Design and Analysis of a Novel Split and Aggregated Transmission Control Protocol for Smart Metering Infrastructure

by

Tarek Khalifa

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Utility companies (electricity, gas, and water suppliers), governments, and researchers recognize an urgent need to deploy communication-based systems to automate data collection from smart meters and sensors, known as Smart Metering Infrastructure (SMI) or Automatic Meter Reading (AMR). A smart metering system is envisaged to bring tremendous benefits to customers, utilities, and governments. The advantages include reducing peak demand for energy, supporting the time-of-use concept for billing, enabling customers to make informed decisions, and performing effective load management, to name a few.

A key element in an SMI is communications between meters and utility servers. However, the mass deployment of metering devices in the grid calls for studying the scalability of communication protocols. SMI is characterized by the deployment of a large number of small Internet Protocol (IP) devices sending small packets at a low rate to a central server. Although the individual devices generate data at a low rate, the collective traffic produced is significant and is disruptive to network communication functionality. This research work focuses on the scalability of the transport layer functionalities. The TCP congestion control mechanism, in particular, would be ineffective for the traffic of smart meters because a large volume of data comes from a large number of individual sources. This situation makes the TCP congestion control mechanism unable to lower the transmission rate even when congestion occurs. The consequences are a high loss rate for metered data and degraded throughput for competing traffic in the smart metering network.

To enhance the performance of TCP in a smart metering infrastructure (SMI), we introduce a novel TCP-based scheme, called Split- and Aggregated-TCP (SA-TCP). This scheme is based on the idea of upgrading intermediate devices in SMI (known in the industry as regional collectors) to offer the service of aggregating the TCP connections. An SA-TCP aggregator collects data packets from the smart meters of its region over separate TCP connections; then it reliably forwards the data over another TCP connection to the utility server. The proposed split and aggregated scheme provides a better response to traffic conditions and, most importantly, makes the TCP congestion control and flow control mechanisms effective. Supported by extensive ns-2 simulations, we show the effectiveness of the SA-TCP approach to mitigating the problems in terms of the throughput and packet loss rate performance metrics.

A full mathematical model of SA-TCP is provided. The model is highly accurate and flexible in predicting the behaviour of the two stages, separately and combined, of the SA-TCP scheme in terms of throughput, packet loss rate and end-to-end delay. Considering the two stages of the scheme, the modelling approach uses Markovian models to represent

smart meters in the first stage and SA-TCP aggregators in the second. Then, the approach studies the interaction of smart meters and SA-TCP aggregators with the network by means of standard queuing models. The ns-2 simulations validate the math model results.

A comprehensive performance analysis of the SA-TCP scheme is performed. It studies the impact of varying various parameters on the scheme, including the impact of network link capacity, buffering capacity of those RCs that act as SA-TCP aggregators, propagation delay between the meters and the utility server, and finally, the number of SA-TCP aggregators. The performance results show that adjusting those parameters makes it possible to further enhance congestion control in SMI. Therefore, this thesis also formulates an optimization model to achieve better TCP performance and ensures satisfactory performance results, such as a minimal loss rate and acceptable end-to-end delay. The optimization model also considers minimizing the SA-TCP scheme deployment cost by balancing the number of SA-TCP aggregators and the link bandwidth, while still satisfying performance requirements.

Acknowledgements

My thanks and praise first and foremost go to Almighty God, for giving me the knowledge, opportunity, and the strength to accomplish this work.

Along the way, several people deserve sincere recognition. I am most grateful to my advisor, Dr. Sagar Naik, for his incomparable guidance and patience over the past years. He has been a great mentor and friend. Over the years, he has always been available to guide my research in the right direction. His valuable advice, his generous help and time, and his constant support were the secret of achieving all this work.

I would like also to extend my appreciation and thanks to my thesis committee members, Dr. Ali Elkamel, Dr. Liang-Liang Xie, and Dr. Mahesh Tripunitara for sparing their time to review this research work. Thanks as well to Mary McPherson of the grad writing center. I would like also to thank Dr. Atef Abdrabou. I am grateful to all of them for their insightful feedback and comments on this work.

I would like to acknowledge the Ministry of Higher Education of Libya for financially supporting this research work. I would like to also acknowledge the financial support that Prof. Sagar Naik has provided for me. Thanks to these generous funds, this thesis was made possible.

Thanks go to my friends for their prayers and support. I am very fortunate to have so many exceptional and genuine people in my life. I thank my dear friends Omar Elfasi, Salah Albiri, Magdi Alnaas, and Jouma Mahfoud for their constant support and encouragement. I thank my friends and colleagues, Maazen AlSabaan, Abdulhakim Abogharaf, Abdurhman Albasir, Majid Altamimi, Mursalin Akon, and Rajish Palit, from the Network Programming Lab, for all our fruitful discussions and for their friendship.

I would like to thank my family for their continuous and unconditional support for all my endeavors. My father, Prof. Masaud Khalifa and my mother, Jumia Ali, and my brothers and sisters, Dr. Nagib, Dr. Manal, Dr. Najwa, Marwa, Suhaib, Faisal, and Gassan, have been the greatest supporters in my life. I am grateful to them for their unconditional love, support, encouragement and the sacrifices they made to raise me.

Cordial thanks go to my beloved wife, Aziza, for the endless understanding, support, patience and care she showed during this hectic period in my career. Despite being extremely busy with her studies, she has always been available to uplift me when I felt down and to congratulate me when I achieved. I am by all means indebted to her.

Finally, I thank all my teachers, colleagues and friends who helped and supported me in different ways throughout my work. God bless you all.

This thesis is dedicated to mum and dad

I love you both

Table of Contents

List of Tables

List of Figures

List of Abbreviations

SMI	Smart Metering Infrastructure
PLC	Power Line Carrier
ToU	Time of Use
RC	Regional Collector
RF	Radio Frequency
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ACK	Acknowledgement
SS	Slow Start
CA	Congestion Avoidance
MSS	Maximum Segment Size
RTT	Round Trip Time
MAC	Medium Access Control

Chapter 1

Introduction

1.1 Overview of Smart Grid

Smart power grids are known to modernize electricity infrastructure by integrating communication and information technologies into every aspect of electricity generation, delivery and consumption. Additions of hardware and software components to the power system make new features possible, for example, (a) metering and monitoring of the power system; (b) communicating the conditions of the grid in real time; and (c) controlling the flow of power to maintain reliable service and stable operation.

Smart Metering Infrastructure (SMI), also called Automatic Meter Reading or AMR, is an essential part of a smart grid. Currently, it is perceived as a system that enables control and automatic data collection and analysis through its bi-directional communication between a utility collection center and smart meters [28] [47] [126]. The SMI system is characterized by the deployment of a large number of smart meters. The meters typically produce data at low rates (*e.g.,* one packet of 200 bytes every minute) either periodically or in response to triggered events [41]. Various terminologies are used in the literature to refer to SMI, including 'smart metering system', 'advanced metering infrastructure', and 'advanced metering system'.

Automation of meter reading and data collection ranges from communicating with meters through an RS-232 interface, via Infrared, or short range radio frequency to transmitting the meter measurements all the way from the meter to the utility company. Electricity providers seem to be at the forefront of this field today, but in fact, all utility providers (*e.g.,* water and gas) are interested in collecting data at high frequencies and ultimately

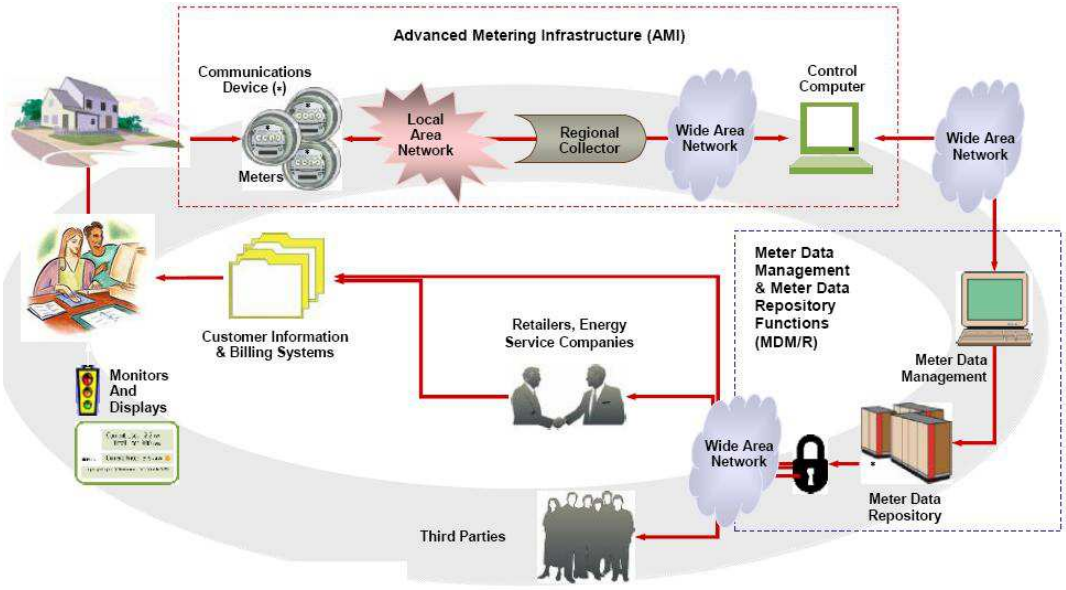


Figure 1.1: Smart Metering System (Source: Ontario Ministry of Energy)

in enhancing the quality of utility provision and service. The SMI network will eventually have to serve all providers' meters together.

Figure 1.1 demonstrates the full cycle of metering. First, measurements get transmitted via a communication network from the smart meters and sensors to the utility provider for processing and management. Sending control signals to the smart meters is also possible as the network is bidirectional. The utility also provides its customers with detailed information about their consumption.

Figure 1.2 illustrates the communication architecture of an SMI. Smart meters form regions of local area networks. Various communication technologies, such as WiFi and PLC, are employed for local communication among meters. For isolated meters, signal repeaters are used to connect them with the rest of the network. Meters themselves can route packets, facilitating a multi-hop communication paradigm. Certain nodes, called advanced metering regional collectors (or RC), are installed at poles at preselected locations in every region. Those RCs act as gateways between meters and the wide area network (WAN), which could be the Internet or the utility's private network. Utility servers on the other side of the network receive the grid's data over WAN, and also send control and information messages in the reverse direction.

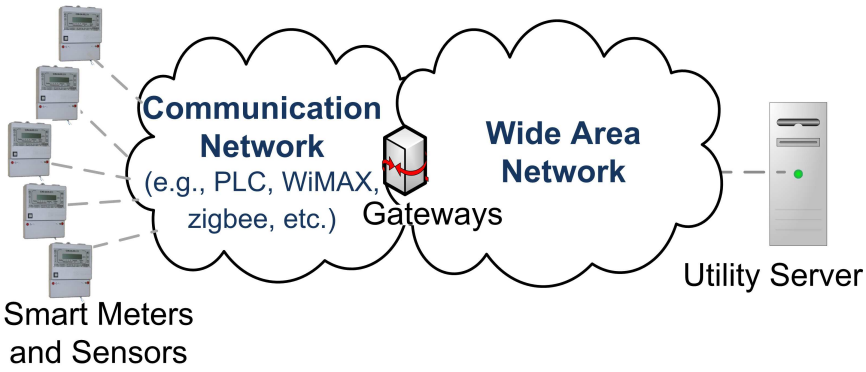


Figure 1.2: Smart Metering Network Architecture

1.1.1 Applications and Services

Supporting the power grid with communication infrastructure has opened up numerous avenues for sophisticated applications and services. The benefits extend not only to utility providers, but to the environment and society also. The application of SMI will greatly simplify and speed up the process of collecting important sensory information and meter readings, which otherwise will require personnel working on site. As more information become available, better quality of service will be made possible. Cost reduction is another motive behind the push for smart meter deployment, as are the following important services:

- Real-time Pricing: Customers are charged tariffs that vary over a short period of time, hourly for example. Smart metering helps customers control their consumption and helps utility providers to better plan for the energy market. Barbose et al. [18] provide an in-depth study of real-time pricing.

- Power quality measurement: Electric utility engineers need more detailed readings than Kwhr so that they can efficiently plan network expansion and deliver a higher quality of supply [28]. Power quality involves the measurement of voltage sags, swells, under and over voltages, harmonics distortion, voltage and current imbalances, and each event's duration [58] [43].

- Automated Billing: Once the metering data is available at the utility company premises, billing, acknowledgement of received payments, and power consumption

reports can be fully automated and made available to customers, on the web for example.

- Load management: This is another industrial area that is feasible with having a smart metering system in place. The service allows sending control signals to appliances, such as air conditioners and heaters. Surrat [136] discusses the importance of load management to electricity providers as well as to customers in terms of power saving.

- Remote Connect/Disconnect: A utility provider can remotely and quickly configure meters to enable or disable energy to certain customers.

- Outage notification: This offers an effective way to improve response times. Liu et al. [95] propose an algorithm that involves two steps: outage locating and outage confirmation through meter polling.

- Bundling with water and gas: The ultimate objective behind a fully functional SMI is to serve all kinds of meters – electricity, water and gas– under one communication technology and one protocol standard.

- Lastly, SMI offers benefits beyond those points mentioned above. Generally speaking, having a two-way communication facility in place definitely enables many sophisticated services. More can be found in [16] and [140].

1.1.2 Evolution of Meters

Metering devices have gone through much improvement over the past years, and are expected to become even more sophisticated, offering more and more services. Meters in the past, and still today in a few countries, were originally electromechanical devices with poor accuracy and lack of configurability. Theft detection was also a challenge. Such meters are limited to providing the amount of energy consumption on site. Today, meters are digital devices enjoying a higher accuracy, added control and configuration functionality, and better theft detection ability. For data collection, the meter can be read through a serial port (e.g., RS232) or wirelessly (Infra Red (IR) or Radio Frequency (RF)). Next generation meters (called smart meters) should make full use of SMI, and numerous sophisticated services would be available through modern communication's facilities available on chip. Data collection, theft reporting, and control can be remotely achieved from the utility company (Table 3.1).

Table 1.1: Development of Meters

	Electro-mechanical	Current Digital Meter	Smart Meter
Accuracy	Poor	High	Very high
Control and Configuration	N/A	Limited	Full
Theft Detection	Poor	At node only	High (at utility premises)
Remote Communication	N/A	Adding communication mudules	Built-in

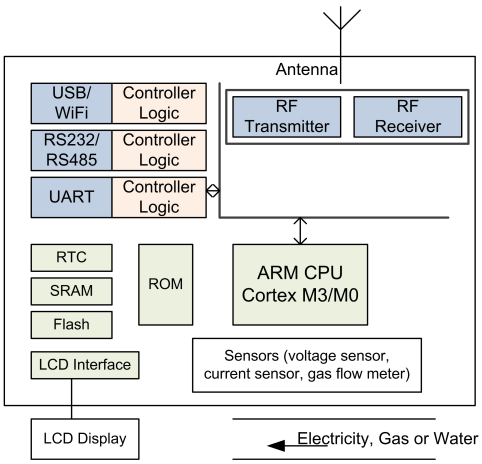


Figure 1.3: Smart Meter H/W Architecture

1.1.3 Specifications

Smart meters enjoy fair hardware/software capabilities that enable them to run TCP/IP suite and have the ability to run applications on top of TCP or UDP. Smart meters are equipped with processing capability ranging from SoC (system on a chip), microcontrollers to 32-bit processors (e.g., Cortex CPU M series and Cirrus Logic's CS7401xx series) (Fig. 1.3). The operating system, supporting an extensive library of routines and applications, has a task scheduler that rotates between a number of tasks, such as communication, measurement and database management [99].

1.2 Smart Metering Infrastructure Design Challenges

This section serves as a general overview, briefly giving the big picture of the design issues concerning smart grid applications. These applications differ from other communication applications in that the grid involves a significantly large number of devices. These devices transmit data at a low rate. Because all the data are destined for the same server, the impact of the resulting traffic is no longer trivial. Rather, the traffic level is high enough to cause disruption in a shared network. In this system, data packets must be delivered reliably and within time bounds to their final destination. Considering these factors (a large number of devices sharing a network, low individual data rates and end-to-end reliability and time constraints), it is important to research the applicability of the existing communication technologies and protocols in supporting smart metering applications, and to design new strategies where existing ones fail. Below is a briefed discussion of such design challenges. Further details are found in Section 2.4.

1.2.1 Network Design for Data Collection

This section describes ways of determining the network entities and protocols needed to collect data from a scattered, large number of energy devices.

- Communication network architecture: Various communication technologies have been proposed. Examples include Power Line Carrier (PLC), Global System for Mobile Communication (GSM), and Radio Frequency (RF). The choice of technology determines the network components involved and its topology (*e.g.,* regional collectors, relays and gateways). However, each technology has certain shortcomings that make it not the appropriate solution in a certain environment. A hybrid solution thus is to be considered to ensure that different demographies are covered and that the solution is still interoperable, scalable and reliable.

- Network Access and Routing: Low-lower communication issues will be faced. For example, with wireless technology, the connectivity of some devices (*e.g.,* in underground floors) can be a challenge due to obstacles blocking wireless signals. Medium Access Control (MAC) is another point to study. Given the highly dense network of energy devices, consideration must be given to what the best MAC protocol is to employ while being energy-efficient. At the IP level, a design is needed for an energy-efficient routing protocol for unicast, multicast, and broadcast data.

- Collection Mechanism: Some data can be scheduled to be reported at fixed periodic times. Other data are event-driven or demand-driven, responding to command signals.

1.2.2 Quality of Service and Network Management

This section briefly discusses handling data with different characteristics and requirements.

- Handling Failure: Reliable delivery of data must be ensured among different entities of the smart grid. If a device breaks down or data get lost, for example, the fault must be detected immediately and automatically recovered from.
- Timeliness of data: The system is required to handle data with different urgency requirements. Some data are urgent and need realtime delivery. Other data are tolerant of delay.
- Security: End-to-end security is required. The utility as well as customers are concerned with data security, such as privacy and integrity. Because the devices are resource-constrained, and the number of meters is significantly large, planning for key distribution and management and selecting the right cryptographic system (*i.e.,* symmetric or asymmetric) are important design matters.

1.2.3 Saving Energy

An important goal of the smart grid is to save energy. This involves techniques at the application level to get customers to adjust their use of electricity, for example, by providing Time of Use (ToU) pricing information. It also involves reducing peak power demand through load management, which requires communication capabilities and smart techniques beyond the smart meter into customer premises. Moreover, all designed communication protocols must be energy-aware and keep energy consumption minimal because any extra consumption will tally up to a large value given the number of energy devices.

1.3 Research Motivation and Objectives

1.3.1 Ineffectiveness of TCP Congestion Control

The smart metering system necessitates reliable delivery of data packets from every source to the final destination (*i.e.,* the utility server). To achieve this end-to-end reliability, Transmission Control Protocol (TCP) must be employed. TCP provides fully reliable, in-order, end-to-end data delivery services by using connection management, congestion flow control, and loss recovery mechanisms (Section 3.1)

TCP has been through many improvements so as to fix its poor performance in certain situations in which an application exhibits extreme behavior with respect to the TCP's initial design assumptions, which were based on host-to-host communication. Examples include the silly window syndrome problem [108] and running TCP in a high speed network or wireless medium.

SMI is shown here as another case of an extreme behavior that TCP reacts to ineffectively due to its congestion control mechanism. TCP protocols achieve congestion and flow control by adjusting a source's congestion window size. A traffic source keeps increasing its transmission speed by enlarging the window size, but if a packet goes unacknowledged, indicating congestion in the network, the source lowers its speed. The general reduction mechanism of TCP [14] occurs in the following manner:

- If the unacknowledged packet times out, the source decreases the transmission rate to the minimum – one segment per round trip time – by setting the congestion window to its initial size (typically one or two segments).
- If the source receives three duplicate ACKs, indicating a missing packet, it halves its sending rate by halving the congestion window size.

With SMI traffic, the typical congestion control mechanism of TCP becomes ineffective for the following reason. The high volume of traffic in SMI does not come from a single source; rather, it comes from a large set of sources, each transmitting at a low data rate [41] [97] [96]. The TCP congestion window always stays at its minimum value of one or two, so reducing the sending rate upon congestion in the network is not viable.

The problem is equivalent to replacing one TCP connection with a large number of TCP sub-connections [51] to deliver the same amount of data, assuming each sub-connection transmits at a low data rate that requires the congestion window size to stay at 1 Maximum

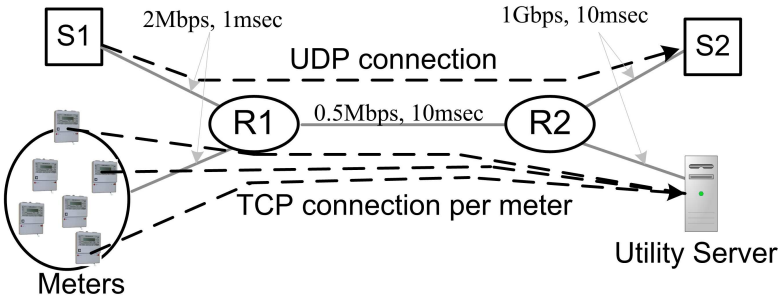


Figure 1.4: Experiment on TCP in SMI

Segment Size (MSS). If congestion occurs in the network, in the case of one TCP connection, the source 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 for all the sub-connections will stay as high as the summation of the individual sub-connection congestion window sizes.

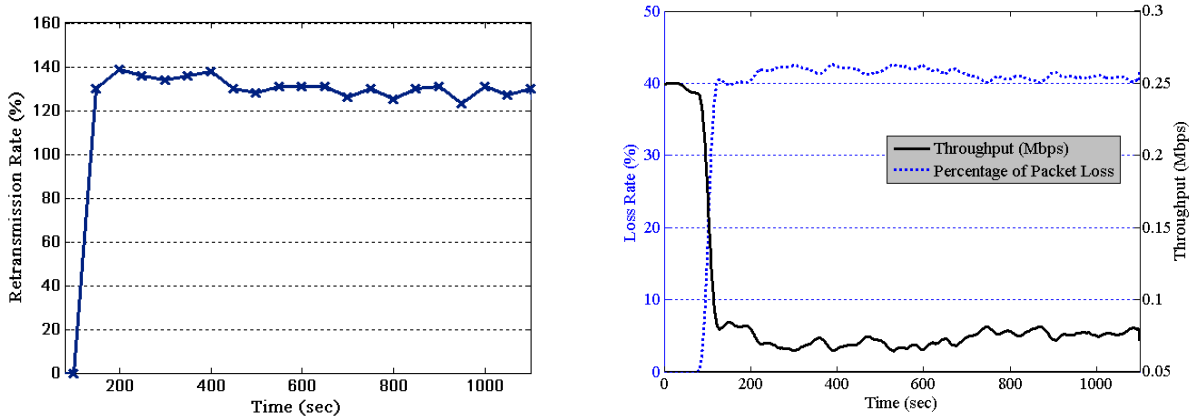
Therefore, the SMI traffic will be highly aggressive as there will be hundreds of thousands of TCP connections transmitting at a low flat rate of around 1 segment/RTT (round trip time). The lack of an effective congestion control mechanism leads to two major problems, namely, congestion collapse and unfairness [51]. Congestion collapse occurs when the network is busy transmitting packets that will be dropped by some congested router, before reaching the final destination. That is, even though SMI traffic may suffer packet drops, the total traffic rate stays unchanged. Consequently, the packet loss rate is high. Unfairness occurs when other competing TCP-friendly flows suffer bandwidth starvation because of the non-rate-reducible SMI traffic, resulting in lower throughput for the competing traffic than for the meters' throughput [75]. Although the meters' throughput seems high, the high loss rate makes the network badly utilized.

To show how performance is degraded in an SMI, an experiment using the ns-2 simulator is performed with each meter maintaining a separate TCP connection with the utility server. Each one of ten thousand meters transmits a small report of 200 bytes every 50 seconds. The experiment setup and parameters are given in Fig. 1.4 and Table 1.2, respectively. The R1-R2 link is shared between the meters' traffic and the external UDP traffic from S_1.

The results show that having independent direct TCP connections between the meters

Table 1.2: TCP Experiment Parameters

Number of meters	10,000
Meter's data rate	1 packet/50 sec.
Packet size	200 bytes
(S1 → S2) UDP traffic	random, up to 250 kbps
Bottleneck buffer	20 packets, DropTail



(a) Meter's Retransmission Rate

(b) Throughput and Loss Rate of UDP Traffic

Figure 1.5: Impact of TCP Congestion Control in Smart Metering Infrastructure

and the utility server would guarantee data reliability but at the expense of congestion control and at the expense of fairness with other applications' flows (*i.e.,* UDP traffic in this case). That is, as overflow occurs in the network, individual meters keep retransmitting lost packets without reducing the transmission rate. The UDP traffic source also continues to transmit at the same speed. The lack of an effective congestion control mechanism leads to more and more packets being dropped at the bottleneck. The result is an excessive retransmission rate from the meter side and extreme loss of UDP data. Figure 1.5a shows the meters retransmission rate as a percentage. A retransmission of 100% means that each meter transmits each packet twice. Figure 1.5b shows the impact on UDP traffic in terms of degraded throughput and percentage of lost packets.

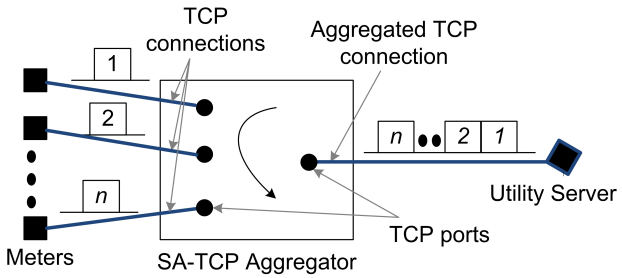


Figure 1.6: Split and Aggregation Mechanism

1.3.2 Objectives

In conclusion from the above argument and experiment results, TCP is found to be ineffective in handling the smart metering traffic. *Accordingly, the objective of this thesis is studying and enhancing the performance of TCP in an SMI.* Besides evaluating whether the existing TCP protocols fit in an SMI, this thesis proposes a novel TCP-based scheme, called *Split- and Aggregated-TCP (SA-TCP)* as the solution. It mathematically models the scheme, analyzes it, and optimizes its design. The scheme enables TCP congestion control to function effectively; consequently, performance improves greatly.

1.4 Summary of Contributions

This thesis makes the following contributions:

- It provides an in-depth study of the smart metering infrastructure from the communication perspective. It extensively studies all the communication technologies and standards that have been proposed as the SMI backhaul network. Furthermore, it highlights the experiences learned from wireless sensor networks (WSN) for their similarity to SMI. Although the two are very similar, this work investigates how the WSN protocols may be of benefit and whether they fit.

- It evaluates the capability of TCP protocols in efficiently handling SMI traffic. In this regard, it analytically and experimentally shows the impact of the large number of meters on TCP performance. Namely, it shows how the TCP congestion control mechanism becomes ineffective in achieving its desired goals. Consequently, meters

suffer high packet loss rate and degraded throughput. Other traffic flows that share the network with meters are also affected, resulting in low throughput.

- It develops a novel TCP-based scheme called *Split- and Aggregated-TCP* (SA-TCP) to enhance the TCP congestion control performance. In this scheme, instead of having the smart meters and sensors communicate over separate TCP sessions with the utility server, we introduce the idea of consolidating those individual TCP connections at intermediate devices we call *SA-TCP aggregators*. An SA-TCP aggregator (depicted in Fig. 1.6) establishes TCP connections with the smart meters on one side, over which the meters' data is received, and establishes another TCP connection with the utility server on the other side, over which the data is forwarded. Existing devices known as regional collectors are exploited for this added functionality. The proposed scheme provides a better response to traffic conditions and, most importantly, makes the TCP congestion control and flow control mechanisms effective, reducing packet loss rates for for meters and enhancing the throughput for competing traffic flows in the network.

- It formulates a mathematical model for capturing the performance achieved by SA-TCP as the meter application and network characteristics change. The mathematical model splits the SMI network into two stages and analyzes each by means of Markov and queueing models. Specifically, in this modelling approach, the meters in the first stage and the SA-TCP aggregators in the second stage are represented by Markov chains. Then, the approach, by means of standard queuing analysis, studies the interaction of smart meters and aggregators with the network. It then finds the overall SMI network performance. Consequently, given the number of meters, the number of SA-TCP aggregators, and the network properties, the model is able to predict the average load offered by a meter, packet loss rate and end-to-end delay. The detailed modelling is shown for two different variants of TCP: Reno and Vegas. The former adjusts the congestion window size according to the packet loss rate, while the latter does so according to the packet delays. For the validation of the model, extensive ns-2 simulations have been conducted under different settings. The comparisons between the analytical results and the simulations show that the model succeeds in accurately representing the metering traffic behaviour.

- It provides extensive performance analysis of the SA-TCP scheme in comparison with a typical one-hop TCP protocol. The impacts of various design and network parameters are considered. Specifically, it shows the impact of SMI link capacity, propagation delay, number of SA-TCP aggregators and their buffering capacities. The analysis is important for understanding how the various parameters change the

TCP performance results so that a better scheme design is achieved. The math model has made it feasible to analyze for a large range of input parameters for both TCP Reno and TCP Vegas variants.

- Finally, it formulates an optimization problem based on the SMI mathematical model. Different objective functions are presented in this part. The goal is to tune the design of the SA-TCP scheme so that satisfactory performance results are guaranteed and deployment cost is minimized. From the performance analysis, it is understood that certain performance metrics such as loss rate and delay conflict as the number of SA-TCP aggregators changes; therefore, the optimization model searches for the optimal point that balances the two metrics. The model 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.

1.5 Proposal Organization

This thesis is organized as follows:

- Chapter 2 provides the background material for this research. SMI is a fairly new area. Therefore, this chapter covers important topics related to SMI, including the conceptual system architecture, the requirements and communication standards, and a general discussion of the major challenges.
- Chapter 3 provides a review of the related work to SMI at different layers, with more emphasis on transmission control protocols.
- Chapter 4 presents the smart metering system model and assumptions. Supported by simulation results and analytical evaluation, this chapter provides a detailed explanation of how the TCP congestion control mechanism would be ineffective in a smart metering infrastructure.
- Chapter 5 describes the architecture and mechanism of the proposed SA-TCP scheme. By means of simulations, a comparison with a one-hop TCP scheme is provided. This chapter concludes with a discussion of the advantages and disadvantages of SA-TCP.
- Chapter 6 develops an analytical formulation for the SA-TCP scheme. Validation with simulation results are provided as well.

- Chapter 7 extends the analytical model of SA-TCP to capture the scheme's behaviour under the Vegas version of TCP. Similar to the previous chapter, simulations are shown to validate the model.
- Chapter 8 provides a comprehensive performance analysis for SMI traffic under SA-TCP and one-hop TCP congestion control set-ups. Besides investigating how various design parameters impact SA-TCP's throughput, packet loss rate and delay, an optimization model to ensure satisfactory performance is formulated in this chapter.
- Chapter 9 summarizes the thesis work and provides interesting and challenging directions for future research.

Chapter 2

Background

Smart metering infrastructure is a fairly new area and is still in a developing phase. This chapter gives the big picture of SMI. First, it introduces the system architecture as presented by utility and communication companies. Next, it describes the requirements of the system and the standards developed specifically for electrical devices with communication capability. Lastly, it provides a general discussion of the major challenges that face an SMI.

2.1 Smart Metering Conceptual System

Figure 2.1) architects an SMI system. This design is taken from Hydro One [115], which is a leading electricity transmitter and distributor in Ontario, Canada, serving a geographic area of 640,000 square kilometers and targeting to deploy 1.3 million smart meters. As shown in the figure, the metering Infrastructure incorporates various entities to achieve two-way end-to-end connectivity. It is composed of a large number of meters connected by means of a wireless communication network. The meters form mesh networks to connect among themselves over multiple hops with gateway nodes, known as advanced metering regional collectors. Meters in less populated areas join the rest of the SMI network through repeaters (*i.e.,* wireless signal extenders). The regional collectors are typically installed on poles at preselected locations within a local area network. They gather the meters' data packets in a defined region and route the packets through the Wide Area Network (WAN) to utility provider [115] [39] [109]. From the other side, the advanced metering control computer (we call it utility server) is also connected to the WAN for the purpose of collecting the meters' data [90].

