TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

KUANYSH KAIRBEK

TCP PERFORMANCE IN 5G MMWAVE SYSTEMS WITH DYNAMIC BLOCKAGE

Master of Science thesis

Examiners:
Sr. Research Fellow Dmitri Moltchanov
Examiners and topic approved by the Faculty Council of Faculty of
Electronics and Communications Engineering
on 30th November 2017

ABSTRACT

KUANYSH KAIRBEK: TCP Performance in 5G mmWave Systems with Dynamic Blockage
Tampere University of Technology
Master of Science thesis, 55 pages
May 2018
Master's Degree Programme in Information Technology
Major: Communication Systems and Networks
Examiners: Sr. Research Fellow Dmitri Moltchanov
Keywords: mmwave, tcp, ns-3, wireless networks, direct communications, prototyping, performance evaluation

The next-generation 5G network is being tested although the specifications of 5G have not been formally defined. Researchers and industry expect that new generation of technology will come to the market much sooner than expected 2020 in this quickly changing world. It may seem as a long ways to go but time flies and so will 5G at speeds of 10 Gbps.

Cisco provides IP Traffic Forecast and other data in charts that show traffic from wireless and mobile devices will account for more than 63 percent of total IP traffic by 2021. By that time, wired devices will account for 37 percent of IP traffic, while WLAN and mobile devices will account for 63 percent of IP traffic. In 2016, wired devices accounted for the majority of IP traffic at 51 percent [1].

Majority of the devices will be wireless, so it causes much overhead to the networks and mobility of the users will make it even more dynamic. Because the networks are dynamic and we need to be sure that our data not only fast but also correctly delivered to the end point. TCP used in dynamic networks. TCP stands for Transmission Control Protocol and one of the the most widely spread protocols in the Internet. TCP guarantees the recipient will receive the packets in order. That is why it is used in dynamic networks where error correction is necessary.

5G will operate on ultra-high frequency millimeter wave spectrum. The most recent 802.11ad Wi-Fi standard operating at 60 GHz showed that the spectrum between microwave and infrared waves can be used for high-speed wireless communications. The spectrum between 30 GHz and 300 GHz has a large amount of available band-

width. This gives mmWave communications a major advantage over the current LTE standard.

The 3rd Generation Partnership Project (3GPP) posted 'Study on channel model for frequency spectrum above 6 GHz', 3GPP TR 38.900 version 14.2.0 Release 14, where model and evaluation the performance of physical layer techniques using the above-6GHz channel models. This document relates to the 3GPP evaluation methodology and covers the modeling of the physical layer of both Mobile Equipment and Access Network of 3GPP systems.

This work targets the TCP performance in 5G mmWave networks with dynamic blockage represented by human bodies and cause additional fluctuations in the channel and signal level. To reduce prices and time of experiments based on mmWave techniques modeling can be used. To achieve that ns-3 was utilized as a tool of modeling and simulations.

PREFACE

The research work presented in this thesis was carried out during the period from November 2017 to May 2018 at the Department of Electronics and Communications Engineering, Tampere University of Technology (TUT), Finland.

First of all, I would like to express my deep gratitude to my supervisor, Sr. Research Fellow Dmitri Moltchanov, whose expertise and motivation helped me a lot during the work.

My sincere thanks go to members of the Department of Electronics and Communications Engineering for their great assistance and guidance through difficulties that I faced. Without them, my progress would have never been as strong during the work, especially, to Doctoral Student Margarita Gapeyenko and Doctoral Student Aleksandr Ometov.

I would like to express my deepest feelings to my family for believing in me, supporting and never letting me down.

Tampere, 20.03.2018
Kuanysh Kairbek

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3GPP	The 3rd Generation Partnership Project
5G	5th Generation Network
AP	Access point
BS	Base Station
CDF	Cumulative distribution function
D2D	Device-to-device
DHCP	Dynamic Host Configuration Protocol
DL	Downlink
E2E	End-to-end
EHF	Extremely high frequency
eNode	Evolved Node
EPC	Evolved Packet Core
IEEE	Institute of Electrical and Electronics Engineers
ISM	Industrial, Scientific, and Medical
ITU	International Telecommunications Union
LoS	Line-of-Sight
LTE	Long Term Evolution
MIMO	Multiple-Input and Multiple-Output
nLoS	non-Line-of-Sight
P2P	peer-to-peer
QoS	Quality of Service
Rx	Receiver
SINR	Signal-to-Interference-Plus-Noise Ratio
Tx	Transmitter
UE	User Equipment
UL	Uplink
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network

LIST OF SYMBOLS

λ	Arrival intensity of blockers entering the LoS
ω_j, η_j	The non-blocked and blocked time interval
ξ	jth time interval equal to $\omega_j + \eta + j$
$F_T(x)$	CDF of LoS zone residence time
$F_\eta(x)$	CDF of blocked time interval
$F_L(x)$	CDF of the residence distance L
d_m	Diameter of blockers
L	Distance walked by a blocker in LoS blockage zone
r_0	Two-dimensional distance between Tx and Rx
α	Angle between Y-axis and the segment Tx-Rx
ω_E, r	Effective width and length of LoS blockage zone
$E[T_l], E[T_n]$	Fraction of time in non-blocked/blocked states
$E[\omega], E[\eta]$	Mean of non-blocked/blocked time interval

1. INTRODUCTION

In recent years we have been witnessing current frequency spectrum limitations and increasing amount of users and devices connected to the Internet. According to Cisco IP Traffic Forecast and other data in charts, traffic from wireless and mobile devices will account for more than 63 percent of total IP traffic by 2021 and total traffic will be three times larger than 2016 [1]. So, the majority of the devices will be wireless, and the current frequency spectrum is occupied very tight. Such challenges could be solved by next-generation 5G networks by the standardization communities. Researchers together with Federal Communications Commission heading towards extending bandwidth for faster delivery with high-quality of the multimedia content and services in the future next-generation 5G networks.

The most recent 802.11ad Wi-Fi standard operating at 60 GHz showed that the spectrum between microwave and infrared waves can be used for high-speed wireless communications [2, 3]. The extremely high frequency spectrum (from 30 GHz up to 300 GHz) has a large amount of available bandwidth [4].

The first thing to keep in mind is a significant amount of available bandwidth [5]. This gives mmWave communications a significant advantage over the current LTE standard. The second thing to consider is that mmWave operating on ultra-high frequencies and affected not only by big obstacles but also by trees, human bodies and even drop of water or air itself.

Majority of applications and services in the Internet using TCP what results that more than 90 percent of traffic on the Internet is handled by TCP. Thus, TCP controls the lion's piece of bytes and packets transmitted over the Internet, so to satisfy user needs and to provide quality of service (QoS) in modern networks predict TCP behavior and optimize its performance in different environments is very important. TCP guarantees the recipient will receive the packets in order. That is why it is used in dynamic networks where error correction is necessary [6].

To predict the behavior of TCP in mmWave systems, we need to evaluate its performance. TCP performance assessment can be divided into empirical studies and analytical modeling. Although analytical modeling is exceptionally accurate and useful to try different scenarios by changing parameters, it is difficult to create such a model, and there can be an option when the entire model has to be recalculated due to the crucial parameter. In this case, simulation is more beneficial because it requires less time and cost. Another advantage of modeling is re-usability once a model is created. By changing input parameters a wide range of scenarios can be executed with further output and results in analysis. TCP performance in 5G mmWave network will be evaluated via modeling to reduce time and price of the experiments.

The primary goals of this work are as follows. The author aims to provide a reliable model to simulate the mmWave behavior and TCP performance in 5G mmWave networks; flexible and repeatable modeling using created simulator.

The thesis is organized as follows. The author firstly presents a technological overview of the TCP and the corresponding challenges in Chapter 2. The Chapter 3 provides insight into TCP performance assessment framework and justifies our choices concerning simulation modeling. Chapter 4 is devoted to the TCP performance evaluation as well as discusses the corresponding results. The last Chapter concludes this thesis work.

2. TECHNOLOGY AND MOTIVATION

Nowadays, Internet data traffic is increasing by 50 percent per year for every subscriber, and the lion's share of data traffic comes from wireless connections [1]. This trend is going to increase even more in the following years due to increasing online video usage and the rise of the Internet-of-Things (IoT) [7, 8].

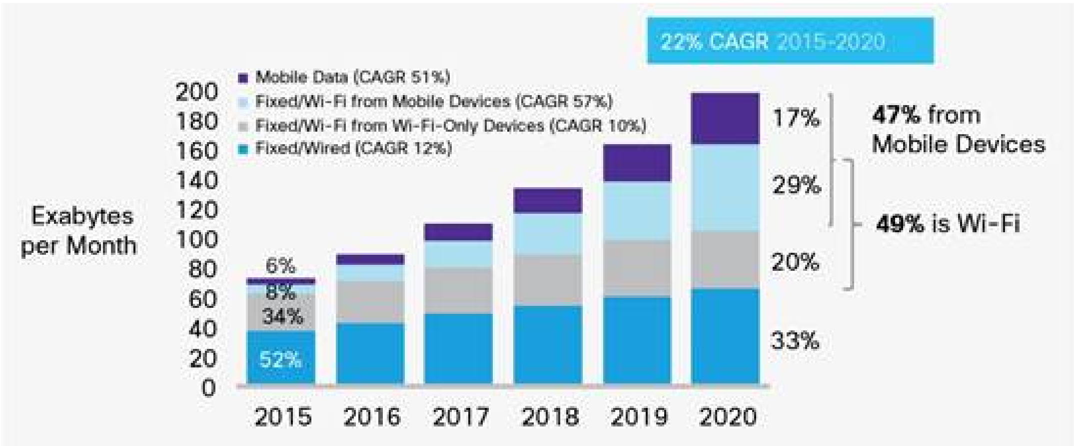

Figure 2.1 *IP Traffic by Access Technology*

Increasing amount of wireless subscribers will lead to massive amount of traffic (see Figure 2.1), so technology need not only to perform on high speed but also be able to support such amount of interference caused by wireless devices. The industry is moving to next-generation cellular communication technology – to the fifth generation, a.k.a. 5G, fulfilling a demand. In addition to increasing amount of traffic and number of subscribers there are limitations of the currently used frequency spectrum (see Figure 2.2). The frequency spectrum of today is already occupied very tight, so there is almost no space for usage.

5G mmWave was introduced to the world as an option to avoid current limitations [9]. The most recent 802.11ad Wi-Fi standard operating at 60 GHz showed that the spectrum between microwave and infrared waves can be used for high-speed wireless communications [2, 3]. The spectrum between 30 GHz and 300 GHz has a

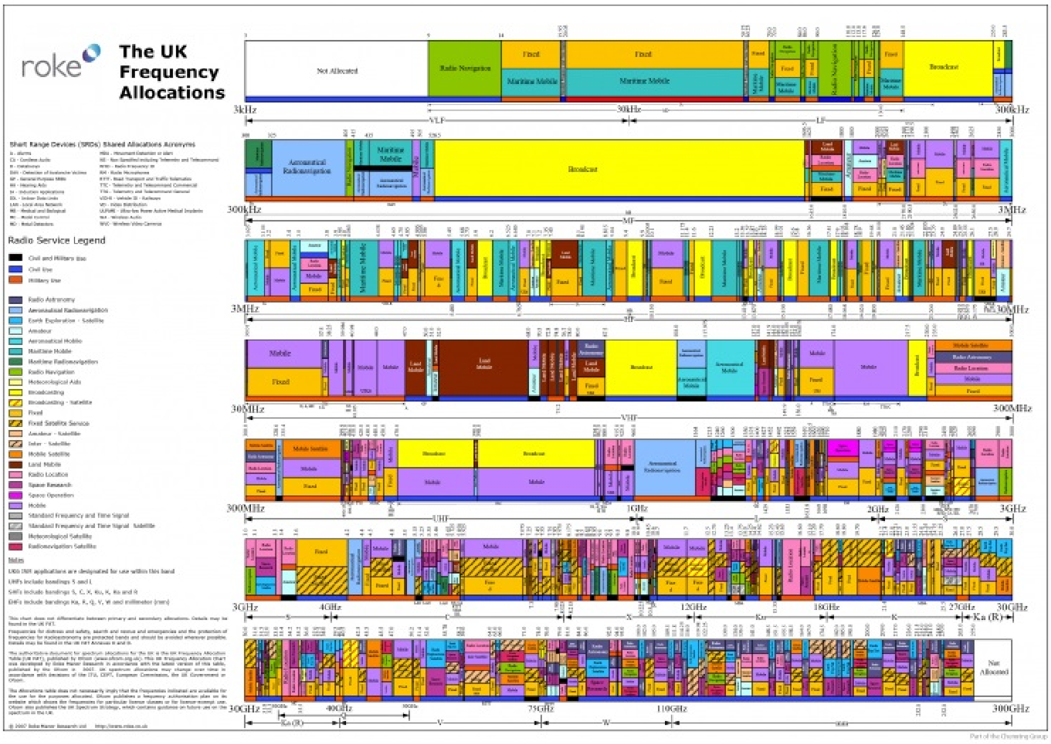

Figure 2.2 Currently used frequency spectrum, reproduced from ofcom.org.uk

large amount of available bandwidth [4].

