Modelling and Performance Analysis of TCP Variants for Data Collection in Smart Power Grids

Tarek Khalifa, Khaled Bashir Shaban, Atef Abdrabou, Abdoulmenim Bilh, Kshirasagar Naik

PII: S0140-3664(16)30716-2
DOI: 10.1016/j.comcom.2016.12.018
Reference: COMCOM 5433

To appear in: *Computer Communications*

Received date: 10 November 2015
Revised date: 14 October 2016
Accepted date: 23 December 2016

Modelling and Performance Analysis of TCP Variants for Data Collection in Smart Power Grids

Tarek Khalifa[a,*], Khaled Bashir Shaban[a], Atef Abdrabou[b], Abdoulmenim Bilh[c], Kshirasagar Naik[c]

[a] *Computer Science and Engineering Department, Qatar University, Doha, Qatar, P.O. Box: 2713*
[b] *Dept. of Electrical Engineering, United Arab Emirates University, Al-Ain, Abu Dhabi, UAE, 15551*
[c] *Dept. of Electrical & Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada, N2L3G1*

Abstract

Smart grid communication networks adopt a variety of communication technologies interconnecting numerous and diverse equipment. The requirement of supporting a large traffic volume over such networks efficiently, reliably, and fairly among various applications calls for the study of the effectiveness of the transmission control protocol (TCP). The question raised in this paper is whether various TCP variants perform differently.

We are particularly interested in comparing the performance of TCP-Reno and TCP-Vegas variants when handling sensory and metering traffic. The former variant follows a loss-based congestion control mechanism while the latter adds delay-based enhancements. Contrary to expectations, our simulation and analytical comparison results show that TCP-Reno outperforms TCP-Vegas in terms of packet loss rate. However, to take advantage of Vegas' added features, a previously proposed scheme named split- and aggregated-TCP (SA-TCP) is recommended. The scheme splits meters' TCP connections at intermediate devices and forwards collected data over an aggregated TCP connection to a data center.

Keywords: smart power grid, smart metering infrastructure, congestion control, telecommunication traffic

1. Introduction

In a smart power grid, information is gathered from a large number of intelligent electronic devices (e.g., smart meters, phasor management units (PMUs), remote terminal units (RTUs), and other sensors) for the purpose of facilitating new capabilities, such as the integration of renewable energy sources, self-healing of the grid, efficient generation, and provisioning of improved power quality [1, 2, 3].

The amount of traffic is indeed significant as metering and sensing devices will be remotely controlled, configured, and read in a periodical, on-demand, and event-driven fashion. A variety of applications with different characteristics and requirements [4, 5] share the grid's communication network (Fig. 1). Thus, regardless of whether a public or utility-owned communication network is employed, it is essential that communication protocols provide services such as fairness among the flows of traffic, congestion control and reliability [6, 4, 7, 8].

*Corresponding author. Tel.:+97466506092
Email addresses: `tarek.khalifa@qu.edu.tr` (Tarek Khalifa), `khaled.shaban@qu.edu.qa` (Khaled Bashir Shaban), `Atef.abdrabou@uaeu.ac.ae` (Atef Abdrabou), `Abilh@uwaterloo.ca` (Abdoulmenim Bilh), `snaik@uwaterloo.ca` (Kshirasagar Naik)

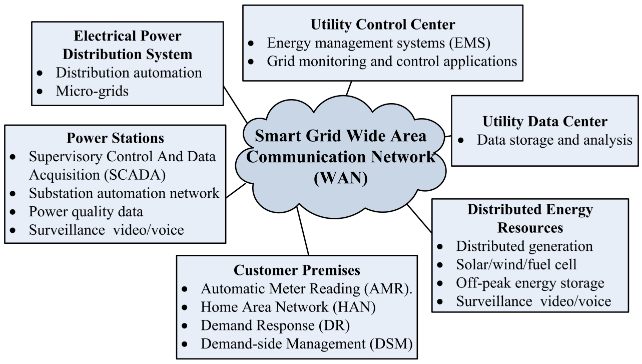


Figure 1: Utility Applications Sharing Grid Communication Network

An Internet Protocol (IP)-based smart grid has been strongly cherished for the above reason, i.e. enforcing reliability and congestion control. In addition, TCP/IP is a well-developed and mature protocol suite, which can be scaled down to operate in small devices such as sensors and meters [9, 10, 11, 12]. TCP/IP supports all power system operator's data, voice, and video requirements [13, 14]. It simplifies network management and gives the network operator wider options of compatible equipment to choose from. Certain application-level protocols for power systems are already compatible with TCP/IP, such as Distributed Network Protocol 3 (DNP3) and International Electrotechnical Commission (IEC) 60870-5-104.

In a network that serves traffic from various applications transmitting at random, the chance for communication bottlenecks to build up is high. The chance is even higher if such a network is formed by heterogeneous technologies with mismatched bandwidths, which is the case with smart grids. Transmission Control Protocols (TCPs) come into picture here to realize vital services. They resolve congestion due to bottlenecks, ensure reliable delivery of data, and share the links fairly [15, 16, 17].

In light of the above, this work focuses on the transport layer protocols performance in terms of communication measures, namely, packet loss rate, end-to-end delay, and throughput. The analysis considers application and transport layers characteristics. At the transport layer, two major variants of TCP, Reno [18] and Vegas [19], are investigated for their performance in a smart grid. Furthermore, the two variants are compared when employed in a proxy-like scheme, called Split- and Aggregated-TCP (SA-TCP) [20], a scheme that improves TCP's ability to effectively perform congestion control. The SA-TCP scheme splits meters' TCP connections at a regional collector (a.k.a concentrator) and unifies all the TCP connections to forward data to the utility server.

It has been reported in [20] that SA-TCP improves the TCP performance in a smart grid regardless of the TCP variant employed. However, in this paper, TCP-Vegas, as opposed to TCP-Reno, has been selected for further investigation for its revolutionary change. Vegas is a proactive protocol that adjusts the transmission speed in response to changes in packets' round trip times, while Reno is a reactive one as it responds to congestion after a packet loss occurs. Although, TCP Vegas was reported in [19] to achieve 37% to 71% better throughput and up to 20% less packet loss rate on the Internet as compared with TCP Reno, surprisingly, Vegas performs worse if adopted by a smart metering infrastructure in a one-hop TCP scheme (Section 5). We show that unless the SA-TCP scheme is applied, the Vegas features are not effective. Indeed, designing a communication network should not be based only on the amount of data to be collected, but also on the number of sources this data is coming from and more importantly the TCP mechanism followed.

In short, this paper makes the following contributions:

- It develops an analytical formulation for data collection in a smart grid under the TCP-Vegas variant and under the SA-TCP scheme. A combination of Markov chain and queuing model has been used for this purpose. As validated by ns-2 simulations, the formulation is able to accurately capture the scheme's behavior under the Vegas version of TCP in terms of packet loss rate and delay.
- It provides performance analysis of data collection under the TCP-Reno and TCP-Vegas protocols. It shows that although TCP-Vegas is known to perform better on the Internet, in a smart metering network, the TCP-Reno protocol, in fact, performs better in terms of packet loss rate and throughput. However, it is shown also that the SA-TCP scheme makes it possible to make the most out of Vegas' added features to improve the overall TCP performance and outperform Reno at some point.