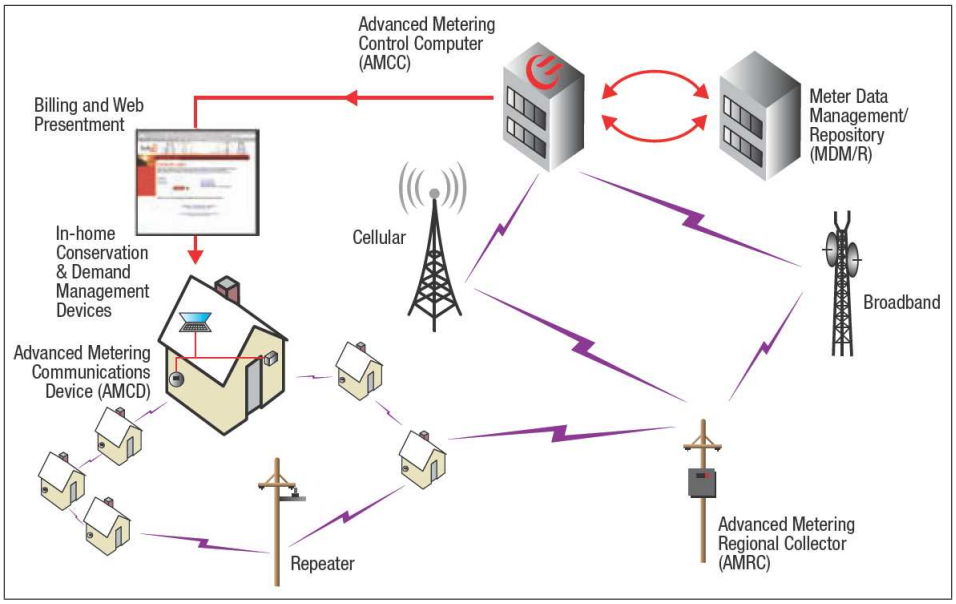


Figure 2.1: Smart Metering Infrastructure. Source: Hydro One, Ontario, Canada

Utilities consider the option of having their own private Wide Area Network (WAN). They justify the costs of building and maintaining their own private WAN (rather than relying on public networks) by the highly critical nature of their applications and their need for a reliable and secure grid. A WAN may include a hybrid mix of technologies, including fiber optics, Power Line Carrier (PLC) [151], a variety of wireless technologies, third generation (3G) networks such as WiMAX, LTE, and HSPA [73], and possibly interconnected with the Internet in some parts. Section 3.2 provides a detailed review of the proposed communication technologies. The WAN network, however, will still have to serve a variety of applications, for example, the traditional SCADA/EMD system, Distribution Automation (DA)/Demand Side Management (DSM), and others (Fig. 2.2 [35]). Such applications are numerous and often have different requirements.

Smart meters and sensors are expected to ultimately number in the hundreds of thousands to millions, whereas the number of regional collectors is expected to remain as small as in the tens or hundreds. The smart metering network is required to operate in both directions – between meters and the utility server. It is expected, however, that the higher volume of data traffic will flow from the meters to the utility server. Occasionally, servers will send control messages and data to the meters.

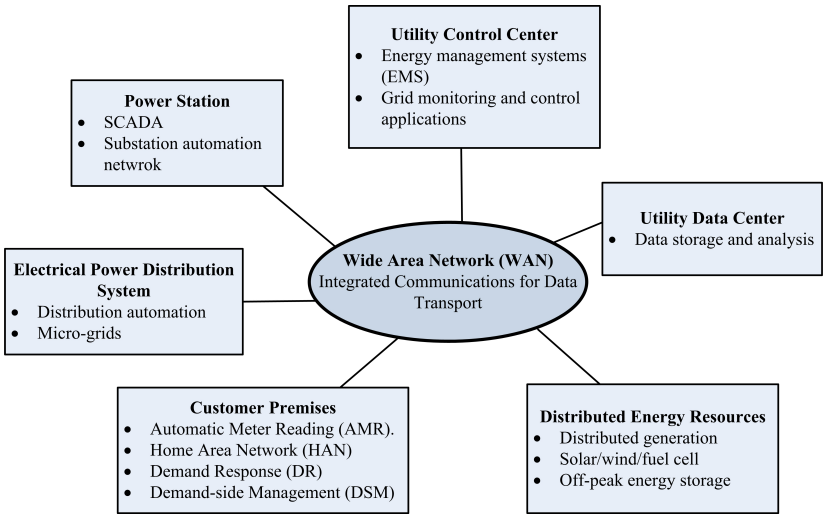


Figure 2.2: Utility Applications

These metering devices fully support the TCP/IP communication stack as defined by the Device Language Message Specification/COmpanion Specification for Energy Metering (DLMS/COSEM) standard [33]. For reliability, the TCP protocol is used. Every meter establishes a direct TCP connection with the utility server. Section 3.1 provides a detailed review of the TCP protocols and how they would fit in an SMI.

2.2 Technical Requirements and Performance Metrics

An SMI network should meet certain quality requirements. Any new design has to be assessed according to a number of quality metrics as follows:

- Reliability: The SMI network must guarantee the arrival of all meter readings as well as all utility server control packets. The success rate or the loss rate performance metric shall give us a fair assessment of the given network.
- Scalability: A designed network should be assessed according to its ability to support a large number of meters covering a large geographical area. Furthermore, the fre-

quency of such readings should be high enough to support the desired SMI services (*e.g.,* real time pricing.)

- Latency: Data reported from a given meter must arrive within a given amount of time. Certain traffic types (e.g. fault detection) mandate a short time delay. A performance metric of end-to-end delay is required to provide a good evaluation of the different traffic types' response times.
- Order: Packets representing different readings should be stamped with the time of measurement so that packet ordering at the receiving station can be guaranteed.
- Security: The level of security can be expressed in terms of the cryptographic tools implemented at different protocol stack layers and the number of key bits used. Hop-by-hop security is implemented at lower layers while end-to-end security is implemented at both ends of the SMI application.

2.3 Smart Meter Communication Protocol Standard

As important as designing a scalable and reliable communication network is ensuring that it conforms the international standards. The International Electro-technical Commission (IEC) is the organization that prepares international standards for all electrical and electronic technologies [3]. Cooperating with IEC, the DLMS User Association [2] takes metering devices (electricity, water, heat, and gas) to be its main focus. The objective is to ensure inter-operability among energy distribution devices so that they can exchange information/control messages under various physical media and communication protocols.

2.3.1 DLMS/COSEM Standard

The metering standard that supports electricity, gas, heater and water equipment is known as the Device Language Message Specification/COmpanion Specification for Energy Metering (DLMS/COSEM) [52]. DLMS is an application layer specification. COSEM presents an object oriented model for meters, providing a view of their functionality through communication interfaces. In COSEM, the physical metering equipment is viewed as a set of logical devices (Fig. 2.3). Every logical device has a world-wide unique identifier and holds certain information, which is modelled by interface objects. The information is organized in attributes and can be accessed through methods, depending on the access rights (Fig. 2.4). These attributes and methods are accessed at the application layer using the

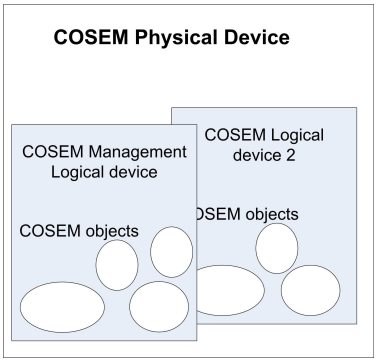


Figure 2.3: COSEM Model

xDLMS protocol services, which arrange the results into data packets (APDU) and delivers them through a stack of layers to the peer application. DLMS/COSEM provides standard codes to reference all the information in a meter device (OBIS codes) and defines a protocol stack for communication, as explained below.

2.3.2 Object Identification System (OBIS)

OBIS provides standard identification codes for all the data items used to configure a meter or obtain information about its behavior. OBIS codes are organized into a hierarchical structure using six value groups of one byte each (A to F in Fig. 2.5). The value group, A, defines the energy type to which the metering is related. Group B defines the channel number, assuming different connections, possibly from different sources. Group C defines the abstract or physical data items related to the information source concerned, for example, current, voltage, or temperature. Group D identifies the processing methods and country-specific codes. Group E is used for identifying rates or can be used for further classification. Last, group F is used for identifying historical values or can be used for further classification. A list of OBIS codes for electricity, gas, and water is available in [34].

2.3.3 DLMS/COSEM Communication Protocol Stack

Data exchange between a metering equipment and data collection system is based on the client/server paradigm, with the meter device acting as the server and the data collection

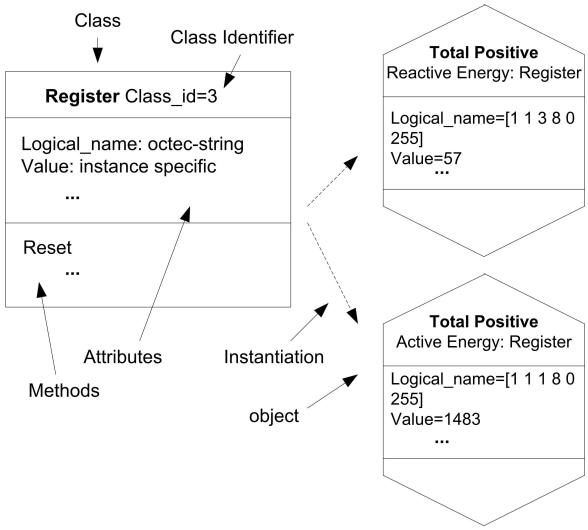


Figure 2.4: An interface Class and its Instances

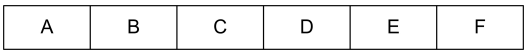
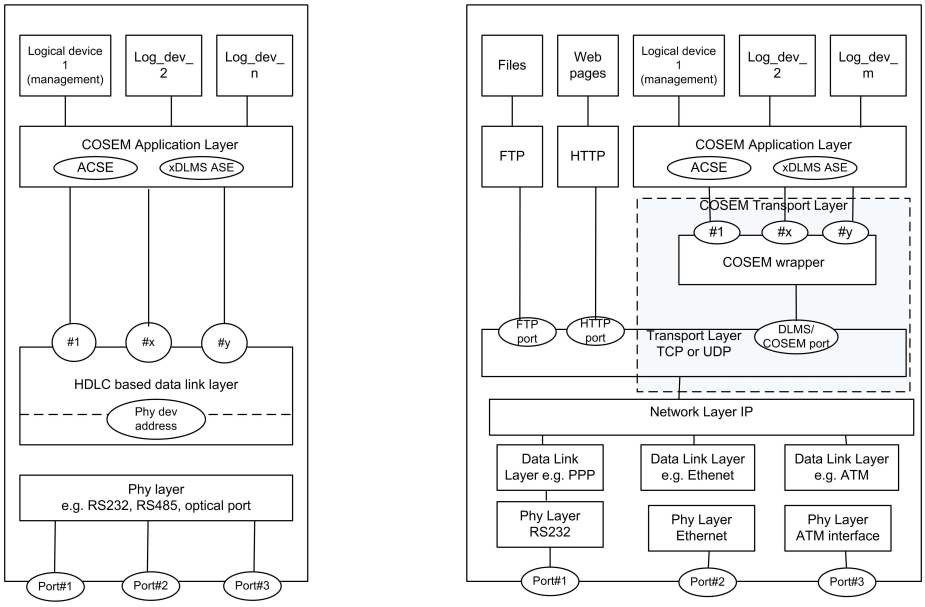


Figure 2.5: OBIS Code Structure

device as the client. An exchange of messages such as *SERVICE.request/.response* goes through a protocol stack. DLMS/COSEM supports different communication profiles (sets of protocol stacks). A single device may support more than one profile so that communication can take place over various communication media (e.g., the Ethernet and GSM)

Figure 2.6 shows two common profiles: the first is the layer, connection-oriented (CO), HDLC-based profile, consisting of the COSEM application layer, the HDLC-based data link layer, and a physical layer for connection-oriented asynchronous data exchange. It supports optical or electrical ports (e.g., RS232.) The second profile is the TCP-UDP/IP based communication profile. At the top is the COSEM application layer. Next is the transport layer, which involves TCP or UDP as well as a wrapper. The wrapper's role is to match the TCP or UDP ports to the logical device address. Since TCP and UDP are supported, other services such as FTP and HTTP can also be implemented. The IP layer is used for addressing the physical device and is supported by different sets of lower layers (data link and physical layers), depending on the media used, e.g., the Ethernet, PPP or



(a) 3-layer,CO,HDLC-based profile (b) TCP-UDP/IP-based Profile

Figure 2.6: Communication Profile Models in DLMS/COSEM

IEEE802 [33].

The support for these profiles as well as others is a strong point of the DLMS/COSEM standard. It enables a data collection system to establish connections with metering devices with different communication protocols and different communication media, thereby allowing a smooth migration from legacy meters to new ones.

2.4 Design Challenges

The SMI network poses certain challenges that come from the need to handle a large amount of data flowing to a centralized location. The data constitutes small packets frequently transmitted from hundreds of thousands of small devices (meters) and control data sent down to the meters. Electricity is at the forefront today, but the challenges apply to all metering data. The SMI application is particularly different in its management of various types of traffic, its tolerance of and reaction to failure, its tolerance of delay, and its security needs. It is important to consider its properties in the application communication design.

These distinctive characteristics are summarized in the following enumeration, followed by discussion showing how they impact a design under their relevant titles.

- Sessions are short (granularity of seconds), with a long waiting period (granularity of minutes) between two sessions.
- Tolerance of delay is different according to the traffic type, ranging from real time delivery to a delay until the next session is due. Constraining jitter delay between successive packets is not necessary.
- Order of data packets can be ignored as long as there is a reordering mechanism at the receiver.
- Data aggregation is not feasible. The collection center must uniquely identify the meter ID and the time at which the consumption measurement is taken.
- Duration of consumption reporting is configurable.
- Loss of messages is not allowed. However, previous energy consumption measurements can be accessed and combined with the current measurement value.
- Identification of and response to failure must be quick.
- Multiple routes exist if a mesh network is created out of the meters. Alternative routes are not available if meters communicate with the base station through one hop only.
- Meters have diverse capabilities. Some meters relax the power constraint, while others feed on a limited source of power.
- SMI is required to be safe from unauthorized access, tampering with data, denial of service, and hijacking of session attacks.

2.4.1 Data Collection Mechanism

All the previous work focuses on gathering power consumption information, a process in which SMI data is pushed from meters into the network at certain fixed times. As utility providers are interested in collecting a large variety of data at frequent intervals, three

modes of communication must be supported: fixed scheduling, event-driven and demand-driven. Each mode is more suitable for a certain kind of data and, thus, the three modes must co-exist. Each mode poses a different design challenge.

Fixed Scheduling: In this mode, a meter reports data at fixed intervals, bya straightforward mechanism with the advantage of guaranteeing a certain rate for every meter under the knowledge of the available bandwidth. However, the traffic that results is significant, which may impact other Internet/Utiltiy traffic at bottleneck nodes. As a result, packets will be dropped and data reports may not meet their delivery deadline. Sbai and Barakat [122] discuss the problem of gathering data from a large number of sources and propose a pulling mechanism at transport level to shorten the duration of a collection session and to reduce the ratio of packet loss.

Event-driven: Data are generated and transmitted as a result of events at meters. Examples of this mode include packets generated when consumption reaches a certain threshold value, power quality when it starts to degrade, and alarm data. This mode may cut down the amount of traffic, though levels vary from time to time; however, a tradeoff must be considered as a contention overhead and delay may be introduced.

Demand-driven: Upon a request from the data collection center, data packets are generated and transmitted back. A utility company uses polling to identify faults, or gets a consumption report at a certain time for a subset of meters. Polling requires extra messaging for the end parties to re-authenticate and set up other communication parameters every time. Demand-driven data typically require realtime response. Therefore, such data should be distinguished from the rest and given a higher level of priority.

2.4.2 Handling Failure

The smart metering system requires a protocol that provides reliable properties such that if failure occurs (*e.g.,* packet loss or device break down), a detection mechanism and preferably an auto-recovery mechanism will be activated. End-to-end reliable data delivery is essential here. A transport protocol such as TCP can guarantee reliability but would incur overhead. Thus, instead of using a generic transport protocol, one tailored to the SMI application has the potential of achieving better results. Device breakdown cannot be handled at the transport layer, although it will disrupt the flow of data. Device failure is to be left to the application layer to recover from. For example, multicast traffic (e.g., control data from the utility server to the meters) may result in a large overhead if a per recipient acknowledgement is employed. Instead, a collective ACK can be considered, in which a gateway node combines the ACKs from the meters and forwards a single ACK

to the utility server. A similar approach to custody in delay tolerant networks (DTN) [44] can be considered. An intermediate node acknowledges reception from the collection center and then takes full responsibility for delivering the packet to the meters.

For upload traffic, consumption reporting is periodical. If the report is not received at the scheduled time, instead of persistent retransmission, the lost consumption report is aggregated with the next one.

2.4.3 Real time and Delay tolerance

SMI Data traffic can be classified into realtime traffic that requires immediate delivery, and delay-tolerant traffic. For upload traffic, consumption data can be delayed until the next scheduled consumption report is due. However, certain event-driven packets must operate realtime. Examples include tamper detection and failure notification. On the the other side, download traffic such as connect or disconnect control packets constitutes realtime traffic.

Typically, Realtime Transport Protocol (RTP) [124] is used for the realtime data, and TCP or UDP for delay tolerant data. However, these transport protocols are irrelevant for SMI. RTP, for instance, is designed to deliver packets while making sure the jitter time is bounded, and if a packet is delayed, it gets dropped. With SMI, jitter is not a requirement, but losing a packet is not permissible. Second, the SMI real time sessions are short and occasional. Other transport protocols are either too light (e.g., UDP) and do not guarantee delivery of packets, or excessively persistent (e.g., TCP) and do not take advantage of SMI characteristics. Thus, they are inefficient.

2.4.4 Unicasting and Multicasting

SMI data constitutes unicast and multicast traffic. The unicast traffic is initiated from both end points: from meters to a utility server and vice versa, with the first type being the dominant. The operation is similar to collecting sensory data in wireless sensor networks (WSN), for which plenty of sensor fusion protocols are available [154] [153]. Nonetheless, in SMI, a meter's data is unique; as such, it cannot be aggregated with other meters' data, thereby making sensor fusion protocols irrelevant. Therefore, the smart metering application must take this challenge into consideration and schedule meter traffic accordingly.

Multicast traffic involves control data that is destined for all or for only a subgroup of meters, either residing in the same region or in different regions. Normally multicasting is

supported at the IP routing level, at which the router creates optimal distribution paths to recipients. At the transport level, UDP can be used, but packet delivery is not guaranteed. For multicast reliability, Pragmatic General Multicast (PGM) [131] can be used, but with extra overhead.

As Fig. 2.1 shows, every group of meters is attached to a gateway (*i.e.,* RC). Thus, the gateway can play an important role in reliably delivering the data to the intended meters. Additionally, data link layer features can be exploited. If WiMAX technology is incorporated, packets can be delivered to a set of meters simultaneously (in the same slot). If the meters form a mesh network, sensor network multicasting protocols can be considered. Lian et al. [92] provide a concise review of the multicasting protocols and propose a geocasting approach that guarantees reliable delivery of messages while keeping transmission cost low.

2.4.5 Network Access and Routing

Meters are stationary nodes distributed at fixed locations such as households. This distribution forms a static topology and makes ensuring connectivity easier than in other wireless networks, including sensor networks. However, although many routing protocols are available in the wired and wireless worlds, choosing or designing one for SMI still requires a closer look at its specifics and requirements. The following points summarize the challenges and special considerations related to SMI:

- Transmission media and data link: Network layer design is closely related to the underlying medium used and the Medium Access Control (MAC) protocol. Routing protocols differ according to the topology of the network and how reliable the MAC protocol is.
- Fault tolerance: MAC and routing protocols must form alternative links and routes when nodes break down or lack the energy to route traffic through. This may involve rerouting traffic or adjusting transmission power levels.
- Scalability: the number of meters in a certain vicinity may be in the order of thousands. MAC and routing protocols must be able to work with such a large network, given that meters are limited in memory and buffer space.
- Quality of service: as introduced in Section 2.4.1, SMI traffic involves information that must be delivered within a certain amount of time; otherwise the data will be useless. Bounded latency for data delivery is a condition to be considered.

- Adaptability: Network conditions are changeable. The routing protocol should be able to recognize a node's state and change its route accordingly. For example, the energy level may change for some nodes (decrease or increase). Additionally, the routing protocol should be able to take advantage of diverse node hardware specifications. It should assign nodes with large memory to store more routing information and let nodes with a fixed energy source (e.g., electricity meters) perform long range communication.

2.4.6 Security

SMI security must be end-to-end to prevent unauthorized access to the metering equipment or any of the SMI intermediate devices and to prevent tampering with data. Adding security cryptosystems, however, imposes extra load on the device processing and impacts energy consumption and bandwidth. Thus, selecting the right cryptographic tool is critical. For example, confidentiality of the SMI data is not as critical an issue as is data integrity. Therefore, a strong message authentication protocol is preferred, while encryption cryptography can be kept simple.

Security is typically implemented at different layers. Taking WiMAX as an example, frame encryption and device authentication are implemented at the link layer, which secures the wireless signal (meter to base station), ensuring that only legitimate devices access the WiMAX network. At the application layer, extra security mechanisms can be implemented to ensure end-to-end security. A COSEM application layer supports three levels of security: 1) No security. 2) Low level security, which uses a password to authenticate the client. 3) High level security, which assumes no encryption is in place, and in consequence, a more complicated authentication procedure is adopted to authenticate both the client and the server.

Key management is another issue to tackle here. Given a large number of meters, how can unique keys be distributed for every meter? Pre-deployment provisioning of keys might be difficult to realize. Asymmetric cryptography might also be impractical to implement in the metering devices, due to the burden that public cryptographic key generation and security primitives add to such a resources-limited device; that is, more processing power would be consumed, more memory storage space would be required, and larger packets would need to be transmitted. Although recent publications such as [137] argue that certain public cryptographic security primitives are viable today on small devices, research is still ongoing to confirm this possibility.

2.5 SMI as a Wireless Sensor Network

Functionally, a meter device is a sensor node that provides energy (electricity, gas, or water) consumption measurement. The number of meters can grow to thousands, and data are typically fused and delivered to a centralized location for processing and decision making. Such characteristics make metering equipment viewable as regular wireless sensors that can form a wireless sensor network (WSN), which is investigated extensively, and for which a good number of protocols have been proposed that can be benefited from for SMI. Wireless sensor networks are diverse in their application objectives, density of nodes, H/W constraints, and nature of traffic (direction and urgency of data). The recent research in the field of WSN typically takes the approach of considering those factors to optimize communication protocols to best satisfy the overall application objectives [61]. SMI as a sensor network is comparable to those of large-scale sensors with the combination of traffic types: sensor to sink (upstream) event-driven data and periodic data gathering, and sink to sensor (downstream) sink-initiated querying. Thus, while referring to WSN, it is important to highlight the special characteristics of SMI that may be involved in choosing or designing the right protocol.

- At the application layer, before data transfer can take place, a connection must be established between the end points, requiring the maintenance of end-to-end reliability semantics. This end-to-end connection is not recommended for large-scale sensor networks because of the lack of unique Internet-like addressing for each node, and because it results in large in-network packets and high end-to-end delays [67]. In such networks, sensors typically send their available readings to the nearest in-range nodes [11] [84]. For that reason, most studies (e.g., [141] [135] [59] [142]) focus on maintaining hop-by-hop reliability. Work on end-to-end reliability semantics work is also available in the literature, however. Dunkles et al. [6] propose a version of TCP/IP that is tailored to sensor networks by maintaining end-to-end semantics combined with hop-by-hop reliability. The protocol caches packets in nodes to reduce the burden of end-to-end retransmission of lost packets. Park et al. [112] propose a downstream reliability protocol for delivery of control data and queries. Another way of addressing reliability semantics that may suite SMI event-driven data is event reliability, that is, to make sure an event is reliably reported to a base station with a certain degree of accuracy (e.g., [10]).

- To achieve scalability and elongated battery life in large-scale sensor networks, instead of having homogenous sensors rotating the role of clusterhead among themselves,

heterogeneity has been introduced [61]. That is, nodes that have sophisticated hardware and higher battery energy take on the role of a clusterhead to perform complex computations and long-range communication. Each clusterhead manages its cluster autonomously. The cluster may consist of nodes with different hardware capabilities. Mhatre et al. [104] explain such a design of heterogeneous networks, and in [103] Mhatre and Rosenberg present a cost-based comparison between homogeneous and heterogeneous networks. The positioning of clusterheads, however, is a question of optimality. In SMI, meters are also diverse in their hardware capabilities. Electricity meters feed on a main power supply. Gas meters feed on batteries (for safety measures.) Water meters feed on both. Thus, electricity meters are good candidates to act as clusterheads that fuse traffic from the other meters that feed on batteries. Positioning of the meters, however, is not controllable.

- To reduce the traffic load of a sensor network and reduce energy cost, the amount of data transmitted in the network is reduced by means of data aggregation. Typically, in large-scale sensor networks (e.g., for habitat monitoring [68]), the sink is not interested in the individual measurements, but requires a distributed computation of some function of the sensor readings. Data aggregation allows nodes to combine multiple readings into one report containing the result of a function such as the average, median, Min or Max [45]. Different algorithms are available to achieve that goal. For example, Tiny Aggregation (TAG [98]) and [12] allow the sink to send queries to a certain set of nodes and let the nodes along the path perform the requested data aggregation type. In [60], in addition to data aggregation, the protocol increases energy saving by increasing the path sharing among different sources. In SMI, however, packets carry unique information identifying a specific meter and the exact time of measurements. Therefore, measurement data from individual meters must reach the collection center while remaining intact.

- In a large WSN with sensors distributed over a large geographical area, because sensors have limited energy and because they sometimes exist in harsh environments, node failure occurs commonly, which leads to service degradation [30]. In sensor network deployments (e.g., glacier monitoring and tracking of military vehicle applications) node failure is tackled in two ways: deployment of redundant nodes and use of algorithms to detect and isolate faulty nodes [19]. In SMI, meters are distributed deterministically, with zero redundancy, at every energy distribution location (e.g., residential houses). However, if a meter ceases to operate or malfunctions, immediate investigation and maintenance must take place. In other words, a fault detection mechanism is essential. Fault detection protocols for sensor networks are available in

the literature, and have a common objective: to be energy- and time-efficient. Jiang [65] presents a review of fault detection protocols and proposes an enhancement to increase accuracy when the number of neighboring nodes decreases. Yamanouchi et al. [149] evaluate the reliability of sensor networks under different weather conditions using a fault detection algorithm that investigates the collected sensory data. SMI characteristics such as the periodicity of data reporting and static topology should be considered to optimize fault detection mechanisms.

- With regard to routing, WSN protocols may be considered for SMI, but attention should be paid to the fact that meters have fixed positions. Thus, a protocol that considers the node location is preferred (e.g., [93]). Nevertheless, if meters communicate to a base station in one hop, then such protocols are not suitable. Moreover, large sensor network routing protocols use attribute-based addressing. The sink issues an attribute-based address composed of attribute-value pair queries. Meters, in contrast, need to be uniquely identified. For example, the control station may need to connect/disconnect energy for a specific customer. Thus, routing should be looked at in light of a different addressing mechanism, which leads to the use of IPv6, as is currently being discussed by the IPv6 over Low power WPAN (6LoWPAN) Working Group in [76].

- SMI must support bidirectional communication to allow for meter set-up and reconfiguration at any time. In most WSN applications, this requirement does not necessarily hold.

- Meters may have similar real time constraints to certain sensor applications. Nonetheless, one should stress that the delay in SMI is tolerable only within a defined time window, determined by the meter measurement schedule.

- Security is a serious concern in both meters and sensors [29]. However, the biggest concern with meters is the provision of data integrity as opposed to data privacy.

The aforementioned differences between meters and sensors must be taken into account when designing a new protocol for SMI. Rather than employing or even modifying a protocol that is generic for wireless sensors, identifying the differences and the unique meter characteristics definitely leads to a more successful and efficient SMI design.

Chapter 3

Literature Review

This chapter's two sections review the literature of smart metering infrastructure communication protocols. The first section provides an in-depth literature review of TCP protocols. In fact, just a few of the existing publications have tackled the transport layer issues of SMI. However, this section brings into discussion transport protocols from other areas that share a common ground with SMI in the context of data collection, for example, in large scale wireless sensor networks and certain collection applications over the Internet. Therefore, a variety of transmission protocols, categorized by the overall objective, are reviewed and critiqued as to how likely they would fit in an SMI.

The second section complements the discussion on TCPs by reviewing the literature on low-layer design proposals. It surveys the work done at the physical level, in which different technologies and models were considered. It also presents the work done at the network access and routing levels. Interestingly, the technologies proposed range in bandwidth from as low as a few hundred kilobytes per second to as high as gigabits per second.

3.1 Transmission Control Protocols (TCPs)

3.1.1 Basics of TCP

TCP is the most widely used end-to-end connection-oriented protocol. It accounts for most Internet traffic, so performance of the Internet relies to a great extent on how TCP behaves. The literature covers numerous TCP solutions. The common goals are to provide fully reliable, in-order, end-to-end delivery service, to eliminate congestion collapse, and

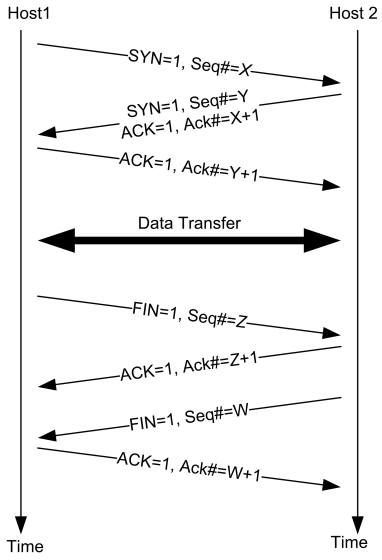


Figure 3.1: TCP Connection Management Signals

also to use the available network resources effectively in different types of environments; such as wired, wireless, high-speed, and long-delay. Effective network utilization does not refer only to how well a single TCP connection uses the network, rather also to how well it cooperates with other TCP connections with which it shares the network resources.

Before citing various TCP congestion mechanisms, it is important to explain the common TCP functions in terms of its connection management, congestion and flow control, and error control mechanisms.

Connection Management: Being a connection-oriented protocol, TCP exchanges a set of control signals to establish a connection between a server and a client, and after data transfer is completed, another exchange of signals takes place to terminate the connection. As Fig. 3.1 shows, a three-way handshake mechanism is used to establish the connection. That is, the TCP sender sends a SYN packet; the receiver responds with a SYN-ACK packet, and the sender finally acknowledges with an ACK packet. For termination, however, the sender and the receiver engage in two-way handshakes through the exchange of FIN and ACK packets. Sequence numbers are used to synchronize both parties.