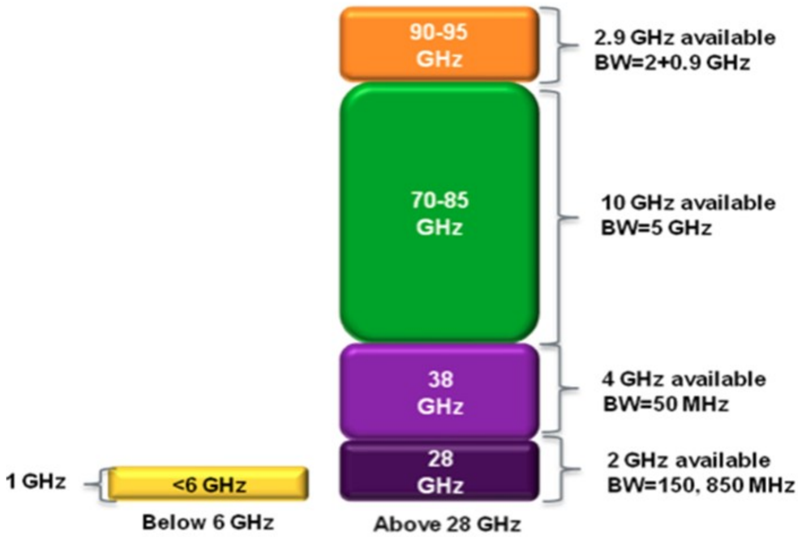

Figure 2.3 Available frequency spectrum

The first thing to observe from Figure 2.3 is the massive amount of available mmWave bandwidth. This gives mmWave communications a significant advantage over the current LTE standard in terms of standardization and general flexibility. The second thing to remember that mmWave operating on ultra-high frequencies and affected not only by big obstacles but also by trees, human bodies and even drop of water or air itself. The higher frequency used leads to increasing range of affecting factors that need to be taken into consideration to predict the behavior of the connection.

2.1 Millimeter Wave technology

The millimeter-wave is a region of the electromagnetic spectrum with wavelengths, depending on the frequency, from 10 millimeters to 1 millimeter [10]. Its spectrum correspond to radio band frequencies between 30 GHz and 300 GHz and marked as the Extremely High-Frequency range by the International Telecommunications Union. The millimeter-wave wedged between microwave and infrared waves, what shows that millimeter waves length is longer than x-rays and infrared waves but shorter than microwaves and radio waves. The high frequency of mmWave makes it useful for a variety of applications including transmitting large amounts of computer data, cellular communications, sensors, and radar [11].

Capability to transfer a large amount of data is one of the essential features of millimeter waves [4]. Cell phone, radio, satellite or any other wireless communication has the specific range of wavelengths or frequencies assigned to it. A particular frequency or a wavelength assigned to each service provider. That range of frequencies available for network provider or a local television called bandwidth. They operate within different bands to avoid interference between them when they communicating at the same period of time. Available bandwidth must to be large enough to provide sufficient communication, because every time of transmission contains particular amount of data to be transferred. For example, TV broadcast need around 6 MHz of bandwidth to carry sound and video simultaneously, but for phone call, that contains sound or voice, 6 kHz is enough, because less data need to be transferred. The use of higher frequencies required by the increased amount of information transmitted. This is where millimeter waves come in.

2.1.1 Applications of the millimeter waves

High speed communications, p2p connections, wireless local area networks (WLANs) and broadband access can be expanded by usage of millimeter waves, which is an undeveloped band of spectrum. Millimeter wave can be used for a variety of services on mobile and wireless networks, due to its capability of high data rates up to 10 Gbps in telecommunications. Another notable use of millimeter waves is radar, and it takes advantage of beamwidth, which is another millimeter wave propagation property [12]. Beamwidth is an angle that shows how wide the signal spreads out at a distance from the source of transmission. Radar systems based on narrow angle beams can *detect* smaller objects at a distance, narrower the beam – more precise the radar. Microwaves can be focused into a narrow beam by a carefully designed antenna, just like a magnifying glass focuses sunlight. Small size of an antenna will allow to place it inside tight airplane cockpit, but narrow beamwidths requires large antenna sizes.

To overcome these antenna problem engineers used millimeter-length microwaves. Using increased frequency helps to reduce size of the antenna with narrow beamwidth. One of the examples of usage radar systems is the collision-avoidance systems build in modern cars. Car manufacturer's specifications require the system to be able to detect a cyclist at a distance of 100 meters, also the system must distinguish the bicycle from other objects such as trees, roadside signs, and other vehicles. The system operating on high frequency, such as millimeter waves do, uses an antennas with size of 10 cm in diameter. The size of the antenna allows to automobile designers to design it into the front end of the vehicle or any part of the vehicle body. The diameter of the antenna would be 78 cm to achieve the same beamwidth at a lower frequency, which will be pretty looking in a silly way antenna mounted on the car.

The 2003-model Mercedes S-class vehicles used a radar sensor. The device was placed under the hood of the car whic part is covered by millimeter waves penetrable materials. The circuits that transmit and receive millimeter waves is covered by 10 cm in diameter *radome*.

2.1.2 mmWave Challenges

Specific characteristics of the earth's atmosphere pose both problems and solutions for millimeter wave applications. As an example, oxygen, contained in the air, molecules will interfere with electromagnetic radiation and absorb the energy at 60 GHz (5 mm wavelength). Therefor 60 GHz can not be efficiently used as frequency for long-range communications, because absorption by the oxygen molecules reduces the electromagnetic radiation and power of the signal. Limited range of millimeter waves gives an opportunity to use it for secure short-range communications. Another application for 60Ghz technologies can be satellites communications, *cross-linking*, in space. 60 GHz works efficiently for communication between satellites because there is almost no oxygen in space.

The optical fiber always can be used to provide high data rates where millimeter waves with short wavelength signals would be absorbed by rain, fog or even dust and smoke. However, if the transmitters or receivers are mobile optical fibers cannot be used, such as cell phones or satellite communication, so wireless communications, including millimeter waves, is usually the best choice.

In general, the spectrum between 3 GHz and 300 GHz known as a mmWave band, even though the band from 26.5 to 300 GHz with 273.5 GHz available bandwidth is mmWave, but due to similarities in propagation characteristics all that band collectively called mmWave with the wavelength range from 1 to 100 mm.

The mmWave frequencies are of big interest for researchers as a part in future 5G cellular networks and potential commercial use since it was published in mmWave communication standards for wireless personal area networks at frequencies of 57-64 GHz and 60 GHz by IEEE 802.15 Task Group 3c (TG3c) and IEEE 802.11ad [13]. Besides, a significant amount of bandwidth at frequencies of 28-30 GHz are investigated for the local multi-point distribution service. The band 60-90 GHz is free and can be dedicated for developments in 5G, what was published in 2003 by the Federal Communications Commission (FCC). Currently, the most of the developments and researches interested in the 28 GHz band, the 38 GHz band, the 60 GHz band [14]. Even though, a number of challenges need to be taken into consideration when comparing characteristics of mmWave with existing systems involved in the microwaves band [15].

It is proposed to merge mmWave communications with LTE, Wi-Fi and other systems due to the limited coverage, which is named as heterogeneous network (HetNet). Different types of base stations are present at the same time in HetNets, such as traditional macro BS, low power BS, etc., in order to increase system capacity. Thus, HetNet is a promising way to investigate, analyze potential problems via cooperation and interaction among different networks to solve them [16], intercell interference and so on. A picocell in 60 GHz band coexists with macrocells and microcells [14], as shown in Figure 2.4. Apparently, cells with microwave bands are smaller but providing more extensive coverage, but the cell operating on 60 GHz frequency will provide higher capacity. As part of a hybrid HetNet [17], features of the spectrum at 60 and 70-80 GHz are used to minimize or even avoid the interference. A combination of the mmWave and 4G system is described in [18, 19], promoted as a candidate for a 5G cellular network with TDMA-based medium access control (MAC) architecture structure.

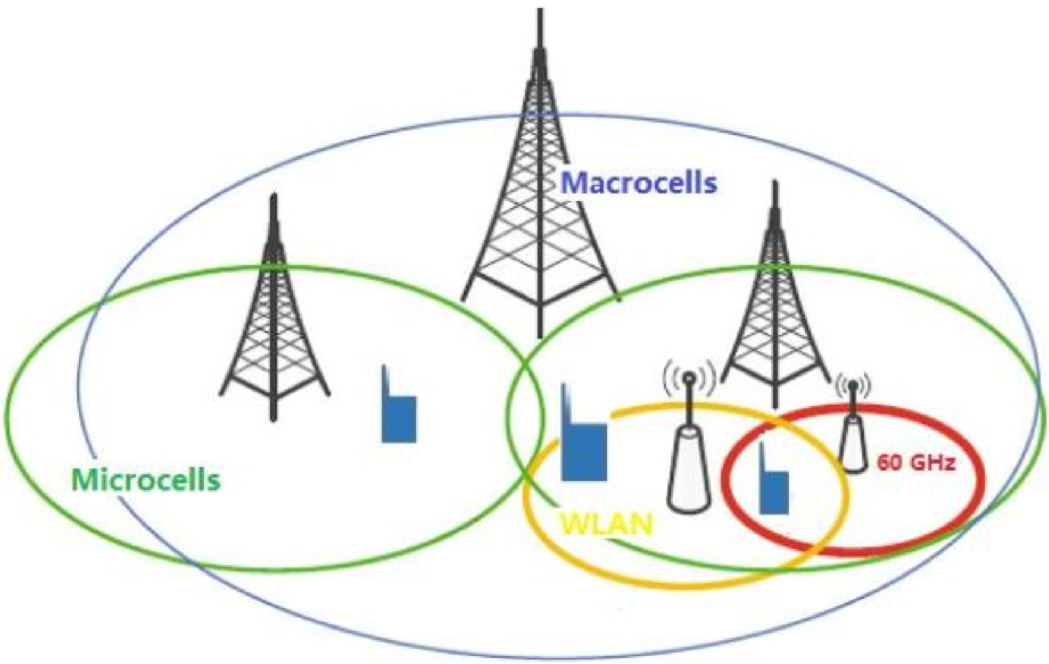

Figure 2.4 *Heterogeneous networks, including macrocells, microcells, WLANs and picocells*

The communication systems operating on mmWave frequencies provide large capacity and capable of taking care of part of the traffic from the macrocells and serve as an enhancement for traffic flow. The control messages will create additional overhead for channel access and coordination at both mmWave and microwave bands [20], because control and service signals, as channel access requests and synchronization, will be transmitted as a broadcast signals. Thus, additional overhead forces to find a compromise between complexity and performance in HetNet [14].