The rest of the paper is organized as follows: Section 2 summarizes the most relevant research work done for smart metering networks. Section 3 presents the smart metering data collection model and describes the mechanism of the SA-TCP scheme. Section 4 provides the analytical formulation of data collection under TCP-Vegas and the SA-TCP scheme. Section 5 provides performance analysis for smart metering traffic under one-hop TCP and under SA-TCP congestion control set-ups. Finally, Section 6 provides summary and concluding remarks.

2. Related Work

In the smart grid literature, reliability and congestion control issues have been addressed in different ways. Niyato et al. [21] provide a study of the impact of communication reliability on the cost of power supply. They show that, due to congestion, a high packet loss rate for data collected from smart meters impacts the estimation of power demand and subsequently increases the electricity cost. In [22], the same authors introduce an optimization model that takes into account the reliability of communication infrastructure to propose a way of supplying electric power economically. In addition, based on the reliability analysis of [22], they propose in [23] adding redundant nodes at certain points of the grid to enhance communication reliability.

Kim et al., in [16] and [24], design a transport layer protocol focusing on reliability and latency of delivered metered data packets. It is an end-to-end lightweight protocol that eliminates many of the TCP's mechanisms that introduce delay while not needed in a smart grid, for example, Nagle's algorithm and delayed acknowledgements. The protocol, however, is not TCP-friendly and does not support congestion control. Similarly, Saputro and Akkaya [25] modify the Retransmission Timeout (RTO) estimation and back-off mechanisms such that the TCP performance is improved in terms of packet delay. However, the modification makes TCP more aggressive by quickly retransmitting lost packets, which could lead to congestion.

Allalouf et al. [26] aim at handling congestion by performing traffic engineering such that the amount of metering and sensory data is reduced at intermediate nodes in the data traffic path. They try to strike a balance between data volume reduction and preserving the needed quality of information at the utility center and at certain nodes along the data collection path. Ren et al. [27] adopt a different approach. They design a receiver-based mechanism in which the receiver is the one that manages congestion control. This approach, however, may not suit smart grid event-based data.

Cherukuri and Nahrstedt [28] argue that even with bandwidth reservation along the traffic path as done in NASPInet [29], congestion can still occur at times of increased traffic demand and bursty traffic situations. This leads to deviations in PMU and IED realtime traffic. Li [30] also stresses the fact that congestion control and quality of service are strongly connected, especially in a heterogeneous network.

Therefore, the author proposes to modify the TCP congestion control mechanism such that an estimate of the bandwidth is involved in adjusting the congestion window.

This work, in fact, complements our previous research on TCP published in [31]. The previous work highlights the shortcomings of TCP in scaling to the large number of data sources and the need for a split and aggregation scheme (*i.e.,* SA-TCP), and also provides an end-to-end analytical model to represent the throughput and delay of a smart grid network under TCP-Reno.

3. Problem Description

This section identifies the shortcomings of TCP in performing congestion control effectively for a large number of data sources in a smart grid. It describes first the system model and the traffic characteristics then explains the control congestion control mechanisms for TCP-Reno and TCP-Vegas. The section also highlights how SA-TCP enhances the TCP congestion control performance.

3.1. System Model

Data collection in smart grids involves dealing with hundreds of thousands to millions of metering and sensing devices. Distributed in regions, these devices report their data by means of wireless communication (WiMAX or LTE) or Power Line Carrier (PLC) [32]. In every region, there exists a gateway to interconnect the meters with the utility wide area network (WAN). The gateways are called advanced metering regional collectors (RCs) or concentrators, typically installed on poles at fixed locations [33, 34, 35]. With this topology, data packets are transmitted from meters to the RC in a single hop and then get forwarded through the utility WAN to the utility server [6]. Figure 2 demonstrates this architecture.

It is assumed that the sources transmit small data packets (a few hundreds of bytes) in response to events or periodically at a low rate (e.g., a packet every a few seconds or minutes) [4, 36]. Automatic meter reading applications, for instance, recommend reporting readings every five minutes. For demand side management or for power quality measurements, data may be reported randomly even more frequently [7]. Packets are delivered to the utility server intact [4, 6], which means that statistical data aggregation techniques (e.g., finding minimum or maximum or averaging [37]) are not applied. Real-time pricing and demand management are some application examples that dictate this requirement. This requirement along with the grid topology and the lack of redundant data packets or redundant metering devices differentiate this area from wireless sensor networks [38].

In other words, smart meters and sensors are TCP/IP devices that have enough capability of communicating data through designated regional collectors. Data is reported by every meter over a separate TCP connection with the utility server (Fig. 3-A) to ensure reliable packet delivery and to realize congestion control in the network. Without effective congestion control, the retransmissions of lost data packets together with the demands of regular ongoing transmissions slow down the network and lead to more packet losses. For the sake of studying TCP, we assume that all the links are loss-free to ensure that any packet loss is indeed due to congestion. Different communication technologies may have their impact on TCP performance, which has been studied widely [39, 40]. The focus of this work, however, is rather on comparing the performance of TCP variants regardless of the underlying communication technology. The details of the underlying communication technologies have been abstracted for two reasons. First, the effect of communication technologies on the performance of different TCP variants has been addressed before in the literature [41, 42]. Second, we focus on comparing the performance of different TCP schemes (single-hop and SA-TCP) with different TCP variants mainly from the point of view of transport layer functionality (the congestion control efficiency) given the considered smart metering system model.

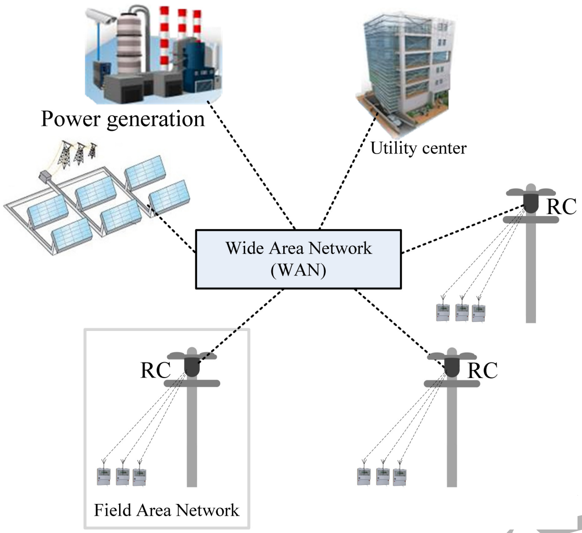


Figure 2: Communication Architecture

3.2. One-hop TCP vs. SA-TCP Scheme

Metering and sensing devices in smart grids typically transmit with low data rate. However, the large number of sources collectively produce a significant amount of data. Therefore, when network congestion occurs, the TCP congestion control mechanism explained above would be ineffective in resolving the congestion. The reason is that the data sources already transmit at a low speed with a congestion window size of one or two segments. Hence, reducing the overall speed upon congestion is infeasible. The problem is equivalent to replacing one TCP connection with a large number of TCP sub-connections to deliver the same amount of data, assuming each sub-connection transmits at a low rate. This keeps the congestion window size at one MSS (Maximum Segment Size). If congestion occurs in the network, the original TCP connection may reset its congestion window to one, which is the minimum size. On the other hand, if a sub-connection is required to reduce its transmission rate, it will reduce its congestion window to one at best. Consequently, the total congestion window will stay as high as the summation of the individual sub-connection congestion window sizes [15]. Similarly, the metering and sensing traffic will be highly aggressive as there will be hundreds of thousands of TCP connections transmitting at a low data rate. The situation is aggravated if the used TCP variant (e.g., Vegas) employs an aggressive congestion control mechanism, a one that is designed to reduce the congestion window slowly and retransmits lost packets hastily. Thus, even though the data traffic may suffer packet drops, the total traffic rate stays unchanged, which intensifies the packet loss rate and leads to congestion collapse [15].