Congestion and Flow Control: Bottleneck links occur in a network as a result of multiple traffic flows trying to penetrate the link. When the buffer capacity is exceeded, packets get lost. Without proper congestion control, the retransmission of lost data to-

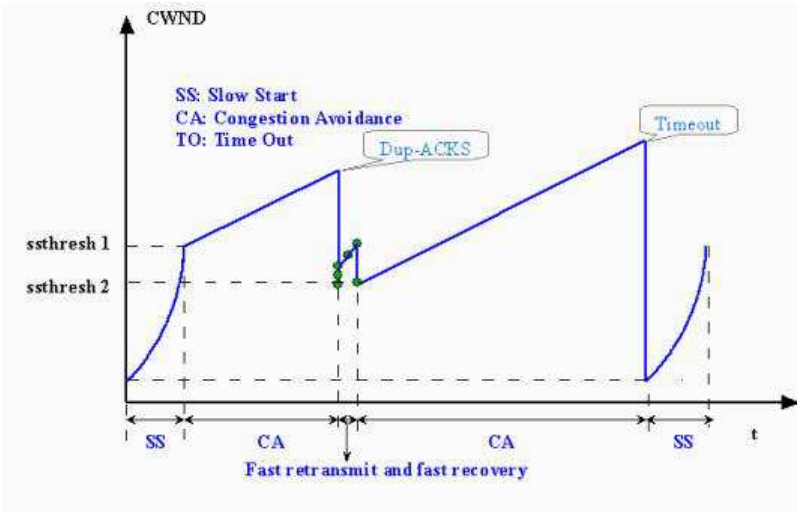


Figure 3.2: TCP Congestion Window Mechanism

gether with the ongoing transmission leads to a worse condition by further slowing down the network and causing more packet loss. As for flow control, the goal is to avoid overwhelming a slow TCP receiver with too many packets. That too would lead to dropping packets at the receiver if its queue exceeds the buffer capacity. In TCP's congestion control scheme, the sender maintains a congestion window that regulates the number of unacknowledged data packets in the network. Each sent packet consumes a slot in the congestion window, and the sender can send a packet only if a free slot is available in the window. When an acknowledgement for an outstanding packet is received, the window is shifted and a slot becomes available (Fig. 3.3). To achieve flow control, the TCP receiver notifies the TCP sender of the amount of free space available in the receiver's buffer through an advertised window. The TCP sender performs congestion and flow control by ensuring that the transmission window does not exceed the size of congestion window and the receiver's advertised window.

The congestion window is dynamically adjusted by the congestion control algorithm. During the life cycle of a TCP connection, the window grows and shrinks in relation to the available link capacities and congestion status. Initially, the window size starts with one (*i.e.,* 1 MSS) and grows exponentially (*i.e.,* addition of 1 MSS with each ACK received, or in other words, doubling the window size every one round trip time). This is referred to as the slow start phase (SS). Once the window size reaches a certain predetermined threshold value, the window size growth becomes linear. That is, it increases by one if all the packets

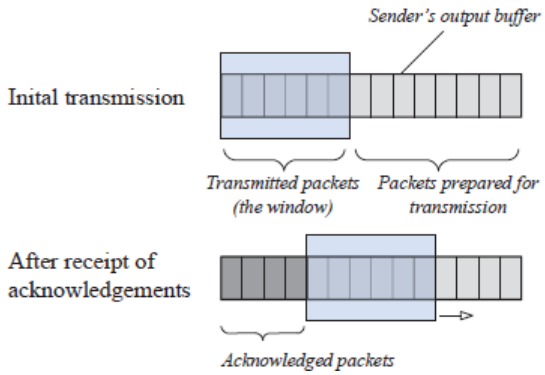


Figure 3.3: Sliding Window Concept

within the last window reach the destination with absolutely no loss. This phase is called Congestion Avoidance (CA). If packet loss happens, the threshold is set to one half of the current congestion window, and the congestion window itself is reset to one to reduce the transmission rate and goes into the SS phase again. TCP recognizes packet loss by either timeout or by receiving duplicate ACKs from the receiver. Duplicate ACKs trigger a phase called fast-retransmit, in which a lost packet is retransmitted immediately. In addition to packet loss, this event suggests that out-of-order packets are received.

The sliding window mechanism is effective but requires optimization so that performance is enhanced a requirement because the growing or shrinking the window size has several conflicting objectives. For instance, to maximize throughput, the congestion window should be enlarged. However, if it becomes too large, the chance of packet loss increases because the network and receiver resources are limited. Thus, to decrease the packet loss rate, the congestion window should be minimized. Consequently, the optimization problem is to find an optimal size for the congestion window that results in the best throughput yet does not overwhelm the network and the receiver.

Error Control: TCP provides data loss recovery through the use of timer-driven and data-driven retransmission mechanisms. In the timer-driven recovery mechanism, the TCP sender maintains a timer. If a positive cumulative ACK for a packet is not received within a certain timeout interval, the sender retransmits the missing packet and backs off exponentially after each unsuccessful retransmission. The timeout interval is normally estimated in relation to round-trip times. In the data-driven recovery mechanism, the sender relies on feedback from cumulative acknowledgements (ACKs). After a packet is lost, the receipt of all later packets generates duplicate ACKs to the TCP sender. The TCP

sender can then detect the lost packet and retransmit it when the number of successfully received duplicate ACKs exceeds a predetermined threshold, which is typically three.

Classification of TCPs

The performance of TCP has been improved at different stages of its development. The TCP protocols can be categorized in accordance with the objectives targeted. Some variants of TCP were proposed to merely fix the congestion collapse problem and later to improve the TCP connection's throughput.

Other mechanisms were proposed to help prioritize certain TCP flows over others. Another set of TCPs aim at certain environments, such as wireless or satellite links. Furthermore, some TCP congestion protocols serve multicast applications rather than unicast ones. Congestion mechanisms for reliable data collection make up another class reviewed here. Recently, SMI transport layer issues have been researched as a new category for its distinct nature and requirements.

3.1.2 Solving Congestion Collapse

A number of proposals have focused on fixing the phenomenon of congestion collapse. The initial TCP protocol did not consider adjusting the transmission rate as the network resources become congested. Congestion collapse is the result of an increase in the network load that leads to a decrease in the useful work done by the network. In other words, the goodput of the network becomes a small portion of what the network can actually offer. Congestion collapse occurs for various reasons. It happens because of the unnecessary retransmission of packets, which is typically fixed by properly adjusting timers and fixing time estimates. Congestion collapse is also caused by undelivered packets, which means packets hopping over nodes not achieving their final destination, resulting in the bandwidth being wasted. This situation occurs mainly due to the increased deployment of TCP-unresponsive applications, called open-loop applications. These applications do not implement end-to-end congestion control mechanisms.

Congestion collapse can be demonstrated as in Fig. 3.4 (discussed by Afanasyev et al. in [9]). Assuming a router is required to deliver traffic at four times its capacity in both directions between networks A and B, the excessive amount of traffic means that the router will have to drop at least 75% of the packets. Thus, only 25% of the packets at most will reach the receiver and trigger ACKs. If the reverse link is congested the same ways, again

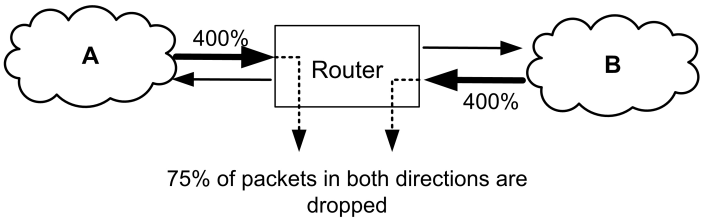


Figure 3.4: Congestion Collapse

only 25% of the ACK packets will reach the sender. Thus, 6.25% of the data packets will be acknowledged successfully. In this illustrative scenario, the consequence is a severe drop of 93.75% of the throughput.

Another scenario, presented by Floyd and Fall [51], shows the problems that result from the absence of an end-to-end congestion control mechanism or what is called TCP-unresponsive flows. Figure 3.5 shows a typical network topology that demonstrates how congestion collapse may occur in the Internet. The scenario assumes that a TCP flow exists between S1 and S3, and a UDP flow exists between S2 and S4, all sharing the link R1-R2. As the UDP source increases its speed, naturally the data arrival increases at router R2, with the UDP taking advantage of the available 1.5Mbps bandwidth. However, the packets get dropped right there because the bandwidth available at R2-S4 is limited to 128Kbps. This traffic causes the TCP source to reduce its transmission speed in reaction to the dropped packets. Although the UDP traffic will eventually be limited by the 128 Kbps bandwidth, the only effect is to imped the TCP traffic. The UDP flow just wastes the R1-R2 link bandwidth that could have been utilized by the TCP flow and reduces network efficiency as a whole to a small fraction of the actual offered bandwidth.

The problem is not related to the bandwidth available at R1-R2. increasing the R1-R2 bandwidth or reserving more than 128 Kbps of its BW does not solve the problem. What is needed is to have an end-to-end congestion control in place to prevent flows from continuing transmission when a large portion of their packets get dropped before reaching their destination.

TCP Tahoe

Tahoe [62], one of the early TCP protocols, was once the standard but is now obsolete. It addresses congestion collapse by modifying the original TCP specification in three manners. First, it enhances the retransmission timeout (RTO) estimate, which impacts TCP

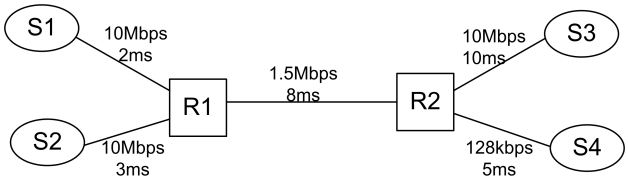


Figure 3.5: Congestion Collapse Scenario

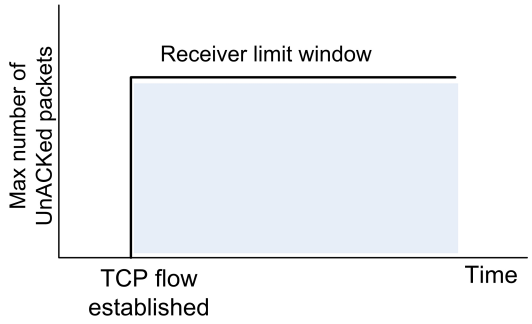


Figure 3.6: Original TCP Specification (RFC793)

performance. If the RTO value is overestimated, the performance of a TCP connection degrades as detection of packet losses takes more time. On the other hand, if RTO is underestimated, the error detection mechanism may cause the congestion window to shrink and cause unnecessary retransmissions. Second, it makes packet loss detection faster by introducing duplicate packets as a loss indicator. A sender as such can detect a loss in much less than the estimated RTO. Third, it introduces the slow start and congestion avoidance algorithms, which allow the sender to adjust its congestion window instead of just sending at the fixed speed specified by the receiver (Fig. 3.6). The sender starts with the slow start phase, in which the transmission rate starts at one segment per round trip time and grows exponentially, but drops to one when packet loss is detected (Fig. 3.2). The congestion avoidance phase is more conservative. It allows the sender to increase its transmission rate linearly, but upon loss detection, the congestion window is halved.

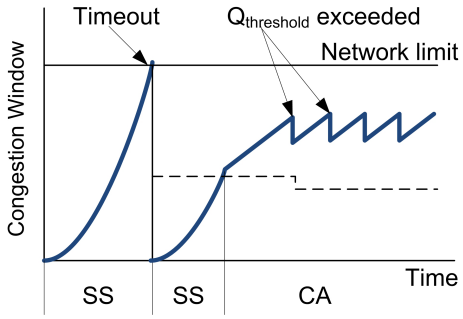


Figure 3.7: TCP DUAL Congestion Window Phases. SS:Slow Start, CA:Congestion Avoidance

TCP Dual

Wang and Crowcroft [144] proposed TCP DUAL to further tune TCP Tahoe so as to mitigate the oscillatory congestion window pattern, an improvement that leads to high variability in transmission rate, round trip time, buffer utilization, and packet loss. TCP DUAL introduces the idea of observing the queuing delay as a proactive congestion detection mechanism. It also makes reaction to congestion events softer.

TCP DUAL keeps track of queueing delays and estimates the delay threshold as half the maximum queuing delay. If queuing exceeds the threshold, the congestion window is decreased by $\frac{1}{8}th$. Fig. 3.7 shows that reduction is not as oscillatory as in Tahoe.

TCP Reno

TCP reno [14] argues that duplicate ACKs do not indicate severe congestion in the way a timeout event does. Therefore, to better utilize a link, it modifies the fast-retransmit phase by reducing only by half rather than resetting to one. Another mechanism, called fast-recovery, was introduced in TCP Reno. In this method, the congestion window is increased by one for each duplicate ACK because each one indicates the successful arrival of another packet. TCP Reno is relatively simple to implement and performs fairly well. For that reason, it is generally considered the congestion control standard for TCP, although it is known for bad performance in a wide range of environments. For example, it suffers great performance degradation in the presence of consecutive and random packet losses and unordered arrival of packets. It is also inefficient in high-speed long-delay networks

TCP Newreno [49] further enhances TCP reno in regard to the case of multiple packet losses. It avoids the need to go into multiple fast-retransmits for the same window of data by letting only one packet be retransmitted per round trip time.

TCP SACK

An ACK messages is limited to acknowledging the last in-order delivered packet only. Consequently, performance is impacted. For example, if consecutive packets are lost, the fast retransmit congestion mechanism lets the sender retransmit only the lost ones. The next ones will be detected in the next duplicate ACKs. TCP SACK [48] solves the multiple-loss problem in a different way from Newreno. It lets the receiver provide more information through selective acknowledgement packets, which specify blocks of packets that have been received successfully. The sender determines then which packets were lost and thereafter quickly retransmits them in a process that requires only one round trip time.

However, TCP SACK has a serious problem because the ACK messages are limited in size, so it cannot declare all the lost packets in a window. The options field is limited to 40 bytes, so it can carry 3 to 8 sequence numbers depending on the required information. Especially in wireless networks where the percentage of packets lost is high, this is not enough.

TCP Vegas

TCP Vegas [21] offers another proactive TCP congestion mechanism. TCP DUAL made the first attempt to resolve the oscillatory problem using an estimate of queuing delay. TCP Vegas also uses RTT to estimate bottleneck buffer occupancy, but it further finds 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.

TCP Vegas has the major advantage of improving transmission rate stability and improving the overall throughput of a TCP connection. However, it has the disadvantage of being unable to get a fair share when competing with TCP flows that use TCP Reno or similar variants. The problem stems from the fact that Vegas is proactive rather than reactive. TCP Vegas+ [57] improves its performance by adapting its aggressiveness according to the situation. It assumes a Vegas-friendly environment, so it uses the bottleneck buffer estimation to control its congestion window, but once it detects a Vegas-unfriendly environment, it switches its congestion avoidance mechanism to the Reno mechanism.

TCP VegasA (Vegas with Adaptation) [133] further enhances TCP Vegas. It makes the argument that RTT changes for other reasons, not only due to buffering. For example, change of routes may change RTT and thus prevent the congestion window from being adjusted properly. VegasA can detect the change in bandwidth, so it adjusts its control boundaries more accurately. The protocol shows significant improvements over Vegas as simulation results suggest. The algorithm, however, is still experimental and has not been proved in a real network.

3.1.3 TCP Protocols for Smart Power Grids

For a long time, most of the research in the SMI field has been on the choice of communication technology and link and routing issues. Transport level issues have just recently caught researchers' attention. To the best of our knowledge, our previous work [75] was the first to highlight the special requirements of SMI applications and the shortcomings of TCP in this field.

Allalouf et al. [13] tackle the problem of congestion caused by the large volume of metering data and limited bandwidth by performing hop-by-hop traffic reduction. They assume that data samples produced by meters are required at certain intermediate devices but not at the utility center. They further assume that the intermediate devices can process data at the application level. Therefore, they propose routing traffic through devices where more detailed data is needed and more reduction can be applied on such data.

Kim and Thottan [78] propose a new transport control protocol that targets mainly delay-sensitive smart grid applications. The authors study all that contributes to increased end-to-end delay in the TCP protocol. In consequence, their proposed protocol avoids all unnecessary delays, thereby making delivery faster. However, they do not consider the impact of congestion control. Furthermore, they modify TCP to a large degree (including the TCP header), which makes it inoperable with the standard TCP.

Kim et al. [79] focus on the security aspect of TCP in smart grid. They take into account the limited resources of meters and sensors. Therefore, they propose the use of symmetric pre-shared keys to achieve secure communication. They compare their work to TLS over TCP Reno. Excluding congestion effect, they show that their protocol is faster. They consider our protocol SA-TCP to be promising in terms of scalability; however, they suggest modifications to make it secure.

3.1.4 TCP in Wireless Networks

Starting in the late nineties, wireless networks have been getting more and more popular. A new issue with TCP appeared right at the advent of wireless networks. Although TCP performs well in wired networks, it suffers from serious performance degradation in wireless networks. The reasons are related to assumptions about TCP, and these are not valid in wireless networks. TCP assumes that packet loss is due to congestion and responds to that by decreasing the transmission rate. In wireless, however, packet loss is mainly due to certain wireless specific reasons (*e.g.,* a high bit error rate in wireless channels, hand-offs between cells, medium contention and route breakages). Although bad connectivity may be temporary, TCP still responds by reducing the congestion window. Additionally, highly variable round trip times (RTTs) in wireless networks can introduce false timeouts, thus unnecessarily degrading TCP throughput.

Solutions have tackled the problem in three ways. A number of proposals have attempted to improve TCP performance by splitting a TCP connection into two at the base station or access point. The second group have aimed at hiding the characteristics of wireless links from TCP by providing a reliable link layer. The last ones have resolved the problems by slightly modifying TCP at the end systems, e.g., by enabling selective acknowledgment or fast retransmission. The following are some of such protocols that target wireless networks.

Indirect TCP (I-TCP)

I-TCP [17] aims at wireless networks to improve the performance of TCP connections between a wireless mobile host and fixed host in the wired side of the network. I-TCP splits the connection between the hosts into two separate connections: one formed between the wireless host and an intermediate support router and the other between the support router and the wired host. The wireless side of the connection can support notifications of events, such as disconnections and moves. The support router, which is in the wired network, can perform much of the communication overhead for the mobile host, including retransmission of those packets dropped in the wired network.

TCP Westwood

The problem with I-TCP was losing the end-to-end semantics. TCP Westwood [100] offers an end-to-end TCP connection that is a modification of NewReno's congestion mechanism.

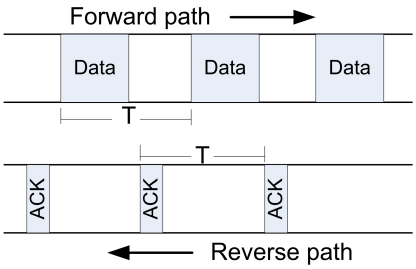


Figure 3.8: Observing forward and backward rate

It adds a heuristic-based procedure for setting the congestion window to an optimal value to achieve a faster recovery. Optimally, the transmission rate should not be reduced if packet loss is due to wireless-related issues, but it should be reduced if loss is due to congestion. TCP Westwood operates on the assumption that the data reception rate observed by the receiver is the exact rate at which the network is capable of delivering the sender's data. To estimate the rate, TCP Westwood observes the rate at which ACKs are received (Fig. 3.8). The bandwidth calculation holds in the long term even if some ACKs are lost or delayed by the receiver. It has been shown through experiments that TCP Westwood gives a good level of precision, but it may be quite deviated in some real environments in which ACKs are grouped or delayed differently. TCP Westwood+ [54] was proposed after in 2004 to further enhance the estimation precision by changing the calculations to achieve RTT granularity.

TCP BBE

TCP BBE (Buffer and Bandwidth Estimation) [127] was proposed for wired-wireless mixed networks to improve fairness with other TCP protocols. It improves TCP's Westwood's policy of reducing the congestion window size upon detecting a loss. It borrows some concepts from TCP DUAL as it uses the queue delay in its estimation of a network's congestion state so that the estimation is neither over- nor underestimated. The protocol has shown good improvement over TCP Westwood in simulation; however, it has not been tested in a real environment.

3.1.5 TCP for Reliable Data Collection

All the TCP protocols presented above are designed for host-to-host communication. They target various environments, different network characteristics and application requirements. Many-to-one applications can be considered another type of scheme, one that poses different transport challenges. In any case, traffic protocols need to be TCP-friendly, but clearly many-to-one schemes are more complex. As the number of receivers or senders increases, the problem becomes more difficult. For example, decreasing the overall speed may not necessarily mean that certain devices using certain paths would approve the decrease.

Data collection is especially important as far as smart metering infrastructure is concerned since it involves a significantly large number of small devices transmitting to a collection center. The following is a discussion of transport protocols for data collection applications.

Data Collection in Wireless Sensor Networks

Reliable data collection has its application in wireless sensor networks. Sensors span a large geographical area, and they produce and transmit their measurement packets to a centralized repository (e.g., sink). Due to WSN nature, an end-to-end transport connection is not recommended for large-scale sensor networks because of the lack of unique Internet-like addressing for each node, and because it results in large in-network packets and high end-to-end delay [67].

In large-scale sensor networks, sensors typically send their available readings to the nearest in-range nodes [11] [84] so that the amount of traffic is kept low and congestion is avoided. Additionally, to ensure a good level of reliability, redundant sensors are deployed. For those reasons, most of the work focuses on maintaining hop-by-hop reliability (e.g., [135] [59] [142]).

Stann et al. [135] examined the different options available for ensuring reliability at the three layers: MAC, transport, application, and the combination of them. They show that hop-by-hop (MAC-based) recovery is the most effective. End-to-end reliability semantics work is also available, however.

Dunkles et al. [6] proposes a version of TCP/IP that is tailored to sensor networks by maintaining end-to-end semantics combined with hop-by-hop reliability. The protocol caches packets in nodes to reduce the burden of end-to-end retransmission of lost packets.

Park et al. [112] propose a downstream reliability protocol for delivery of control data and queries. Another way of addressing reliability semantics is event reliability, which may suit meters' event-driven data, that is, by making sure an event is reported to a base station with a certain degree of accuracy (e.g., [10]).

Although SMI seems similar to large-scale WSN, a transport protocol for SMI would be different for a number of reasons. In SMI, packets carry unique information identifying a specific meter and the exact time of the measurement. Thus, reducing traffic through WSN data aggregation techniques cannot be implemented here. Achieving reliability through redundant deployment of meters is not suitable. A meter typically is attached to a certain service point. Moreover, transport fairness is not required for WSN as normally a sensor network performs a common task for a single entity, but meter traffic shares the communication network with other applications' traffic.

Data Collection Applications in the Internet

On the Internet, there is no transport layer that is specialized in collecting data from a large number of machines. TCP is designed for one-to-one machine communication, so it is not suitable.

SNMP [24] is a protocol that provides delay and topology measurements through collecting data from routers and hosts. However, the process of probing is done at a low rate, so congestion is not expected to occur.

Another application of data collection is reliable multi-casting, specifically the gathering of NACK packets. Congestion may occur as a result of a large number of NACK packets in the network, especially close to the multicast source. For that reason, proposals try to reduce the number of NACK packets in the network. Tan et al. [139] place proxies in the network that can aggregate NACK packets.

Lacher et al. [86] lets nodes wait a random amount of time before sending a NACK packet. This approach cancels sending NACKs if another node has done so. For smart meters, combining data is not possible as a packet should preserve the unique information belonging to each meter.

Lately, Sbai and Barakat [123] have shown the need for a transport protocol to reliably collect data such as statistics, votes and quality of reception in multicast applications from hosts in the Internet. The proposed protocol, called Transport Information Collection Protocol (TICP), provides congestion and error control. For congestion control, the collector maintains a congestion window size that corresponds to the maximum number of requests

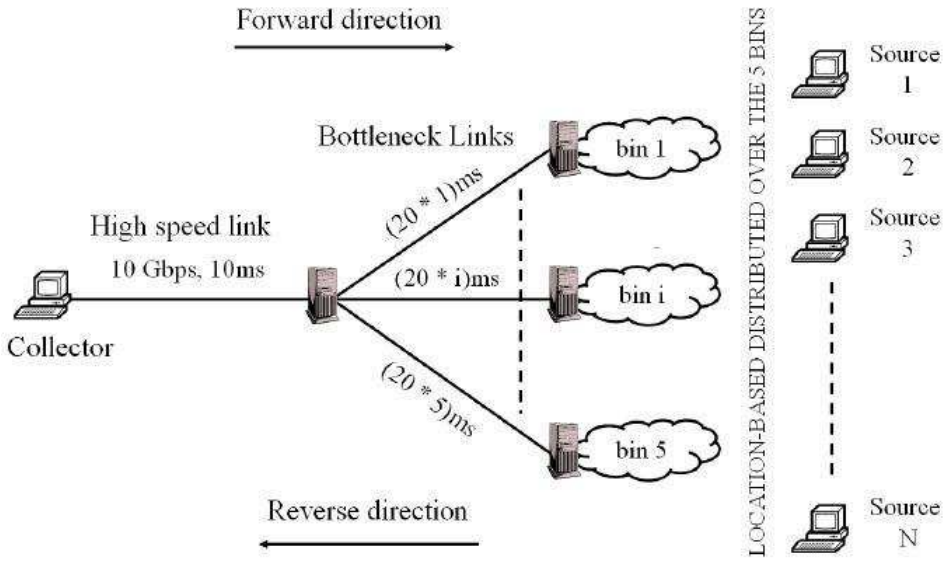


Figure 3.9: TICP Simulation Setup

that can be sent to hosts. As a report is received, the congestion window slides to allow more requests to be retransmitted. For error control, those hosts whose reports were lost are probed in a second round. Simulation is provided in a topology as shown in Fig. 3.9. It assumes thousands of nodes clustered behind bottlenecks with different propagation delays. A collector probes all the sources to receive 1000-byte reports. The simulation shows an improvement in terms of throughput and speed of collection duration when compared to collection using multiple regular TCP connections.

For smart meters, the collection transport protocol should account for the fact that meters exist in the wireless part of the network, which can lead to the high loss and delay of the probe packets sent from the collector machine.

3.1.6 TCP in High speed Networks

High speed networks [134] refer to the networks enjoying high data rates in the range from a few Mbps to Gbps. None of the above variants considered these high speed networks. TCP's Linear growth phase leads to a relatively long packet-loss recovery time. TCP would require a packet loss probability to be as low as 10^{-1} in order to achieve a throughput of 1

Gbps [50], a reasonable objective in high speed networks. A higher probability value would lead to an inefficient utilization of links.

Proposals such as [50] [66] and [116] have targeted such networks with large bandwidth-delay products. These proposals suggest new window updating functions with the common objective of letting the congestion window size grow more aggressively.

3.1.7 Priority-based TCP Protocols

To satisfy applications' QoS requirements, there has been a number of attempts at an IP level. However, the solutions still face deployment problems due to the heterogeneity of the Internet and the need to implement the solutions entirely in the backbone. Applying QoS at TCP level is easier to deploy since it can be implemented in the end systems. The following two TCP protocols offer priority-based congestion mechanisms.

TCP Nice

In 2002 [7], minimizing interference between high priority and low priority TCP connections was proposed. For example, automatic updates, data backups, and file sharing applications typically have a lower priority. TCP Nice was based on TCP Vegas because of the latter's proactive nature. Vegas can already provide some level of prioritization, although it cannot compete with reactive protocols, such as Reno. TCP Nice relies on estimating the queuing delay through measured RTTs. It counts the number of times that the queuing delay exceeds a certain threshold within an RTT period. If that number exceeds another threshold, then the congestion window is halved, as demonstrated in Fig. 3.10. As for all the standard TCP flows, TCP Nice considers them to be high priority, so it does not apply the same aggressive reduction. Additionally, Nice allows fractional window sizes. For example, $\frac{1}{48}$, which is the minimum in Nice, allows only one packet to be sent in 48 RTTs. This makes Nice even more aggressive.

TCP LP (Low Priority) [85]

This protocol is based on Newreno's congestion mechanism, but its objective is to provide low priority TCP connections to background services, such as software updates. TCP LP uses the Timestamp option to keep track of the minimum and maximum delays during a connection's lifetime. The idea is to detect early congestion events. When the first congestion event is detected, TCP LP reduces its window by half and starts a timer. If

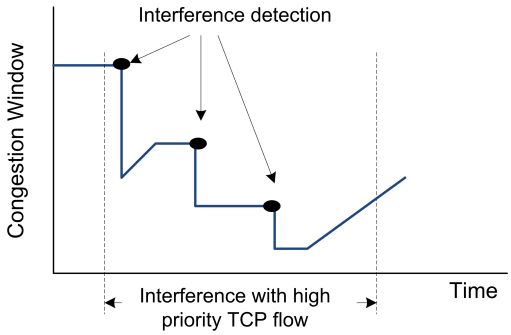


Figure 3.10: Demonstration of TCP Nice Congestion Window Mechanism

another early congestion event occurs before the timer elapses, TCP LP concludes that a high priority flow exists, so it reduces the low TCP connection's window to the minimal value.

3.2 Low-layer Design Models

Research for automating meters started a few decades back, targeting mostly low-layer issues. The major part of an SMI system is the underlying communication technology over which packets can be delivered from both sides. There are four major types of SMI communication networks: power line carriers (PLCs), cellular networks, telephone/the Internet, and radio frequency. This section surveys these proposed technology models and presents the work done at the routing level.

3.2.1 Power Line Carrier (PLC)

In this technology, data is transmitted over voltage transmission lines along with electrical power. Factors such as the choice of frequency, propagation speed, voltage level carried, distance between the two communicating points and the existence of transformers affect the PLC communication properties.

PLC has gained great interest as the SMI backhaul network because no extra cabling is required. Kerk [69] and Soh [129] argue that a PLC SMI combined with a wireless technology network is the only solution to reduce the tariff price and be able to serve more houses in India and Singapore.

Table 3.1: Summary of TCP Variants

TCP Variant	Year	Base	Main Features	Status
TCP Tahoe	1988	RFC793	slow start, congestion avoidance, fast retransmit	obsolete
TCP DUAL	1992	Tahoe	queuing delay for congestion prediction	Experimental
TCP Reno and NewReno	1990 & 1999	Tahoe	Fast recovery	Standard
TCP SACK	1996	RFC793	Timestamp option in ACK	Standard
TCP Vegas A	2005	Vegas	RTT to predict congestion and adapt	Experimental
TCP Nice	2002	Vegas	delay to indicate congestion	Experimental
TCP LP	2002	NewReno	Early congestion detection	Experimental
I-TCP	1994	NewReno	splits connection	Experimental
TCP Westwood	2001	NewReno	estimates available BW	Experimental
TCP BBE	2003	Westwood	bottleneck buffer capacity estimation	Experimental
TICP	2009	Reno	Receiver controls congestion	Experimental

Park et al. [111] discuss the technical features and the available standards for PLC modems, and propose a combination of a PLC network and data network (Fig. 3.11). Every electricity meter is connected to a PLC modem through an RS232 data port. Multiple PLC modems, corresponding to a group of houses under the same pole transformer, connect to a single concentrator modem. The concentration modem bridges the PLC network to a data network. Meters report their measurement when they are polled. The PLC modem buffers the frames until an ACK is received, or otherwise the frame is retransmitted. No evaluation of the system is provided.

Choi et al. [31] propose the use of PLC as a means of delivering electricity, gas, and water measurements to the utility providers. The system involves various devices and different communication technologies (Fig. 3.12); water, gas, and electricity meters transmit their measurements over wireless links to a device called the Home Concentration Unit (HCU), which is to be installed in every household. A number of HCUs, normally from different households, send the measurements to a device called the Data Concentration Unit (DCU), which eventually sends the metering data in Device Language Message Specification (DLMS) format via a PLC modem to the utility company. The traffic direction is only from the meters to the utility provider. No metrics for evaluation or comparison with other designs are provided.