2.1.3 5G with mmWave band

Nowadays, a new generation 5G network concept with mmWave is still in the prototyping and trial stage. mmWave already showed its potential as part of the 5G cellular system. For example, an antenna with beamforming in the mmWave band that able to transmit data at distance of 2km away with a speed of 1 Gbps was created by researchers from SAMSUNG. However, to establish a reliable connection and perform cellular mmWave communications in 5G networks, there are specific concerns left regarding the extremely high frequency of the transmission, materials penetrability, such as walls and windows, etc, air and water absorption, huge path loss. Typically, mmWaves cannot go further than other systems operating at lower frequencies, higher frequency communications suffer from high transmission loss. The Friis transmission law describes the limitations with the following equation:

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi R}\right)^2 \tag{2.1}$$

where:

- P_t is transmit power,
- P_r is received power,
- G_t is transmitter antenna gain,
- G_r is receiver antenna gain,
- λ is the wavelength,
- R is the distance from the transmitter to the receiver.

Friis law 2.1 with isotropic antennas ($G_t = G_r = 1$) shows that increased frequency leads to increased effective area and λ is related to path loss value. The high carrier frequency in mmWave bands leads to high path loss. Nevertheless, it is possible to put more antennas in the same are because of short wavelength. If the signal is sent through narrow directed beams with high gains, then transmission and reception will be more effective [21, 20]. The received power for 30 GHz antenna setup can be even larger than the 3 GHz one if the size and number of antennas are designed accordingly [20].

Note, mmWave signals do not penetrate through solid materials, like walls or buildings, as good as lower frequency signals do. In Table 2.1, values of attenuation for the most used materials are given by [22, 23]. When penetrating through buildings, the signals will suffer propagation losses, and articles show that signals with higher frequencies penetrating different material with more loss. The limited distance of propagation for mmWave, besides to the attenuation, causes an outage or termination of the transmission from outside to the inside of the building and vice versa. Of course, some signals may go through windows, but the signal becomes so weak that its energy will be out of the receiver sensitivity.

Table 2.1 *Attenuation for different materials [dB], reproduced from [24]*

		Dry Wall	**Clear Glass**	**Mesh Glass**	**Wood**	**Concrete**
Thickness (cm)	**Carrier frequency**	**2.5**	**0.3/0.4**	**0.3**	**0.7**	**10**
Attenuation	<3 GHz	5.4	6.4	7.7	5.4	17.7
	40 GHz	–	2.5	–	3.5	175
	60 GHz	6.0	3.6	10.2	–	–

Water drops, rain in particular, also causes dramatical attenuation in mmWave transmission by scattering because the size of raindrops roughly similar to radio wavelengths in mmWave. The rate of precipitation is used to identify category of the rain:

- 0.25mm-1mm per hour is a light rain;
- 1mm-4mm per hour is a moderate rain;
- 4mm-16mm per hour is a heavy rain;
- 16m-50mm per hour is very heavy rain.

Specifically, for a very heavy rain with the rate of precipitation at 50 mm per hour, signal suffer from 14 dB attenuation per kilometer at 30 GHz carrier frequency [24].

The authors in [24] declare that the cell with 200 meters radius of coverage with the mmWave based communications will operate well in the rain and attenuation is going to be minimized.

In addition to natural factors affecting the signal, the performance of millimeter wave systems operating in the extremely high frequency band is also seriously affected by the obstacles in line-of-sight. Contradictory to lower frequencies, millimeter waves can not penetrate and go through multiple objects, which is producing *shaded* locations, besides diffuse scattering and reflection effects in the channel. It causes the system to be in non-line-of-sight (nLoS) communication, what results in significant degradation of performance of the system [25]. As a result, more difficult to provide QoS and fulfill needs of the users when line-of-sight (LoS) is unavailable because more computational and power resources are needed. In realistic conditions of a user moving across the area within the a mmWave access point cell coverage, LoS and nLoS time intervals alternate. Those changes in state of the channel affecting the resource allocation [26]. The latest requirements published by 3GPP shows that the small-scale mobility should be taken into consideration for scenarios such as d2d communication [27]. The temporal and spatial consistency of the line-of-sight and non-line-of-sigh states is important for possible massive multiple-input and multiple-output (MIMO) implementation and beam tracking. The channel model in question should be able to capture a smooth change in the LoS/nLoS states as a function of time [27, 28]

2.2 Transmission Control Protocol

The Transmission Control Protocol (TCP) is one of the most common protocols of the Internet Protocol Suite. TCP invented by Vinton Cerf, and Robert Kahn who introduced to the World the basic principles of Transmission Control Program in 1974, which was an internetwork protocol [29]. The protocol specifications published there supported overall addressing, fragmentation, and defragmentation, in addition to basic flow control, link error control, and loss recovery. At that time all networks had different addressing mechanisms, different size of the packets, different view on performance, such as total throughput and delay, which caused many difficulties to develop such reliable protocol. The principal goal was to reliably transmit data and provide stable communication over heterogeneous networks between remote hosts. There wasn't any agreement on traffic communication and packets retrieval what resulted as a risk of corruption, drop or delay of the packets, also packets could be misrouted or delivered out of order, in addition to duplicates that appeared due to lack of synchronization, because data delivery over such communication networks is not error-free. Therefore, the new protocol was designed to handle sequencing functions, synchronization mechanisms with acknowledgments (ACKs), when the error happened, and the packet is dropped or lost, retransmitting unacknowledged packets and perceive duplicates to resolve all these problems. Order of the packets being delivered maintained to detect the error and retransmit data. Moreover, flow control is one of the features of the protocol. It is preventing hosts that communicated with each other to be overloaded with traffic amount coming in and out. To perform all of the operations the Transmission Control Program created logical link from source to destination and destroyed the connection at the end of the transmission. The new protocol was demonstrated to the world in July 1977 [30].

The protocol was modified several times next years [31, 32, 33, 34]. Division of solid Transmission Control Program into two separate parts was the most crucial change. The Transmission Control Protocol (TCP) and the Internet Protocol (IP) protocol were created based on Transmission Control Program. Later in was called TCP/IP Internet Protocol Suite. Four layered Department of Defense model placed TCP just above IP in the third layer [34, 35], so TCP segments are encapsulated in IP packets to provide delivery of those segments. Header of the IP packet consist of address and control information that used by routers find an end point to forward packets. Fragmentation and defragmentation of packets are also done by IP. It is required when the size of the packets are bigger than the Maximum Transmission Unit (MTU). TCP perform peer-to-peer flow and error control on top of IP functions. It is required for applications that sensitive to sequential data delivery over reliable connections. Currently, TCP is situated in the transport layer, the fourth layer of the Open Systems Interconnection (OSI) reference model [36]. The United States Department of Defense used TCP for reliable data delivery over the network in their the ARPANET (Advanced Research Project Agency Network) as the main protocol in 1982 [30]. Later on, TCP has become the most popular protocol used in the worldwide Internet, a standard protocol for best-effort networks.

The Internet suffered a series of congestion collapses because of fast growth of computer network during following years that caused severe performance problems [37]. In fact, TCP mechanisms that provide reliability mainly caused these performance problems. The constant increase in the load by sending additional packets in the network caused congestion collapse. More packets peers send - more packets are queued, as a result, the routers suffer from the buffer overflow. Congestion leads to timeouts because of increased delays and lost data, what triggered retransmissions. It became even worse because of overload of the network caused by a big number of retransmissions. At some moment the network throughput drops very low that there is almost no packets flow with minimum throughput. The lack of e2e congestion control together with rate adaptation are the main reasons of congestion collapse at that time Internet. Only flow control existed as end-to-end rate control in early TCP [34]. TCP flow control could not prevent congestion anywhere in the middle of the network, because it was developed to restrict the TCP sender from overwhelming the TCP receiver. There is a need to have something to control the number of packets sent into the network as a mechanism to manage or, at best, bypass congestions.

Congestion control algorithms were added to TCP to avoid congestion collapse [38, 39]. Agreement of the rules and amount of data sent in the network used as a base for those algorithms. The principle based on preventing to send a new packet to the network unless the previous one leaves the network. It works as follows, the TCP receiver an ACK as an sign of the packets have been received and acknowledged and TCP sender initiates the next packet transmission because the previous packet has left the network without increase of congestion level. Because of this the basis why TCP is also called ACK-clocked. The reasons for the delivery failure of the packets either they are corrupted in transit, that cause rejection, or the network is congested somewhere on the path, and buffer overflow forced to drop extra packets. Packet loss is used as a reliable congestion indicator by congestion control algorithms of TCP with presumption that rate of the packet loss is far less than 1 percent, so data corruption supposed to be very low [39]. The congestion control algorithms multiplicatively reducing the sending rate or exponentially increasing the TCP retransmission timer if triggered when TCP sender detects a packet loss. TCP congestion control algorithms developed back at that time is used nowadays that prevent congestion collapse on the Internet. Fairness with other traffic, to some degree, giving the prospect to divide the available bandwidth between concurrent flows approximately evenly [6].

Today, TCP is providing reliable data delivery over unreliable networks. The *must have*, standard transport protocol, in client-server applications and services on the Internet are using TCP [40]. Majority of the applications and services in the Internet are client-server what results that about 90 percent of the current Internet traffic is transferred by TCP [41, 42]. It is essential to analyze TCP behavior in various environments to be able to optimize the performance of TCP, because the lion's share data transmitted via TCP over the Internet, to provide quality of service (QoS) in nowadays networks that result in user's needs satisfaction. Different issues related to TCP have been actively investigated the last years since TCP flow control drastically affect on QoS received by users and the performance of the overall Internet. Researchers are interested in TCP performance over wireless and wired-to-wireless environments because of increasing deployment of wireless networks. Wireless networks have very different characteristics from wired ones. TCP behavior might be different due to dynamic nature of wireless communications and wireless channels suffer from a higher bit error rate (BER). If the same algorithms used in the wireless channel, then TCP keep assuming that network congestion is the reason for packet loss and triggers congestion control algorithms, thus, decreases the sending rate to resolve the congestion. As a result, TCP slows down the transmission of new data because of packets drop, even if it happened due to data corruption, not because these packet drops signaling congestion in the network. As a result, TCP operate very badly over unstable wireless channels [43, 44, 45].

2.2.1 TCP Functions

TCP is a compound and fully-featured protocol. RFCs (Request For Comments) published by the Internet Engineering Task Force (IETF) describes the specification for TCP. Several RFCs [39, 34, 37] describe the core specifications, but the number RFCs about TCP is rather more bigger [6].

TCP has the following mechanisms:

- congestion control.
- flow control;
- error control;
- ordered data transfer and data segmentation;
- multiplexing/demultiplexing.

2.2.2 Data delivery

Data ordering and segmentation

TCP does not structure the data from the application layer but sending it in particular order of bytes because it is a byte-oriented protocol. TCP sends data from one host to another in a sequence of bytes, also called as a segment. In order to route segment to TCP receiver, the TCP sender encapsulates segment in IP packets on IP layer. To deliver the data to the destination application, the TCP receiver receives and obtain arriving segments from the IP layer and process it in the same order.

TCP sender and receiver need to make a mutual agreement on the maximum segment size (MSS) during connection establishment that will be sent via the established connection in each direction to be able to process it. MSS option is used only once during a TCP connection is the establishment to specify the maximum size of the segment that the host can obtain. That agreement sent in segments marked with the SYN flag. The TCP sender sends MSS size in the MSS value option field. MSS size is equal to the size of MTU without the IP header size and the TCP header size [46].

$$MSS = MTU - IPHeader - TCPHeader.$$

So, the MSS is 40 bytes smaller than the MTU. There are a few exceptions happens when the operating system is able to handle segments with lengths of multiples of 512 bytes only. As an example, when the MTU of 1500 bytes is possibly could be sent, the segments 1024 bytes only will be sent instead of 1460 bytes that is possible without exceeding MSS size.

Error Control

The packets might be corrupted, misrouted, dropped, delayed, duplicated or delivered in wrong order during the transmission. TCP error control provides end-to-end data integrity via the TCP checksum calculation and alignment of sequence numbers with acknowledgments. TCP checksum field is 16 bit long and serves as a verification of the segment to detect errors in the received data. Checksum hashes the TCP segment with header and payload, in addition to routing part of the IP header. That part of the IP header also called pseudo header. Information about IP addresses of the source and destination, transport layer protocol code, in addition to the size of the segment, are taken in the checksum calculation.