The SA-TCP scheme [20] comes as a solution to make TCP congestion control effective. SA-TCP does not change the internal congestion control mechanism, which could be based on any TCP variant; rather, as depicted in Fig. 3-B, SA-TCP introduces the idea of aggregation at the transport layer. The regional collectors (RCs) are upgraded to operate at the transport layer rather than being restricted to the IP layer. Each data source initiates a TCP connection with an RC in its respective region, over which data packets are reliably transmitted. An RC, in turn, creates an aggregated TCP connection with the utility server and forwards the data packets over this unified TCP connection. Consequently, the TCP connections between the meters and the utility server are no longer one-hop, but rather, two-hop TCP connections.

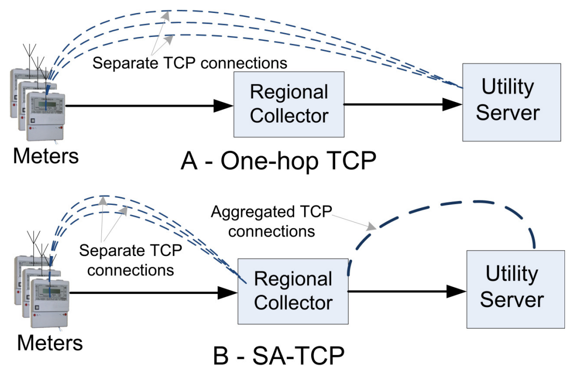


Figure 3: One-hop TCP vs. SA-TCP Schemes

3.3. Reactive vs. Proactive TCP Variants

The Reno [18] and Vegas [19] variants of TCP have been chosen in this study of data collection over one-hop TCP and over SA-TCP schemes because these variants represent two diverse mechanisms of achieving congestion control. TCP-Vegas is an example of a class of proactive TCP variants (*e.g.,* DUAL [43], CARD [44] and Nice [45]) that observes packet delay, not packet loss, to predict congestion. On the other hand, TCP-Reno and similar ones, *e.g.,* Tahoe [46], SACK [47] and NewReno [48]), are reactive protocols that adjust the congestion window size after packet loss is detected.

The common behavior of TCP in achieving congestion and flow control is to adjust the source's congestion window. A traffic source keeps increasing its transmission speed by enlarging the window size, but if a packet goes unacknowledged, which is an indication of congestion in the network, the source lowers its speed. The reduction mechanism, in TCP Reno and the others like it, occurs in the following manner. If an unacknowledged packet times out, the source decreases the transmission rate to the minimum, which is one segment per round trip time, by resetting the congestion window to its initial size (typically one or two segments). Nonetheless, If the source receives three duplicate ACKs, indicating a missing packet, it halves its sending rate by halving the congestion window size.

On the other hand, TCP-Vegas uses round trip times (RTT) to estimate bottleneck buffer occupancy. It calculates the absolute number of packets enqueued at the bottleneck router as a function of the expected transmission rate. If the rate falls below a certain threshold, the congestion window is decreased by one or otherwise increased by one. The aim is to adjust the congestion window size before packet loss occurs and also to reduce the number of timeouts, and hence the packet loss rate is reduced and the throughput is increased. If packet loss occurs, TCP Reno's timeout and fast retransmit/fast recovery mechanisms are initiated. TCP-Vegas outperforms TCP-Reno in terms of improving the transmission rate stability and the overall throughput of a TCP connection. Section 4 further details the mechanism, which is needed in the analytical model.

3.4. Objectives

To be able to make an informed decision about the SA-TCP variant selection is the objective. Having learned that the TCP congestion control mechanism is ineffective, the question raised here is whether different TCP variants would make a difference. This paper investigates the performance of one-hop TCP in a smart grid setting under two major variants, Reno and Vegas, since they represent two different ways (loss-based and delay-based) of achieving congestion control. Furthermore, the two variants when operating

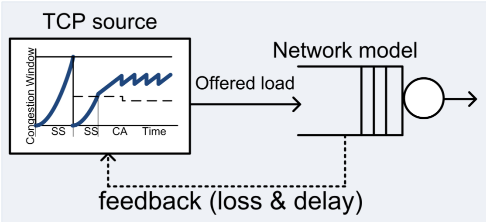


Figure 4: Source - Network Interaction

under the SA-TCP scheme have been compared against different performance metrics. The comparison has been done analytically and validated by extensive simulation results.

4. Analytical Model

The objective of mathematically modelling SA-TCP is to compare its performance with a one-hop TCP scheme under both Reno and Vegas TCP variants. The focus of this paper is particularly on modelling TCP-Vegas in the context of SA-TCP in order to compare it with SA-TCP-Reno. The reader is referred to [31] for a detailed analytical model for data collection under SA-TCP-Reno. The analytical model below takes into account the number of data sources and regional collectors, the application behavior, the TCP-Vegas congestion control mechanism, and network characteristics, such as the capacity, propagation delay, and queuing. The model allows one to reproduce the actual behavior of collected data, depicted in terms of the following performance metrics:

- *Packet loss rate:* It is the probability of packet loss due to buffer overflow in the network.
- *Offered load:* It is calculated as the rate at which data segments are produced by a TCP source.
- *Throughput:* It is the number of segments that are successfully delivered per unit time.
- *Packet end-to-end delay:* It is the time a packet takes to arrive at its destination.

Table 1 provides a summary of the notations and variables used in the modelling.

4.1. Modelling Technique

The end-to-end performance measures are obtained by combining two stages of the traffic, over the first and second TCP connections. As revealed in Fig. 3-B, data packets are first sent to the corresponding regional collector in that area according to the TCP behaviour of the sources (*i.e.,* meters or sensors) and the regional collector status. In the second stage, the packets are forwarded through the network to the utility server according to the regional collectors' (acting as sources) TCP mechanism and the grid's wide area network status.