Moghavvemi [106] focuses on digitizing the meter and detecting tampering. The author uses an optical encoder to generate signals and counts them as the electro-mechanical disk

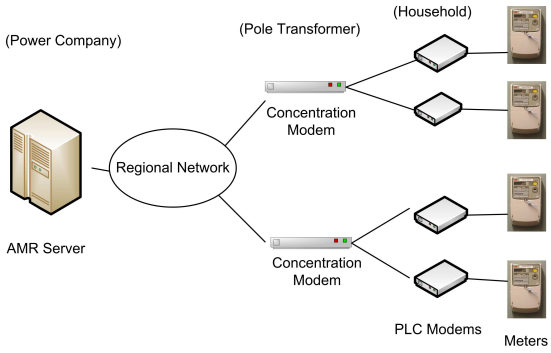


Figure 3.11: PLC SMI Diagram

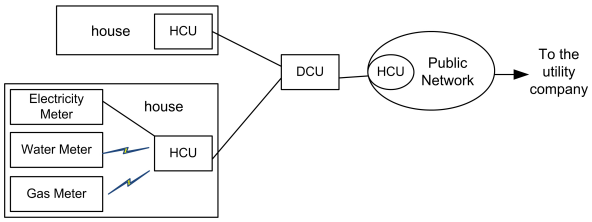


Figure 3.12: IMR System Diagram

rotates. For tamper detection, the data concentrator analyzes the data received and checks whether abnormalities appear. Similarly, Raja and Sudhakar [114] focus on the technical design of PLC modems. However, they do not show how to communicate between the modems from a long distance and how to bypass the transformers. Oska et al. [110] provide testing results for using one-hop communication over power lines. They conclude that the length of the cable and the structure of the electrical network affect the throughput, causing a reduction of 65% when the cable length reaches 10 meters.

Selga et al. [125] work on the Medium Access Control (MAC) layer. They borrow a wireless sensor network MAC protocol called "Ripple Control" for PLC-based SMI networks. They assume a chain of meters ending at a concentrator and forming a star-like topology. The concentrator acts as a controller and aggregator. They argue that the best capacity can be achieved when a receiver initiates the connection.

Yu et al. [150] study the problem of the so-called silent node. When a base station (BS) polls all the metering nodes, it may fail to communicate with certain nodes due

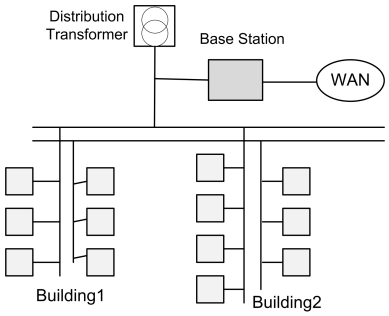


Figure 3.13: PLC-based SMI in Singapore

to environment noise. The paper proposes modifying the polling mechanism to resolve this issue. The system is modelled as a number of buildings (50-80 apartments per each) connected to the same distribution transformer, thus having the same BS (Fig. 3.13). In simple polling, where the silent node problem may occur, the BS polls all the meters in a cyclic order. Each meter responds immediately with its available data. If a certain meter does not respond, the Enhanced Polling (EP) mechanism is used. The BS re-polls the node in question after finishing the cycle. A third proposed mechanism, called Neighbor Relay Polling (NRP), which lets the BS attempt to communicate with the not-responding node through a neighboring node. Three metrics are used to evaluate the simulation: the data collection success rate, collection delay, and number of additional polls. In the simulation, 504 units are assumed. A complete cycle to poll all the meters is as high as 30 minutes.

PLC technology however faces a number of challenges: a noisy medium, high signal attenuation, and susceptibility to interference from nearby devices, leading to a high loss rate. The scalability of PLC-based SMI is also in question, for example, whether it can support frequent readings. PLC has already been deployed for broadband services in many countries. However, in certain countries such as Australia, Russia, and the United States, such deployments have been terminated. The reason is the high cost involved and the fact that other means of communication of higher stability and reliability are available.

3.2.2 Messaging over GSM Network

Short Message Service (SMS) has become a communication protocol that allows parties to exchange delay-tolerant short text messages. It is supported by different standards: namely, the Global System for Mobile communications (GSM), Code-Division Multiple

Access (CDMA2000) and Digital Advanced Mobile Phone Service (D-AMPS). The popularity and wide coverage of cellular networks have attracted researchers to consider the use of SMS service.

Tan et al. [138] propose an SMI system design that utilizes a GSM network. The system constitutes at the consumer site a digital meter with an RS232 interface and a GSM modem containing an SIM card dedicated for only SMSing; and at the energy provider site, an SMS gateway to send and receive messages. Measurements are reported once a month. An SMS message contains six digit KWh with one decimal point of energy consumption. The SIM card number acts as a unique number to identify a customer. To boost reliability, the meter stores the latest reading in an Electrically Erasable Programmable Read-Only Memory (EEPROM), and it keeps trying to send the SMS multiple times. Nevertheless, metrics are not identified for evaluating the reliability and strength of the system.

Abdollahi et al. [8] also suggest the use of GSM networks. Communication can be either one way or two way. In the uni-direction setup, meters send readings at predefined intervals and switch off otherwise to conserve energy. In the bidirectional setup, the energy provider can have more control over the meter but requires the meter to be active all the time. Measurements are reported once a month as Object Identification System (OBIS) codes. However, no evaluation of the system performance or comparison with other designs is provided.

The scalability and reliability of such a network however is questionable, especially under high load. Zerfos et al. [152] have analyzed real data taken from a real GSM network in India. The SMS delivery success rate was found to be 94.9%; 73.2% of the successfully delivered messages reached their destination within 10 seconds; about 5% of them required more than an hour and a half. Using SMS for SMI service will definitely increase the flow of messages tremendously. Meng et al. [102] provide analyses of latency and failure ratio under high load. For example, on a New Year's eve, the volume of SMSing increased eight times. Consequently, latency grew from several minutes to an hour. The failure rate shows an increase to 20% as well. All in all, SMS should be further investigated before being used for SMI. For example, can cellular networks support a messaging frequency of up to one message every 15 minutes? How reliable is the network in that case?

3.2.3 Telephone Lines

Telephone lines are desirable because they offer a highly reliable, relatively inexpensive, and simple-to-operate solution. An SMI system can use telephone lines for inbound, outbound, or bidirectional communication. The connection is initiated from the customer site in the

inbound mode, and initiated from the energy provider in the outbound mode. In the bidirectional mode, connection is initiated from either site, enabling more services such as sending out queries and collecting measurements. Utilizing telephone lines for this purpose is an old proposal ([89]), but it continues to interest developers such as COMETECH M2M [32] and as proposed recently in [77].

Lee et al. [89] describe a smart metering system that utilizes the public switched telephone network (PSTN). They describe the hardware of the two end points: the Remote Reading Unit (RRU) and Communication Front End (CFE). At the customer site, RRU is installed, where it can connect up to three meters, possibly of different kinds. At the utility company site, the CFE is installed. It consists of a regular computer and a modem. The RRU and CFE communicate with each other through the telephone network in both directions, allowing the RRU to send data frames, and the CFE to send commands. Measurement reporting can take place either on demand or periodically. The CFE collects the information sent by all the RRUs, and transmits it to processing and billing servers. Having two-way communication allows the utility company to reprogram the RRUs, for example, to change the reporting schedule. An RRU can store the measurements until they are successfully delivered to the CFE. NACK and timeout are used to learn about any failure. However, no evaluation or performance metrics was included.

Kim [77] provides a design description of a telephone line SMI system. In this system, the meter device comprises the following parts: an interface module to connect to the remote control center through a telephone line, a main control unit (MCU) to generate control signals that embed the Caller ID (CID), a CID decoder to decode the meter reading request signal and CID, and a memory to temporally store measurements. Meter data and control signals from the control center are transmitted in the form of dual tone multi-frequency signals (DTMF).

The availability of a telephone line at each meter is a requirement that cannot be always satisfied, especially in developing countries. However, telephone lines can be considered for distant and isolated locations, in which wireless coverage is missing.

3.2.4 Short Range Radio Frequency

Short range Radio frequency (RF) in this context refers to a low-power RF facility at the customer site. A number of technologies can be classified under RF: Bluetooth, WiFi, Zigbee, depending on the signal power and frequency band. It is unlikely that gas and water meters will share the same power line communications infrastructure because utility companies may not want to share their network infrastructure [23]. Koay et al. [83] propose

equipping electricity meters with Bluetooth modules to deliver the readings wirelessly to a nearby PC (or PDA) directly. Metering data is then forwarded through a dial-up connection to the energy provider or collected by a walking-by person. Meters transmit their data either periodically or whenever they are polled. Bluetooth, as a solution to SMI, is not plausible anymore today; however, it stays a viable solution in certain circumstances (e.g., at SMI early deployments).

Wesnarat and Tipsuwan's work [145] aims at networking water meters as a wireless sensor network. Because the meters feed on a battery, and because fusion (aggregation) is not possible, the problem here is how to arrange these meters such that power consumption is kept low, mainly by avoiding long packets. Meters form sub-trees with a base station being the root node. Every meter reports its measurement through other meters. The BS then sends all the received measurements from all the sensors to the final control station using SMS or GPRS. Spencer [132] claims that compression can also reduce the packet length due to two facts: (i) Data reported from different households tend to follow a certain probability distribution. (ii) Successive data from the same meter correlate. However the reduction is only three bits.

Zhu and Pecen [155] propose to let meters create a wireless mesh network with IEEE802.15.4 as the underlying technology and Zigbee standard for the upper-layer protocols. The authors claim that this combination of protocols and network setup guarantee real time collection of data, but no experimental validation is provided. Zigbee has received significant attention as a solution to SMI because the technology is already designed for low rate applications and consumes minimal energy, enabling a device to last for a number of years. It also supports a variety of strong routing protocols. However, it is worth noting a number of drawbacks. The bandwidth is very low (20 kbps at 868 GHz and 250 kbps at 2.4 GHz) [88]. With an increase of the number of nodes, interference increases dramatically, making its connections and routing paths unstable and incuring high delay. Thus, the technology is barely reliable and scalable for SMI.

RF mesh networks are the leading solution in SMI in north America [90]. Mesh networks are easy and quick to deploy, yet scalable to millions of meters [5], and reliable by providing redundant communication paths. Mesh networks are self-configuring and self-healing. However, the downside is that they may not be able to support highly frequent meter readings, and that they cause high latency.

3.2.5 Third Generation (3G) Networks

Technologies such as PLC, GSM, telephone, and RF have not been completely satisfactory. To support more services, SMI demands investigation into more sophisticated technologies with higher bandwidth. 3G wireless technologies are getting more attention [90] [73] today for their flexibility, easiness and high speed of deployment, cost-effectiveness, scalability, and business needs.

Various 3G wireless technologies have been introduced to the community, some of which had actual implementations. Long Term Evolution (LTE), High Speed Packet Access (HSPA) and IEEE 802.16 (known as Worldwide Interoperability for Microwave Access or WiMAX) comply with International Mobile Telecommunication (IMT-2000). WiMAX was added in 2007, giving it significantly larger global popularity. The solutions engineered tend to be similar since the goal and the underlying technical solutions are fundamentally the same: wideband transmission, high-order modulation, fast scheduling, advanced receivers, and multi-carrier. What will make either of the technologies more popular than the others will be determined by industry. Currently, WiMAX and LTE are atop the list, and with regards to SMI both of them are good candidates.

3.2.6 Link Access and Routing

The network layer is well coupled with the underlying layers. In a multi-hop setup, the traditional problems associated with a wireless channel (e.g., interference and fading) may affect meter-to-meter communication, especially in areas with highly dense meters. Some isolated meters (e.g., in rural areas) need to have repeaters to connect to the rest of the SMI network. At the Medium Access Control (MAC) level, an energy efficient protocol is required. TDMA-based protocols are more energy efficient for flat network architecture than Carrier sense multi-access (CSMA). However, if the SMI network is clustered, then more work should be done to accommodate inter-cluster communication and to adapt the intra-cluster MAC in terms of the number of nodes involved and the MAC parameters such as frame length and slot assignment.

Recent ongoing work by IETF has targeted smart metering infrastructure and similar applications. The Routing Over Low-power and Lossy networks (ROLL) group ([39]) provides a comprehensive discussion of the routing requirements of Urban Low-Power and Lossy Networks (U-LLNs), which applies to smart meters. The network architecture considered is a mesh network. All nodes (meters, actuators and metreological sensors) can provide measurements as well as perform routing. Routes lead to a sink or a gateway node

that is connected to the Internet. Levis et al. [91] evaluate the suitability of standard protocols such as the Routing Information Protocol (RIP), Open Shortest Path First (OSPF) and Ad hoc On-Demand Distance Vector Routing (AODV) to act as the routing protocol for U-LLNs. Winter et al. [148] specify a Routing Protocol for Low Power and Lossy Networks (RPL). The protocol forms a directed acyclic graph. Edges form paths that are oriented toward and terminate at a root node, which could be a sink or a gateway to the Internet. Another network architecture option would be to let meters connect directly to a third-generation base station. The issues to be investigated in this case are purely related to lower layers' functionalities (MAC and data link). To achieve scalability, scheduling the meters should be considered. If multiple simultaneous transmissions are allowed, low level interference must be taken into account.

Chapter 4

Problem Description

As stated in Chapter 1, the objective of this work is to study and enhance the performance of TCP in an SMI. This chapter identifies the shortcomings of the TCP in achieving a scalable and efficient smart metering infrastructure. Experiment results and mathematical evaluation are presented here to support the claims made in Chapter 1 that TCP's congestion control is ineffective in an SMI.

4.1 System Model and Assumptions

This section deals with the SMI communication model and traffic assumptions applied in this chapter and in the rest of this thesis.

4.1.1 SMI Communication Model

Smart Metering Infrastructure constitutes the deployment of a large number of meters installed at fixed locations. The meters form a mesh topology by means of wireless communication or a Power Line Carrier (PLC) [55]. Isolated meters (e.g., in rural areas) connect to the rest of the network through repeaters that boost their signals. Thus, the meters act as sources of data and as routers. In every region, there exists a gateway to interconnect the meters with the Wide Area Network (WAN). The gateways are called advanced metering regional collectors or concentrators (abbreviated as RC), typically installed on poles at fixed locations [115] [39] [109]. With this topology, data packets route

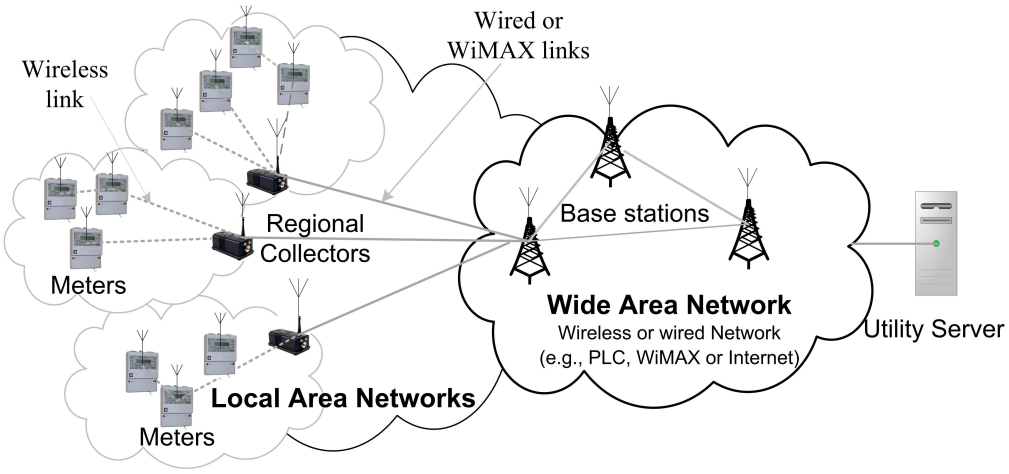


Figure 4.1: Smart Metering Communication Architecture

over multiple hops through the meters to reach the RC and then get forwarded through the WAN to the utility server [90]. Figure 4.4 demonstrates this architecture.

Smart meters and sensors number in the hundreds of thousands to millions, whereas the number of regional collectors remains small. These devices are assumed to fully support the TCP/IP communication stack as defined by the Device Language Message Specification/COmpanion Specification for Energy Metering (DLMS/COSEM) standard [33]. The network is required to operate in both directions between meters and utility servers. The bulk of the traffic, however, will flow from the meters to the utility server [42]. Although wireless and PLC links are employed, 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.

In short, smart meters and sensors are TCP/IP devices distributed in regions, forming local area networks. Each region routes traffic through a regional collector. For reliability, the TCP protocol is used. Every meter establishes a direct TCP connection with the utility server. Figure 4.2 shows a simplified communication model. Direct TCP connections are depicted by the dashed lines.

4.1.2 Traffic Assumptions

The following summarizes the assumptions used in this research about the smart metering infrastructure [41] [71].

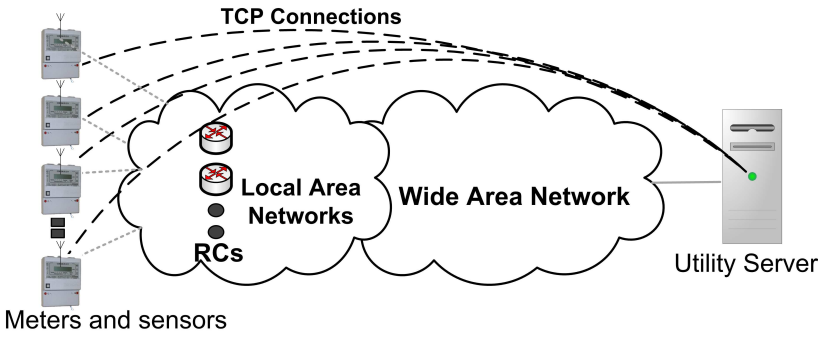


Figure 4.2: Simplified SMI Communication Model

- 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.
- 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.
- Data aggregation is not applicable. Because the utility provider is interested in each measurement, rather than some statistical summary of the data, data aggregation techniques, such as mean, median, maximum and minimum [45], cannot be employed here. Meeting this requirement is essential for major applications such as Real-time pricing and demand side management; that is for the utility provider to be able to calculate and present the cost to every customer and to control a customer's power consumption as needed [90] [41] [70].
- Latency of end-to-end packet delivery is tolerable within a time window D. 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, depicted in Fig. 4.3.
- The meters are already configured with a routing protocol that enables them to route data to the utility server [91] [148].

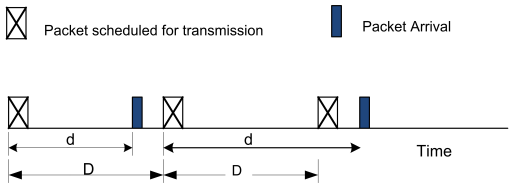


Figure 4.3: Meter's Packet Schedule

- Data packets vary in size from tens of bytes to a few hundreds of bytes, depending on the information carried and security system employed [41] [37]. Typical fields include meter id, data and time stamp. DLMS/COSEM provides standard codes for a meter's data items (called the Object Identification System or OBIS). These codes are used for configuration and for obtaining information about the behaviour and status of a meter.

4.2 Ineffectiveness of TCP Congestion Control

The TCP congestion control mechanism is an essential functionality used to keep a network running efficiently and is thought to be the one behind the success of the Internet today. Without proper congestion control, the retransmission of lost data together with the demands of regular ongoing transmission would slow down the network and cause more packet losses.

The common behavior of the transport control protocols 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 [14] and others like it, for example, 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. This decrease is achieved by setting the congestion window to its initial size (typically one or two segments).
- If the source receives three duplicate ACKs, indicating a missing packet, it halves its sending rate by halving the congestion window size.

With SMI traffic, the typical mechanism of TCP becomes ineffective. The high volume of traffic in SMI does not come from a single source; rather it comes from a large set of sources, each transmitting at a low data rate (*e.g.,* one packet per round trip time). The TCP congestion window always stays at its minimum value of one or two. As such, reducing the sending rate upon congestion in the network is not viable.

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, and so keeping the congestion window size at one MSS (Maximum Segment Size). If congestion occurs in the network, the one 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.

Therefore, the SMI traffic will be highly aggressive as there will be hundreds of thousands of TCP connections transmitting at a low flat rate of around one segment/RTT (round trip time). The lack of an effective congestion control mechanism leads to two major problems, namely, congestion collapse and unfairness [51]. Congestion collapse occurs when the network is busy transmitting packets that will be dropped at some congested router before reaching the final destination. That is, even though SMI traffic may suffer packet drops, the total traffic rate stays unchanged. Unfairness occurs when other competing TCP-friendly flows suffer bandwidth starvation because of the non-rate-reducible SMI traffic.

4.3 Mathematical Evaluation of TCP Congestion Control

Further to the above discussion of why TCP is ineffective in handling SMI traffic, this section provides an analytical approach to explain the situation. The analysis is performed according to the network diagram demonstrated in Figures 4.4 and 4.5. The diagram topology has been modified from [51], which was introduced in the literature by Floyd and Fall to illustrate the negative impact of having TCP-unfriendly traffic in a network.

In the same manner, the diagram considers a smart metering infrastructure with a bottleneck link shared between smart meters and certain external traffic.The link, $R_1 - R_2$, represents a bottleneck with a service rate of μ packet/sec. The router R_1 has a buffering capacity of B packets. A number of meters, n, transmit data to the utility server at a rate

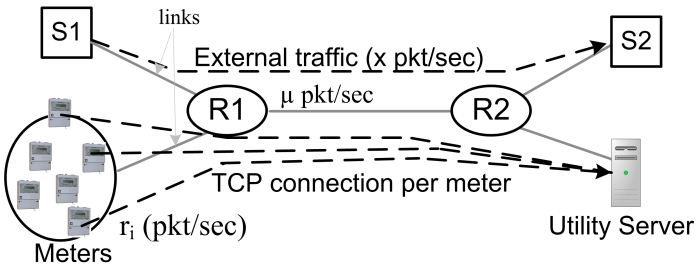


Figure 4.4: Smart Metering Network Architecture

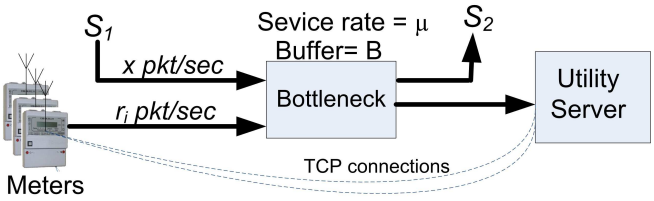


Figure 4.5: Smart Metering Model

of r_i pkt/sec each. An external traffic (UDP or TCP) from S_1 to S_2 is transmitted at rate x packet/sec. A propagation delay τ is assumed to be constant and is equal for both flows.

Assuming all meters are configured to transmit data at the same rate, the total traffic then is the summation:

$$\sum^{n} r_i = nr_i.$$

The maximum number of packets that can be in the network are those not yet unacknowledged by the utility server, that is, the total of the buffered packets and the ones still in transit. This number can be calculated as:

$$\mu T + B, \text{ where } T \text{ is the round trip time}$$
$$(T = \tau + \tfrac{1}{\mu})$$

To have an overflow-free traffic flow, the following inequality must be satisfied.

$$nr_iT + xT \leq \mu T + B,$$

where μT represents the packets in transit, B the buffer space, and the left hand side represents the number of packets generated during T seconds, which is limited by the

window size. That is,

$$\text{Source's sending rate} = \frac{W}{T}$$

To keep traffic below the buffer capacity, the transport protocol slows down the transmission speed at the sources (*i.e.,* r_i and S_1) by reducing the transmission window size. Nonetheless, because r_i corresponds to a fixed window size of typically one or two, the meters' transmission rate is already at a minimum; thus, it cannot be lowered any more.

With regard to competing traffic, if it is over a TCP connection, then it drops its transmission rate, resulting in an unfair drop of its traffic throughput. If the traffic is a UDP one, then it continues to transmit at the same speed. This situation leads to an unstable condition because more and more packets will be dropped at the bottleneck due to overflow. Since meters use TCP, they will keep retransmitting lost packets, which again will make the UDP source lose more packets. The result is excessive retransmission from the meter side and extreme loss of UDP data.

If meters have higher amounts of data, does that make the situation better? In fact, congestion control stays ineffective. As stated above, this is equivalent to replacing one large traffic source with multiple TCP sub-connections, e.g., N, each roughly receiving $\frac{1}{N}$th of the original throughput. A single packet drop causes the corresponding connection to drop its sending rate to 50% of its current speed. Thus, in the one-TCP connection case, the new sending rate is half of the original. However, in the equivalent N sub-connections, a packet drop leads only one sub-connection to reduce its sending rate. The total new rate becomes as follows:

$$50\% * \frac{1}{N} + (N-1) * \frac{1}{N} = \frac{2N-1}{2N}$$

Thus, for example, with 100 sub-connections, the new rate is 99.5%, which means the reduction is too little to resolve congestion.

Furthermore, in the case of one TCP connection, the congestion window size is increased by one segment every round trip time. However, with N sub-connections, each increasing its window by one segment in a round trip time, the congestion windows collectively grow by a total of N segments every round trip time. The case with SMI traffic flows is more aggressive as there will be hundreds of thousands of TCP connections.

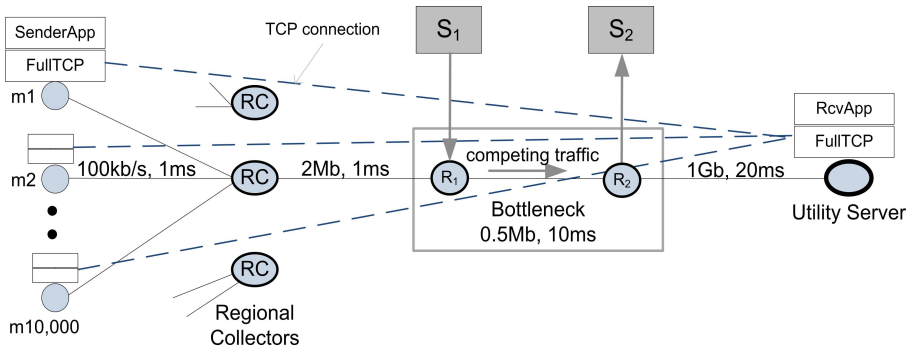


Figure 4.6: NS2 Simulation Setup

4.4 Experimental Evaluation of TCP

By means of the network simulator ns2 [101], the smart metering scenario presented in Fig. 4.4 is implemented with every meter establishing a direct TCP connection with the utility server. We refer to this type of TCP setup as *one-hop TCP*. The goal is to show that in this one-hop TCP configuration, the congestion control mechanism is not able to achieve its desired objectives, namely, to reduce the loss rate in a network and to let traffic flows share a link fairly.

4.4.1 Simulation Setup

Figure 4.6 illustrates the experiment setup in ns2. Meters are implemented as IP devices that send small data reports to the utility server periodically. The meters, $m1, .., m10,000$, connect through a single regional collector (RC). The RC here acts as a router and has no TCP-level role. The router R_1 has a buffering capacity of 20 packets configured as DropTail. The utility server gathers all the reports and returns acknowledgement packets to the meters. The bottleneck bandwidth $(R_1 - R_2)$ is shared among the meters and certain competing traffic (UDP or TCP) from S_1. All the links are simulated as wired with high capacities to be able to assess the effectiveness of TCP congestion control since packet losses are indeed due to congestion at the bottleneck, not due to media issues. In reality, meters would be equipped with a wireless facility, which would introduce other wireless-related causes for packet loss, e.g., interference and collision. Table 4.1 summarizes the simulation setup parameters.

Table 4.1: TCP Experiment Parameters

Number of meters	10,000
Meter's data rate	1 packet/50 sec.
Packet size	200 bytes
(S1 $\rightarrow$ S2) traffic	Random, up to 250 kbps
Bottleneck buffer	20 packets, DropTail
Simulation duration	1200 seconds

4.4.2 Simulation Results and Discussion

To evaluate the effectiveness of TCP congestion control in this one-hop TCP scenario, a number of performance metrics are observed, namely, throughput, loss rate and delay. To differentiate between TCP and UDP traffic, the term retransmission rate is used here for the former and loss rate for the latter. The distinction is necessary because a TCP source retransmits lost packets, while a UDP one does not.

Naturally, in a network with a functional congestion control, it is expected that the bottleneck link, $R_1 - R_2$, would be shared somewhat equally between the set of meters and the external TCP traffic since they produce data at a similar rate. Since both the meters and the source S_1 use TCP, they adjust their speeds according to the available bandwidth in order to minimize the network's loss rate. If the external traffic is UDP, it is expected that only the meters would reduce their transmission speed. Nevertheless, the results are different from expected in both cases.

The results show that the meters are impacted. Figure 4.7 shows the impact in terms of a high retransmission rate. The Retransmission rate is calculated in the following equation as a percentage of retransmitted packets with respect to the number of originally transmitted ones.

$$\text{Percentage of retransmission} = \frac{Number\ of\ retransmitted\ packets}{Number\ of\ originally\ transmitted\ packets} * 100\%$$

Because meters use TCP, even when the network gets congested and packets get dropped, a meter will continue retransmitting those lost packets according to its congestion window. Since a meter's traffic is very low (with a large gap between consecutive transmissions), the meter cannot reduce the transmission rate any further. Figure 4.8 shows that a meter's congestion window stays at its minimum value regardless of the congestion situation of the network.

As Fig. 4.7 shows, the retransmission percentage is between 120% to 140%. According

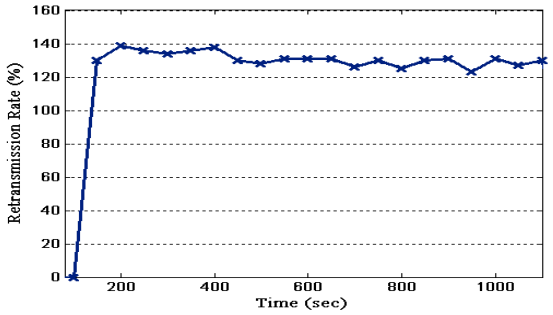


Figure 4.7: Meter Retransmission Percentage

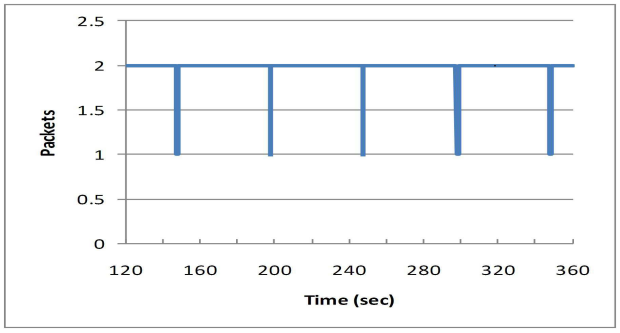


Figure 4.8: Meter Congestion Window

to the equation, a retransmission rate of 100% means a packet is sent twice to get to the collection node. In other words, for meters to deliver data reliably, twice the amount of meter traffic must be accommodated. The impact here is twofold: First, the high retransmission rate leads to significant energy cost for the meters. Second, meter traffic becomes similar to UDP traffic in that it keeps penetrating the network and overtaking the highest possible bandwidth. The TCP congestion control mechanism fails to enforce fair bandwidth sharing with other flows.

The lack of an effective congestion control and the impact of the smart metering traffic on other traffic in the network can be noticed in the rise of packet loss rate and low throughput of competing UDP and TCP flows (Fig. 4.9 and Fig. 4.10). For example, the loss rate of the UDP traffic in Fig. 4.9 reaches 40% of the traffic. TCP and UDP are both affected, giving a low throughput, hardly reaching 100 Kbps out of the link capacity of 500

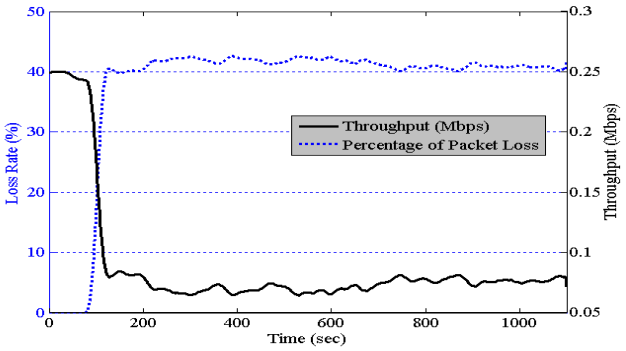


Figure 4.9: Throughput and Loss Rate of Competing UDP Traffic

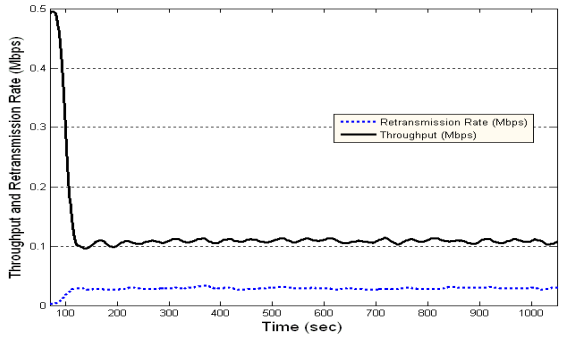


Figure 4.10: Throughput and Retransmission Rate of Competing TCP Traffic

Kbps. The following equation shows how packet loss is calculated as a percentage.

$$\text{Percentage of packet loss} = \frac{Number\ of\ dropped\ packets}{Number\ of\ transmitted\ packets\ from\ source} * 100\%$$

As for delay, packets are affected since a packet has to be retransmitted multiple times to reach the destination. However, delay here is calculated differently. Percentage of delayed reports is the metric used and refers to the percentage of packets that do not get delivered within a certain time limit. The deadline is assumed to be the time when the following report is due for transmission. This notion of delayed reports is used for comparative purposes with the proposed Split- and Aggregated-TCP scheme in the next chapter. Figure 4.11 shows the percentage of such reports that are delayed beyond a 50-second deadline.