Each and every TCP segment is marked by a 32 bit long sequence number identifier. This number represents the offset of the beginning of this segment in byte stream of the TCP sender. The TCP receiver sends an ACK to the sender when a segment received successfully. ACK segment includes the number of the next sequence number the TCP receiver is waiting to receive. TCP sender presume loss of the segment if the ACK has not been returned back within retransmission timeout interval and retransmit it [6].

Extremely high-speed networks might be at possibility of multiple different segments having the identical sequence number because the total amount of sequence numbers is finite and it will start from 0 when the limit is exceeded. An algorithm with the TCP timestamps option provides protection against wrapped sequences (PAWS) [47]. To avoid having additional sequence number field in the header, these timestamps help to clearly distinguish new segments from the same sequence numbered old duplicates.

During the connection establishment TCP sender and receiver propose the initial sequence number for the transmission when the SYN/ACK segments are sent. Besides, TCP sender and receiver harmonize all the TCP options, such as window scale, the MSS size, timestamps, etc, in the phase of connection establishment. Thus, connections have to be established when needed and closed when the transmission is over because TCP is a connection-oriented protocol.

2.2.3 Data ordering

Multiplexing/Demultiplexing

TCP has a multiplexing and demultiplexing mechanisms to provide simultaneous access for several processes to the network via the same application layer interface. For operation, it requires additional assign of 16-bit port numbers, served as an IDs, to each process that is using TCP in the application layer. The combination of source and destination port numbers, in addition to source and destination IP addresses, uniquely identifies a TCP connection. TCP able to send data of different processes via single link by multiplexing them using port numbers and demultiplex incoming segments. Then, knowing IP address and port number, deliver it to the correct destination processes.

Fast Retransmit and Fast Recovery

There are two options to detect a segment loss for TCP. The first option is a timeout of the acknowledgment. When TCP sender transmits the data to the TCP receiver, it waits a particular period for confirmation of the successful reception of the data (ACK), if no ACKs returned it is the reason to consider that data has been lost during transmission. If loss of the data is detected the TCP sender sets ssthresh to the value equal to 50 percent of current CWND size or size of two MSS segments, whatever is bigger:

$$ssthresh = \max(CWND/2, 2MSS). \tag{2.2}$$

TCP connection goes to slow start state and the CWND is also set to the size of one segment. As a result, TCPS sender needs to use slow start algorithm after retransmission of the lost segment, because the congestion window became the size of one segment. Again the same steps to perform: increase CWND exponentially unless it will exceed the ssthresh, then when it happens the TCP sender uses linear increase the congestion avoidance algorithms [6].

Another option is the arrival of duplicate ACKs. The TCP sender cannot distinguish a duplicate ACK is received because of the loss of the data segment or due to delayed reception by the TCP receiver what caused the wrong sequence of the segments. In either of scenarios, the TCP sender awaits for four identical ACKs in a row to be received, three duplicate ACKs. The fast retransmit algorithm is triggered. Hence, TCP performs retransmission of the missing segments after receiving the last duplicate ACK, without waiting for the retransmission timer to run out [43].

As specified in [39], a new fast recovery algorithm can help the recover transmission speed faster after congestion. For best performance, fast recovery algorithm accompanied with fast retransmission algorithm. After reception of the third duplicate ACK, the ssthresh is set to the value from equation 2.2. The TCP sender retransmits lost segment, and the CWND is set to the ssthresh plus three segments size. The CWND is incremented by additional one segment for every extra duplicate ACK received. The TCP sender sends a new segment if it is possible by the updated values of the CWND and RWND recorded in the duplicate ACKs. The TCP receiver acknowledges new data the next ACK is sent, the ssthresh value is assigned to CWND as an updated value, congestion avoidance algorithm performed by TCP sender further. Detection of duplicate ACKs shows not only a packet loss but also that there are some data left in the link between the TCP sender and receiver. That is why slow start is not used in this case. The algorithm of fast retransmission with slow start published with TCP Tahoe. One of the main features of the TCP Reno implementation was fast recovery algorithm. Subsequently, several new features to improve TCP performance and loss detection have been proposed. SACK and NewReno are the The most acquired and widely known modifications of the loss recovery [48] and fast recovery algorithms [49], respectively.

2.2.4 Transmission management

Flow Control

Congestion and flow control is one of the TCP features. TCP flow control guarantees that data rate is aligned with the capacity of TCP links in the e2e connection to provide efficient use of the network with the best possible throughput and use of the available bandwidth. TCP flow control mechanism prevents overload of a slow receiver by a fast transmitting sender. The sliding window algorithm is used to accomplish data transfer between TCP sender and receiver to increase or decrease the amount of data that can be sent during one particular moment. The window size is the amount of data in bytes that the TCP sender is allowed to send by the TCP receiver. Only this amount of bytes of the sequence is allowed to send to the TCP sender that can fit inside the window size. TCP sender allowed to send new data only after receiver giving permission. Thus, the TCP sender can only transmit segments that less or equal to the window size. The amount of data allowed to send can be changed after receiving ACK feedback from the TCP receiver. When the TCP receiver acknowledges new data, the window slides across the byte stream [6].

TCP flow control operates on both receiver-side and sender-side. Sender-side flow control is limiting the data flow in response to the receiver-side flow control provided information about the free buffer space in the receiver-side. TCP header has a window field to indicate the receiver's free buffer space, a.k.a the receive window or the receiver advertised window (RWND). Receive window afford the information to a flow control about available space in the TCP connection. That window field is utilized by the TCP receiver to define an amount of data the TCP sender is allowed to transmit over the last acknowledged sequence number. The window size changing dynamically during data transmission via the TCP connection due to constant changes in the receiver's free buffer space, depending on the number of segments present in the buffer.

The silly window syndrome (SWS) happens when very small segments are transmitted over the TCP connection. The size of those segments is a lot smaller than the size of MSS [50]. Either TCP sender or TCP receiver can cause it. In one hand, It can happen when the TCP receiver increase RWND to small value just after the buffer space slightly freed for data reception, what forces the TCP sender to send small segments in return. On the other hand, transmission of small segments by the TCP sender caused by small increments of the sending data without delay for additional data to be packed into more significant segment altogether to be send later. Consequently, transmitting tiny segments occupy more network resources because a header size of the protocol is larger than the actual payload of the segment what causes unnecessary computational overhead on both peers in the network. The Nagle algorithm was developed to prevent sending and receiving small TCP segments, in addition to the SWS avoidance algorithm for the efficiency of data transmission. The Nagle algorithm [39] accumulate small segments of data in the TCP sender's buffer and do not send them before reception of an acknowledgment that previous data is delivered correctly, that prevents small segments transmission. The SWS avoidance algorithm [48] allows the TCP receiver to wait unless the free space attains less than one MSS or 50 percent of the total buffer size, whatever will be less, to prevents small window advertisements in the case of slow buffer processing by TCP receiver.

The window field is 16 bits long in the TCP header so that the maximum RWND size can be $2^{16} - 1 = 65535$ bytes. As a result, the TCP sender allowed transmit 65535 bytes of data at a moment, what limits the maximum possible throughput to $RWND/RTT$. The time that a segment takes to move all the way from source to destination and back called round-trip time (RTT). According to [47], the TCP window scale option where a window size larger than 65535 bytes was allowed for the TCP receiver to advertise to improve performance in long-distance and high-speed networks has been announced. The window scale factor is a power of two multipliers applied to the window field in the TCP header. It is in range of 0 when no scaling performed, and 14, referring to maximum scaling is done. Thus, the maximum possible window size is equal to $65535 \times 2^{14} = 1073725440$ bytes. The same as the MSS value, the window scale value communicated and agreed in the SYN and ACK segments when the establishment of the connection is going on. Therefore, the scale option value is unalterable in both directions after establishment of the TCP connection.

Congestion Control

Standard TCP congestion control relies on inordinately delayed packets or packet losses to trigger congestion control algorithms, such as behavior called reactive congestion control. A linear increase of the window size is done by the TCP sender until a packet loss occurs in the steady state of the TCP congestion control. The TCP sender divides the window size by two and, as a result, sending data rate too, when a packet loss is happened and noticed. Because of that, the strategy with such behavior is also called additive increase/multiplicative decrease (AIMD).

TCP uses an ACKs feedback mechanism to fulfill rate control and provide reliable delivery of the transmitted segments. As a result, TCP congestion control operates with the stream of correct order data reception acknowledgments sent back by the TCP receiver to notify the TCP sender, so the TCP sender can process these ACKs to justify the state of the network, information about lost data, congestion.

TCP receiver sends an ACK back to the TCP sender as soon as a segment from the TCP sender is received successfully. However, the majority of TCP implementations using the delayed ACK algorithm [48]. These implementations allow the TCP receiver to hold ACKs when some conditions are met, what permit to send a new data and ACK wrapped in one segment. As written in [48], the TCP receiver has to send an ACK segment if any of these events happened:

- no ACK was sent for the previously received segment and a new segment inbound;
- no new segment adopted within the certain timeout (usually 200 ms) after last segment is accepted;

- a new segment arrives and completes entirely or partially the aperture in the order of the byte series of TCP receiver.

In addition, if *duplicate*, an out-of-order segment, ACK is received the TCP receiver should generate an immediate ACK. *Duplicate* ACK send to warn the TCP sender that the segment was received out of order, the expected sequence number is not equal to received one.

Cumulative TCP ACKs number represents a successfully received sequence of bytes up to acknowledged sequence number. If received segments are not in the expected range of the numbers, they were rejected, because old TCP algorithms used this cumulative ACK mechanism only. If several segments of data get lost during the transmission, TCP performance becomes worse. Many losses in short period of time in general precipitate TCP to skip ACK-based clock, the overall performance degrades and, therefore, throughput is decreased. As introduced in [51], TCP performance can be improved by the selective acknowledgment, in addition to usage of the SACK for acknowledging duplicate segments and loss recovery algorithms during loss recovery when multiple drops of the segments happened. The TCP sender can be notified that all sequence of data has been received successfully, so the transmitter aware than only lost segments need to be retransmitted, acknowledgment works with segment out of order or duplicates.

Initial Window

TCP receiver permit to TCP sender to transmit only one window size sequence of segments at the beginning of the data transmission in early TCP implementations, what forces the TCP sender fit initial segments to the size of the window. It is highly efficient within the one network. In contrast, if TCP sender and TCP receiver are situated in different networks, and the path lies over other networks that work differently, without synchronization with the link between peers, it will cause instant congestion in the network. Initial window (IW) is used to start TCP data flow, in order to prevent an enormous number of segments being sent into the network, and then increments the IW during the transmission. Previously the IW contained one or two segments that were a maximum amount of data allowed in the MSS [39, 48].

Modern standard [52] specifies the maximum size for the IW as

$$IW = \min(4MSS, \max(2MSS, 4380bytes)).$$

TCP can transmission starts with up to three segments when standard settings with packets of 1500 byte are used.

Slow Start

TCP using the congestion window (CWND) in addition to the available buffer space advertised in ACKs (RWND), which has the same mechanism as the RWND. Incrementation or decrementation is done when the network can process the number of segments being injected into the network, without taking in consideration free space in TCP receiver's buffer. So

$$\min(RWND, CWND)$$

represent maximum quantity of data that TCP sender can send in the network without acknowledgement at a particular moment of the time [6].

TCP uses slow start algorithm to examine the capacity of the link all over the path from source to destination [39]. Initially, TCP will send one segment and wait for a response (ACK), if that is positive, it will transfer two segments and then wait for the response again, if that confident it will transfer four segments and so on, as shown in Figure 2.5. So initially the TCP transfer relatively small amount of data and then it quickly doubles up each time until congestion within the network is reached, that fast growth has an exponential trend.

This doubling of the transmitted number of segments continues unless either a packet loss happens or the size of congestion window attains the value of the slow start threshold (ssthresh). TCP enters the congestion avoidance mode when the congestion window goes over the ssthresh. 65535 bytes is a standard value of the ssthresh [39]. A packet loss interpreted as the evidence that there is a congestion in the network and reduces the size of the congestion window, usually it will be 50 percent of the last acknowledged congestion window size.