Figure 4 demonstrates the process to be modelled in each stage of SA-TCP. In each stage, the sources and the network are modelled separately, and then a common operating point for both models, *i.e.,* source and network, is found. This operating point is expressed in terms of the network average offered load rate, packet loss rate, and packet delay. It is obtained by following the fixed-point approximation method [49, 31], which is briefed in the following iterative approach:

Table 1: Notations and Variables

Variable	Description
N_M	Total number of meters
N_A	Total number of SA-TCP aggregators
d_1	Packet latency on the meter-regional collector side
d_2	Packet latency on the regional collector-utility server side
Ed	Expected value of packet delay
D	Tolerated latency
T_p	Propagation delay
T_q	Queuing delay
ET_q	Expected value of queuing delay
T_r	Average round trip time
T_{on}	Average time the meter being active
T_{off}	Average time the meter being inactive
L	Maximum packet loss rate tolerated
w	Congestion window size
w_t	Slow start window size threshold
W_M	Maximum congestion window
P_i^w	Probability of loss of i segments in a window of size w
q_w	Probability of successful delivery of all segments in a window of size w
P_m	Probability of packet loss on the meter-regional collector side
P_a	Probability of packet loss on the regional collector-utility server side
ρ_m	Queue utilization factor on meter-regional collector side
ρ_a	Queue utilization factor on regional collector-utility server side
π_i	Stationary probability for Markov model state i
s	A state in Markov chain
W_s	Congestion window size at Markov state s
λ_i	Average traffic rate generated by a meter
λ_t	The total traffic rate formed by meters in a region
γ_j	Average traffic rate of an SA-TCP aggregator
γ_t	Total traffic rate of SA-TCP aggregators in the second stage

1. A continuous-time Markov chain is used to model source's offered load as a function of the network parameters: probability of loss and delay. It is assumed that each source follows an on-off model, explained in Section 4.4. Consequently, the aggregation of a large number of such independent on-off sources follows a Poisson distribution [50].
   In the first stage of SA-TCP, each source produces data at rate λ_i. Individual data sources may have different application characteristics, for example, an active or inactive time duration or propagation delay to the regional collector. The total rate is $\lambda_t = \sum_{i=1}^{n} \lambda_i$. Section 4.4 explains how λ_i is calculated.
   In the second stage, the regional collectors act as data sources transmitting data segments to the utility server. Similarly, an SA-TCP aggregator's segment generation mechanism is represented by a Markov chain model mimicking the TCP congestion window dynamics. The individual rate is γ_j, and the total is $\gamma_t = \sum_{j=1}^{N_A} \gamma_j$. Section 4.3 explains how λ_j is calculated.
2. The network parameter pair (packet loss and packet delay) can be calculated using queuing theory as functions of the offered traffic load of the sources for the first and second stages.

$$(P_m, d_1) = g(\lambda_t)$$
$$(P_a, d_2) = g(\gamma_t)$$

   where P_m and P_a are the probability of packet loss in the first and second stages of SA-TCP, respectively; d_1 and d_2 are the first and second stage average RTTs. The queuing model represents the network bottlenecks in the first stage and in the WAN network. In the first stage, a regional collector is the one queuing the meters data packets. Section 4.5 explains how these network parameters are calculated.
3. Starting with an initial value for the sources' offered load. The packet loss rate and delay are calculated. The obtained values are used to calculate a new value for the offered load. Iteratively, every time new values of loss and delay or offered load are calculated, they are fed back to obtain another set of new values. This iterative procedure is performed in each stage separately and is stopped when no further change occurs, marking the fixed-point solution. Note that the obtained offered load of the second stage is used as the queuing model service rate of a region in the SA-TCP first stage.

The next subsections show how the three components (the traffic generation process of data sources, the traffic generation process of regional collectors, and the network model) are mathematically modelled under the assumption that TCP-Vegas is employed.

4.2. TCP-Vegas Congestion Control Mechanism

As indicated in Section 3.3, TCP-Vegas adds new delay-based techniques to the slow start and congestion avoidance phases. The following is an outline of the TCP-Vegas main congestion control techniques that are mathematically modeled.

- TCP-Vegas keeps the estimated number of backlogged packets in the network between two thresholds, a and b where $a \leq b$. The number of backlogged packets N_b is approximated by this equation.

$$N_b = \frac{w}{RTT}(RTT - RTT_{min}) \tag{1}$$

  where w is the congestion window size, RTT is the actual round trip time, and RTT_{min} is the minimum value learned from previous transmissions.

If $N_b < a$, TCP-Vegas concludes that it is safe to increment the congestion window size by one. If $N_b > b$, then it decides that the window size gets decremented by one. If $a \leq N_b \leq b$, the window size is not changed.

- TCP-Vegas adjusts the congestion window continuously, so it does not reduce the window size harshly when losses occur. Therefore, upon losses that cause fast retransmit, TCP-Vegas reduces the congestion window only to three quarters its current size.
- TCP-Vegas does not wait for three duplicate ACKs. Instead, a sender keeps a record of RTT durations of every packet on transit. When the first or second duplicate ACK is received, TCP-Vegas checks whether the elapsed time duration for the possibly missing packet exceeds RTT. If so, Vegas decides to retransmit it. This mechanism helps reduce the time to detect a lost packet from the third duplicate to the first or the second. More importantly, it reduces the number of timeouts by allowing detection even though there may be no second or third duplicate ACK, for example, when the congestion window is small.
- In the slow start phase, TCP-Vegas avoids congestion by doubling the congestion window size only every other RTT. That is done to accurately estimate RTT and N_b. If $N_b > \frac{a+b}{2}$, Vegas enters the congestion avoidance phase.

4.3. Model of Regional Collectors

The aim of this part is to be able to calculate the load that an SA-TCP aggregator (i.e., regional collector) can offer (γ_j). A continuous-time Markov chain is designed to mimic the congestion control window dynamics. The model here follows TCP-Vegas as explained in [19] and [51]. Every state of Markov model (Figure 5) represents the congestion window size, the slow start threshold, and the congestion control phase. The numbers in the states correspond to the number of segments TCP sends in an RTT. States numbered as zero represent timeout events and mean that there are no segment transmissions. The rightmost column states, with dashed numbers, model the fast retransmit/fast recovery phase. The following summarizes the transition rates among the chain states. Variables are defined in Table 1.

- Assuming an exponential distribution of RTT ($T_p + T_q$) with average T_r, the transition rate becomes $\delta = \frac{1}{T_r}$, where T_p is the propagation delay, T_q is the queuing delay, and T_r is the average round trip time. The assumption that RTT is exponential is justified in networking by the fact that network traffic is random, which means that it is difficult to tell how much time is left for an ACK to arrive. This means that the probability of the ACK to arrive after time t does not change as some of the time passes, which is the memoryless inherent characteristic of the exponential distribution.
- For the states with $w < w_t$, transition is made to the same window size with rate δ, reflecting the fact that TCP-Vegas does not change the congestion window size every RTT.
- During the slow start phase, doubling the congestion window size is achieved with rate $P(N_b \leq \frac{a+b}{2})q_w\delta$.
- Transitioning from a state in the slow start phase to a state in the congestion avoidance phase with $w = w + 1$ is achieved at rate $P(N_b \geq \frac{a+b}{2})q_w\delta$.
- For states of $w < 4$, the loss of any segment causes transition to a timeout state with $w_t = \frac{w}{2}$. The rate of transition is $\delta(1 - q_w)$. During timeout, no segments are transmitted.