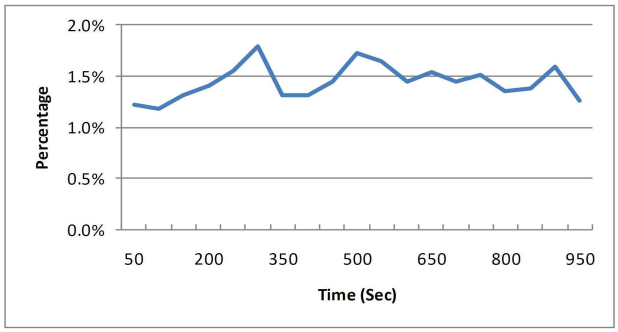


Figure 4.11: Percentage of Delayed Reports

4.5 Summary

This chapter has identified the shortcomings of TCP in handling the smart metering infrastructure traffic. Supported by mathematical explanations and experimental results, it is evident that the degraded performance is due to the lack of an effective TCP congestion control mechanism. Therefore, there is a need to consider a new approach at the transport level to mitigate such performance issues, which is the topic of the next chapter.

The same simulation scenario will be visited in the next chapter but with a slight modification. A new service that splits and aggregates TCP connections at regional collectors is added. With that modification, new simulation results will be compared with the above ones to observe the improvements gained.

Chapter 5

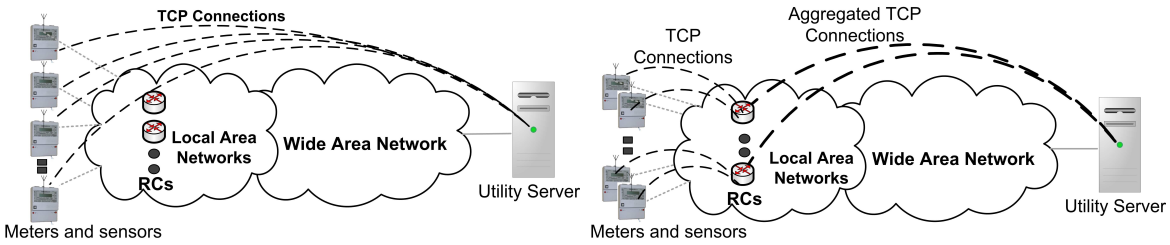
Proposed Split- and Aggregated-TCP (SA-TCP) Scheme

A one-hop TCP scheme results in serious performance problems as discussed in Chapter 4, shown in terms of elevated retransmission rates and degraded throughput for the meters and for competing applications. This chapter presents our solution: a TCP-based scheme under the name *Split and Aggregated TCP*. SA-TCP greatly improves the TCP performance since the congestion control mechanism is reworked to function with its full range of congestion window dynamics. Comparison with the one-hop TCP scheme is provided in this chapter by applying certain modifications and performing the same experiments described in the previous chapter. Furthermore, a critique of SA-TCP is given, highlighting the gains as well as the scheme's downside.

5.1 Split- and Aggregated-TCP Scheme(SA-TCP)

The proposed SA-TCP scheme makes TCP congestion control effective and so enhances performance. SA-TCP introduces the concept of aggregation at the transport layer. The regional collectors (RC) are upgraded to operate at the TCP layer rather than being limited to the IP layer. Consequently, they take on the task of splitting and aggregating TCP connections. We call those RCs *SA-TCP Aggregators*.

Figure 5.1 illustrates this modification to the TCP scheme. Each meter initiates a TCP connection with an SA-TCP aggregator in its region, over which data packets are reliably transmitted. An SA-TCP aggregator, in turn, creates an aggregated TCP connection with



(a) One-hop TCP Scheme (b) SA-TCP Scheme

Figure 5.1: SMI TCP Architecture: One-hop Vs. SA-TCP

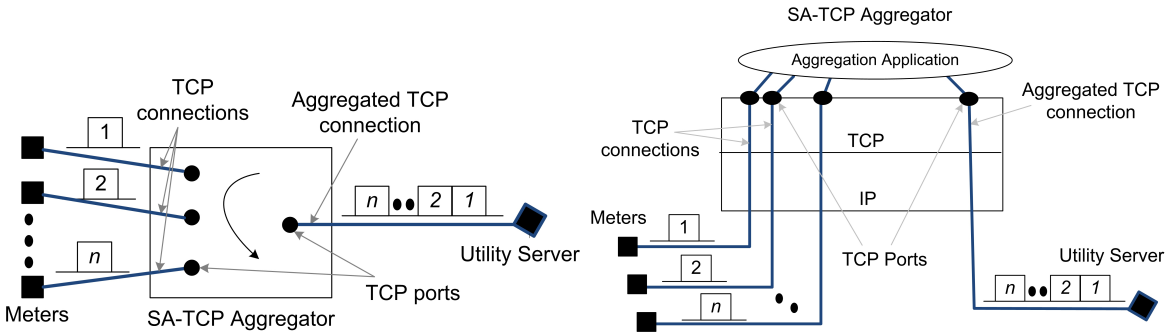
the utility server and forwards the data packets over this unified TCP connection utility server. In other words, the TCP connections between the meters and the utility server are no longer one-hop, but rather, two-hop connections.

More precisely, as Fig. 5.2b shows, every n meters establish n TCP connections with an SA-TCP aggregator. The meters' data packets are received at the application layer and get forwarded by the aggregation application over a single TCP connection with the utility server. No change occurs to the TCP Protocol mechanisms at 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 (Fig. 5.2a). For example, a priority-based or time-based scheduling can be applied to enable urgent data (*e.g.,* alerts) to be delivered quickly. This work, however, does not address such policies. It assumes that statistical multiplexing is performed to immediately forward an arriving packet.

The proposed scheme makes TCP congestion control effective. 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. As a result, the packet loss rate is reduced, resulting in better link utilization.

At first glance, our scheme seems to have some resemblance to Indirect-TCP (I-TCP) [17] in splitting 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 may also seem similar to [27]. This work [27], however, was introduced for General Packet Radio Service (GPRS), to control the bandwidth distribution among cellular devices and to improve link utilization. The aggregation there is in the form of unifying the TCP state information (*e.g.,* round trip time and congestion window size) of a set of TCP connections that a single mobile device



(a) Split and Aggregation Mechanism (b) Layered Architecture

Figure 5.2: A Regional Collector Acting as an SA-TCP Aggregator

initiates.

5.2 SA-TCP vs. One-hop TCP Experiment

To demonstrate the gain achieved by implementing the SA-TCP scheme, a simulation similar to that in Section 4.4 is performed again, but with the inclusion of an SA-TCP aggregator. All the network parameters are kept the same so that performance can be compared to the case of one-hop TCP. The performance comparison is in terms of throughput, loss rate and delay for both the meters' and the competing source's data.

5.2.1 Simulation Setup

Figure 5.3 and Table 5.1 show the simulation setup in ns-2 and the experiment parameters, respectively. All the 10,000 meters connect to a single RC over separate TCP connections. The RC operates as an SA-TCP aggregator, so it connects with the utility server on a separate TCP connection. To ensure that all packet losses are due to congestion at the bottleneck ($R_1 - R_2$), the RC's buffer capacity is made infinity and all the links are set up as wired. The bottleneck bandwidth is shared between the meters and the competing traffic source, S_1.

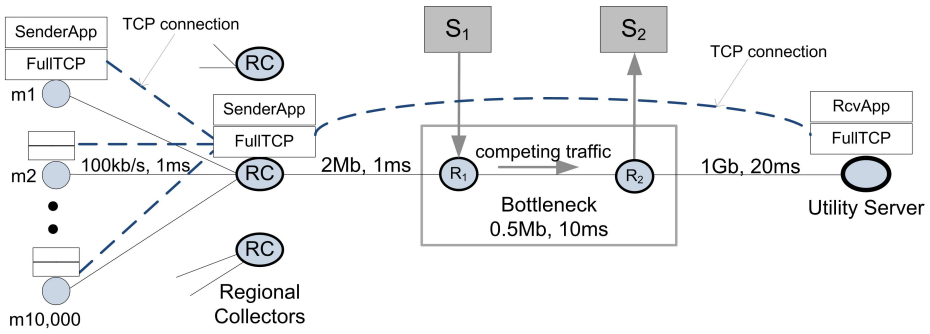


Figure 5.3: NS2 Simulation Setup

Table 5.1: SA-TCP Experiment Parameters

Number of meters	10,000
Meter's data rate	1 packet/50 sec.
Packet size	200 bytes
(S1 → S2) traffic	Random, up to 250 kbps
Bottleneck buffer	20 packets, DropTail
RC Buffer	unlimited
Simulation duration	1200 seconds

5.2.2 Simulation Results and Discussion

The simulation results are compared to those obtained from the one-hop TCP scheme (Section 4.4). The results compare the following performance metrics: throughput, packet loss rate and delay. Packet loss rate is calculated as a percentage of dropped packets with respect to the total number of transmitted packets, including retransmitted ones in the case of TCP traffic. Thus, packet loss rate is used here for UDP traffic, but it is expressed as a retransmission rate for TCP traffic.

The consequence of having one-hop TCP connections is a high retransmission rate, reaching up to 120% of the original traffic. With SA-TCP, the retransmission rate drops to 1%, as shown in Fig. 5.4, a significant enhancement. Having enough data, the SA-TCP aggregator is able to increase the transmission rate if the bandwidth allows and to decrease the transmission rate if congestion occurs. It is also worth noting that retransmission is performed at the SA-TCP aggregators rather than at the meters, which is a second

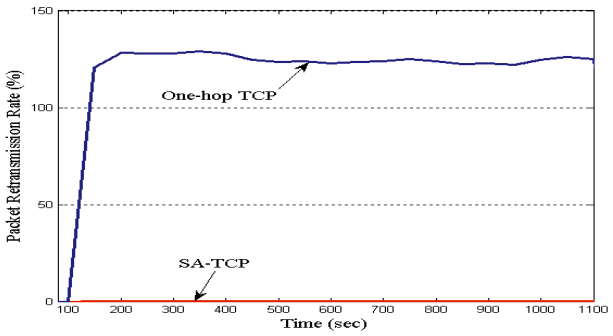


Figure 5.4: Retransmission Rate of Meters' Packets

advantage, especially if meters connect wirelessly.

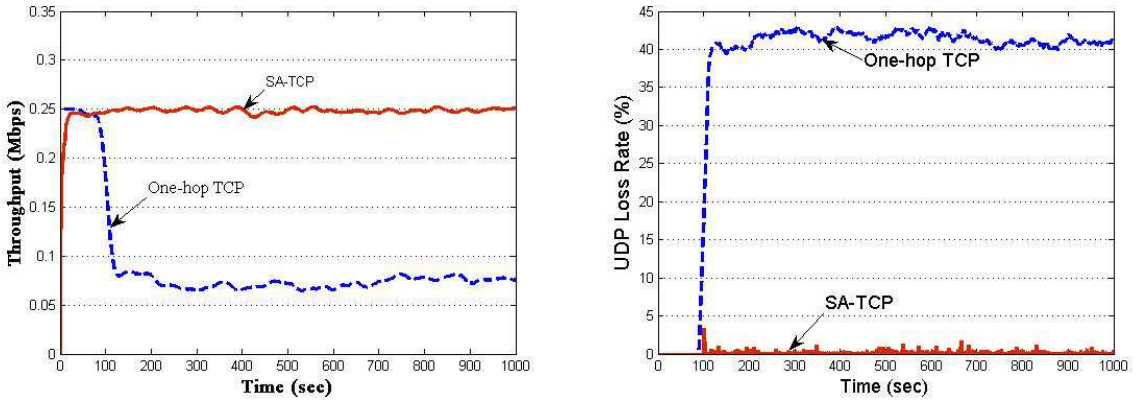
The second problem with One-hop TCP is the degraded throughput. Figure 5.5 shows the gain achieved by SA-TCP. The throughput of the competing UDP traffic ($S_1 - S_2$) has risen from 100 kbps to 250 kbps, with a packet-loss reduction from 40% to only 1%. This drop indicates that the SA-TCP aggregator has successfully reduced its speed to share the link with competing applications.

Fairness is also successfully enforced between the smart metering system and competing traffic. Fig. 5.6 shows the case of competing TCP traffic. The congestion window size of both the SA-TCP aggregator and of the external TCP traffic is shown to move into the same range. This is possible because with enough data, the SA-TCP aggregator is able to enlarge the congestion window and work at the different phases of TCP.

However, the SA-TCP aggregator, being a store-and-forward point, causes a slight increase in delay. Figure 5.7 shows the latency as a percentage of those reports arriving after the next report is due to be transmitted (*i.e.,* 50 seconds in this experiment). The percentage rises from around 1% to as high as 34%. The increase happens when there is high congestion in the network forcing the SA-TCP aggregator to shrink its congestion window size, and so it enqueues the arriving packets for a longer time.

5.3 Advantages of SA-TCP

Although the focus has been on resolving the problem of congestion control ineffectiveness, there are other serious issues that SA-TCP can tackle. Below is a discussion of such issues and of how SA-TCP can be of benefit.



(a) Traffic Throughput

(b) Loss Rate

Figure 5.5: Competing UDP Traffic Performance Comparison

1- Data Segment Size: SMI traffic is typically carried in small segments. From a TCP perspective, three unwanted issues result:

- With the TCP/IP headers being 40 bytes, overhead can be as high as 50 percent of the packet length.
- Small segments contribute to latency at the sender side because of TCP's use of Nagle's algorithm [108]. This algorithm is meant to resolve the inefficiency due to small segments; however, in smart meters, it worsens delay enormously.
- The receiver also contributes to delay because of TCP's delayed ACK mechanism [20], whose original purpose was to lessen the number of transmitted ACKs. Instead, it adds to the latency of SMI packet delivery since the number of arrived packets to be ACKed is infrequent. Delayed ACK and Nagle's algorithm may add up to 500 msec in delay. Furthermore, this mechanism along with the SMI's low data rate prevents the sender from taking advantage of triple duplicate ACKs to shorten retransmission delay, leaving the TCP congestion control algorithm to use only timeout.

In SA-TCP, since the metering data are destined for the same utility server, it is possible to combine segments and compress headers, increasing efficiency and bandwidth utilization.

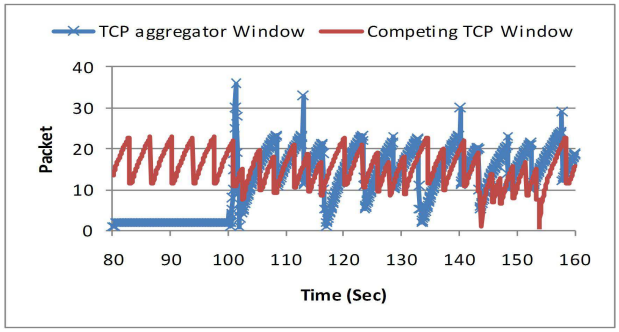


Figure 5.6: Congestion Window Size

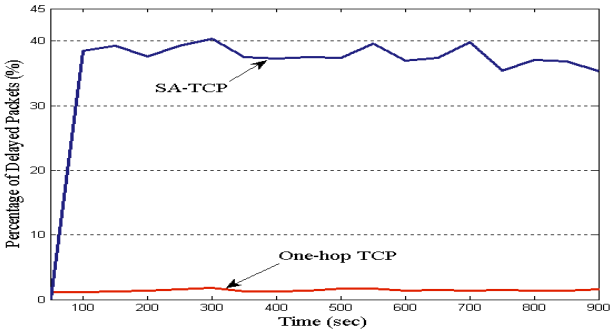


Figure 5.7: Packet Delay as a Percentage

2- Impact of combining different technologies: As SMI develops, it is expected to have a heterogeneity of communication technologies (*e.g.,* PLC and wireless) forming networks with mismatched bandwidths connected to one another. This situation increases the possibility of congestion. Additionally, it disrupts TCP's congestion control because it introduces high variability of certain parameters, such as round trip times and bit error rates. This variability causes false timeouts and hence unnecessary retransmissions. SA-TCP separates a region's network from the wide area network. Thus, variability of such TCP parameters is minimized. Another benefit is that retransmission takes place only on the side where loss occurs. If loss occurs in the WAN, then retransmission is performed by SA-TCP aggregators, rather than by meters.

3- Fairness with competing TCP flows: Inability to force congestion control properly

is a serious problem that leads to unfairness. To clarify the severity of SMI traffic, it can be compared to other types of traffic flow at transport level in the following way. TCP-friendly traffic is controllable through congestion control. If the transmission rate becomes high and the public network gets disrupted, the congestion mechanism lets the congestion window shrink and thus traffic speed drops. As for UDP flows, there is no congestion mechanism, but when UDP packets exceed the rate that the network can handle, the packets get dropped and never get retransmitted. Nevertheless, unlike TCP-friendly and UDP traffic, SMI traffic cannot be made slower, and if packets get dropped they will have to be retransmitted regardless of the network congestion condition. That retransmission leads to even more traffic and to unstable condition. In other words, even though SMI uses TCP connections, the result is still harmful to the network. In SA-TCP, the aggregator is able to adjust its congestion window size to achieve fairness with other flows.

5.4 Drawbacks of SA-TCP

Drawbacks should be noted too. One consequence of SA-TCP is that end-to-end semantics are not maintained. The end points do not know when data is received at the other party. An application-level mechanism may be required to provide error recovery and acknowledgements. This situation is similar to certain TCP-based applications that have built-in reliability mechanisms, *e.g.,* , FTP.

Another consequence is the inability of maintaining end-to-end secure data transfer because having an intermediate node to redirect transport segments necessitates unwrapping packets and modifying or rearranging the contents. However, because those intermediate nodes (TCP aggregators) are the property of the utility company, security can still be preserved.

Finally, the TCP aggregator must maintain enough buffering space for the packets received from meters, which introduces another source of delay to the data, especially when the TCP aggregator is forced to slow down transmission. In our experiments (Fig. 5.3), we found that with UDP competing traffic, the percentage of delayed meter reports may be as high as 40% of the total transmitted reports. For this purpose, an application-layer signaling protocol may be employed to reduce the report-generation rate. It is advantageous to have less frequent reports rather than expired reports.

5.5 Summary

This chapter has described the proposed SA-TCP scheme. Through ns-2 simulations, it has been shown that the scheme is successful in improving TCP performance. Making congestion control effective has been the main goal; however, as discussed above, the proposed scheme makes improvements in other ways too; for example, it improves the efficiency of data segment size, improves fairness with competing flows, and isolates wireless-related issues from affecting TCP. A slight increase in packet delay due to queuing in RCs has been observed too. Therefore, there is a need to optimize the scheme so as to ensure performance constraints are met. The next chapter introduces a mathematical model for SA-TCP. The model captures the performance of SA-TCP under various setup parameters. Through the model, optimization is achieved.

Chapter 6

SA-TCP Analytical Model

The objective of mathematically modelling SA-TCP is to conduct performance analysis and ultimately to optimize the SA-TCP architecture design, for example, by deciding on the optimal number of SA-TCP aggregators and other parameters for satisfactory performance results. The analytical model presented in this chapter considers various components at different communication layers, mainly, application and transport behavior. It also takes into account the network characteristics, such as the capacity, propagation delay, queuing, and number of smart meters and SA-TCP aggregators. The model allows one to reproduce the actual behavior of the meter traffic in SMI, depicted in terms of the following performance metrics: packet loss rate, offered load, and packet end-to-end delay.

- *Packet loss rate* is the probability of packets getting dropped due to buffer overflow in the network.
- *Offered load* is calculated as the rate at which data segments are produced by a TCP source. *Throughput* is the portion of those segments that are successfully delivered.
- *Packet end-to-end delay* is the time in seconds a packet takes to arrive at its destination.

6.1 Modelling Approach

Figure 6.1 gives a big picture of the SA-TCP model. SMI traffic passes through two stages, the first recognized by the first-hop TCP connection between the meters and the SA-TCP

aggregators, and the second by the aggregated TCP connections between the SA-TCP aggregators and the utility server. In each stage, the fixed-point approximation method [105] [25] [107] is applied. This method corresponds to the operating point of the network expressed in terms of the average offered load rate, loss rate, and delay. The method is based on the idea of modelling the sources and the network separately and then finding a common solution that satisfies both models. In brief, the procedure is as follows:

1. The traffic load offered by the sources is modeled as a function of the network parameters: probability of loss and delay, *i.e.,* $E[W] = f(p, T_r)$
2. These network parameters are modeled as functions of the sources' offered traffic load, *i.e.,* $(P, T_r) = g(E[w])$

   where $E[W]$ is the average congestion window size, P is the probability of packet loss, T_r is the mean RTT, and $f()$ and $g()$ denote functional relationships.
3. Starting with an initial value for the sources' offered load, the same network parameters 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 continued until no further change in the performance parameters occurs. The solution of this last iteration corresponds to the fixed-point solution.

In the first stage, meters are grouped into regions, with each region connecting to the same SA-TCP aggregator. The model of a single region is demonstrated in Fig. 6.2. The aggregated traffic from the meters in a region goes through a network bottleneck modelled as a queuing model. The meter's offered load, packet loss rate and packet delay are determined by each meter's application characteristics, TCP congestion window dynamics, and the network bottleneck parameters.

It is assumed here that each source produces data segments whereby the arrival rate follows Poisson distribution at rate λ. Consequently, the total rate seen by the network bottleneck is $\lambda_t = \sum_{i=1}^{n} \lambda_i$ since the summation of independent Poisson processes is Poisson [80]. The combined behaviour of the application and TCP congestion mechanism is represented as a continuous-time Markovian model. Individual meters may have different application characteristics, for example, an active or inactive time duration or propagation delay to the SA-TCP aggregator. The network is modelled as a queue with a certain buffer size and service rate. The input rate to the queue is λ_t. The queue service rate, however, is calculated from the second stage, which is the rate at which the SA-TCP aggregator is

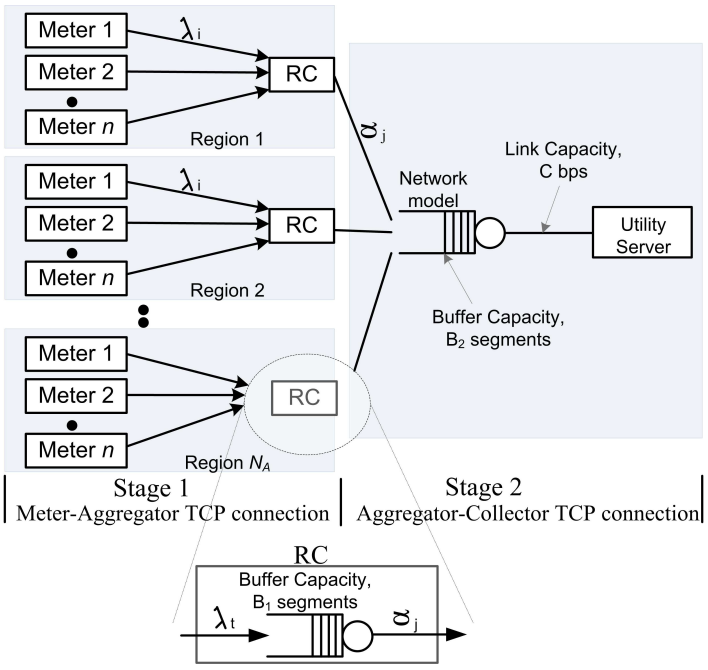


Figure 6.1: SA-TCP model Architecture

able to serve the received segments. Packets entering the queue are subject to a certain dropping probability and queuing delay. The Markovian model (mimicking the congestion window dynamics) takes those two measures into account to recalculate the segment generation rate accordingly. Our goal is to find the network operating point, so this interaction between the meters and the network is continued iteratively until a fixed point is reached, as explained in the procedure above. The fixed point is determined when there is no more change in the input rate λ_t. Thereby, we get the approximate average offered load (λ_t) and probability of loss (P_m), and end-to-end delay (d_1).

In the second TCP stage, each region's data is received at the corresponding SA-TCP aggregator, then sent collectively to the utility server through a wide area network (WAN) link. The same modelling approach is applied: the SA-TCP aggregator's segment generation mechanism is represented as a Markov chain model (mimicking the TCP congestion window dynamics) and the WAN bottleneck as a queue model. Iterating between the aggregator's segment generation process and the queue model is performed until a fixed-point solution is reached. At that point, the necessary parameters are obtained, including the second stage's loss rate P_a, delay (d_2) and the SA-TCP aggregators' offered load (γ_t). The

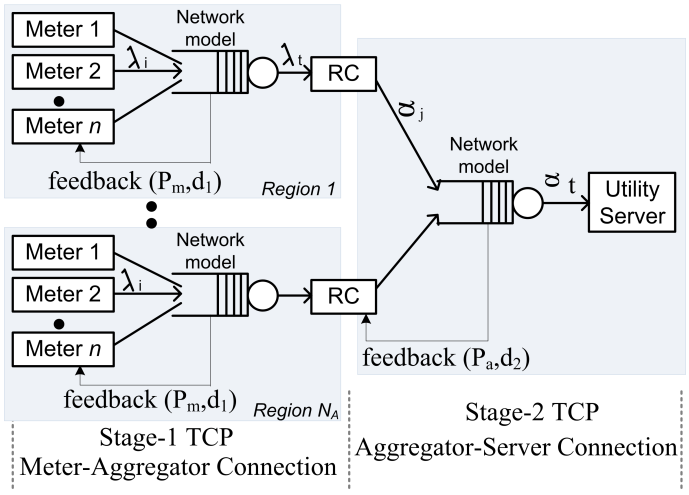


Figure 6.2: Source - Network Interaction

obtained offered load is used as the service rate of a region in the first stage.

In summary, the main components to be mathematically modelled are the meters' traffic-generation process, the SA-TCP aggregator's traffic generation process, and the network. The details of each component are provided in the following subsections. Table 6.1 provides a summary of the major notations and variables used in the modelling.

6.2 Model of Meters

We are interested in finding the average number of segments generated by a meter (or sensor), which is the offered load performance metric. The process depends on the application layer and the TCP layer. Therefore, we model the two layers together to capture their interaction.

In this model, the TCP connections are assumed to be long lived, so there is no need for repeated connection setup. Moreover, Nagle's and the delayed ACK algorithms are turned off. A meter is expected to generate data at a low rate, with relatively long pauses. Therefore, the application behavior is modelled as a process alternating between on and off periods. We assume that a meter's application stays inactive (i.e., no data is produced) for T_{off} seconds and follows an exponential distribution with parameter $\alpha = \frac{1}{T_{off}}$. Similarly,

Table 6.1: Summary of 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-aggregator side
d_2	Packet latency on the aggregator-collector side
Ed	Expected value of delay
D	Tolerated latency
T_p	Two-way 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 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-aggregator side
P_a	Probability of packet loss on the aggregator-collector side
ρ_m	Queuing utilization factor on meter-aggregator side
ρ_a	Queuing utilization factor on aggregator-collector 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

it spends T_{on} time as active, following an exponential distribution with parameter $\beta = \frac{1}{T_{on}}$. Although there is no traffic model for meters in the literature, we follow this approximation because it is close to the anticipated real activity. Additionally, by adjusting the on and off durations, we can easily adapt the application's behavior.

When the meter's application is active, TCP encapsulates segments and transmits them over the TCP connection in accordance with the congestion window size (w). Regardless of the TCP variant used, the meter's congestion window will not exceed 2, for two reasons: (i) the amount of data is low (Small T_{on}); (ii) T_{off} is large, which causes the congestion window to reset to its initial size.

Fig. 6.3 depicts the state diagram of the Markov chain describing the combined model of the application and TCP behavior. 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 system spends T_{off} time in the idle state, during which it sends zero segments. Then it moves to State 1 with rate α, corresponding to sending one segment every RTT. If the segment is delivered successfully, the window size grows to two segments (State 2). If a segment is lost while in State 2 or State 1, it enters into State 0, which corresponds to an acknowledgement (ACK) packet timeout. The congestion window is then reset to one (State 1) for retransmission. As T_{on} expires, the system moves to State 'idle' (with rate β). The transition rates are further explained as follows:

- For $w = 1$ segment, the probability of successfully sending the segment is $q_1 = (1 - P_m)$, where P_m is the probability of packet loss on the meter-aggregator side.
- For $w = 2$ segments, assuming independent losses, the probability of sending the two segments successfully is $q_2 = (1 - P_m)^2$.
- 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 transition rate from State 1 to State 2 is δ, weighted by q_1, the probability of the successful delivery of one segment.
- The transition rate from State 1 or State 2 to State 0 is rate δ, weighted by the probability of the unsuccessful delivery of either one of the segments.
- Acknowledgement timeout is approximated to be $5T_r$, based on the assumption that the TCP protocol estimates timeout as the average RTT plus 4 times its standard

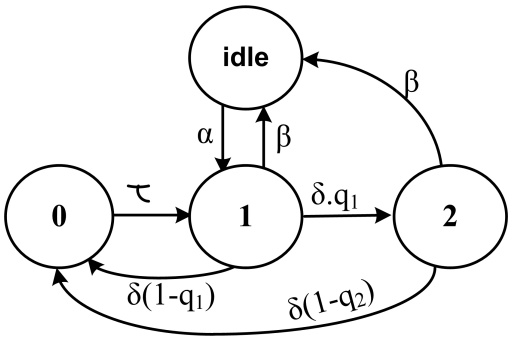


Figure 6.3: Meter Markov Model

deviation. Since RTT is assumed to be exponentially distributed, the average and standard deviation are the same (*i.e.,* T_r). Thus, the transition rate from State 0 to State 1 is calculated as $\tau = \frac{1}{5T_r}$ [25].

We get the following balance equations from the Markov chain of Fig. 6.3:

$$(\delta + \beta)\pi_1 - \tau\pi_0 - \alpha\pi_{idle} = 0$$
$$\delta q_1\pi_1 - (\beta + \delta(1 - q_2))\pi_2 = 0$$
$$\delta(1 - q_1)\pi_1 + \delta(1 - q_2)\pi_2 - \tau\pi_0 = 0$$
$$\beta\pi_1 + \beta\pi_2 - \alpha\pi_{idle} = 0$$
$$\pi_1 + \pi_2 + \pi_0 + \pi_{idle} = 1$$

where, π_1, π_2, π_0 and π_{idle} are the stationary probabilities of the four states. We solve the above linear equations to find the stationary probabilities; specifically we are interested in π_1 and π_2, which correspond to the transmission of 1 and 2 segments, respectively. Thus, the average traffic generated by a single source is the following:

$$\lambda_i = \delta\pi_1 + 2\delta\pi_2 \tag{6.1}$$

The total traffic generated by all the meters in a region, each with its On/Off and RTT characteristics, is as follows.

$$\lambda_t = \sum_{i=1}^{n} \lambda_i \tag{6.2}$$

6.3 Model of SA-TCP Aggregators

In the second stage in Fig. 6.1, the SA-TCP aggregators act as data sources for the utility collector. An SA-TCP aggregator establishes a long-lived TCP connection with the utility collector, over which it continuously and immediately forwards the received packets from the first stage in accordance with its congestion window dynamics. The congestion window grows to large sizes, and the mechanism depends on the TCP variant, which is assumed here to follow the Reno version [14].