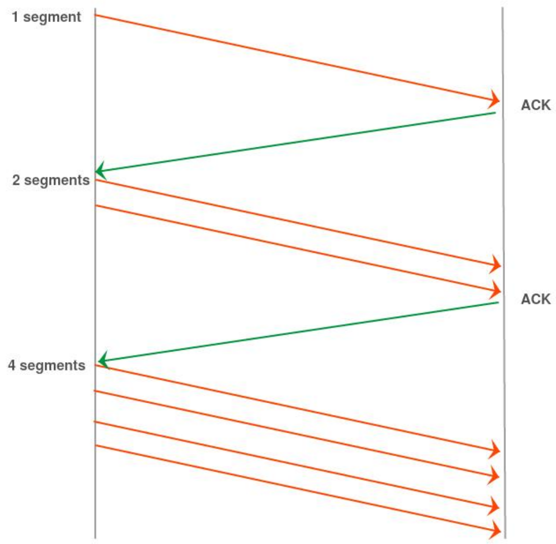

Figure 2.5 *TCP Slow Start process*

Congestion Avoidance

TCP enters the congestion avoidance mode when the congestion window goes over the ssthresh value [48]. In that state, the CWND is incremented by

$$MMS \times MSS/CWND$$

bytes for every non-duplicate ACK. Consequently, whenever TCP sender receives ACK for the previous transmission, it will linearly increase the size of CWND by one segment. When the TCP receiver allowed to send ACKs to every second transmission and do not acknowledge all of them every time, it is called the delayed ACK algorithm. Because TCP receiver does not send ACKs so often, this algorithms reduces overhead in the network, but in the same time, TCP sender increases CWND size less often what can result not the best performance of the TCP [39]. The linear increase continues unless congestion is detected, then TCP will be in slow start phase or linear increase, depending on the implementation of TCP.

There is a simplified example of congestion control with most of the features visible on the figure 2.6. It is a comparison between TCP Tahoe and TCP Reno performance. As it can be seen, fast retransmit and fast recovery affecting overall performance very much, what leads to higher throughput of the network.

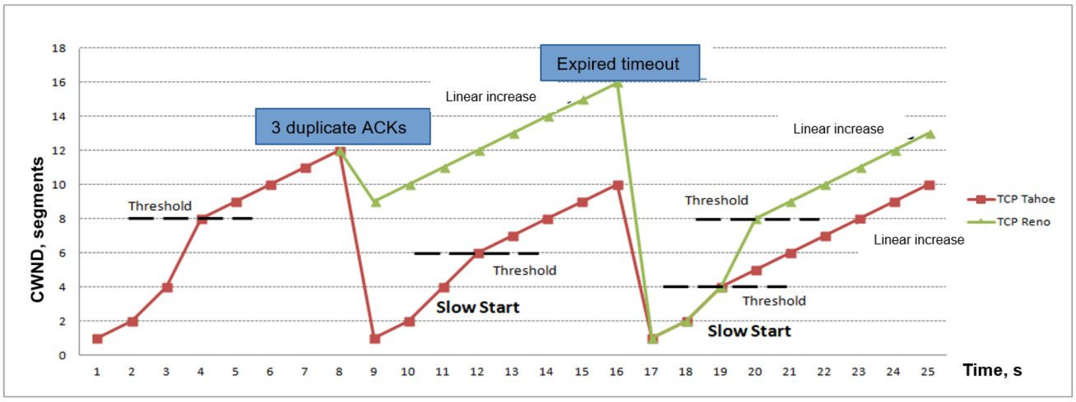

Figure 2.6 *TCP Congestion control example*

2.3 TCP in mmWave

TCP performance is highly dependent on conditions of the network and the link where it is operating. Interaction with other protocols, implementation of TCP used, the setting of the devices is already affecting the overall bandwidth usage, throughput values in legacy technologies. mmWave will add sensitivity for movement, air, fog, water absorption, additional delay and extremely high speed to the previous list. When new technology is used we need to know how it will behave and what is pros and cons of usage in particular cases.

Due to dynamic nature of wireless communications, mmWave in particular, current flow control, congestion control, congestion avoidance, etc., might be inefficient. mmWave provide incredibly high data rates, where modern algorithms will perform poorly. For example, slow start will not be able to reach maximum throughput fast enough, linear increase during congestion avoidance will not exploit the majority of available throughput due to the slow increase of CWND, the network will be very dynamic due to users mobility, what will result in unpredictable delays, errors or outage of the signal.

Currently, TCP utilizes around 75 percent of available bandwidth due to congestion avoidance algorithms and forms TCP sawtooth. It is shown in Figure 2.7. Maximum bandwidth consumption was sacrificed for reliable data transmission.

If we add extra conditions in the system, Figure 2.7 will change a bit. Because the user will move, it will create LoS and nLoS periods where the signal is strong enough to provide the best throughput possible and huge attenuation due to obstacles,

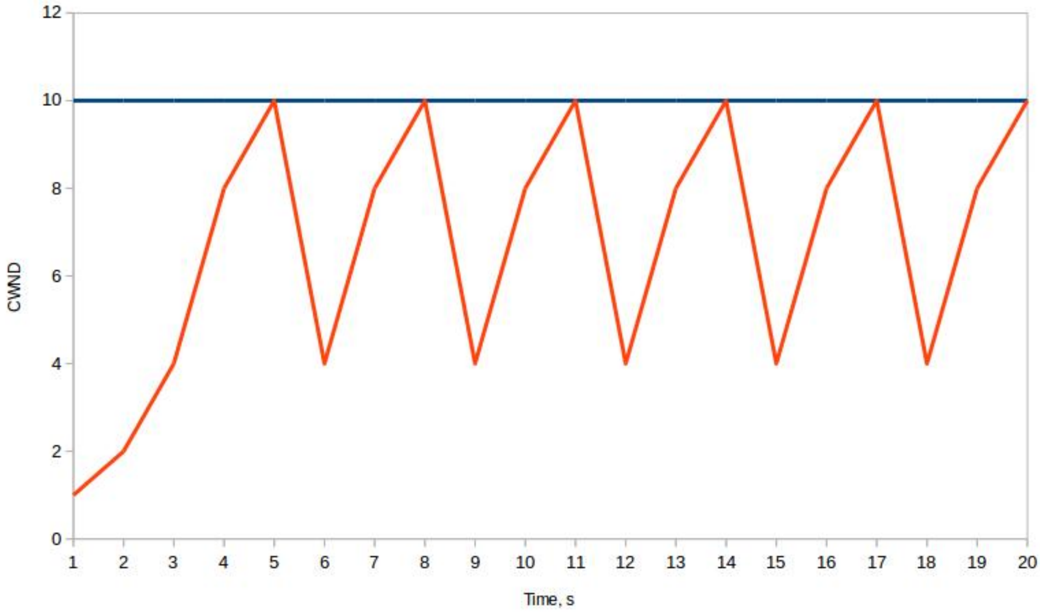

Figure 2.7 *TCP sawtooth, red curve represent the network capacity*

respectively. Such conditions are shown in Figure 2.8, in this case, TCP will be loosing rather more than 25 percent of bandwidth. This scenario shows that due to the slow linear growth of the CWND size sometimes transmission could be better and faster if another algorithm used to increment size of CWND.

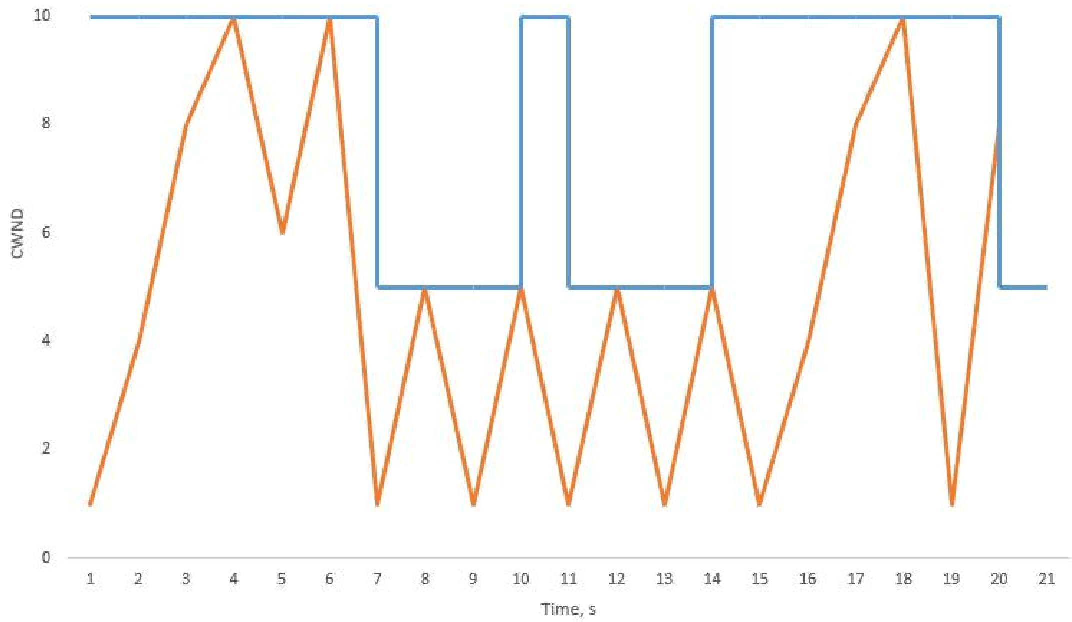

Figure 2.8 *TCP behavior in the dynamic networks*

It is just a glimpse of the wide range of factors and scenarios that need to be studied and taken into consideration for the future 5G communication systems for the best user satisfaction.

3. TCP PERFORMANCE ASSESSMENT FRAMEWORK

One of the most affecting factors in mmWave is line-of-sight blockage due to the sensitivity of shortwave mmWave to obstacles such as human bodies. The state when human as a mobile blocker is blocking line-of-sight called blocked LoS state, in contrast to the absence of human body obstacle causes the channel to be in non-blocked LoS state. In the [26] a new prediction of blocked/non-blocked state of a user model published according to 3GPP requirements and dynamic blockage scenarios. Blocked time intervals are generally distributed while non-blocked time intervals follow an exponential distribution, what produces an alternating renewal process of blockage effect. The model can be used in system-level simulations of mmWave cellular systems characteristics such as placement of access points to satisfy required throughput for users.

3.1 Analytical blockage model

Model-based on several metrics. Total and mean time intervals of blocked and non-blocked states, remaining blocked or non-blocked time and the time-dependent conditional probability of the user being in blocked/non-blocked at time t_1 if previously (at time t_0) the state was blocked/non-blocked, are metrics used in the model. The last metric calculated as a function of intensity, height and width of the blockers, distance from mmWave access point, the height of the access point, height of the user device. Numerical analysis and system-level simulations from [26] confirm that usage of the method reduces the complexity of modeling. In the Figure 3.1 shown the scenario of blockage in the analytical model.

According to [26], every blocker can spend different amount of time in LoS zone. The renewal process is shown in the Figure 3.2, where the residual time of blockers is overlapping with each other and causing long blockage time intervals.

