cost of the scheme. The three formulations help the utility provider to optimize the SMI based on the system parameter of interest.

5.1. Minimizing Packet Loss Rate

In certain SMI models, metering traffic is considered delay tolerant [38]. Therefore, one optimization objective is to minimize the scheme’s packet loss rate, while relaxing the delay performance constraint, as presented in the following problem formulation.

\[
\begin{align*}
\min & \quad P_t(N_A) \\
\text{Subject to} & \quad P_t = 1 - (1 - P_m)(1 - P_a) \\
& \quad N_A \leq N_M \\
& \quad N_A \in \{1, 2, \ldots, N_M\}
\end{align*}
\]  

(9)

Equation (10) calculates the total loss rate according to Algorithm 1 and Equation (5), which takes as input all network parameters (e.g., number of meters and all network set-up parameters). The number of SA-TCP aggregators is an integer variable constrained by the number of meters. Figure 9b demonstrates the total SMI loss rate as a function of the number of SA-TCP aggregators. Loss rate continues to decrease as we increase the number of aggregators up to a certain point where loss rate starts to rise again. A large number of SA-TCP aggregators leads to the same problem of ineffective congestion control on the aggregator-utility server side.

The solution assumes that the variable of interest for finding a minimum \(P_t\) is the number of SA-TCP aggregators, \(N_A\). The number of meters and their traffic characteristics, buffering capacities, and network parameters (e.g., link capacities and propagation delay) are given as inputs to the model. The formulation makes an integer non-linear optimization model.

Algorithm 1 finds \(N_A\) that minimizes the loss rate. This algorithm is a modified version from the Rosenbrock optimization method [39]. Because derivation of the SA-TCP model is not viable, this gradient-free direct search method is used. The algorithm is configured by the expansion factor \(\beta_1\) and the contraction factor \(\beta_2\) to be 2 and \(\frac{1}{4}\). Accordingly, the search advances in large steps by doubling the step size \(\Delta\), and when a step goes beyond the optimal point, the algorithm changes its directions and contracts to the middle of the last two points. Every time the algorithm changes its direction, the step size \(\Delta\) shrinks.

The program stops when the step size \(|\Delta|\) becomes smaller.
than the termination tolerance \( \epsilon \), since the search is for an integer value \( (N_A) \). A smaller value of \( \epsilon \) would not change \( N_A \) in Line 6 as a floor value is taken here. Asymptotically, the algorithm takes \( O(\log_2 N_M) \). Figures 10 and 11 show the benefit of the optimization at different number of meters and at different meter traffic rates, respectively. Both figures calculate the packet loss rate at the optimal \( N_A \) value in comparison with the loss rate at other \( N_A \) values and with the loss rate using the conventional one-hop TCP. The dotted curve in both figures show that as the number of meters or their traffic rate increases, the packet loss rate increases rapidly. However, with the optimization model, the packet loss rate can be kept low, as shown by the solid line curve.

### Algorithm 1: Finding the Optimal \( N_A \)

1. \( \epsilon = 0.5 \) // Termination factor
2. \( \beta_1 = 1 \) // Expansion factor
3. \( \beta_2 = \frac{1}{2} \) // Contraction factor
4. \( \Delta = 1 \) // Initial step size
5. if \( (P_f = f(N_A + \Delta) < f(N_A)) \) {
   // Successful move
   set \( N_A = [N_A + \Delta] \)
   set \( \Delta = \beta_1 \Delta \)
} else {
   // Unsuccessful move
   set \( \Delta = \beta_2 \Delta \)
} endif
6. if \( (|\Delta| \leq \epsilon) \) Stop, Return \( N_A \)
7. Go to Step 5

### 5.2. Tuning Number of SA-TCP Aggregators

In the previous formulation, delay is relaxed. It is possible, however, to re-write the optimization model such that delay and packet loss rate are constrained by certain threshold values while varying the number of SA-TCP aggregators. To keep the cost of SMI deployment low, however, it is important to keep the number of SA-TCP aggregators as small as possible. In the following, we formulate an optimization problem to minimize the number of SA-TCP aggregators \( N_A \) for certain requirements on the packet loss rate and packet delay. Tuning \( N_A \) is motivated by Figures 9b and 9c. Since changing \( N_A \) results in a one minimum point, finding this point at first indicates whether a feasible solution exists in terms of packet loss rate. Then Looking at Fig 9c, we see that packet delay can be reduced by decreasing \( N_A \).