Figure 6.4 details the Markov chain model of the aggregator's TCP congestion mechanism. The diagram demonstrates how TCP behaves and shows how Markov states and transitions are designed. This design is inspired by [25]. The states are numbered to represent the congestion window size. A state S, for instance, corresponds to the transmission rate of S segments per RTT. State 0 represents the period of timeout, in which no segment transmission takes place. Numbers with dashes in the right-most column states represent the fast retransmit phase but do not correspond to actual transmissions.

We assume that the maximum congestion window is W_M (determined by the receiver window), and the initial slow start threshold is w_t (e.g., $W_M = 16$ and $w_t = 8$ as in the diagram). The TCP protocol starts with an initial congestion window size of 1 and operates in three phases: Slow Start (SS), Congestion Avoidance (CA), and Fast Retransmit/Recovery (FR). In the diagram, the initial states correspond to the left-most column ($w_t = 8$), starting at State 1. During SS, the window doubles every RTT, so it moves to states 2, 4, then 8. After that, it enters into the CA phase, in which the window grows by only one MSS every RTT, so the next states to be visited are 9, 10, and upward up to W_M as long as all the segments are acknowledged successfully each time. However, as packet loss occurs (depicted by dashed lines), the window shrinks in two ways. If no ACK arrives for the lost packets before the timeout, then the congestion window resets to one and w_t is halved. This situation corresponds to the loss of multiple packets in a window without giving a chance for triple duplicate acknowledgements (3Dup). For example, if the window is less than four, 3Dup does not occur, forcing the window to reset to the SS phase. This situation is modelled as a transition to State 0 to capture the time duration of an ACK timeout. In the case of 3Dup, the FR phase is triggered, in which the congestion window size (w) as well as w_t reduce to half (i.e., $\frac{w}{2}$), and then TCP enters into the CA phase again. In the diagram, this case corresponds to the transitions to the right-most column of states. For example, with the loss of only one segment when the window is 4 (State 4), the transition is made from State 4 to State 2' and then to State 2 under $w_t = 1, 2$ so that the window grows linearly after that.

Further to the above discussion, state transitions can be explained as follows:

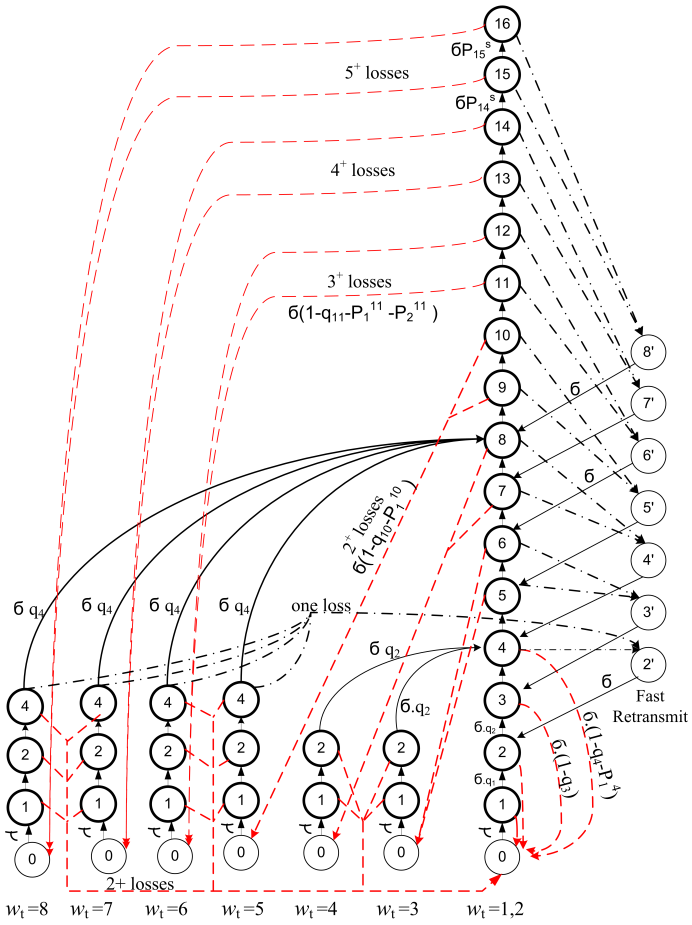


Figure 6.4: SA-Aggregator Markov Model

- The probability of successfully delivering all segments while in a state of congestion window size w is calculated as $q_w = (1 - P_a)^w$, where P_a is the probability of packet loss on the aggregator-collector side.
- State transitions representing window growth (whether doubling or linearly incrementing) are calculated as δq_w.
- Assuming losses occur independently, the probability of the loss of i segments while in a state of window size w is calculated as $P_i^w = \binom{w}{i}.P_a^i.(1 - P_a)^{w-i}$
- For states of $w < 4$, the loss of any segment causes transition to State 0 of column

$w_t = \frac{w}{2}$. The rate of transition is $\delta(1 - q_w)$

- For states of $4 \leqslant w \leqslant 10$, the loss of one segment causes transition to FR states. The rate of transition is δP_1^w.

  However, for two or more losses, the transition is made to State 0, and the transition rate is calculated as $\delta(1 - q_w - P_1^w)$

- For states of $w = 11$ or 12, one or two losses lead to a transition to the FR states numbered $\frac{w}{2}$. The transition rate is $\delta(P_1^w + P_2^w)$.

  If more than two losses occur, the transition is made to a state with w and $w_t = \frac{w}{2}$. The rate in this case is $\delta(1 - q_w - P_1^w - P_2^w)$.

- For states of $w = 13$ or 14, a loss of 1, 2, or 3 segments causes a transition to an FR state of $\frac{w}{2}$ with a transition rate of $\delta(P_1^w + P_2^w + P_3^w)$. However, in the case of more than three losses, the transition is for a state with w and $w_t = \frac{w}{2}$ at a rate $\delta(1 - q_w - P_1^w - P_2^w - P_3^w)$.

  The same form of analysis is performed on the higher states (> 14).

- Transitions from the FR states take the rate δ. This transition captures the TCP Reno's recovery mechanism. The FR states do not correspond to the transmission of actual segments.

- Transition from 0 states takes the rate $\tau = \frac{1}{T_r + 4\sigma} = \frac{1}{5T_r} = \frac{\delta}{5}$, assuming an exponential distribution for RTT with an average and standard deviation of T_r [25].

- For simplicity, $\frac{w}{2}$ is approximated as $\lfloor \frac{w}{2} \rfloor$ or $\lceil \frac{w}{2} \rceil$.

- Unlike a meter, an SA-TCP aggregator does not have an idle state because it is unlikely that an aggregator stays inactive with a large number of meters sending asynchronously. The probability of an aggregator being idle (P_{idle}) is the probability of none of the meters send data. To calculate that, assume that the probability of a meter transmits is $P_{active} = \frac{T_{on}}{T_{off} + T_{on}}$. Probability of j meters in a region of n meters send data is binomially distributed.

$$P(X = j) = \binom{n}{j} P_{active}^j (1 - P_{active})^{n-j} \tag{6.3}$$

$$P_{idle} = (1 - P_{active})^n \tag{6.4}$$

For example, if T_{on} = 0.1 sec and T_{off} = 1 minutes, and n = 10,000, then P_{idle} = $5.8X10^{-8}$, which is negligible.

Next, to calculate the load served by an aggregator (γ_j), we solve the balance equations of the Markov chain model to obtain the stationary probabilities π_s, where s is a state. Equation (6.5) calculates the average traffic load on the jth SA-TCP aggregator (γ_j). The FR states are excluded because they do not correspond to the actual transmission of segments.

Thus,

$$\gamma_j = \delta(\sum_s W_s \pi_s) \tag{6.5}$$

where W_s is 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 total traffic from all the SA-TCP aggregators going into the stage 2 bottleneck is as follows.

$$\gamma_t = \sum_{j=1}^{N_A} \gamma_j \tag{6.6}$$

6.4 Network Model

In reference to Fig. 6.1, we assume that multiple bottlenecks determine network performance. In every region, a bottleneck link exists between the meters and the corresponding SA-TCP aggregator. In the second stage, a bottleneck link exists between the set of SA-TCP aggregators and the utility server. We examine two queuing models (M/M/1/B and M/D/1/B) to model the network bottlenecks. Because packets have little variability in size, both models fit, but in terms of computational complexity, M/M/1/B is more efficient [81].

We are particularly interested in calculating the probability of dropping packets (i.e., packet loss rate) and the average queuing delay, which impact a source's segment-generation process. The model here assumes that ACK packets in the reverse direction are loss-free. A queue is characterized by three parameters: the queue size, input traffic rate, and service rate. In the SMI's first stage (Fig. 6.1), a region's network model is characterized by queue size (B), input traffic rate (λ_t), and service rate (μ_1). In the network model of the SMI's second stage, these parameters become B, γ_t, and μ_2, respectively. The input traffic rates,

λ_t and γ_t, are derived from the Markov models of the meters and SA-TCP aggregators, respectively.

The service rate, μ_1, is equal to an SA-TCP aggregator's offered load (γ_j). The service rate, μ_2, is computed as the link capacity (C) over the segment size (MSS) in bytes (*i.e.*, $\mu_2 = \frac{C}{MSS}$ segment/sec) [81]. The loss rate and expected queuing delay are given by:

$$P(\rho, B) = \frac{\rho^B(1-\rho)}{1-\rho^{(B+1)}} \tag{6.7}$$

$$ET_q(\mu, \rho, B) = (\frac{1}{\mu}).\frac{1-(B+1).\rho^B + B.\rho^{(B+1)}}{(1-\rho)(1-\rho^B)} \tag{6.8}$$

$$Ed(\mu, \rho, B) = T_p + ET_q(\mu, \rho, B) \tag{6.9}$$

$$\rho_m = \frac{\lambda_t}{\mu_1}, \;\; \mu_1 = \frac{\gamma_t}{N_A}, \;\; \rho_a = \frac{\gamma_t}{\mu_2}, \tag{6.10}$$

$$P_m = P(\rho_m), \;\; P_a = P(\rho_a) \tag{6.11}$$

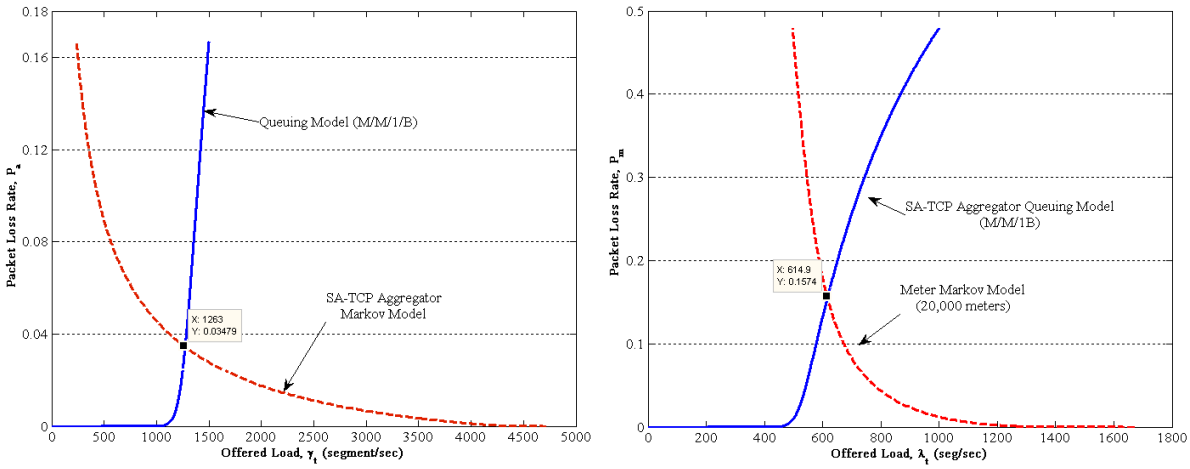
$$d_1 = Ed(\mu_1, \rho_m, B_1), \;\; d_2 = Ed(\mu_2, \rho_a, B_2) \tag{6.12}$$

where ρ_m and ρ_a are the queue utilization factors in the first and second stages of the SMI model, respectively. $ET_q(\mu, \rho)$ and $Ed(\mu, \rho)$ are the expected values of the queuing delay and end-to-end delay, respectively.

6.5 Convergence and Existence of Unique Solution

Here we show that a unique fixed-point solution that solves the meter model and the network model exists. The equations that control the interaction between the network bottleneck model and the source TCP model are (6.9), (6.7) and (6.5). Solving them together leads to a unique fixed-point, which corresponds to the network operating point. Equations (6.7) and (6.9) characterize the queue model as a function of the input rate (γ). Equation (6.5) describes the behaviour of the TCP Markov model in terms of the loss rate and RTT.

Equation (6.7) is non-decreasing in γ. As γ increases, the number of enqueued packets increases, leading to a higher packet loss rate. Equation (6.5) is non-increasing in packet loss rate. The higher the probability of loss, the less the offered load. Figure 6.5 plots the equations simultaneously as a relation between the offered load (γ_t) and packet loss rate (P_a). It is evident that the two curves intersect at a single point, which is the fixed point



(a) Stage 2:SA-TCP Aggregator-Utility Server (b) Stage 1:Meters - SA-TCP Aggregator

Figure 6.5: Existence of Unique Solution

solution (γ^*, P^*, T_r^*). Figure 6.5a considers 15 SA-TCP aggregators communicating with the utility server. The obtained fixed point solution is validated by ns-2 simulations. A similar experiment is performed for a region of meters connecting to an SA-TCP aggregator. Plotting λ_t versus P_m also proves that the network operating point is obtainable in the same manner (Fig 6.5b)

Algorithm 6.1 illustrates the search procedure by which we find the fixed-point solution for each region of the first stage (*i.e.,* meters to SA-TCP aggregator) and second stage (*i.e.,* SA-TCP aggregators to the utility server) models.

6.6 Model Validation

The SA-TCP scheme analytical model is validated through extensive computer simulations using the network simulator ns-2. The focus is mainly on three measures: source traffic offered load, packet loss rate, and packet end-to-end delay. The simulation parameters are presented in the corresponding subsections below.

Algorithm 6.1: Finding the Fixed-Point Solution

Step 1: The offered load γ^* is expected to be close to the bottleneck service rate, μ, because the TCP protocol tries to utilize the available capacity, so we start initially with $\gamma = \mu$.
Step 2: We linearly increment or decrement γ by a small amount $\Delta\gamma$ and calculate P and d from the queue model (Equations (6.11) and (6.12)).
Step 3: We apply P and d to the TCP Markov model (Equation (6.6)) to compute $\bar{\gamma}$. If the absolute relative error ($|\gamma - \bar{\gamma}|$) is close to zero (*e.g.,* 0.01), then this is the fixed-point solution. If not, we repeat Step 2.
Remark: The first two iterations lead to the right search direction (*i.e.,* whether γ^* is higher or lower than μ). To speed up the search, we start with a relatively large $\Delta\gamma$ in Step 2 and then reduce it as the relative absolute error shrinks. On average, the fixed point solution is reached in 400 to 500 iterations.

6.6.1 Validation of Meter's Model

The meter Markov model presented in Fig. 6.3 considers two input parameters, packet loss rate P_m and round trip time RTT, in order to compute the network throughput λ_t. Thus, validating the meter model can be achieved in two steps. First, ns-2 simulation is performed to calculate P_m, RTT and the throughput (λ_t). By feeding P_m and RTT, obtained from the simulation results, into the model (Equation (6.2)), we get the analytical throughput value. Figure 6.6a shows a comparison of the analytical and the simulated throughput values. Validation is done for a large range of meters (15,000 to 25,000) sharing a link of 1 Mbps bandwidth and 50 msec propagation delay. Each meter is characterized by an average $T_{on} = 100$ msec and $T_{off} = 1$ minute. As the figure shows, the analytical model provides a close match to the simulation results.

6.6.2 Validation of SA-TCP Aggregator

The SA-TCP aggregator Markov model (Fig. 6.4) in the second stage block is validated in the same way as the meters' model. Figure 6.6b shows the SA-TCP aggregator throughput γ_t that is calculated by the model (Equation (6.6)) using the simulation results of P_a and RTT as inputs and the throughput measured by the simulation. Figure 6.6b shows a match of within 2% of the simulation results. In this experiment, the implemented model is assumed to have $W_M = 16$ and an initial $w_t = 8$, thus forming 46 Markov chain states. The SA-TCP aggregators share a network in the second stage (Fig. 6.4) limited by an E1

link (2.048 Mbps data rate).

6.6.3 Validation of Network Model

The queuing model (Equations (6.11) and (6.12)) takes the traffic load from the sources (*i.e.,* meters or aggregators) as an input and estimates the packet loss rate and delay. Therefore, we test the network model by measuring the throughput by the simulator and feeding it to the model. The calculated values of P_m and d_1 for a region in the first stage and P_a and d_2 for the second stage are compared against those obtained from the simulator. Figures 6.6c, 6.6e, 6.6d and 6.6f show the results for M/M/1/B (B = 40 MSS) for the first and second stages. It is observed from the figures that M/M/1/B gives good matching results as the number of sources increases. This is explained by the fact that the large number of TCP connections makes the assumption that the statistical independence (*i.e.,* less correlation) of the input traffic to the queue is realistic. Thereby, the assumption that the network sees a segment generation process given by a Poisson distribution is valid [25].

6.6.4 Validation of Fixed-point Approach in a Region

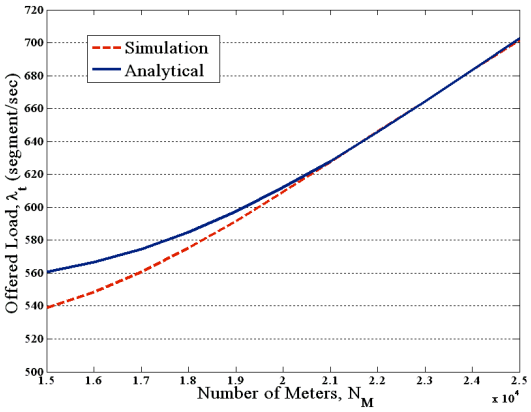
In Fig. 6.1, a region is a collection of meters transmitting their data to an SA-TCP aggregator. Figures 6.7a, 6.7b and 6.7c show that our model is highly accurate in predicting all region performance measures, namely, offered load λ_t, packet loss rate P_m and delay d_1, taking into account the interactions between the sources' Markov model and the queue model. In other words, the performance results (*i.e.,* the network operating point) are obtained using the fixed-point algorithm (Section 6.5). In this experiment, meter traffic is modelled as independent on-off sources where the on and off times follow exponential distributions with parameters T_{on} = 100 msec and T_{off} = 1 minute, respectively. During the on time, the source produces data packets of size 240 bytes (40 bytes TCP header). The shared link is characterized by a bandwidth of 1 Mbps, buffer capacity B = 40 MSS, and propagation delay of 50 msec. In the simulation tests, the sources use FTP and run for 12,000 seconds.

6.7 Summary

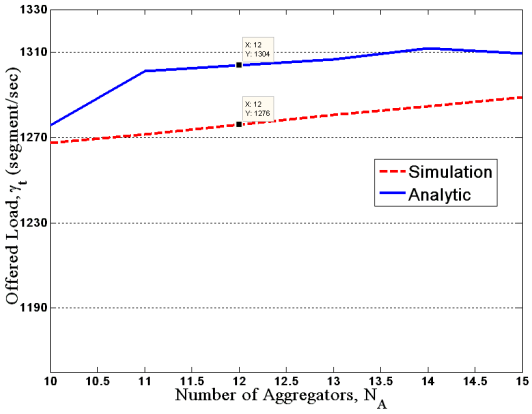
Mathematical modelling of SA-TCP has been the focus of this chapter. Predicting performance of the SA-TCP scheme under various scheme setups and network parameters is crucial for better understanding and optimal design of the scheme. The model applies Markov

chains to represent the meters' and SA-TCP aggregators' segment-generation process and applies queuing systems to represent the network performance. Such model combinations allow one to determine the so-called stationary point of a network describing three metrics of interest: the data rate at which the meters transmit, packet loss rate, and delay.

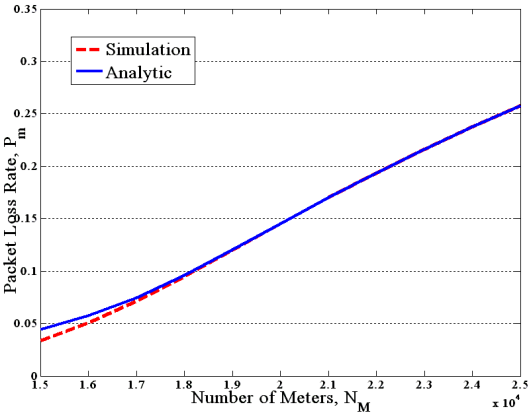
The math model studies the effect of various scheme and network parameters, including the meter active/inactive timings, number of meters and SA-TCP aggregators, available bandwidth, buffering capacity of SA-TCP aggregators and network bottlenecks, and propagation delays. The TCP Reno variant has been assumed as the TCP version employed in all the involved devices. The next chapter expands the model to study the difference that TCP Vegas makes if used instead in all the SA-TCP communicating devices.



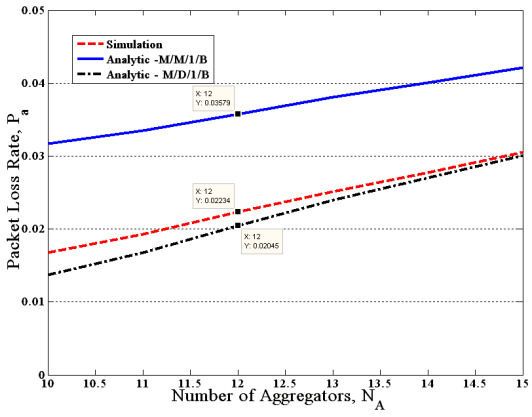
(a) Meter Offered Load



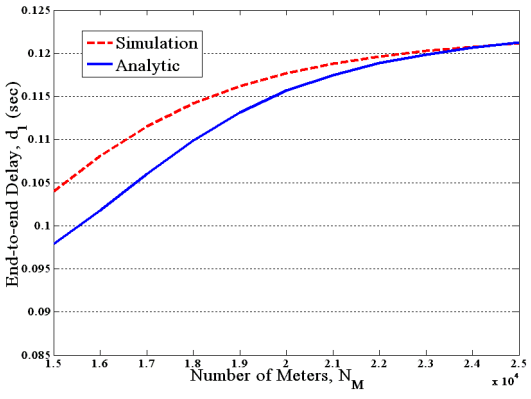
(b) SA-TCP Aggregator Offered Load



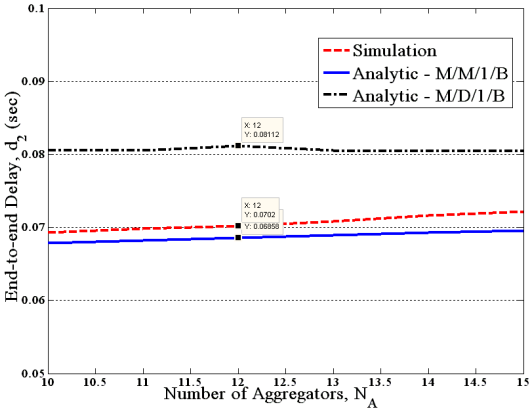
(c) Meter Loss Rate



(d) SA-TCP Aggregator Loss Rate

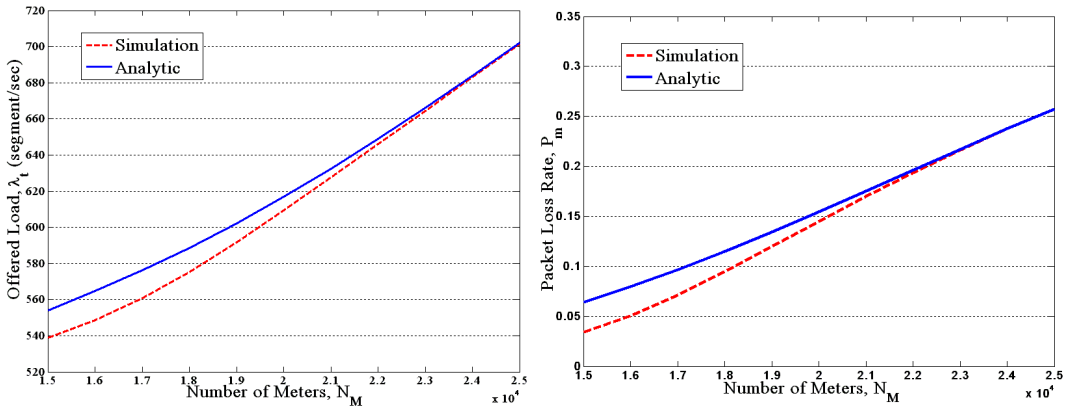


(e) Meter Delay



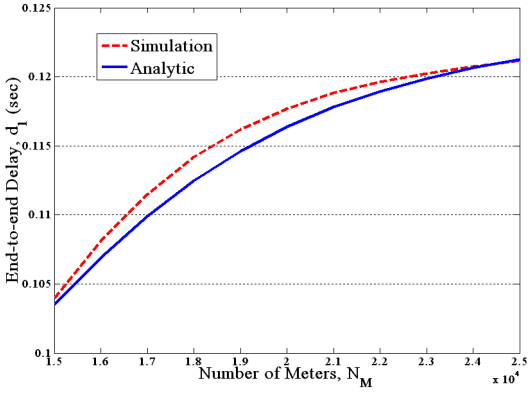
(f) SA-TCP Aggregator Delay

Figure 6.6: Validation of First and Second Stage Markov and Network Models



(a) Offered Traffic Load in a Region

(b) Packet Loss Rate in a Region



(c) End-to-end Delay in a Region

Figure 6.7: Validation of fixed-point approach in a region. Performance metrics are obtained using fixed-point solution

Chapter 7

SA-TCP Vegas Analytical Model

This chapter mathematically models SA-TCP assuming both meters and SA-TCP aggregators run the Vegas [21] implementation of TCP. TCP Vegas has been chosen in this extended study of SA-TCP for its revolutionary change to TCP Reno. TCP Vegas represents a class of TCP variants (*e.g.,* DUAL [144], CARD [64] and Nice [7]) that observes packet delay, as opposed to loss, to predict congestion. TCP Reno and similar ones (*e.g.,* Tahoe [62], SACK [48] and NewReno [49]) are reactive protocols adjusting the congestion window size after losses are detected.

As indicated in Section 4.2, a one-hop TCP congestion control mechanism is ineffective, regardless of the TCP version. This chapter confirms this claim. Although, TCP Vegas was reported in [21] to achieve 37% to 71% better throughput and up to 20% less loss on the Internet as compared with TCP Reno, surprisingly, Vegas performs worse if adopted by SMI in a one-hop TCP scheme. This drop in performance occurs because Vegas is more aggressive in the slow start phase, as will be shown later in this chapter.

7.1 TCP Vegas

TCP Vegas adds new delay-based techniques to the slow start and congestion avoidance mechanisms. The aim is to adjust the congestion window size before losses occur and to reduce the number of timeouts so that the packet loss rate is reduced and throughput is increased. If loss occurs, however, TCP Reno's timeout and fast retransmit/fast recovery mechanisms are used. Before discussing the modifications applied to the math models of meters and SA-TCP aggregators, below is an abstract of the added techniques in TCP Vegas.

- The first technique gives TCP the ability to predict congestion and, consequently, to adjust the transmission rate accordingly. The technique works by keeping 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.

$$N_b = \frac{w}{RTT}(RTT - RTT_{min}) \tag{7.1}$$

  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.

- Because TCP Vegas adjusts the window continuously, it does not reduce the window size harshly when losses occur. Therefore, upon losses that cause fast retransmit, instead of slashing the congestion window to half its size as in TCP Reno, reduction here is only to three quarters its current size.

- In trying to achieve a more timely decision in retransmitting a dropped packet, TCP Vegas does not wait for three duplicate ACKs. Instead, a sender keeps record of RTT durations of every packet on transit, so when the first or second duplicate ACK is received, Vegas checks whether the 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.

7.2 Modelling Approach

The same methodology presented in Section 6.1 is followed here. One addition, though, is the use of delay distribution, based on which TCP Vegas decides about resizing the

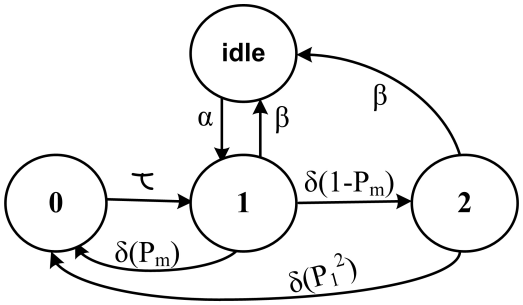


Figure 7.1: Meter Markov Model Under TCP Vegas

congestion window. The queuing model in Section 7.5 provides this delay distribution. Again, the main components to be modeled are meters, SA-TCP aggregators, and the network. In each of the two TCP stages of SA-TCP (Fig. 6.2), the fixed point approach (Algorithm 6.1) is performed to find the network operating point in the context of the source offered load, packet loss rate and delay.

7.3 Model of Meters

The states and transition rates of a meter's Markov chain, combining the application and TCP behaviors, are depicted in Fig. 7.1. Variable definitions are available in Table 6.1. The numbers in the states correspond to the congestion window size. The chain transits between active and idle states with exponentially distributed rates of $\beta = \frac{1}{E[T_{on}]}$ and $\frac{1}{E[T_{off}]}$, respectively. The window is limited by two, reflecting the assumption that a meter sends data for a short period of time (*e.g.*, $T_{on} = 100$ msec).

The model is mostly similar to TCP Reno as provided in Section 6.2. Differently, however, the transition representing a timeout from State 2 to State 0 occurs if only one segment is lost, rather than any of the two segments. This difference reflects the case in which one duplicate ACK is enough to trigger a retransmissions, 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 the fraction of time that TCP spends at each state, π_1 and π_2. The offered load of meter i is, therefore, $\lambda_i = \delta(\pi_1 + 2\pi_2)$.

7.4 Model of SA-TCP Aggregators

Figure 7.2 details the Markov chain model of SA-TCP aggregator's congestion window dynamics. The numbers in the states match the congestion window size, corresponding 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 right-most column states, with dashed numbers, model the fast retransmit/fast recovery phase, so they do not correspond to actual segment transmissions. The model here follows TCP Vegas as explained in [21] and [146].

For estimation of the number of backlogged packets in the network, 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 (7.1) becomes as follows:

$$N_b = \frac{wT_q}{T_p + T_q} \tag{7.2}$$

where w is the congestion window size, corresponding to the chain state. Section 7.5 shows how probabilities such as $P(N_b \leq a)$ and $P(N_b \geq b)$ are calculated.

The following bullets summarize the transition rates among the chain states. Variables are defined in Table 6.1.