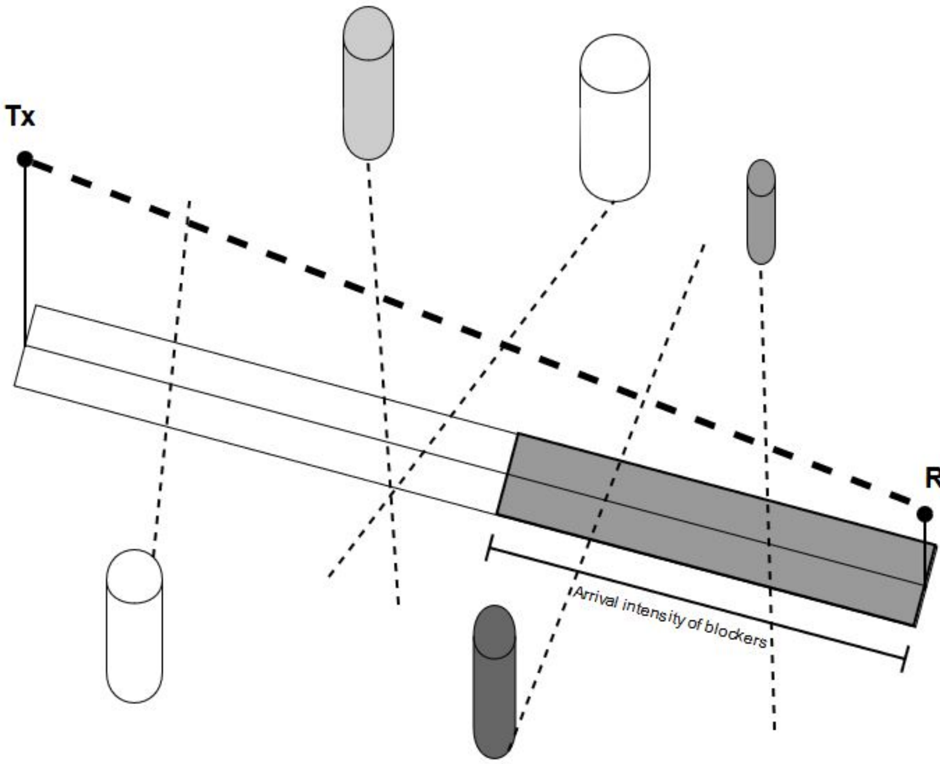

Figure 3.1 *Blockage scenario in analytical model*

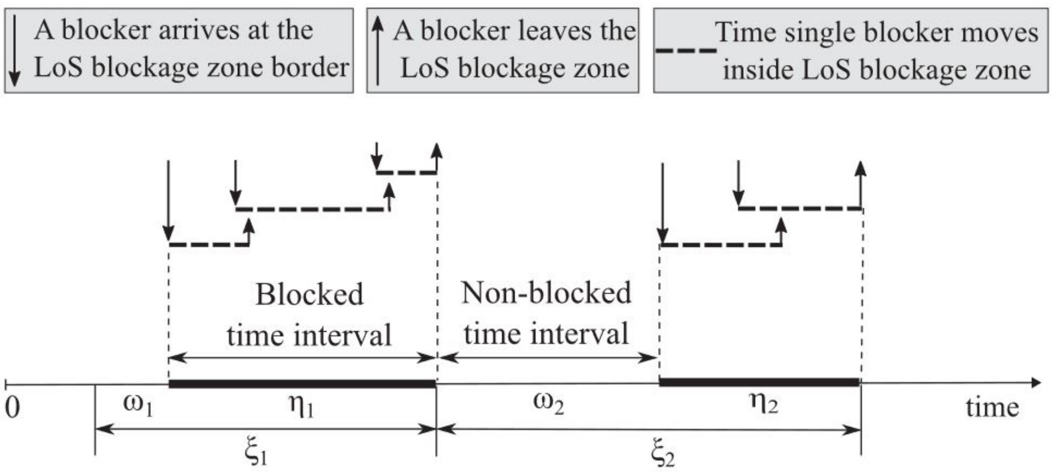

Figure 3.2 *Blockage/non-blockage intervals, reproduced from [26]*

Arrival time of the blockers in the LoS zone assumed to be Poisson distributed in time. Time interval between blockers arrival is exponentially distributed, because distribution of arrival time is Poisson. Length of the time intervals of non-blocked will be exponentially distributed [26]. Define $\xi_j = \omega_j + \eta_j$ from the Figure 3.2. If the points 0, ξ_1,$\xi_1 + \xi_2$, and $\xi_1 + \xi_2 + \xi_3$ will be taken as a renewal moments, applying [53], then the density of renewal process will be

$$f(x) = \lambda F_T(x) e^{-\lambda \int_0^x [1-F_T(y)]dy}, \tag{3.1}$$

Blockers entering the system and renewal process leads to proposal of an analogy with $M/GI/\infty$ queuing system, then the CDFs of the time in the blocked intervals can be calculated from following

$$\begin{aligned} F_\eta(x) =& 1 - \Big([1 - F_T(x)] \big[1 - \int_0^x (1 - F_\eta(x-z)) e^{-\lambda F_T(z)} \lambda dz \big] + \\ &+ \int_0^x (1 - F_\eta(x-z)) |de^{-\lambda F_T(z)}| \Big) \end{aligned} \tag{3.2}$$

where $F_T(x)$ is cumulative distribution function of time of single blocker being in LoS zone.

In the Figure 3.3 shown the geometry of blockage zone in the model, where L is the distance passed by a blocker within the LoS zone. To calculate cumulative distribution function of that distance 3.3 is used.

$$F_L(x) = \begin{cases} 0, & x \leq 0 \\ \omega_1 F_{L_1}(x) + \omega_2 F_{L_2}(x), & 0 < x \leq \sqrt{d_m^2 + r^2} \\ 1, & x > \sqrt{d_m^2 + r^2} \end{cases} \tag{3.3}$$

where the weights ω_1 and ω_2 are the probability for a blocker to enter from the longer side (AD or CB, see Figure 3.3) and to leave from another longer side (AD or CB), and the probability for a blocker to enter from the short side (DC) and to leave from the long side (AD or CB), respectively. Those weights calculated as

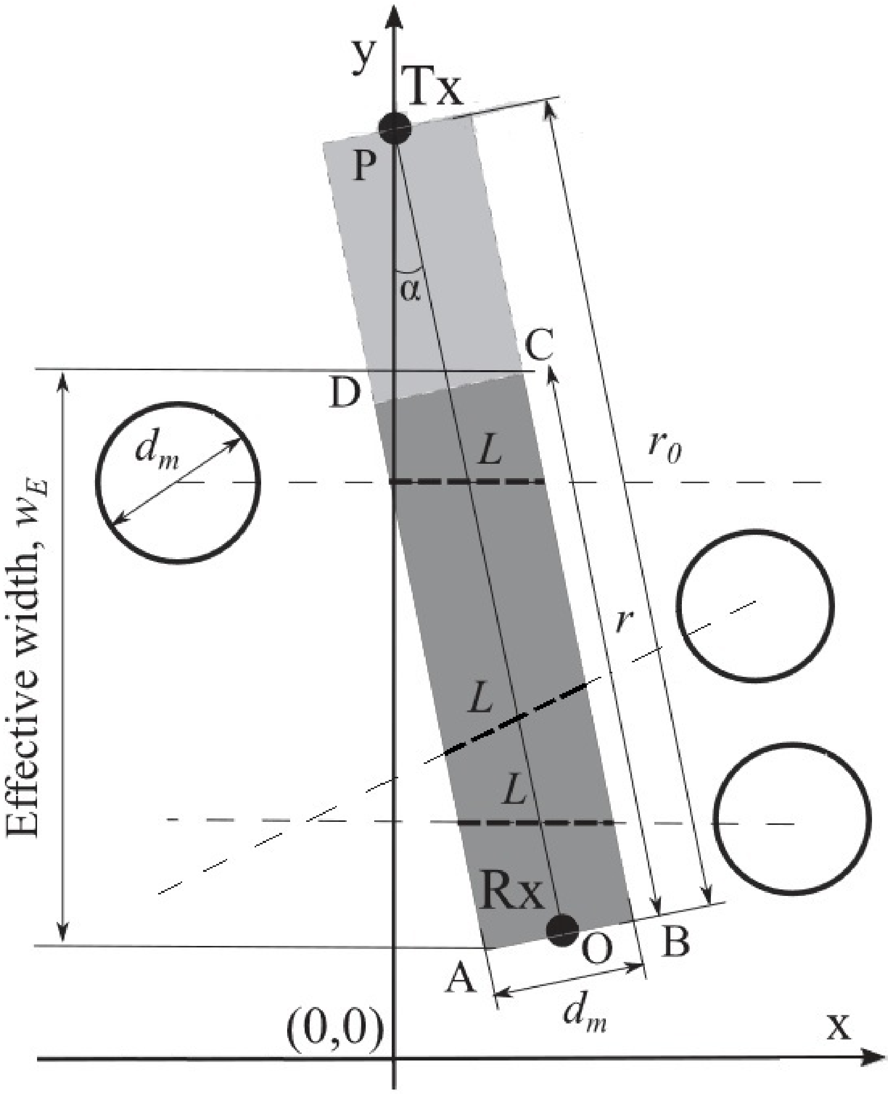

Figure 3.3 *Geometry of the LoS blockage zone*

$$
\begin{aligned}
\omega_1 &= \frac{d_m^2 + 3d_m r}{d_m^2 + 3d_m r + 2r^2}, \\
\omega_2 &= \frac{2r^2}{d_m^2 + 3d_m r + 2r^2},
\end{aligned}
\tag{3.4}
$$

and the corresponding cumulative distribution functions are

$$F_{L_2}(x) = \begin{cases} 0, & x \leq d_m \\ \frac{x^2 - d_m^2 + 2r\sqrt{x^2 - d_m^2}}{r^2} & d_m < x \leq \sqrt{d_m^2 + r^2} \\ 1, & x > \sqrt{d_m^2 + r^2} \end{cases} \tag{3.5}$$

$$F_{L_1}(x) = \begin{cases} 0, & x \leq 0 \\ \frac{\pi x^2}{4rd_m}, & 0 < x \leq min(r, d_m) \\ \frac{1}{2rd_m}\Big(min(r,d_m)\sqrt{x^2 - min(r,d_m)^2} & \\ + x^2 \arcsin\left(\frac{min(r,d_m)}{x}\right)\Big), & min(r,d_m) < x \leq max(r,d_m) \\ \frac{1}{2rd_m}\Big(min(r,d_m)\sqrt{max(r,d_m)^2 - min(r,d_m)^2} & \\ + d_m\big(\sqrt{x^2 - d_m^2} - \sqrt{max(r,d_m)^2 - d_m^2} & \\ + r\big(\sqrt{x^2 - r^2} - \sqrt{max(r,d_m)^2 - r^2} & \\ + max(r,d_m)^2\Big(\arccos\left(\frac{r}{max(r,d_m)}\right) & max(r,d_m) < x \leq \sqrt{d_m^2 + r^2} \\ + \arcsin\left(\frac{min(r,d_m)}{max(r,d_m)}\right) - \arcsin\left(\frac{d_m}{max(r,d_m)}\right)\Big) & \\ + x^2\Big(\arcsin\left(\frac{d_m}{x}\arccos\left(\frac{r}{x}\right)\right)\Big)\Big), & \\ 1, & x > \sqrt{d_m^2 + r^2} \end{cases} \tag{3.6}$$

The arrival intensity of the blockers in LoS zone then will be $\lambda = \lambda_I$.

The mean interval of time of the blocked/non-blocked state used to calculate blockage fraction of time

$$\begin{aligned} E[T_l] &= \frac{E[\omega]}{E[\omega] + E[\eta]}, \\ E[T_n] &= \frac{E[\eta]}{E[\omega] + E[\eta]}, \end{aligned} \tag{3.7}$$

where $E[\omega]$ and $E[\eta]$ are the mean values of the non-blocked/blocked intervals, which can be calculated by recalling exponential nature of ω and derivation from [26] as follows

$$E[\omega] = \frac{1}{\lambda},\ E[\eta] = \frac{1}{\lambda}(e^{\lambda E[T]} - 1). \tag{3.8}$$

The analytical model will be used as an efficient and reliable source of data for simulation. 3.7 and 3.8 will be precalculated in Matlab to be used as an input parameter to derive parameters of Gamma distribution for the blocked time interval and exponential distribution for the non-blocked time interval.

Exponential distribution require only one parameter λ, which can be derived from $E[\omega]$ as

$$\lambda = \frac{1}{E[\omega]}.$$

Gamma distribution require two parameters k as shape and θ as scale parameters. Using conditional blockage probability, CDF of non-blocked interval together with the values of non-blocked interval from [26] we can calculate mean and variance of non-blocked intervals knowing that mean value $\mu_x = \sum x_i p_i$ and variance $\sigma_x^2 = \sum (x_i - \mu_x)^2 p_i$ directly related to those parameters ($\mu = k\theta$ and $\sigma^2 = k\theta^2$). Thus, coefficients of Gamma distribution can be written as follows:

$$k = \frac{\mu}{\theta},\ \theta = \frac{\sigma^2}{\mu}. \tag{3.9}$$

3.2 Simulation framework

As the primary tool for simulation ns-3 (discrete-event network simulator) was chosen. It is free software and available for development, research, and education. The main target of the ns-3 project is to create a stable environment for research and development in networking, which is easy to use, well documented and can be used to simulate realistic scenarios to provide the ns-3 capability to be used as a real-time emulator reflecting real-world technologies, modern implementations of protocols.