\[
\min N_A(C) \quad (13)
\]

Subject to

\[
Ed(\mu_1, \rho_m, B_1) + Ed(\mu_2, \rho_m, B_2) \leq D \quad (14)\\
1 - (1 - P_m)(1 - P_a) \leq L \quad (15)
\]

Inequality (14) constrains the total average time for delivering a packet. It is computed as the average time a packet spends in both queues and the two-way propagation delay. The parameter \( D \) is the maximum delay allowed. Equations (3) and (2) define how delay is calculated. Inequality (15) is defined in Equation (10) constraining the maximum percentage of packet loss allowed, which is \( L \).

Figures 9b and 9c show the performance results pertaining to packet loss rate and end-to-end delay. Clearly, the loss rate is high when no aggregators are used. As we increase the number of aggregators, the packet loss rate decreases, approaching zero; however, the latency increases. When the number of aggregators is too small (e.g., \( N_A = 1 \) and 2), the packet loss rate is greater than in the case of zero aggregators due to the limited buffer capacity. Latency increases in response to congestion since packets tend to wait longer in the SA-TCP aggregator queue. The optimal value of \( N_A \) is found using Algorithm 1. The algorithm searches for the minimum loss rate first. When the loss rate constraint is satisfied, then the delay constraint
the number of available rates in the set. Inequality (19)
represents the link capacity, which is selected from
a set of discrete values \( S \).

C sophisticated and powerful the device is in terms of process-
ing. SA-TCP aggregators and the link capacity, respectively.

where \( N_A \) is checked. At this point, delay peaks. For this reason and
for the sake of minimizing the packet delay metric, \( N_A \) is
decremented. Every time \( N_A \) is decremented by one,
the loss and delay constraints are checked. Decrementing
continues until an infeasible point is hit. Thereafter,
the search stops and returns the last feasible \( N_A \) point. Figure
12 shows that by this optimization model, the packet
loss rate and delay can be guaranteed at a low link ca-
pacity. The figure shows that for a one-hop-TCP scheme
to meet the same constraints, a significantly higher link
capacity is required.

5.3. Minimizing Deployment Cost

Another perspective on optimization is to minimize the
cost of deploying SA-TCP aggregators while taking into
account the cost of link capacity cost. Again, the solution
must meet acceptable loss and delay metrics.

\[
\min \, \zeta_A N_A + \sum_i x_i \zeta_i C_i \\
\text{Subject to} \\
Ed(\mu_1, \rho_m, B_1) + Ed(\mu_2, \rho_a, B_2) \leq D \\
1 - (1 - P_m)(1 - P_a) \leq L \\
\sum_i x_i \leq 1 \\
N_A \leq N_M \\
N_A, \zeta_i \in \mathbb{Z}
\]

where \( \zeta_A \) and \( \zeta_i \) represent the cost of the deployment
of SA-TCP aggregators and the link capacity, respectively.
The cost of an SA-TCP aggregator depends on how so-
plicated and powerful the device is in terms of processing.
\( C_i \) represents the link capacity, which is selected from
a set of discrete values \( S = \{C_1, C_2, ..., C_K\} \), where \( K \) is
the number of available rates in the set. Inequality (19)
ensures that only one capacity value is selected by setting
only one of the \( x_i \)s to 1. Inequalities (17) and (18) rep-resent constraints on the delay and packet loss rate.