- The transition rates from the slow start and congestion avoidance states to the fast retransmit/fast recovery and timeout states are similar to the SA-TCP aggregtor's model under TCP Reno (Section 6.3).
- For 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 SS, doubling the congestion window size is achieved with rate $P(N_b \leq \frac{a+b}{2})q_w\delta$.
- Transitioning from an SS state to a CA state with $w = w + 1$ is achieved at rate $P(N_b \geq \frac{a+b}{2})q_w\delta$.
- During CA, 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

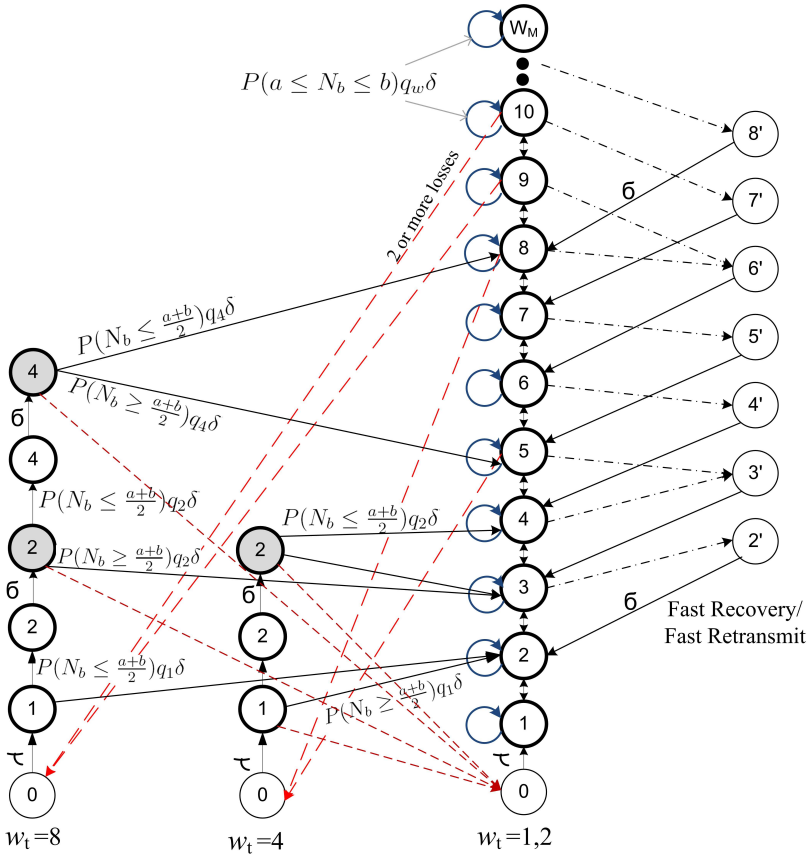


Figure 7.2: Meter Markov Model Under TCP Vegas

rate $P(a \le N_b \le b)q_w\delta$. 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 $P(N_b \le b)q_{W_M}\delta$.

The Markov chain is solved for the limiting probabilities of the fraction of time that TCP spends at each state, π_1 and π_2. The offered load of meter i is, therefore, $\gamma_j = \delta(\sum_s W_s\pi_s)$.

7.5 Network Model

The queuing system, modelling the network, determines delay distribution along with probability of packet loss and average delay. The queuing model takes into account all network-related characteristics, including the queuing capacity, link capacity, packet arrival pattern (assumed poisson), queue management scheme (assumed DropTail), and network environment (fixed or wired links).

Assuming an M/M/1/B queuing system, the Equations (6.7) to (6.12) are used, in addition to the delay distribution, needed specifically for the TCP Vegas model as shown above.

$$Pr(T_q \leq t) = 1 - \frac{e^{-\mu t}}{1-\rho^B} \sum_{i=0}^{B-1} \frac{(\lambda t)^i}{i!} + \frac{\rho^B e^{-\mu t}}{1-\rho^B} \sum_{i=0}^{B-1} \frac{(\mu t)^i}{i!} \tag{7.3}$$

The probabilities $P(N_b \leq a)$ and $P(N_b \geq b)$, required for the SA-TCP aggregator model, are calucated by Equation 7.3. This calculatin is achieved by manipulating Equation (7.2) such that T_q is put on one side and then applying Equation 7.3.

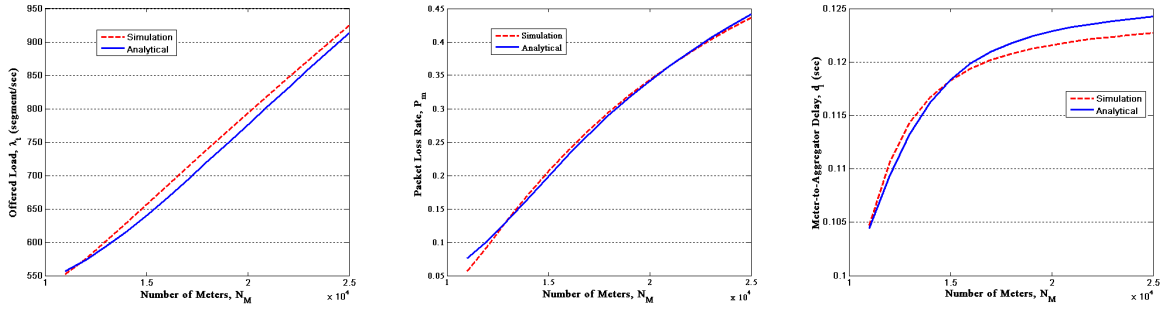
$$P(N_b \leq a) = P(Tq \leq \frac{aT_p}{w-a}) \tag{7.4}$$

$$P(N_b \geq b) = 1 - P(T_q \leq \frac{bT_q}{w-b}) \tag{7.5}$$

7.6 Model Validation

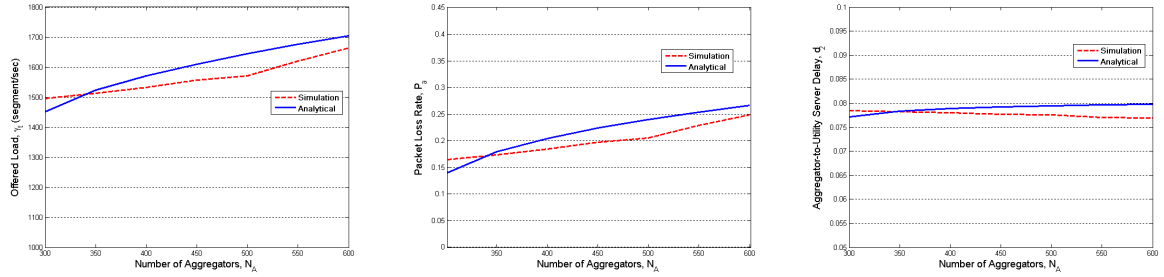
The analytical model is validated against ns-2 simulations. Validation shows that the math nodel can accurately predict the packet loss rate, delay, and traffic load offered by meters and SA-TCP aggregators.

Unlike the validation of TCP Reno, it is not possible here to seperately validate the Markov chain model of meters or aggregators without the involvement of the network model because of the need for the delay distribution. However, it is sufficient to show that the model is able to estimate the network operating point using the fixed point approach presented in Algorithm 6.1. In the experiments below, the TCP-Vegas parameters are configured as in the ns-2 default values, $a = 1$ and $b = 3$.



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

Figure 7.3: Validation of TCP-Vegas in a region. Performance metrics are obtained using fixed-point solution



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

Figure 7.4: Validation of TCP-Vegas in in the 2^{nd} hop. Performance metrics are obtained using fixed-point solution

Figure 7.3 validates the TCP Vegas for a region of meters reporting data to an SA-TCP aggregator. 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, and the available link bandwidth is 1 Mbps. The figures show that the model estimates the three performance metrics within less than 10% of the simulations.

Similarly, Fig. 7.4 valides the three performance metrics in the second TCP hop formed by SA-TCP aggregators and the utility server. In this experiment, the propgation delay is 50 msec, and the available link bandwidth is 2 Mbps with a 40-packet buffer capacity.

7.7 Summary

Unlike TCP-Reno, the Vegas version of TCP is a proactive protocol in that TCP-Vegas adjusts a source's congestion window before losses take place. TCP-Vegas keeps track of round trip time variations, and so it changes the window size accordingly. The major difference between the two variants was the motive behind studying the variant. This chapter has managed to capture the behaviour of meters and SA-TCP aggregators mathematically. Thus, given the SMI setup and network settings, the math model is able to predict the traffic load generated, packet loss rate, and delay.

The next chapter is dedicated to evaluating SMI under the change of various SMI parameters, including the comparison between TCP-Reno and TCP-Vegas.

Chapter 8

Performance Analysis and Optimization

This chapter provides performance evaluation for the smart metering infrastructure in terms of throughput, loss rate and delay. Throughput is the portion of generated packets that successfully arrive at the utility server. Packet loss rate is the probability of packets getting dropped due to buffer overflow in the network. Packet end-to-end delay is the time in seconds a meter's packet takes to arrive at the utility server. The varying parameters are link capacity, buffering capacity, propagation delay, and number of SA-TCP aggregators. The analysis is based on the mathematical model presented in Chapters 6 and 7. The purpose of the evaluation is to investigate how the SA-TCP scheme design is affected by the different parameters, and how this scheme compares to the one-hop TCP scheme.

Based on understanding what impacts the scheme's performance and using the math model, optimization comes into the picture. Section 8.3 formulates an optimization model to enhance the performance of the SA-TCP scheme.

8.1 Performance Analysis

Figure. 6.1 depicts the meter-to-utility server architecture with RC devices acting as SA-TCP aggregators. The figure shows the network and SA-TCP scheme parameters. The link capacity, C, in *bps* is the bandwidth available on the WAN network. Propagation delay is the time 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

Table 8.1: Experiment Parameters

Number of meters, N_M	400,000
Number of Aggregators, N_A	0 to 1000
Meter's T_{on} /T_{off}	100 msec/ 1 minute
Packet size	200 bytes
Link bandwidth, C	1 Mbps to 3 Mbps
SA-TCP aggregator buffer capacity, B_1	20 to 80 packets, DropTail
Bottleneck buffer, B_2	100 packets, DropTail
Meter-to-Aggregator bandwidth	1 Mbps
Meter-to-Utility server propagation delay	80 to 260 msec

aggregator's. The second stage network bottleneck is assumed to be large, B_1. Finally, N_A represents the number of SA-TCP aggregators (*i.e.,* RCs). The SMI network setup and the variable parameters to be analyzed are summarized in Table 8.1. In the following experiments, TCP-Reno is assumed.

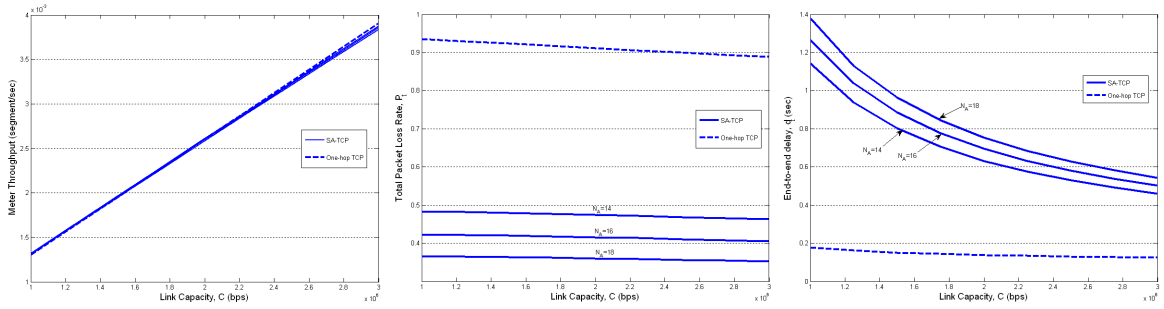
8.1.1 Varying Link Capacity

Figure 8.1 shows the impact of the WAN shared link bandwidth C on the SMI performance. As expected, increasing the link capacity reduces the packet loss rate and delay and allows a traffic source to increase its offered traffic load. However, it is worth noticing that with SA-TCP, the impact of changing the link capacity is higher than that of one-hop TCP. For example, Fig. 8.1c shows that increasing C decreases delay faster than the case with one-hop TCP.

Figure 8.1a shows that applying the SA-TCP scheme benefits the SMI system by keeping the meter throughput unchanged, while improving the packet loss rate (Fig. 8.1b).

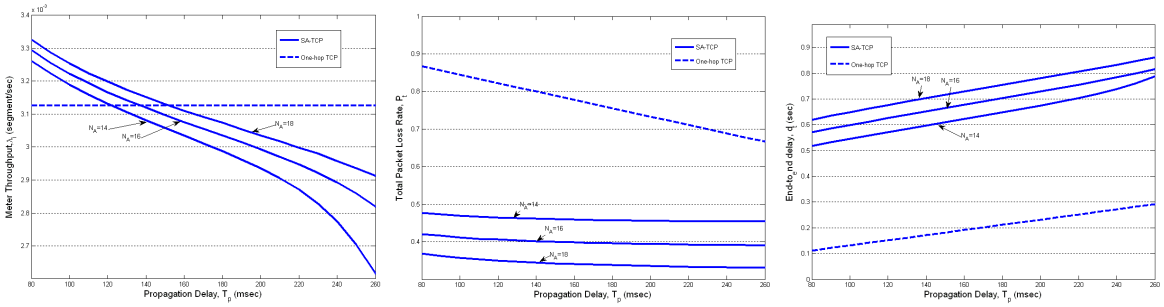
8.1.2 Varying Propagation Delay

Figure 8.2 shows the relation between propagation delay and the performance metrics, offered load, loss rate and delay. As propagation delay increases, traffic load decreases as expected because it takes longer to increase the congestion window. Less traffic load leads to a smaller loss rate (Fig. 8.2b). End-to-end delay increases, which is a reflection



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

Figure 8.1: Impact of Link Capacity on performance



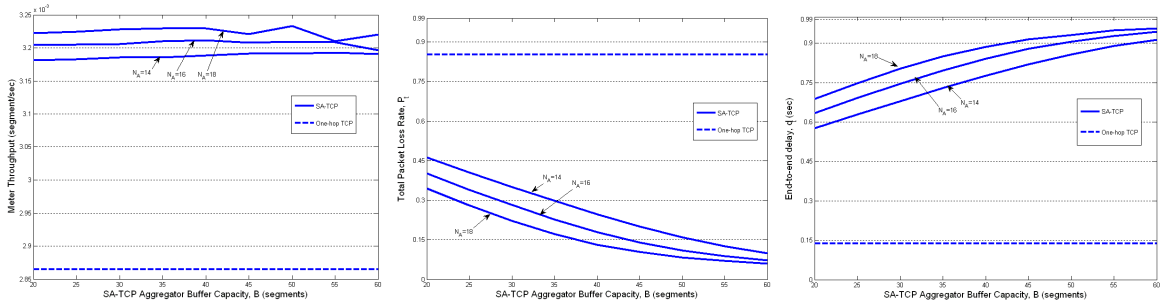
(a) Meter Throughput (b) Total Packet Loss Rate (c) End-to-end Delay

Figure 8.2: Impact of propagation delay on performance

of the longer propagation delay a packet experiences. 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 delay to increase but loss to decrease. A meter's offered load slightly decreases as a result of a longer end-to-end packet delay.

8.1.3 Varying SA-TCP Aggregator Buffer Capacity

The effect of the buffering capacity of SA-TCP aggregators is shown in Fig. 8.3. In one-hop TCP, none of the performance metrics changes since there is no aggregator in the architecture. The situation shows that SA-TCP improves the response to network congestion. Meters keep transmitting a high traffic load despite the high loss rate that results. Fig. 8.3a clearly indicates that SA-TCP improves meter throughput, especially with a higher num-



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

Figure 8.3: Impact of SA-TCP Aggregators Buffer Capacity

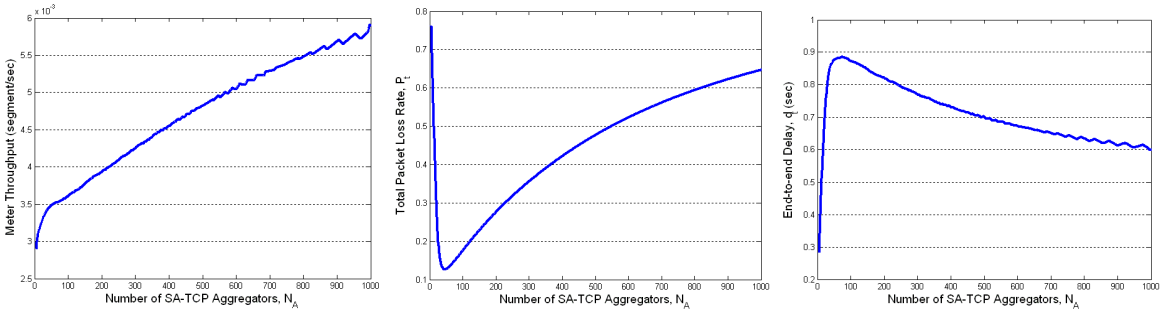
ber of SA-TCP aggregators. The impact of buffering capacity is shown clearly in terms of packet loss rate and delay (Figures 8.3b and 8.3c); the bigger the buffer size, the greater the delay and smaller the packet loss rate.

8.1.4 Varying Number of SA-TCP Aggregators

As shown in Fig. 8.4, the number of SA-TCP aggregators impacts SMI performance. Increasing the number of SA-TCP aggregators reduces the loss rate in a meter's regions; therefore, the meter's throughput naturally increases (Fig. 8.4a). However, that does not mean the more aggregators, the better, because as N_A increases, loss in the second stage of SMI increases (low throughput), and makes TCP congestion control ineffective again. We notice in Figures 8.4b and 8.4c that the total loss rate decreases and total delay increases as N_A is incremented, but after a certain point, the loss rate starts to increase while delay decreases. The increase of loss and decrease of delay indicate that congestion control becomes ineffective as a result of a large number of SA-TCP aggregators, similar to the case of having a large number of meters.

8.2 TCP-Vegas Vs. TCP-Reno in SMI

Both versions of TCP, and all other similar ones, lead to the same problem: ineffective TCP congestion control. As explained in Section 4.2, the problem occurs because all variants are based on the idea of reducing the congestion window size when a source senses congestion in the network, but the ability of traffic reduction is not viable in SMI. However, interesting



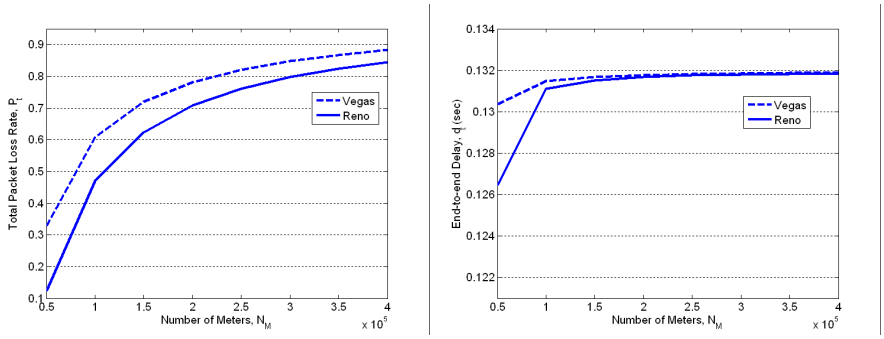
(a) Meter Throughput (b) Total Packet Loss Rate (c) End-to-end Delay

Figure 8.4: Impact of Number of SA-TCP Aggregators on performance

differences in performance exist. The experiments below contrast Reno and Vegas in an SMI environment.

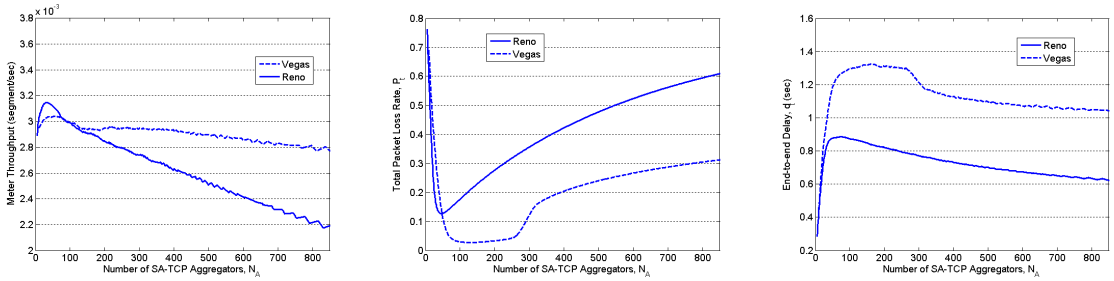
TCP-Vegas is surprisingly found to perform worse in a one-hop-TCP SMI. Figure 8.5 demonstrates the performance in terms of loss rate and delay. This One-hop-TCP experiment (depicted in Fig. 4.2) is set up here to vary the number of meters from 50,000 to 400,000. A shared bottleneck is configured as 2 Mbps and a 100-packet buffer capacity. The propagation delay is 100 msec. While both variants result in the same delay, TCP-Reno causes fewer losses. Although TCP-Vegas has been shown to outperform TCP-Reno in the Internet [21], certain properties of Vegas make it even less effective than Reno in SMI, 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 and makes congestion control in SMI even less effective, thus leading to more losses.

For evaluating SA-TCP under TCP-Vegas in contrast with TCP-Reno, an experiment with the same parameters as in the above one-hop-TCP experiment is conducted, but the number of meters is fixed to 400,000, and the number of SA-TCP aggregators, N_A, is varied from 1 to 1000. Here, TCP-Vegas performs better than TCP-Reno, as Fig. 8.6 suggests. A closer look, however, reveals interesting behavior. SA-TCP under Vegas achieves a lower loss rate, but the delay stays slightly higher than with Reno as N_A is changed. It is noted that Vegas keeps the loss rate low for a longer range of N_A. However, it is noticed also that with Reno, there is a faster decrease in the loss rate (Fig. 8.6b) than with Vegas. This rapid decrease explains why throughput goes higher than that of Vegas in the beginning (Fig. 8.6a). This behaviour is explained by the fact that TCP-Vegas performs better in a



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

Figure 8.5: TCP-Vegas Vs. TCP-Reno Performance in One-hop-TCP SMI



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

Figure 8.6: TCP-Vegas Vs. TCP-Reno Performance in SA-TCP SMI

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, and better congestion control is achievable on the WAN side. As N_A turns large, the effectiveness of congestion control becomes weaker on the WAN side as well; therefore, the loss rate increases again.

8.3 Optimizing SA-TCP Architecture

Several input variables impact the performance of the SA-TCP scheme as described in the math model. In what follows, optimization is presented in three ways. The first model

minimizes packet loss rate; the second minimizes the number of SA-TCP aggregators; and the third model minimizes the deployment cost of the scheme.

8.3.1 Minimizing Packet Loss Rate

In certain SMI models, metering traffic is considered delay tolerant [39]. Therefore, one optimization objective is to minimize the scheme's loss rate, while relaxing the delay performance metric, as presented in the following formulation.

$$\min\ P_t(N_A) \tag{8.1}$$

Subject to

$$P_t = 1 - (1 - P_m)(1 - P_a) \tag{8.2}$$

$$N_A \leq N_M \tag{8.3}$$

$$N_A \in \{1, 2, ..., N_M\} \tag{8.4}$$

Equation (8.2) calculates the total loss rate according to Algorithm 6.1 and Equation 6.11, the latter of 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 (Inequality 8.3). Figure 8.4b demonstrates the total SMI loss rate as a function of the number of SA-TCP aggregators. The loss rate continues to decrease as we increase the number of aggregators, up to a certain point, where the 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, 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 8.1 finds N_A that minimizes the loss rate. This algorithm is a modified version from the Rosenbrock optimization method [119]. Because derivation of the SA-TCP model is not viable, this gradient-free direct search method is used. The algorithm is initialized by the expansion factor β_1 and and the contraction factor β_2 to be 2 and $\frac{-1}{4}$. Accordingly, the search advances in large steps by doubling the step size Δ, 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 Δ shrinks. The program stops when the step size $|\Delta|$ becomes smaller than the termination tolerance ϵ. Asymptotically, the algorithm takes $O(log_2^2 N_M + log_2 N_M)$.

Figure 8.7 shows the benefit of the optimization at different number of meters and at different meter traffic rates. Both sub-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 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 8.1: Finding the Optimal Number of Aggregators

1: $\epsilon = 0.5$ // Termination factor
2: $\beta_1 = 1$ // Expansion factor
3: $\beta_2 = \frac{-1}{4}$, // Contraction factor
4: $\Delta = 1$ // Initial step size
5: $if\ (P_t = f(N_A + \Delta) < f(N_A))\ \{$
// Successful move
6: set $N_A = \lfloor N_A + \Delta \rfloor$
7: set $\Delta = \beta_1 \Delta$ }
8: $else\ \{$
// Unsuccessful move
9: set $\Delta = \beta_2 \Delta$ }
10: $endif$
11: $if(|\Delta| \leq \epsilon)$ Stop, Return N_A
12: $else$ Go to Step 5

8.3.2 Optimizing 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 low as possible. In the following, we formulate an optimization problem to minimize the number of SA-TCP aggregators N_A for certain requirements on loss rate and packet delay.

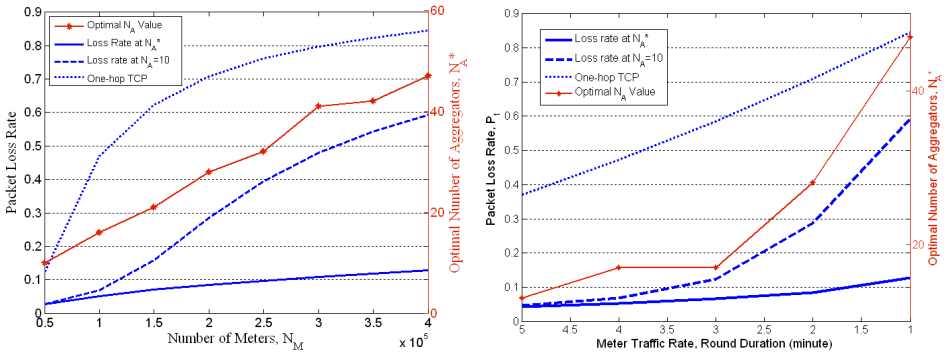


Figure 8.7: Optimized Loss Rate

$$\min N_A \tag{8.5}$$

Subject to

$$Ed(\mu_1, \rho_m, B_1) + Ed(\mu_2, \rho_a, B_2) \leq D \tag{8.6}$$

$$1 - (1 - P_m)(1 - P_a) \leq L \tag{8.7}$$

$$N_A \in \{1, 2, ..., N_M\}$$

Inequality (8.6) 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. Equations (6.9) and (6.8) define how delay is calculated. Inequality (8.7) is defined in Equation (8.2) constraining the maximum percentage of packet loss allowed.

Figures 8.4b and 8.4c 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 8.1. The algorithm searches for the minimum loss rate first. When the loss rate constraint is satisfied, then the delay constraint is checked. At this point delay peaks. For this reason and for the sake of minimizing, 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.

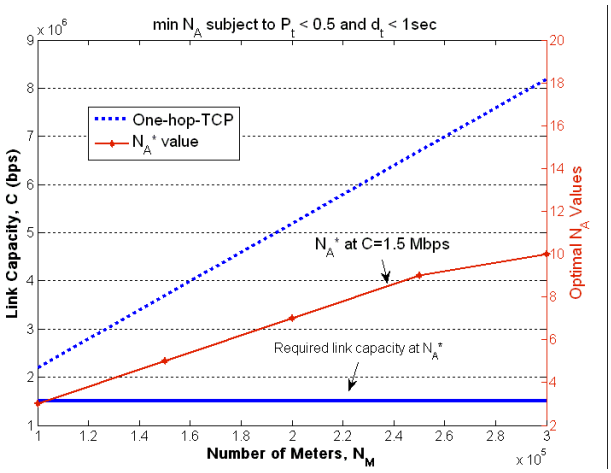


Figure 8.8: Optimized Number of Aggregators Example

Figure. 8.8 shows that by this optimization model, the packet loss rate and delay can be guaranteed at a low link capacity. The figure shows that for a one-hop-TCP scheme meet the same constraints, a significantly higher link capacity is required.

8.3.3 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. Again, the solution must meet acceptable loss and delay metrics.

$$\min \zeta_A N_A + \sum_i x_i \zeta_i C_i \tag{8.8}$$

Subject to

$$Ed(\mu_1, \rho_m, B_1) + Ed(\mu_2, \rho_a, B_2) \leq D \tag{8.9}$$

$$1 - (1 - P_m)(1 - P_a) \leq L \tag{8.10}$$

$$\sum_i x_i \leq 1 \tag{8.11}$$

$$N_A \leq N_M$$

$$N_A, \zeta \in \mathcal{Z}$$

where ζs represent the cost of the deployment of SA-TCP aggregators and reservation of link capacities, respectively, and C_i is the capacities of the link (*e.g.,* 128 Kbps, 256 Kbps, 1 Mbps, etc.). The cost of an SA-TCP aggregator depends on how sophisticated and powerful the device is (*e.g.,* as powerful as a circuit-level hardware-based proxy server). Inequality (8.11) ensures that the solver evaluates the cost with a specific link capacity by setting only one of the x_is to 1. Inequalities (8.9) and (8.10) constrain the delay and loss rate.

The objective function is a monotonically increasing function in the number of SA-TCP aggregators and the link bandwidth, and their corresponding costs. The impact of the number of SA-TCP aggregators and link capacity on the loss rate is shown in Fig. 8.9. The packet loss rate can be enhanced by increasing the link capacity, while by increasing the number of aggregators, it decreases to a certain point then increases again. For the delay metric, however, Fig. 8.10 shows that it gets worse with the increase of the number of aggregators while it is enhanced by increasing the link capacity. In both figures, it is evident that the impact of link capacity is not as effectual as that of the number of aggregators. From Fig. 8.8, it is clear that having the right number of SA-TCP aggregators saves on the cost of installing large link capacities.

Solving this optimization problem can be achieved by any non-linear integer programming solver (*e.g.,* Genetic Algorithm). Alternatively, we use our understanding that the optimal minimum cost can be obtained by increasing the number of aggregators and the link capacity (discretized to 128kbps or as appropriate per step) in iterations. The stopping criteria must satisfy both delay and loss constraints. Algorithm 8.2 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 feasible 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 minimum. This algorithm seems to take $O(N_A C)$; however, as Figures 8.9 and 8.10 demonstrate, the number of aggregators makes a noticeable quick change in loss and delay at small numbers. Additionally, given that the link capacity is a small range, it is expected to reach the solution much faster than the worst case analysis of $O(N_A C)$.

Algorithm 8.2: 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 $(N_A \leq N_M)$ { Go to Step 3 }
12: else { $C_i = C_{i+1}$, Go to Step 2 } }
13: Compare recorded costs and choose the minimum

8.4 Summary

Based on the mathematical model described in Chapters 6 and 7, this chapter has provided in-depth performance analysis of the proposed SA-TCP scheme. It has shown the impact of various design and network parameters and how the proposed scheme compares with the one-hop TCP scheme. This chapter also shows the variation the two famous versions of TCP – Reno and Vegas – make to SA-TCP. The performance results are shown in terms of throughput, packet loss rate and packet delivery delay.

The performance analysis has shown that tuning the scheme design is possible. Therefore, further use of the math model is achieved by formulating different optimization problems. The objective has been to minimize packet loss rates and to find the optimal deployment cost of the scheme. For those models, algorithms for solving the optimization problems have been described.

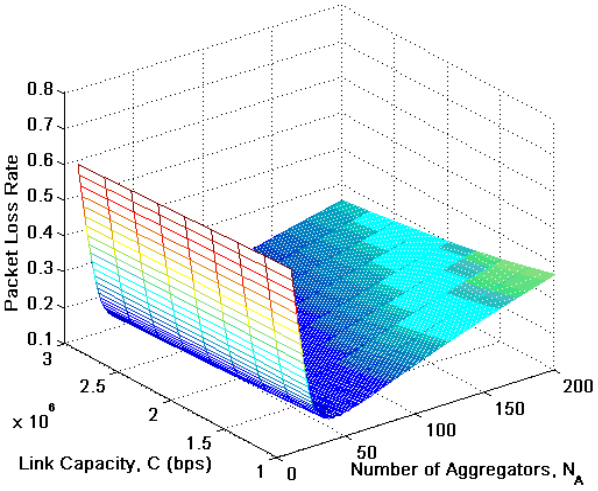


Figure 8.9: Impact of Number of SA-TCP Aggregators and Link Capacity on Loss Rate

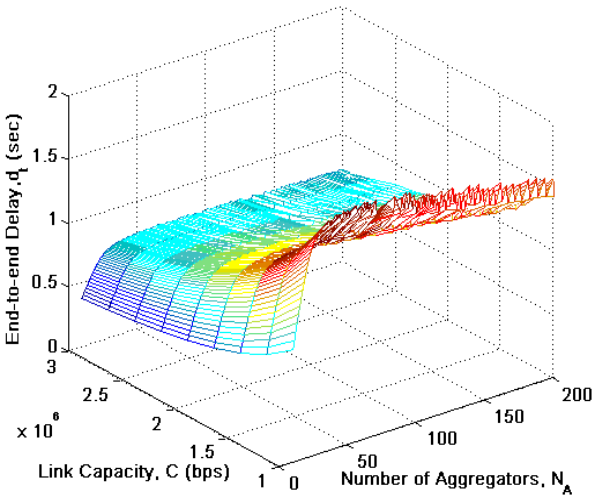


Figure 8.10: Impact of Number of SA-TCP Aggregators and Link Capacity on Delay

Chapter 9

Conclusion and Future Work

9.1 Summary and Conclusions

Smart power grids are gradually becoming a reality. The intelligence that will be added to the electricity grid makes the system highly desirable. In short, a smart power grid increases reliability and power quality, increases efficiency, improves responsiveness to failures, reduces cost for the utility provider and consumers, and enables handling current and future demand.