Development of the simulator contributed by the vast amount of people from users community. Those people not only creating modules, modifying existing modules, but also verifying official software development results via mailing. Every several months, ns-3 project team roll out a new release of the software with new features, modules, and fixes, that mostly done by enthusiastic collaborators.

One of the recent modules developed by the community was 3GPP propagation module [27] and mmWave propagation module [54]. They are using a stochastic model of human and vehicular blockage scenario published by 3GPP in their technical report [55], where two alternative models (Model A and Model B) are provided for the blockage modeling. Both approaches have their use cases. Model A is applicable when a generic and computationally efficient blockage modeling is desired. Model B is applicable when a specific and more realistic blocking modeling is desired.

In the Figure 3.4 UML diagram of the classes in ns-3 is shown. All required basic modifications already combined in [56] open source project. The subject was to modify existing classes or to create own to implement new blockage model, described in the analytical model. ns-3 capable of simulating quite complex systems, such as mesh networks, but the main target was to create accurate and simple modeling. So mmWave propagation loss model was modified with new blockage model, in addition to changed interfaces of the class to be able to change the state of the channel on the fly.

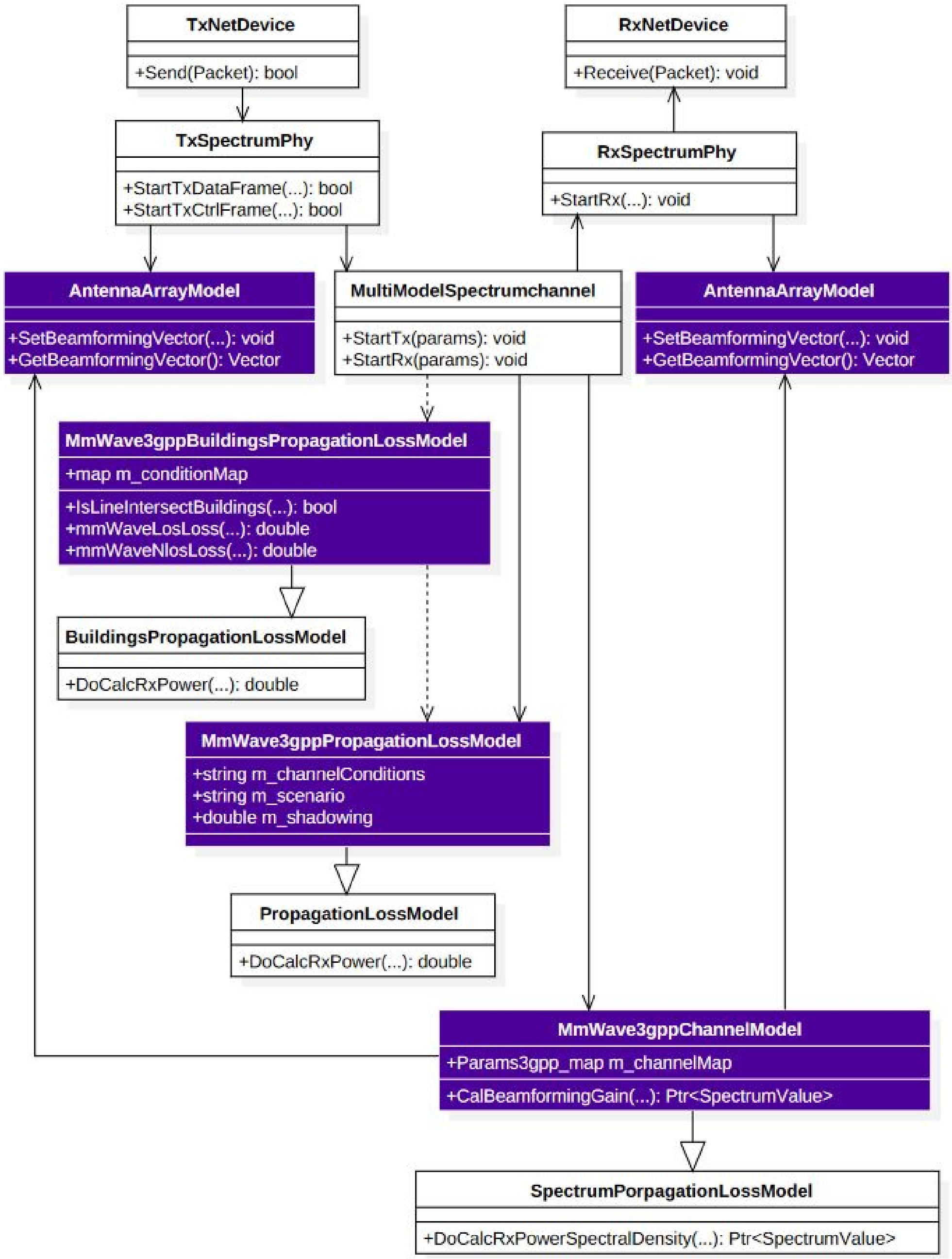

Figure 3.4 *UML diagram of the classes in ns-3, colored blocks are 3GPP related*

3GPP channel class modified with additional member function to add 20 dB of attenuation to the channel if blockage flag is triggered.

```
}
if(m_blockage)
{
    for(uint8_t cInd = 0; cInd < clusterNum; cInd++)
    {
        std::cout << "Cluster["<<(int)cInd<<"] the loss WAS ["<<powerAttenuation.at(cInd)<<"] dB" << std::endl;
        powerAttenuation.at(cInd) += 20; //human body blockage attenuation
        std::cout << "Cluster["<<(int)cInd<<"] is blocked by BLOCKER, "
            "the loss is ["<<powerAttenuation.at(cInd)<<"] dB" << std::endl;
    }
}
```

Figure 3.5 *Additional attenuation caused by human body blockage*

To change the state of the channel shared mutable scenario map added to mmWave channel class and function shown in Figure 3.6 need to be used in the primary function of the simulation. In the end, it became a combination of stochastic and geometrical models, which combine the computational speed of stochastic model and precision of geometric model. Computationally complex blocked/non-blocked probability, input parameters of Gamma and Exponential distributions precalculated in Matlab to reduce simulation run time.

```
void
ChangeBlockageState (Ptr<MmWaveHelper> mmChannel)
{
    double timePeriod;
    double ProbLos = uniformVar->GetValue();
    std::cout << "PRef(given)= " << PRef << ", ProbLos(rand)= " << ProbLos << std::endl;

    if (PRef >= ProbLos)
    {
        timePeriod = time_Block->GetValue();
        mmChannel->SetChannelBlockage(false);
        Simulator::Schedule (Seconds(timePeriod), &ChangeBlockageState, mmChannel);
    }
    else
    {
        timePeriod = time_nBlock->GetValue();
        mmChannel->SetChannelBlockage(true);
        Simulator::Schedule (Seconds(timePeriod), &ChangeBlockageState, mmChannel);
    }
    std::cout << "Time interval: " << timePeriod << std::endl;
    RngSeedManager::SetSeed (myRandomNo);
    myRandomNo = randomSeed->GetInteger ();
}
```

Figure 3.6 *Function to change blockage state*

The mmWave module allows to choose and configure a wide range of parameters, with possible two channel models. The first consist of MmWavePropagationLossModel and MmWaveBeamforming classes written in according to the latest 3GPP requirements. That model, recently, was extended with building propagation loss model to simulate mobility of the user inside and around the buildings with LoS and nLoS dynamic transition. List of mmWave parameters and their values shown in the Table 3.1.

Table 3.1 *mmWave module configuration parameters*

Parameter Name	Default Value	Description
SubframePerFrame	10	Number of subframes in one frame
SubframeLength	100	Length of one subframe in µs
SymbolsPerSubframe	24	Number of OFDM symbols per slot
SymbolLength	4.16	Length of one OFDM symbol in µs
NumSubbands	72	Number of sub-bands
SubbandWidth	13.89×10^6	Width of one sub-band in Hz
SubcarriersPerSubband	48	Number of subcarriers in each sub-band
CenterFreq	28×10^9	Carrier frequency in Hz
NumRefScPerSymbol	864 (25% of total)	Reference subcarriers per symbol
NumDlCtrlSymbols	1	Downlink control symbols per subframe
NumUlCtrlSymbols	1	Uplink control symbols per subframe
GuardPeriod	4.16 µs	Guard period for UL-to-DL mode switching
MacPhyDataLatency	2	Subframes between MAC scheduling request and scheduled subframe
PhyMacDataLatency	2	Subframes between TB reception at PHY and delivery to MAC
NumHarqProcesses	20	Number of HARQ processes for both DL and UL

3.3 Illustrative examples

The main focus of this work is human body blockage of TCP over the mmWave link. So we have changing in time *blocked* and *non-blocked* periods, which different affecting parameters of the channel. All scenarios run with the set of variable input data to analyze the behavior of the channel and evaluate the performance of TCP in contrast to *clear* channel without destructing factors. In Figure 3.7 shown TCP performance during first 10 seconds of simulation. As you can see, there is one blocked and one non-blocked period, which corresponds with the drop of the throughput.

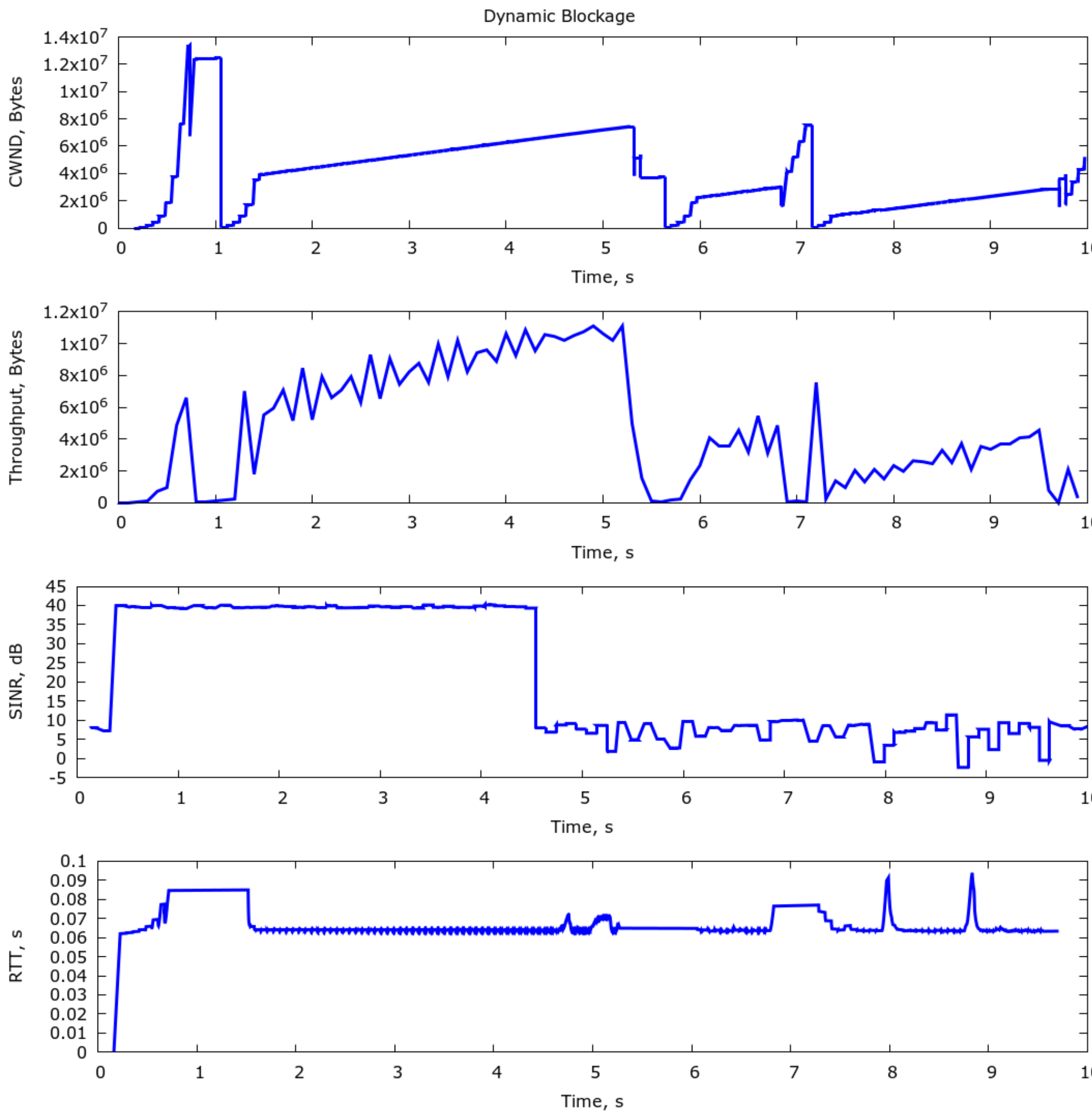

Figure 3.7 *TCP characteristics with blockage*

Those plots from Figure 3.7 quite useful already, but more reflective will be to compare the same scenario without happened blockage to evaluate the behavior of TCP to dynamic blockage. Comparison is shown in Figure 3.8.