The objective function is a monotonically increasing
function in the number of SA-TCP aggregators, link band-
width and their corresponding costs. Thus, it is desirable
to deploy the metering network with the least possible
number of aggregators and minimum bandwidth capacity,
but that may not satisfy the packet loss rate and packet de-
lay thresholds. The figures below help us understand how
the network behaves in relation to \( N_A \) and \( C_i \). Figure 13
shows the impact on packet loss rate as the number of SA-
TCP aggregators and link capacity change. Clearly, packet
loss rate can be enhanced by increasing the link capacity.
An increase of capacity from 1 Mbps to 3 Mbps reduces
packet loss rate by around 0.02. Nevertheless, by increas-
ing the number of aggregators (from 0 to 200), packet loss
rate decreases from around 0.8 down to a certain point
(around 0.12) then increases again. For the delay metric,
however, Fig. 14 shows that packet delay gets worse with
the increase of the number of aggregators while it is en-
hanced by increasing the link capacity. In this scenario,
for example, increasing the link capacity from 1 Mbps to 3
Mbps reduces delay to less than half. In both figures, it is
evident that the impact of link capacity is not as effectual
as that of the number of aggregators.

Solving this optimization problem can be done by any
non-linear integer programming technique (e.g., genetic
algorithm). Alternatively, we use our understanding that
the optimal minimum cost can be obtained by increas-
ing the number of aggregators and the link capacity (from
\( C_i \) to \( C_{i+1} \) per step, where \( i \in \{1, ..., K\} \) in iterations.
The stopping criteria is to satisfy both delay and loss con-
straints. Algorithm 2 finds the minimum objective cost
starting with small values for \( C_i \) and \( N_A \). For every link
capacity, \( C_i \), it iterates through \( N_A \) until it finds a feasi-
ble solution. As a feasible solution is found, it records the
cost and jumps to test with a new value of \( C_i \). Finally,
it compares all the recorded costs to recommend the mini-
imum. For example, lets assume packet loss rate and packet
delay are required to be less than 0.2 and 0.4 seconds, re-
spectively. With a link capacity of 2 Mbps, a number of
aggregators of 50 could be the solution. But if it is more
costly to add an aggregator than to increase link capacity,
the algorithm may choose to increase link capacity to 3
Mbps and reduce the number of aggregators.

This algorithm’s complexity is \( O(N_A C) \). Figures 13
and 14 demonstrate that the number of aggregators makes
a noticeable quick change in loss and delay at small num-
bers. Additionally, given that the link capacity is a small
range, it is expected that Algorithm 2 reaches the solution
much faster than the worst case analysis of \( O(N_A C) \).

6. Conclusion

The SA-TCP scheme has been proposed in an earlier
publication with the objective of making TCP scalable to
the large number of smart meters and sensors. The TCP
Figure 13: Impact of Number of SA-TCP Aggregators and Link Capacity on Loss Rate

Figure 14: Impact of Number of SA-TCP Aggregators and Link Capacity on Delay

Algorithm 2: Finding the Optimal Cost

1: \( C_i = C_1 \)
2: \( N_A = 1 \)
3: Calculate \( P_t = 1 - (1 - P_m)(1 - P_a) \)
4: if \( (P_t \leq L) \) \{ 
5: Calculate \( d_t = Ed(\mu_1, \rho_m, B_1) + Ed(\mu_2, \rho_a, B_2) \)
6: if \( (d_t \leq D) \) \{ 
7: Record objective function total cost
8: Repeat Line 2 with \( C_i = C_{i+1} \) \}
9: else \{
10: \( N_A = N_A + 1 \)
11: if \( (N_A \leq N_M) \) \{ Repeat Line 3 \}
12: else \{ \( C_i = C_{i+1} \), Repeat Line 2 \}
13: Compare recorded costs and choose the minimum

performance has been improved as a result. This paper makes further in-depth performance analysis of the proposed scheme in comparison with the conventional one-hop TCP scheme. Specifically, an analytical model was used to study the impact of the number of SA-TCP aggregators and their buffering capacity, shared link bandwidth, and propagation delay. The performance results are shown in terms of throughput, packet loss rate and packet delivery delay.

The performance analysis of the SA-TCP parameters suggests that the scheme can be tuned for better performance. Therefore, we make further use of the math model by formulating and solving different optimization problems. The objective is to minimize packet loss rate and to find the optimal deployment cost of the scheme.

Further studies, however, will be done to assess the impact of clustering smart meters in various ways. Ongoing research is being done in this regard, taking into account the diversity of metering applications and network characteristics.

Acknowledgement

This work was made possible by NPRP 6 - 711 - 2 - 295 grant from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References