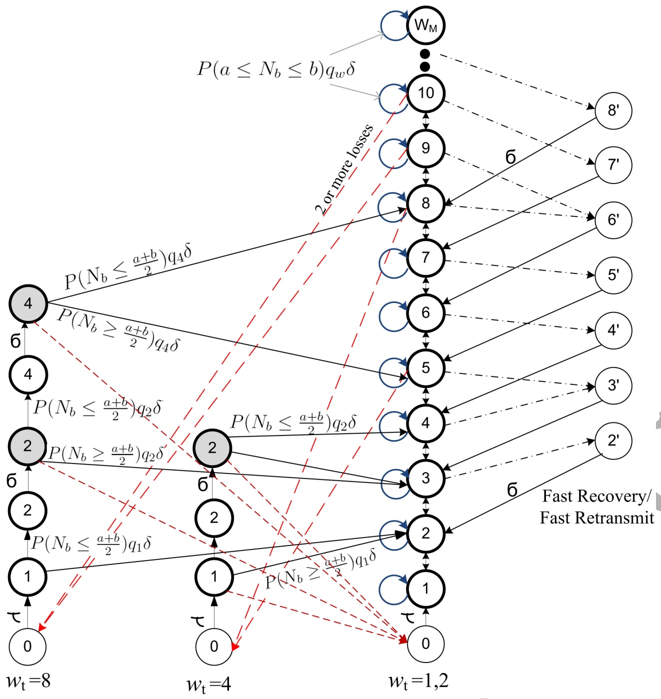


Figure 5: Regional Collector Markov Model Under TCP-Vegas

- For states of $w \geq 4$, the loss of one segment causes transition to the fast retransmit states with rate δP_1^w.

  However, for two or more losses, the transition is made to a timeout state with rate $\delta(1 - q_w - P_1^w)$

- During the collision avoidance phase, for states with $1 < w < W_M$, the window increases linearly with rate $P(N_b < a)q_w\delta$; it decreases with rate $P(N_b > b)q_w\delta$; and it stays unchanged with rate $P(a \leq N_b \leq b)q_w\delta$. This mechanism tries to keep buffer occupancy between a and b.

- For the state with $w = 1$, the window increases with rate $P(N_b < a)q_1\delta$ and remains unchanged with rate $P(N_b > a)q_1\delta$.

- For the state with $w = W_M$, the window decreases with rate $P(N_b > b)q_{W_M}\delta$ and remains unchanged with rate $P(N_b \leq b)q_{W_M}\delta$.

For the estimation of N_b, the minimum RTT is approximated as a two-way propagation delay (T_p), and the actual RTT is approximated by the addition of the queuing delay to the propagation delay ($RTT = T_p + T_q$). Therefore, Equation (1) becomes as follows:

$$N_b = \frac{wT_q}{T_p + T_q} \tag{2}$$

The probabilities $P(N_b \leq a)$ and $P(N_b \geq b)$ are derived from the queuing model, which is shown in Section 4.5.

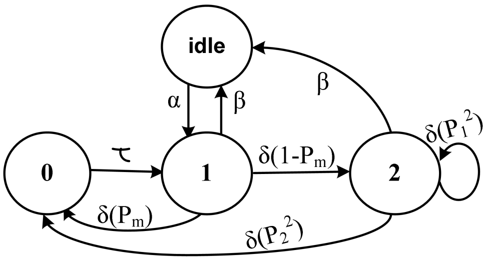


Figure 6: Meter Markov Model Under TCP-Vegas

Next, The Markov chain is solved for the limiting probabilities, π_s, where s is a state. Equation (3) calculates the average traffic load on the jth regional collector (γ_j). The states of the fast retransmit phase are excluded because they do not correspond to the actual transmission of segments.

$$\gamma_j = \delta(\sum_s W_s \pi_s) \tag{3}$$

where W_s denotes the congestion window size of State s. The summation ($\sum_s W_s \pi_s$) calculates the mean size of the congestion window. Multiplying all that by δ results in the offered load because δ corresponds to the average time between two successive segments. The overall traffic from all the regional collectors going into the WAN bottleneck can be given as follows.

$$\gamma_t = \sum_{j=1}^{N_A} \gamma_j \tag{4}$$

4.4. Model of Data Sources

Similar to the regional collector, a data source's throughput model captures the dynamics of the TCP congestion control window but with considering the application layer behavior. The sources are assumed to produce packets at a low data rate with periodical durations of inactivity keeping the congestion window size as small as one or two. It is also assumed that the data sources are not synchronized. Therefore, even if every meter is scheduled to send data deterministically, the meters can be considered as on-off sources. The combination of a large number of such on-off sources leads to a Poisson distribution [50]. Such assumptions apply to meters and sensors in a power distribution system [4].

The modelled Markov chain in this case transits between active and idle states with exponentially distributed rates of $\beta = \frac{1}{E[T_{on}]}$ and $\frac{1}{E[T_{off}]}$, respectively. Through setting the on-off periods, the system is made flexible as per a certain application's characteristics. That is at the application level. At the transport level, the congestion window size is limited to two segments, reflecting the assumption that a meter sends data for a short period of time (*e.g.,* T_{on} = 100 msec) [4, 36].

The states and transition rates of a meter's Markov chain, combining the application and TCP behaviors, are depicted in Fig. 6. Timeout is represented by the transitions to State 0 when one segment is lost. Different from the TCP-Reno model ([31]), in TCP-Vegas, one duplicate ACK is enough to trigger a retransmission, as in State 2, which explains why TCP-Vegas is more aggressive than TCP Reno in small window sizes.

The Markov chain is solved for the limiting probabilities of π_1 and π_2. The offered load of a source i is, therefore, $\lambda_i = \delta\pi_1 + 2\delta\pi_2$

The total traffic generated by all the data sources in a region, each with its On/Off and RTT characteristics, is $\lambda_t = \sum_{i=1}^{n} \lambda_i$.

4.5. Network Model

The network model determines the packet delay distribution along with probability of packet loss and average delay. It employs a queuing model that takes into account all network-related characteristics, including the buffer size, link capacity, packet arrival process (assumed poisson), queue management scheme (assumed DropTail), and network environment (wired or wireless links).

Assuming an M/M/1/B queuing system, (5)-(10) are used, in addition to the delay distribution [52], needed specifically for the TCP-Vegas model as shown above.

$$P(\rho, B) = \frac{\rho^B (1-\rho)}{1-\rho^{(B+1)}} \tag{5}$$

$$ET_q(\mu, \rho, B) = (\frac{1}{\mu}).\frac{1-(B+1).\rho^B + B.\rho^{(B+1)}}{(1-\rho)(1-\rho^B)} \tag{6}$$

$$Ed(\mu, \rho, B) = T_p + ET_q(\mu, \rho, B) \tag{7}$$

$$\rho_m = \frac{\lambda_t}{\mu_1}, \;\; \mu_1 = \frac{\gamma_t}{N_A}, \;\; \rho_a = \frac{\gamma_t}{\mu_2}, \tag{8}$$

$$P_m = P(\rho_m), \;\; P_a = P(\rho_a) \tag{9}$$

$$d_1 = Ed(\mu_1, \rho_m, B_1), \;\; d_2 = Ed(\mu_2, \rho_a, B_2) \tag{10}$$

$$Pr(T_q \le t) = 1 - \frac{e^{-\mu t}}{1-\rho^B} \sum_{i=0}^{B-1} \frac{(\lambda t)^i}{i!} + \frac{\rho^B e^{-\mu t}}{1-\rho^B} \sum_{i=0}^{B-1} \frac{(\mu t)^i}{i!} \tag{11}$$

The probabilities $P(N_b \le a)$ and $P(N_b \ge b)$, required for the regional collector model, are calculated by (11). This calculation is achieved by manipulating Equation (2) such that T_q is put on one side and then applying Equation (11).