In order to have a successful and reliable end-to-end smart grid, one that is able to run applications between the utility and the consumer and the different components of the grid, an end-to-end communication network is needed. Over the past decades, a great deal of research has focused on lower layer issues, studying and proposing various communication technologies and routing solutions. **Chapters 2** and **3** review those proposals and discuss the major challenges that face scalable end-to-end communication.

This thesis makes a move upward to study the TCP layer of the TCP/IP suite. This research investigates the suitability of TCP in a smart metering infrastructure. **Chapter 4** provides an evaluation of the TCP protocol in an SMI. Supported by simulations and mathematics, the chapter has shown, in particular, the ineffectiveness of TCP congestion control. The significant number of smart meters make the flow of traffic TCP-unfriendly. The consequences are high packet losses and unfairness to other applications' traffic.

To fix the shortcomings of TCP in an SMI, **Chapter 5** has introduced the proposed novel scheme, called Split- and Aggregated-TCP (SA-TCP). The proposal argues that upgrading regional collectors to operate at TCP layer, combining TCP meters' TCP connections and forwarding data over a separate TCP connection to the utility server enhances

the situation. Simulations have shown that this scheme enables TCP congestion control to function effectively; consequently, performance has been improved. In addition to making congestion control effective, the proposed scheme achieves other improvements, such as improving the efficiency of data segment size and isolating wireless-related issues. However, observing how the performance changes, the chapter concludes that improving the scheme's performance is a matter of optimization and tuning of the design parameters. For this reason, representing the scheme mathematically was important.

Chapter 6 describes a mathematical model taking into account various parameters that impact SA-TCP's performance. The math model includes the application and TCP layers (following TCP-Reno) and the network characteristics, such as link capacities, propagation delay, buffering capacities, and the number of meters and SA-TCP aggregators. The model allows us to predict the actual behavior of the meters' traffic in SMI, depicted in terms of packet loss rate, offered load, and end-to-end delay performance metrics. In the same line, **Chapter 7** extends the model to investigate SA-TCP's performance under TCP-Vegas, which is another major TCP variant.

Based on the SA-TCP mathematical model, **Chapter 8** presents an in-depth performance analysis of the scheme. The results have made it clear how SA-TCP compares to a one-hop TCP scheme, and how SA-TCP reacts and performs under varying a number of setup parameters. Additionally, the chapter has shown the variation in performance TCP-Reno and TCP-Vegas bring. Although Vegas is proactive in handling congestion, because it is more aggressive than Reno, it does not perform as well as TCP-Reno in a one-hop-TCP scheme or in the first hop of an SA-TCP scheme. Knowing that TCP-Vegas causes more losses, meters should run less aggressive TCP variants. Based on understanding what impacts the scheme's performance and using the math model, optimization comes into the picture. The chapter has introduced the idea of minimizing packet loss rate and minimizing the deployment cost of the scheme, while satisfying certain performance constraints.

All in all, this thesis contributes to the development of SMI in the following manner. It evaluates the use of two major TCP versions (Reno and Vegas) in an SMI. It concludes that TCP does not scale to a significant number of smart meters and results in serious performance degradation. SA-TCP is presented as a solution that addresses the ineffectiveness of TCP congestion control. However, to ensure satisfactory performance, the SA-TCP scheme has been mathematically modelled end-to-end, and then comprehensively investigated, and lastly, optimized.

Although SMI has been the focus, the proposal of SA-TCP is applicable to other applications that are similar in nature to SMI, that is, any application characterized by the existence of a large number of Internet Protocol (IP) devices sending data packets at a

low rate to a centralized server. Examples include collecting broadcast quality information for TV over Internet Protocol (IP) (e.g., loss rate and average delay) or collecting measurements from network monitoring devices or meteorological sensors (*e.g.,* monitoring of weather, pollution and allergy conditions).

9.2 Future Research Work

The overall objective of this research is to enhance the performance of TCP in an SMI. This objective has been achieved through the design and optimization of the SA-TCP scheme. The work presented in this thesis, however, still has the potential for various extensions and directions for future work, as listed in the following bullet points.

- Because SMI is a heterogeneous network, the analytical model needs to be extended to involve environment-specific issues. For example, wireless and PLC links are described as noisy, so losses are not limited to congestion. Therefore, further study is needed to assess TCP and SA-TCP under a heterogeneous setting and investigate other TCP variants that are more suitable for these environments.

- Fairness of SA-TCP is yet to be defined and studied. First, the study should examine whether every meter receives the same share of bandwidth. Every meter in a region receives as much bandwidth as its regional aggregator offers. The question then is whether meters in different regions receive equal shares of the SMI bandwidth, and whether certain meters or certain types of data should, in fact, be given higher priorities. An SA-TCP aggregator's scheduler (depicted in Fig. 5.2a) can play a role in the topics of fairness and prioritization. The second point of interest is to examine SA-TCP fairness with other types of traffic in the smart grid. It is apparent that as the number of aggregators increases, a larger share of bandwidth is taken by the smart meters.

- The optimization model presented here takes into consideration the number of SA-TCP aggregators and link capacities. It is possible to extend this model to include other parameters such as buffering capacities and distances to meters. The dimensionality of the problem increases, but it is worth studying how every parameter affects performance.

- TCP congestion control can still be improved by looking into other TCP techniques and other TCP variants that suit the smart grid environment. In SA-TCP, reducing

the chance of congestion in a region of meters is achieved by reducing the number of meters. A possible future direction is to develop a technique to let congestion control be managed by an SA-TCP aggregator, even though it is a receiver, instead of meters.

References

[1] CellNet + Hunt Data Systems Inc. In *www.cellnethunt.com*.

[2] DLMS user association. In *www.dlms.com*.

[3] IEC - international electrotechnical commission. In *www.iec.ch*.

[4] Leach industries. In *www.leachindustries.com*.

[5] Pacific gas & electricity. In *http://www.pge.com/*.

[6] A. Dunkels A, J. Alonso, and Voight T. Making TCP/IP viable for wireless sensor networks. In *European Workshop on Wireless Sensor Networks*, Jan. 2004.

[7] R. Kokku A. Venkataramani and M. Dahlin. Tcp nice: A mechanism for background transfers. *Operating Systems Review*, 36:329–344, 2002.

[8] A. Abdollahi, M. Dehghani, and N. Zamanzadeh. SMS-based reconfigurable automatic meter reading system. In *IEEE int. conf. on Control Applications*, Oct. 2007.

[9] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock. Host-to-Host congestion control for TCP. *IEEE Communications Surveys Tutorials*, 12(3):304–342, 2010.

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

[11] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. In *IEEE Commun. Magazine*, volume 40, pages 102–114.

[12] A. Al-Yasiri and A. Sunley. Data aggregation in wireless sensor networks using the SOAP protocol. In *Journal of Physics: Conference Series 76 012039*, 2007.

[13] M. Allalouf, G. Gershinsky, L. Lewin-Eytan, and J. Naor. Data-quality-aware volume reduction in smart grid networks. In *IEEE Int. Conf. on Smart Grid Communications*, pages 120–125, oct. 2011.

[14] M. Allman, V. Paxson, and E. Blanton. TCP congestion control. *IETF RFC 5681*, Apr 2009.

[15] J. Andrews, A. Ghosh, and R. Muhamed. Fundamentals of WiMAX. In *book, ISBN: 0-13-222552-2*. Prentice Hall, 2007.

[16] M. Baker. Added value services through the use of AMR in commercial and industrial accounts. In *Int. Conf. on Metering and Tariffs for Energy Supply*, May. 1999.

[17] A. Bakre and B. Badrinath. I-TCP: Indirect tcp for mobile hosts. In *Distributed Computing Systems*, 1995.

[18] G. Barbose, C. Goldman, and B. Neenan. A survey of utility experience with real time pricing. Technical report, Lawrence Berkeley National Laboratory, Dec. 2004.

[19] Jacir L. Bordim and Koji Nakano. Fundamental protocols to gather information in wireless sensor networks. In *Sensor Network Protocols*, chapter 6. Springer, 2006.

[20] R. Braden. Requirements for internet hosts communication layers. In *IETF RFC 1122*, Oct 1989.

[21] L. Brakmo and L. Peterson. Tcp vegas: end to end congestion avoidance on a global internet. *IEEE Journal Selelected Areas in Comm.*, 13(8), 1995.

[22] L.S. Brakmo and L.L. Peterson. TCP Vegas: end to end congestion avoidance on a global internet. *Selected Areas in Communications, IEEE Journal on*, 13(8):1465–1480, Oct 1995.

[23] C. Brasek. Urban utilities warm up to the idea of wireless automatic meter reading. *Computing and Control Engineering*, 15(6):10–14, Jan. 2005.

[24] J. Case, M. Fedor, M. Schoffstall, and J. Davin. A simple network management protocol (snmp). In *IETF RFC 1157*, May 1990.

[25] Claudio Casetti and Michela Meo. An analytical framework for the performance evaluation of tcp reno connections. *Computer Networks*, 37(5):669–682, 2001.

[26] Ann Cavoukian. Operationalizing privacy by design: The ontario smart grid case study. In *Information and Privacy Commissioner, Ontario, Canada*, Feb 2011.

[27] Rajiv Chakravorty, Sachin Katti, Ian Pratt, and Jon Crowcroft. Flow aggregation for enhanced tcp over wide area wireless. In *INFOCOM*, 2003.

[28] T. Chandler. The technology development of automatic metering and monitoring systems. In *IEEE International Power Engineering Conference*, Dec. 2005.

[29] Xiangqian Chen, Kia Makki, Kang Yen, and Niki Pissinou. Sensor network security: a survey. *IEEE Communications Surveys and Tutorials*, 11(2):52–73, 2009.

[30] S. Chessa and P. Santi. Crash faults identification in wireless sensor networks. *Computer Communication*, 25(14):1273–1282, Sep. 2002.

[31] M. Choi, S. Ju, and Y. Lim. Design of integrated meter reading system based on power line communication. In *IEEE International Symposium on Power Line Communications and Its Applications*, Apr. 2008.

[32] COMETECH M2M. Machine-to-machine (m2m) communication solutions: Monitor, control and manage any remote equipment. 2008.

[33] Companion Specification for Energy Metering. DLMS/COSEM architecture and protocols. In *Green Book, DLMS UA, 1997-2007.*

[34] Companion specification for energy metering. Identification system and interface classes. In *Blue book, DLMS User Association, 1997-2007.*

[35] James G. Cupp and Mike E. Beehler. Implementing smart grid communications. In *Burns & McDonnell TECHBriefs*, 2008.

[36] E. Dahlman, S. Parkvall, J. Skold, and P. Beming. 3G evolution HSPA and LTE for mobile broadband. In *book ISBN: 978-0-12-372533-2.* Elsevier, 2008.

[37] G. Deconinck. An evaluation of two-way communication means for advanced metering in flanders (belgium). In *IEEE Instrumentation and Measurement Technology Conference Proceedings*, pages 900–905, 2008.

[38] N. Degrande, K. Laevens, D. De Vleeschauwer, and R. Sharpe. Increasing the user perceived quality for iptv services. *IEEE Communications Magazine*, 46(2), Feb 2008.

[39] M. Dohler, T. Watteyne, T. Winter, and D. Barthel. Routing requirements for urban low-power and lossy networks. In *IETF RFC 5548*, May 2009.

[40] EarthLink Research and Development. SIPshare: SIP beyond voice and video. In *http://www.research.earthlink.net/p2p/*, 2004.

[41] Engage Consulting Ltd. High-level smart meter data traffic analysis. In *Document Ref.: ENA-CR008-001-1.4*, May 2010.

[42] Engage Consulting Ltd. Smart metering system requirements update. In *ENA-CR006-002-1.1*, Apr 2010.

[43] M. Faisal and A. Mohamed. A new technique for power quality based condition monitoring. In *17th Conference of Electrical Power Supply Industry*, Oct. 2008.

[44] K. Fall and S. Farrell. DTN: An architectural retrospective. In *IEEE Journal on Selected Areas in Communications*, volume 26, pages 828–836, Jun. 2008.

[45] K. Fan, Sha Liu, and Prasun Sinha. Data aggregation in wireless sensor networks. In *Wireless Sensor Networks and Applications*, Signals and Communication Technology, pages 331–347. Springer, 2008.

[46] Z. Fan, P. Kulkarni, S. Gormus, C. Efthymiou, G. Kalogridis, M. Sooriyabandara, Z. Zhu, S. Lambotharan, and W. Chin. Smart grid communications: Overview of research challenges, solutions, and standardization activities. *Communications Surveys Tutorials, IEEE*, PP(99):1–18, 2012.

[47] Zhong Fan, Parag Kulkarni, Sedat Gormus, Costas Efthymiou, Georgios Kalogridis, Mahesh Sooriyabandara, Ziming Zhu, Sangarapillai Lambotharan, and Woon Hau Chin. Smart grid communications: Overview of research challenges, solutions, and standardization activities. *CoRR*, 2011.

[48] S. Floyd. Issues of tcp with sack. *Tech. Report*, Jan 1996.

[49] S. Floyd, T. Henderson, and A. Gurtov. Rfc 3782 the newreno modification to tcps fast recovery algorithm. In *RFC*, 2004.

[50] S. Floyd, S. Ratnasamy, and S Shenker. Modifying tcp's congestion control for high speeds. *Rough Draft*, May 2002.

[51] Sally Floyd and Kevin Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Trans. on Networking*, 7(4):458–472, August 1999.

[52] P. Fuchs and T. Schaub. DLMS user association - co-ordination between applications and channels. In *Int. Conf. on Metering and Tariffs for Energy Supply*, Aug. 1999.

[53] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1994.

[54] L. A. Grieco and S. Mascolo. Performance evaluation and comparison of westwood+, new reno and vegas tcp congestion control. In *ACM Computer Communication Review*, Apr 2004.

[55] Vehbi C. Gungor and Frank C. Lambert. A survey on communication networks for electric system automation. *Computer Networks*, 50(7):877–897, 2006.

[56] M. Handley and V. Jacobson. SDP: Session description protocol. In *IETF RFC 2327*, Apr. 1998.

[57] G. Hasegawa, K. Kurata, and M. Murata. Analysis and improvement of fairness between tcp reno and vegas for deployment of tcp vegas to the internet. In *IEEE ICNP*, 2000.

[58] G. T. Heydt. Virtual surrounding face geocasting in wireless ad hoc and sensor networks. *Electric Power Quality: A Tutorial Introduction*, 11(1):15–19, Jan. 1998.

[59] B. Hull, K. Jamieson, and H. Balakrishnan. Mitigating congestion in wireless sensor networks. In *ACM Sensys 04*, Nov. 2004.

[60] Chalermek Intanagonwiwat, Deborah Estrin, Ramesh Govindan, and John Heidemann. Impact of network density on data aggregation in wireless sensor networks. In *ICDCS*, 2002.

[61] Aravind Iyer, Sunil Kulkarni, Vivek Mhatre, and Catherine Rosenberg. A taxonomy-based approach to design of large-scale sensor networks. In *Wireless Sensor Networks and Applications*, Signals and Communication Technology, chapter 1, pages 3–33. Springer, Feb. 2008.

[62] V. Jacobson. Congestion avoidance and control. *ACM SIGCOMM*, pages 314–329, 1988.

[63] V. Jacobson. Modified tcp congestion avoidance algorithm. *end2end-interest mailing list*, Apr 1988.

[64] Raj Jain. A delay based approach for congestion avoidance in interconnected heterogeneous computer networks. *CoRR*, cs.NI/9809093, 1998.

[65] Peng Jiang. A new method for node fault detection in wireless sensor networks. *Sensors*, 9(2):1282–1294, 2009.

[66] C. Jin, D. Wei, and S. Low. Fast tcp: Motivation, architecture, algorithms, performance. In *IEEE INFOCOM 2004*, May 2004.

[67] Justin Jones and Mohammed Atiquzzaman. Transport protocols for wireless sensor networks: State-of-the-art and future directions. *International Journal of Distributed Sensor Networks*, 3(1):119–133, Jan. 2007.

[68] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *IASPLOS-X*, Oct 2002.

[69] S. Kerk. An AMR study in an Indian utility. In *IEEE Power Engineering Conference*, Dec. 2005.

[70] William H. Kersting. *Distribution System Modeling and Analysis*. CRC Press, 2012.

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

[72] Tarek Khalifa, Atef Abdrabo, Kshirasagar Naik, Maazen Alsabaan, Amiya Nayak, and Nishith Goel. Design and analysis of split- and aggregated-transport control protocol (sa-tcp) for smart metering infrastructure. In *IEEE SmartGridComm*, Nov 2012.

[73] Tarek Khalifa, K. Naik, and A. Nayak. A survey of communication protocols for automatic meter reading. *IEEE Communications Surveys and Tutorials*, 13(2):168–182, 2011.

[74] Tarek Khalifa, Kshirasagar Naik, Maazen Alsabaan, and Amiya Nayak. A transport control protocol suite for smart metering infrastructure. In *Malaysia Power Electronics, Industrial Electronics & Industrial Applications*, Apr 2011.

[75] Tarek Khalifa, Kshirasagar Naik, Maazen Alsabaan, Amiya Nayak, and Nishith Goel. Transport protocol for smart grid infrastructure. In *IEEE Int. Conf. on Ubiquitous and Future Networks (ICUFN)*, Jun 2010.

[76] E. Kim, D. Kaspar, C. Gomez, and C. Bormann. Problem statement and requirements for 6LoWPAN routing. In *6LoWPAN Working Group Internet Drafts*, Jul. 2009.

[77] Sung Kyung Kim. Automatic meter reading system and method using telephone line. In *United States Patent 7102533*, Sep. 2006.

[78] Y.-J. Kim and M. Thottan. SGTP: Smart grid transport protocol for secure reliable delivery of periodic real time data. In *Bell Labs Tech. Journal*, volume 16, pages 83–99, Dec. 2011.

[79] Young-Jin Kim, V. Kolesnikov, Hongseok Kim, and M. Thottan. SSTP: A scalable and secure transport protocol for smart grid data collection. In *IEEE Int. Conf. on Smart Grid Communications*, pages 161–166, Oct. 2011.

[80] J.F.C. Kingman. *Poisson Processes*. John Wiley & Sons, Ltd, 2005.

[81] Leonard Kleinrock. *Queuing Systems, Vol.1: Theory*. Wiley & Sons, New York, NY, USA, 1975.

[82] Donald Knuth. *The TEXbook*. Addison-Wesley, Reading, Massachusetts, 1986.

[83] B. Koay, S. Cheah, Y. Sng, P. Chong, P. Shum, Y. Tong, X. Wang, Y. Zuo, and H. Kuek. Design and implementation of bluetooth energy meter. In *Information Communication and Signal Processing*, Dec. 2003.

[84] B. Krishnamachari. Networking wireless sensors. In *Cambridge University Press*, Dec. 2005.

[85] A. Kuzmanovic and E. Knightly. Tcp-lp: low-priority service via end-point congestion control. *IEEE/ACM Transactions on Networking*, 14(4), 2006.

[86] Martin S. Lacher, Jörg Nonnenmacher, and Ernst W. Biersack. Performance comparison of centralized versus distributed error recovery for reliable multicast. *IEEE/ACM Trans. Netw.*, 8(2):224–238, 2000.

[87] Leslie Lamport. *LATEX — A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.

[88] Benoit Latre, Pieter De Mil, Ingrid Moerman, Bart Dhoedt, Piet Demeester, and Niek Van Dierdonck. Throughput and delay analysis of unslotted IEEE 802.15.4. *Journal of Networks*, 1(1):20–28, May. 2006.

[89] S. Lee, C. Wu, M. Chiou, and K. Wu. Design of an automatic meter reading system. In *Proceedings of the IEEE IECON*, Aug. 1996.

[90] David J. Leeds. The smart grid in 2010: Market segments, applications and industry players. *GTM Research*, July 2009.

[91] P. Levis, A. Tavakoli, and S. Dawson-Haggerty. Overview of existing routing protocols for low power and lossy networks. In *Networking Working Group*, Apr. 2009.

[92] J. Lian, Y. Liu, K. Naik, and L. Chen. Virtual surrounding face geocasting in wireless ad hoc and sensor networks. *IEEE/ACM Trans. on Networking*, 17(1):200–211, Feb. 2009.

[93] J. Lian and K. Naik. Skipping technique in face routing for wireless ad hoc and sensor networks. *International Journal of Sensor Networks*, 4(1/2):92–103, Jul. 2008.

[94] Antonio Liotta, Danil Geelen, Gert van Kempen, and Frans van Hoogstraten. A survey on networks for smart-metering systems. *International Journal of Pervasive Computing and Communications*, 8(1):23–52, 2012.

[95] Y. Liu, R. Fischer, and N. Schutz. Distribution system outage and restoration analysis using a wireless AMR system. In *IEEE Power Engineering Society Winter Meeting*, Aug. 2002.

[96] Belvin Louie. MDMS meeting the meter data management challenge. In *Metering International*, volume 2, 2009.

[97] Gary Lutz and Matt Schwarz. Aami and mdms deployment best practices. In *EcoLogic Analytics*, May 2009.

[98] Samuel Madden, Michael Franklin, Joseph Hellerstein, and Wei Hong. TAG: A tiny aggregation service for ad-hoc sensor networks. In *ACM SIGOPS Operating Systems Review*, volume 36, 2002.

[99] Michal MAJCHRAK, Jozef HEINRICH, Peter FUCHS, and Vladimir HOSTYN. Single phase electricity meter based on mixed-signal processor msp430fe427 with PLC modem. In *17th Int. Conf. Radio elektronika*, Apr. 2007.

[100] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang. Tcp westwood: Bandwidth estimation for enhanced transport over wireless links. In *ACM MOBICOM*, 2001.

[101] S. McCanne and S. Floyd. Ns network simulator. http://www.isi.edu/nsnam/ns/.

[102] X. Meng, P. Zerfos, V. Smanta, S. Wong, and S. Lu. Analysis of the reliability of a nationwide short message service. In *IEEE INFOCOM*, May. 2007.

[103] V. Mhatre and C. Rosenberg. Homogeneous vs heterogeneous sensor networks: A comparative study. In *Int. Conf. on Communications (ICC)*, Jun. 2004.

[104] V. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff. A minimum cost surveillance sensor network with a lifetime constraint. In *IEEE Transactions on Mobile Computing*, Jan. 2005.

[105] A. Misra, T. Ott, and J. Baras. The window distribution of multiple tcps with random loss queues. In *Global Telecommunications Conference, 1999. GLOBECOM '99*, volume 3, pages 1714–1726, 1999.

[106] T. Moghavvemi. PIC-based automatic meter reading and control over the low voltage distribution network. In *Student Conference on Research and Development*, Jul. 2002.

[107] Dmitri Moltchanov. A study of tcp performance in wireless environment using fixed-point approximation. *Computer Networks*, 56(4):1263–1285, 2012.

[108] J. Nagle. Congestion control in IP/TCP internetworks. In *IETF RFC 896*, Jan 1984.

[109] D. Niyato, Lu Xiao, and Ping Wang. Machine-to-machine communications for home energy management system in smart grid. *Communications Magazine, IEEE*, 49(4):53–59, Apr 2011.

[110] P. Oksa, M. Soini, L. Sydanheimo, and M. Kivikoski. Considerations of using power line communication in the AMR system. In *IEEE International Power Line Communications and Its Applications*, Oct. 2006.

[111] B. Park, D. Hyun, and S. Cho. Implementation of AMR system using power line communication. In *IEEE/PES Transmission and Distribution Conference and Exibition*, Oct. 2002.

[112] Seung-Jong Park, Ramanuja Vedantham, Raghupathy Sivakumar, and Ian F. Akyildiz. A scalable approach for reliable downstream data delivery in wireless sensor networks. In *MobiHoc '04*, May. 2004.

[113] R. Pries, D. Staehle, and D. Marsico. IEEE 802.16 capacity enhancement using and adaptive TDD split. In *IEEE Vehicular Technology Conference*, May. 2008.

[114] G. Raja and T. Sudhakar. Electricity consumption and automatic billing through power line. In *Internation Power Engineering Conference*, Dec. 2007.

[115] Danny Relich. Smart meters on a roll in canada. In *Hydro One Networks Inc.*, Sep 2008.

[116] I. Rhee and L. Xu. Cubic: A new tcp-friendly high-speed tcp. In *Workshop on Protocols for Fast Long-Distance Networks*, Feb 2005.

[117] A. Roach. SIP-specific event notification. In *IETF RFC 3265*, Jun. 2002.

[118] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. In *IETF RFC 3261*, Jun. 2002.

[119] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.

[120] Smruti R. Sarangi, Partha Dutta, and Komal Jalan. IT infrastructure for providing energy-as-a-service to electric vehicles. *IEEE Trans. Smart Grid*, 3(2):594–604, 2012.

[121] B. Sarikaya and X. Zheng. SIP paging and tracking of wireless lan hosts for VoIP. *IEEE/ACM Trans. on Networking.*, 16(3):539–548, Jun. 2008.

[122] M. Sbai and C. Barakat. Experiences on enhancing data collection in large networks. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 3(7):1073–1086, May. 2009.

[123] Mohamed Sbai and Chadi Barakat. Experiences on enhancing data collection in large networks. *Comp. Networks*, 53(7):1073–1086, 2009.

[124] H. Schulzrinne, S. Casner, R. Fredrick, and V. Jacobson. RTP: A transport protocol for real-time applications. In *IETF RFC 3550*, Jul. 2003.

[125] J. Selga, A. Zaballos, G. Corral, and J. Vives. Lessons learned from wireless sensor networks with application to AMR and PLC. In *IEEE International Symposium on Power Line Communications and Its Applications*, Mar. 2007.

[126] Mohammad Shahraeini, Mohammad Sadegh Ghazizadeh, and Mohammad Hossein Javidi. Co-optimal placement of measurement devices and their related communication infrastructure in wide area measurement systems. *IEEE Trans. Smart Grid*, 3(2):684–691, 2012.

[127] H. Shimonishi, M. Sanadidi, and M. Gerla. Improving efficiency-friendliness tradeoffs of tcp in wired-wireless combined networks. In *IEEE ICC*, May 2005.

[128] K. Singh and H. Schulzrinne. Peer-to-peer Internet telephony using SIP. In *Workshop on Network and operating systems support for digital audio and video*, Jun. 2005.

[129] S. Soh and S. Kerk. The electricity and metering trends in Singapore. In *IEEE Power Engineering Conference*, Dec 2005.

[130] Oliver Spatscheck, Jørgen S. Hansen, John H. Hartman, and Larry L. Peterson. Optimizing tcp forwarder performance. *IEEE/ACM Trans. Netw.*, 8(2):146–157, Apr 2000.

[131] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sumanasekera, and L. Vicisano. PGM reliable transport protocol specification. In *IETF RFC 3208*, Jul. 2007.

[132] Q. Spencer. An information-theoretic analysis of electricity consumption data for an AMR system. In *IEEE International Symposium on Power Line Communications and its Applications*, Apr. 2008.

[133] K. Srijith, L. Jacob, and A. Ananda. Tcp vegas-a: Improving the performance of tcp vegas. *Computer Communications*, 28(4), 2005.

[134] W. Stallings. high-speed networks and internets performance and quality of service. In *2nd edition, Pearson Education*, 2004.

[135] F. Stann and J. Heidemann. RMST: Reliable data transport in sensor networks. In *IEEE International Workshop on Sensor Network Protocols and Applications*, May. 2003.

[136] J. Surrat. Integration of cebus with utility load management and automatic meter reading. In *IEEE Transaction on Consumer Electronics*, volume 37, pages 406–412, Aug. 1991.

[137] P. Szczechowiak, A. Kargl, M. Scott, and M. Collier. On the application of pairing based cryptography to wireless sensor networks. In *WiSec '09: Proceedings of the second ACM conference on Wireless network security*, Mar. 2009.

[138] H. Tan, H. Lee, , and V. Mok. Automatic power meter reading system using GSM network. In *IEEE International Power Engineering Conference*, Dec. 2007.

[139] Liansheng Tan, Li Jin, and Yi Pan. Efficient placement of proxies for hierarchical reliable multicast. *Comp. Comm.*, 31(9):1842–1855, 2008.

[140] The Independent Electricity System Operator. Smart metering start-up guide. Apr. 2009.

[141] C. Y. Wan, S. B. Eisenman, and A. T. Campbell. CODA: Congestion detection and avoidance in sensor networks. In *The First Int. Conf. on Embedded Networked Sensor Systems (SenSys03)*, 2003.

[142] C. Wang, K. Sohraby, and B. Li. SenTCP: A hop-by-hop congestion control protocol for wireless sensor networks. In *IEEE INFOCOM*, 2005.

[143] P. Wang, J.Y. Huang, Y. Ding, P. Loh, and L. Goel. Demand side load management of smart grids using intelligent trading/metering/ billing system. In *IEEE Power and Energy Society General Meeting*, July 2010.

[144] Z. Wang and J. Crowcroft. Eliminating periodic packet losses in 4.3 tahoe bsd tcp congestion control. *ACM Computer Communication*, 22(2), 1992.

[145] A. Wasnarat and Y. Tipsuwan. A power efficient algorithm for data gathering from wireless water meter networks. In *IEEE Int. Conf. on Industrial Informatics*, Aug. 2006.

[146] A. Wierman and T. Osogami. A unified framework for modeling tcp-vegas, tcp-sack, and tcp-reno. In *MASCOTS*, pages 269–278, Oct. 2003.

[147] Michael I. Taksar Winfried K. Grassmann and Daniel P. Heyman. Regenerative analysis and steady state distributions for markov chains. *Operations Research*, 33(5):1107–1116, 1985.

[148] T. Winter and ROLL Design Team. RPL: Routing protocol for low power and lossy networks. In *Networking Working Group Internet drafts*, Jul. 2009.

[149] M. Yamanouchi, S. Matsuura, and H. Sunahara. A fault detection system for large scale sensor networks considering reliability of sensor data. In *Applications and the Internet SAINT '09*, Jul. 2009.

[150] J. Yu, P. Chong, P. So, and E. Gunawan. Solution for the 'silent node' problem in automatic meter reading system using power line communications. In *IEEE International Power Engineering Conference*, Dec. 2005.

[151] Agustin Zaballos, Alex Vallejo, Marta Majoral, and Josep Selga. Survey and performance comparison of AMR over PLC standards. *IEEE Trans. on Power Delivery*, 24(2):604–613, 2009.

[152] P. Zerfos, X. Meng, S. Wong, V. Samanta, and S. Lu. A study of the short message service of a nationwide cellular network. In *ACM SIGCOMM Internet Measurement Conference*, Oct. 2006.

[153] Z. Zhang, M. Ma, and Y. Yang. Energy efficient multi-hop polling in clusters of two-layered heterogeneous sensor networks. *IEEE Transactions on Computers*, 57(2):231–245, Feb. 2008.

[154] M. Zhao, M. Ma, and Y. Yang. Mobile data gathering with space-division multiple access in wireless sensor networks. In *IEEE INFOCOM*, Apr. 2008.

[155] J. Zhu and R. Pecen. A novel automatic utility data collection system using IEEE 802.15.4-compliant wireless mesh networks. In *Proceedings of the IAJC-IJME Int. Conf.*, Nov. 2008.