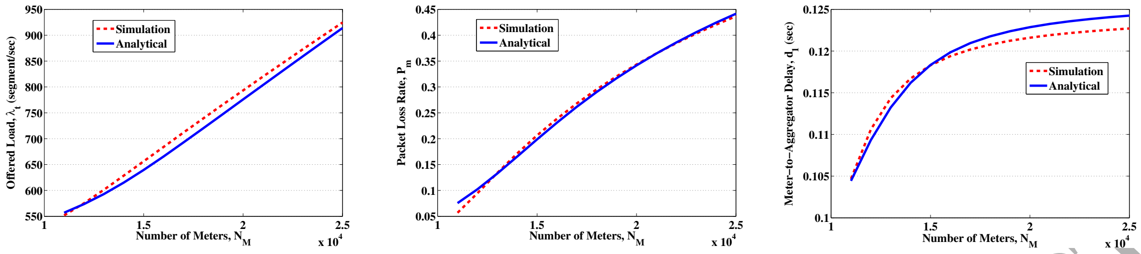
$$P(N_b \le a) = P(Tq \le \frac{aT_p}{w-a}) \tag{12}$$

$$P(N_b \ge b) = 1 - P(T_q \le \frac{bT_q}{w-b}) \tag{13}$$

4.6. Model Validation

The analytical model is validated against ns-2 simulations for a variety of scenarios testing network parameters over wide ranges. The results demonstrate that the math model can accurately predict the packet loss rate, delay, and the offered load of data sources and regional collectors. The model estimates the network operating point using the fixed point approach explained above. In the experiments below, the TCP-Vegas parameters are configured as $a = 1$ and $b = 3$.

The analytical model of data sources running TCP-Vegas is validated for a wide range of number of meters in a given region reporting data to a single regional collector. The meters are configured to operate as on-off sources with T_{on} =100 msec and T_{off} =1 minute. The two-way propagation delay is 50 msec,



(a) Offered Traffic Load (b) Packet Loss Rate (c) End-to-end Delay

Figure 7: Validation of TCP-Vegas in a region

and the available link bandwidth is 1 Mbps. The analytical results are obtained by following the fixed-point approximation method (Section 4.1), which combines in this case the Markov model of Fig 6 and the network model. Figure 7 shows that as the number of meters in a region changes, the model's estimation of the three performance metrics stays in a close match with the simulation results.

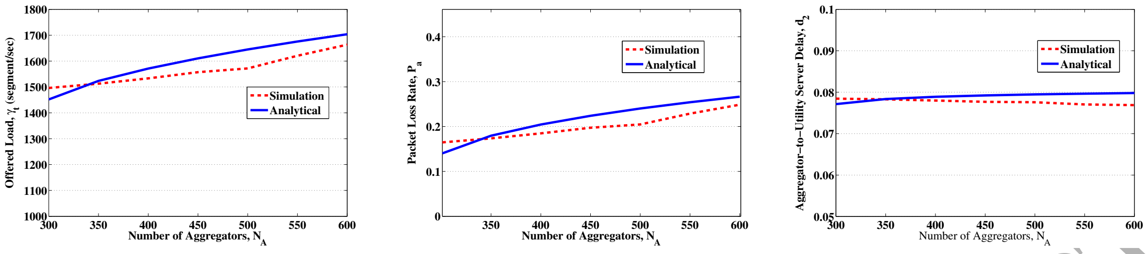
Similarly, Fig. 8 validates the three performance metrics in the second TCP hop, which is formed by regional collectors (i.e., SA-TCP Aggregators) and the utility server (Fig. 3). This experiment assumes that data is transmitted by a group of regional collectors through a shared network to the utility server. The link capacity between the regional collectors and the utility server is 2 Mbps with a 40-packet buffer capacity (240 bytes per packet), and the propagation delay is 50 msec. The analytical model is obtained by the TCP-Vegas Markov chain model presented in Fig. 5 (Section 4.3) . Clearly, as the number of regional collectors changes, it is shown that the analytical model curve matches the simulation results.

The above experiments prove that our modelling technique is accurate in predicting the behavior of both metering devices and regional collectors. This gives us the flexibility of testing different combinations of smart grid setups. For example, in one scenario, meters report to a utility server directly over one-hop TCP. Alternatively, every group of meters connect to a given regional collector, and that regional collector forwards the data to the utility server. The number of meters and number of regional collectors can be changed in every scenario, as done in the next section.

5. Performance Results and Discussion

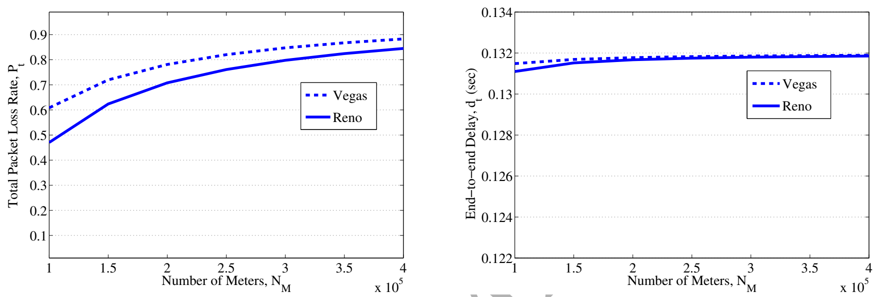
TCP-Vegas and TCP-Reno, and other variants working similarly, suffer from ineffective congestion control in a data collection infrastructure with a large number of low data rate sensors/meters. The problem occurs because all variants are based on the idea of reducing the congestion window size of a data source when the source senses congestion in the network, but the ability of traffic reduction is not viable in a smart metering and sensing infrastructure. However, interesting differences in performance exist. The experiments below contrast Reno and Vegas in a smart metering and sensing system environment and show the improvement that the splitting and aggregating scheme brings.

TCP-Vegas is surprisingly found to perform worse in a one-hop-TCP set-up. Figure 9 demonstrates the performance in terms of loss rate and delay. This One-hop-TCP experiment, depicted Fig. 3-A, is set up to vary the number of meters. A shared bottleneck is configured as 2 Mbps and a 100-packet buffer capacity. The propagation delay is 100 msec. While both variants result in the same delay, TCP-Reno



(a) Offered Traffic Load (b) Packet Loss Rate (c) End-to-end Delay

Figure 8: Validation of TCP-Vegas in the 2^{nd} hop

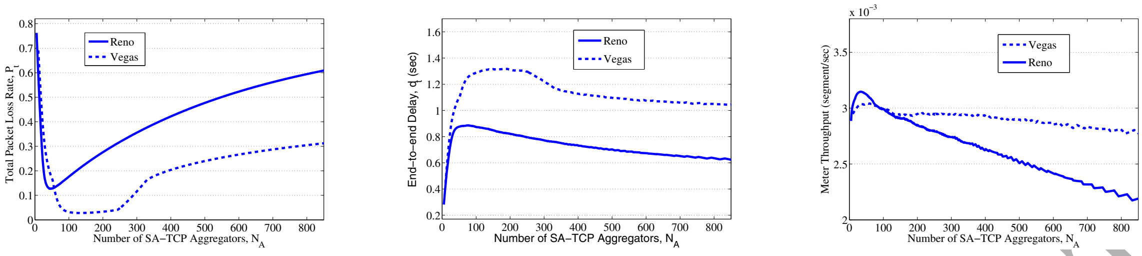


(a) Total Packet Loss Rate (b) End-to-end Delay

Figure 9: TCP-Vegas Vs. TCP-Reno Performance in One-hop-TCP

causes fewer losses. Although TCP-Vegas has been shown to outperform TCP-Reno in the Internet [19], certain properties of Vegas make it even less effective than Reno in a smart metering network, rationalized as follows. To achieve a faster decision in retransmitting a dropped packet, TCP-Vegas does not wait for three duplicate ACKs. Instead, it retransmits a possibly missing packet if its time duration exceeds RTT when a single or two duplicate ACKs arrive. This difference with Reno in the mechanism explains why Vegas is more aggressive (*i.e.*, quick to retransmit) and makes congestion control in a smart metering infrastructure even less effective, which leads to more packet losses.

In order to compare the performance of SA-TCP-Vegas with SA-TCP-Reno, an experiment with the same parameters as in the above one-hop-TCP experiment is conducted. In this experiment, the number of meters is fixed to 400,000, and the number of SA-TCP aggregators, N_A, is varied from 1 to 1000. It is assumed that the bandwidth available for the first TCP hop is the offered traffic rate of the regional collector to which the meters of that region connect, and the buffer capacity is 100 packets (a packet size is 240 bytes). Figure 10 shows the results of the experiment, which suggest that TCP-Vegas performs better than TCP-Reno. A closer look, however, reveals interesting behavior. SA-TCP under Vegas achieves a



(a) Total Packet Loss Rate (b) End-to-end Delay (c) Meter Throughput

Figure 10: TCP-Vegas Vs. TCP-Reno Performance in SA-TCP

lower loss rate, but the delay stays slightly higher than that with Reno as N_A is increased. It is clear that Vegas keeps the loss rate low for a longer range of N_A. However, it is noticed also that, with SA-TCP-Reno, there is a faster decrease in the packet loss rate (Fig. 10a) than with SA-TCP-Vegas. This rapid decrease explains why throughput goes higher than that of Vegas in the beginning (Fig. 10c). This behaviour is explained by the fact that TCP-Vegas performs better in a normal network setting, which exists on the second-hop-TCP side, but it is not as effective on the first-hop-TCP side. As N_A increases, the impact of meters on the congestion control effectiveness in a region is less because the number of meters is reduced in that region, while effective congestion control is achievable on the WAN side (the second hop). As N_A is increased, the effectiveness of congestion control becomes weaker on the WAN side as well; therefore, the loss rate increases again. TCP Vegas seems to handle congestion well in the second stage, but as the number of aggregators exceeds around 300, it fails, and the same scalability issue starts to show up also in the second stage. At low numbers of N_A, the total loss rate is dominated by the loss occurring in the first stage. For that reason, we notice that the loss rate increase in the range 100 to 300 is slow. After around $N_A = 300$, the loss rate starts to raise quickly as it starts to be dominated by the second stage's loss rate caused by the ineffectiveness of Vegas congestion control from that point upward.

One more observation is drawn from Fig. 10 by comparing the packet loss rate and packet delay when N_A is zero, which corresponds to the one-hop TCP scheme, and when N_A is higher than zero, which corresponds to the SA-TCP scheme. As claimed above, in either case, under TCP-Reno or TCP-Vegas, SA-TCP always outperforms one-hop TCP. For example, in Fig. 10a, the packet loss rate exceeds 0.7 in both cases, one-hop-TCP-Reno and one-hop-TCP-Vegas (i.e., $N_A = 0$). Nevertheless, the packet loss rate effectively decreases to values below 0.1 when deploying a number of regional collectors between 50 and 250. As expected, packet delay is higher in the SA-TCP scheme as a result of queuing delay in the regional collectors when the network becomes congested.

6. Conclusion

This paper analyzes the performance of two distinctive TCP variants, TCP-Reno and TCP-Vegas, for high density sensing/metering infrastructure. We develop an analytical model that successfully captures the behaviour of data collection traffic. The devised model is able to predict the load generated, packet loss rate, and packet delay with sufficient accuracy compared with results of extensive simulations. The analytical and simulation results show that TCP-Reno outperforms TCP-Vegas when collecting data over end-to-end TCP connections with a utility server. However, under the SA-TCP scheme, TCP-Vegas performs better than TCP-Reno because the low-rate smart grid data sources do not benefit from the full range of congestion control dynamics. However, SA-TCP-Vegas lends itself to an improved congestion control performance sensing/metering infrastructure of smart grids. For the future work, our research is ongoing to examine the gain of incorporating other TCP variants, including a hybrid of different ones in the SA-TCP scheme.

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

[1] S. R. Sarangi, P. Dutta, K. Jalan, IT infrastructure for providing energy-as-a-service to electric vehicles, IEEE Trans. Smart Grid 3 (2) (2012) 594–604.

[2] The Independent Electricity System Operator, Smart metering start-up guide, 2009.

[3] M. Faisal, A. Mohamed, A new technique for power quality based condition monitoring, in: 17th Conference of Electrical Power Supply Industry, 2008.

[4] Engage Consulting Ltd, High-level smart meter data traffic analysis, in: Document Ref.: ENA-CR008-001-1.4, 2010.

[5] Engage Consulting Ltd., Smart metering system use cases, 2010.

[6] D. J. Leeds, The smart grid in 2010: Market segments, applications and industry players, GTM Research.

[7] W. H. Kersting, Distribution System Modeling and Analysis, Third ed. Boca Raton, FL: CRC Press, 2012 (2012).

[8] A. Cavoukian, Operationalizing privacy by design: The ontario smart grid case study, in: Information and Privacy Commissioner, Ontario, Canada, 2011.

[9] N. Bui, M. Rossi, M. Zorzi, Networking technologies for smart grid, in: Ieee Vision For Smart Grid Communications: 2030 And Beyond, IEEE Smart Grid Research, IEEE Standards Association, 2013, Ch. 6, pp. 169–194.

[10] A. Liotta, D. Geelen, G. van Kempen, F. van Hoogstraten, A survey on networks for smart-metering systems, Int. J. Pervasive Computing and Communications 8 (1) (2012) 23–52.

[11] L. Ploeg, Small tcp/ip stacks for micro controllers, in: PhD thesis, Universiteit Twente, Enschede, 2006.

[12] Y.-C. Li, M.-L. Chiang, Lyranet: A zero-copy tcp/ip protocol stack for embedded operating systems., in: RTCSA, IEEE Computer Society, 2005, pp. 123–128.

[13] Cisco Systems, Why IP is the right foundation for the smart grid, in: Cisco Publication, White Paper, 2010.

[14] Companion Specification for Energy Metering, DLMS/COSEM architecture and protocols, in: Green Book, DLMS UA, 1997-2007.

[15] S. Floyd, K. Fall, Promoting the use of end-to-end congestion control in the internet, IEEE/ACM Trans. on Networking 7 (4) (1999) 458–472.

[16] Y.-J. Kim, M. Thottan, SGTP: Smart grid transport protocol for secure reliable delivery of periodic real time data, in: Bell Labs Tech. Journal, Vol. 16, 2011, pp. 83–99.

[17] K.-H. Mak, B. Holland, Migrating electrical power network SCADA systems to TCP/IP and ethernet networking, Power Engineering Journal 16 (6) (2002) 305–311.

[18] V. Jacobson, Modified tcp congestion avoidance algorithm, end2end-interest mailing list.

[19] L. Brakmo, L. Peterson, Tcp vegas: end to end congestion avoidance on a global internet, IEEE Journal Selelected Areas in Comm. 13 (8).

[20] T. Khalifa, A. Abdrabo, K. Naik, M. Alsabaan, A. Nayak, N. Goel, Design and analysis of split- and aggregated-transport control protocol (sa-tcp) for smart metering infrastructure, in: IEEE SmartGridComm, 2012.

[21] D. Niyato, P. Wang, Z. Han, E. Hossain, Impact of packet loss on power demand estimation and power supply cost in smart grid., in: WCNC, IEEE, 2011, pp. 2024–2029.

[22] D. Niyato, Q. Dong, P. Wang, E. Hossain, Optimizations of power consumption and supply in the smart grid: Analysis of the impact of data communication reliability., IEEE Trans. Smart Grid 4 (1) (2013) 21–35.

[23] D. Niyato, P. Wang, E. Hossain, Reliability analysis and redundancy design of smart grid wireless communications system for demand side management., Wireless Commun. 19 (3) (2012) 38–46.

[24] Y.-J. Kim, V. Kolesnikov, H. Kim, M. Thottan, SSTP: A scalable and secure transport protocol for smart grid data collection, in: IEEE Int. Conf. on Smart Grid Communications, 2011, pp. 161–166.

[25] N. Saputro, K. Akkaya, An Improved TCP for Reduced Packet Delay in IEEE 802.11s-Based Smart Grid AMI Networks, Springer International Publishing, Cham, 2014, pp. 86–97.

[26] M. Allalouf, G. Gershinsky, L. Lewin-Eytan, J. Naor, Data-quality-aware volume reduction in smart grid networks, in: IEEE Int. Conf. on Smart Grid Communications, 2011, pp. 120–125.

[27] Y. Ren, J. Li, S. Shi, L. Li, G. Wang, B. Zhang, Congestion control in named data networking a survey, Computer Communications 86 (2016) 1 – 11. doi:http://dx.doi.org/10.1016/j.comcom.2016.04.017. URL `http://www.sciencedirect.com/science/article/pii/S0140366416301566`

[28] N. Cherukuri, K. Nahrstedt, Cooperative congestion control in power grid communication networks, in: Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on, 2011, pp. 587–592.

[29] P. Myrda, K. Koellner, Naspinet - the internet for synchrophasors, in: System Sciences (HICSS), 2010 43rd Hawaii International Conference on, 2010, pp. 1–6.

[30] L. Li, An improved congestion control strategy algorithm for heterogeneous network, JNW 8 (9) (2013) 2107–2113.

[31] T. Khalifa, A. Abdrabou, K. Naik, M. Alsabaan, A. Nayak, N. Goel, Split-and aggregated-transport control protocol (sa-tcp) for smart metering infrastructure, in: IEEE Transactions on Smart Grid, Vol. 5, 2014, pp. 381–391.

[32] V. C. Gungor, F. C. Lambert, A survey on communication networks for electric system automation., Computer Networks 50 (7) (2006) 877–897.

[33] D. Relich, Smart meters on a roll in canada, in: Hydro One Networks Inc., 2008.

[34] M. Dohler, T. Watteyne, T. Winter, D. Barthel, Routing requirements for urban low-power and lossy networks, in: IETF RFC 5548, 2009.

[35] D. Niyato, L. Xiao, P. Wang, Machine-to-machine communications for home energy management system in smart grid, Communications Magazine, IEEE 49 (4) (2011) 53–59.

[36] M. Kezunovic, V. Vittal, S. Meliopoulos, T. Mount, The big picture: Smart research for large-scale integrated smart grid solutions, IEEE Power and Energy Magazine 10 (4) (2012) 22–34.

[37] K. Fan, S. Liu, P. Sinha, Data aggregation in wireless sensor networks, in: Wireless Sensor Networks and Applications, Signals and Communication Technology, Springer, 2008, pp. 331–347.

[38] O. Akan, I. Akyildiz, Event-to-sink reliable transport in wireless sensor networks, IEEE/ACM Trans. on Networking 13 (5) (2005) 1003–1016.

[39] B. Nguyen, A. Banerjee, V. Gopalakrishnan, S. Kasera, S. Lee, A. Shaikh, J. Van der Merwe, Towards understanding tcp performance on lte/epc mobile networks, in: Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications and Challenges, ACM, 2014, pp. 41–46.

[40] J. Hwang, S. W. Son, B. H. Rhee, Improving tcp performance over wimax networks using cross-layer design, in: Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on, Vol. 2, 2008, pp. 83–88.

[41] S. A. Nor, A. N. Maulana, Performance of TCP variants over LTE network, AIP Conference Proceedings 1761 (1) (2016) –.

[42] K. Tsiknas, G. Stamatelos, Comparative performance evaluation of tcp variants in WiMAX (and WLANs) network configurations, Journal of Computer Networks and Communications 2012 (2012) –.

[43] Z. Wang, J. Crowcroft, Eliminating periodic packet losses in the 4.3 Tahoe BSD TCP congestion control, ACM Computer Communication 22 (2) (1992) 9–16.

[44] R. Jain, A delay based approach for congestion avoidance in interconnected heterogeneous computer networks, CoRR (1998) –.

[45] R. K. A. Venkataramani, M. Dahlin, Tcp nice: A mechanism for background transfers, Operating Systems Review 36 (2002) 329–344.

[46] V. Jacobson, Congestion avoidance and control, ACM SIGCOMM (1988) 314–329.

[47] S. Floyd, Issues of tcp with sack, Tech. rep. (Jan 1996).

[48] S. Floyd, T. Henderson, A. Gurtov, Rfc 3782 the newreno modification to tcps fast recovery algorithm, in: RFC, 2004.

[49] A. Misra, T. Ott, J. Baras, The window distribution of multiple tcps with random loss queues, in: Global Telecommunications Conference, 1999. GLOBECOM '99, Vol. 3, 1999, pp. 1714–1726.

[50] J. Kingman, Poisson Processes, John Wiley & Sons, Ltd, 2005.

[51] A. Wierman, T. Osogami, A unified framework for modeling tcp-vegas, tcp-sack, and tcp-reno, in: MASCOTS, 2003, pp. 269–278.

[52] L. Kleinrock, Queuing Systems, Vol.1: Theory, Wiley & Sons, New York, NY, USA, 1975 (1975).