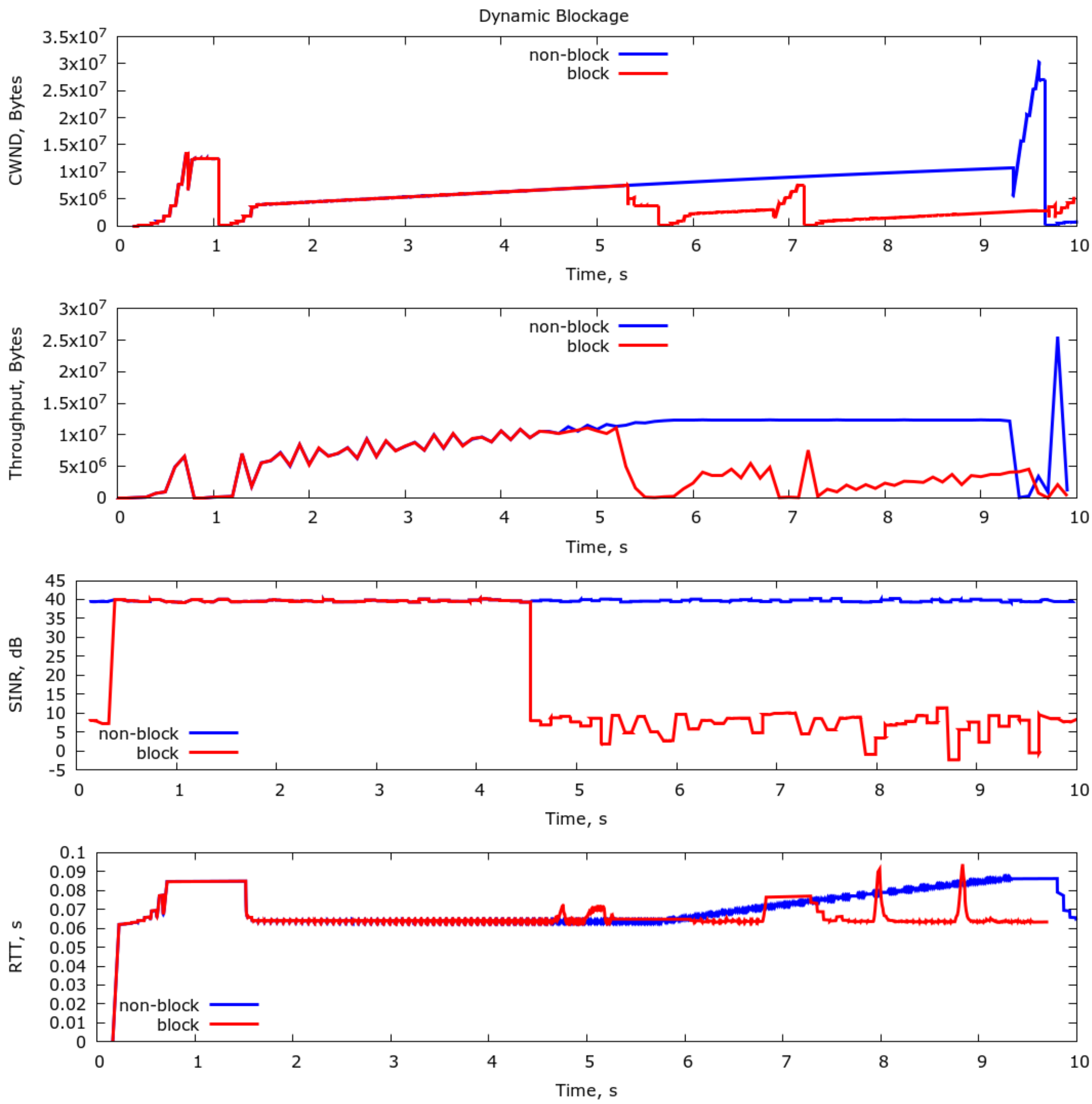

Figure 3.8 *Comparison of TCP performance with dynamic blockage*

As you can see in Figure 3.8, in non-blocked case CWND continue increasing till the end of simulation time, but blocked case CWND size dropped to the size of one segment. More detailed description of this case and others written in the next Chapter.

4. TCP PERFORMANCE EVALUATION AND RESULTS

As it was described in Chapter 2 wireless communications and TCP in particular affected by a broad range of factors that need to be taken into consideration. Some of them might be helpful, others not, each of the characteristics of the channel can be a pro the same as a con of the total performance of TCP.

Performance evaluation require metric that will show *good* or *bad* connection perform within given conditions. In this work, throughput is taken as a metric to evaluate TCP performance in mmWave 5G systems. Throughput or *speed* is the most interesting characteristic for the end users, because they do not need to know about placement of the APs or beamforming of the signal, etc.

4.1 Scenario and initial parameters

The simplified scenario is shown in Figure 4.1 and Figure 4.2 with the blocked and non-blocked condition of the channel, respectively. Model consist of constant and variables used to determine main parameters of the scenarios. Those parameters are shown in the Table 4.1.

Table 4.1 *ns-3 model parameters*

Constants		**Variables**
Frequency	28GHz	Position of the UT
Speed of the blockers	1 m/s	Distance from eNB
Height of the eNB	25m	LOS/nLOS link state
Height of the UT	1.5m	Density of the blockers
Height of the Blocker	1.7m	Time of Block/nBlock period
Human body attenuation	20 db	

Blocked period of time generated with Gamma distribution, a non-Blocked period of time generated with Exponential distribution, where computationally complex input parameters for Gamma and Exponential distributions precalculated in Matlab, in addition to LOS/nLOS probability, to reduce simulation run time.

Initially, channel assumed to be undestructed by obstacles, in our case it is a human, and in LoS state as shown in Figure 4.1. UE is situated in a certain distance from eNB and receiving/downloading data via 5 Gbps channel.

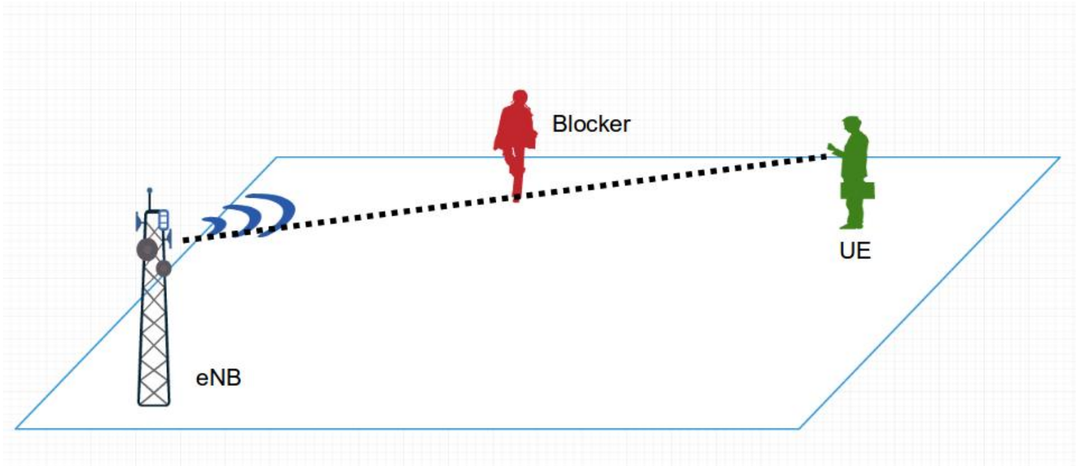

Figure 4.1 *Scenario for non-blocked state in LoS*

In some moment random blocker destruct the signal causing additional attenuation of 20 dB and transaction of the channel to nLoS state, what will result in a chain reaction of performance drops in other levels.

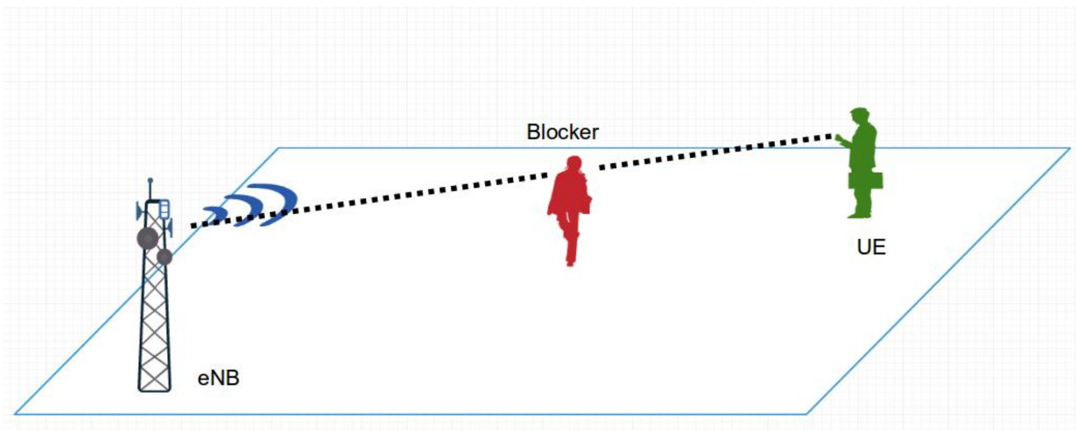

Figure 4.2 *Scenario for blocked state in LoS*

4.2 Simulations outcome

Figure 4.3 represent used topology of the network in scenarios. The p2p link is limited to 100 Gbps, to give maximum bandwidth for mmWave transmissions, the LTE mmWave link is using the mmWave frequency band with download capacity of 10 Gbps. A P2P connection has no delay (0 ms), and LTE mmWave has dynamic latency, with reference delay of 80 ms, depending on the distance of UE from eNB.

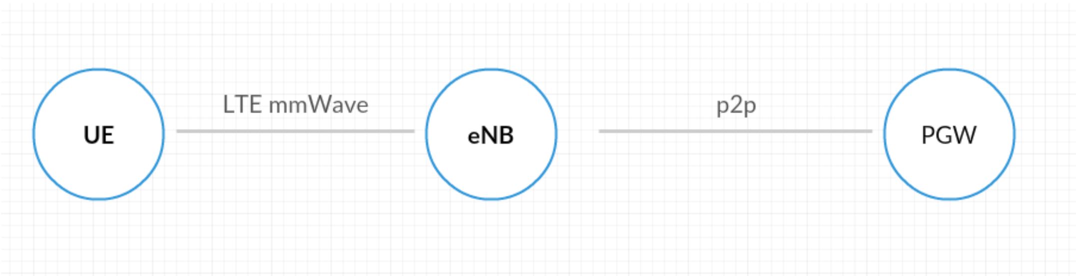

Figure 4.3 *Evaluation topology*

In Figure **??** different characteristics of the TCP NewReno performance with dynamic blockage enabled are shown. Average throughput is 2.67 Gbps with no significant drops. In this scenario distance of UT from eNB configured to 150 meters, the intensity of the blockers set to 1 person per second, what corresponds to very high blockage intensity, all buffer sizes set to 25 MB to be able process such a huge packets arrival rate, HARQ and RLC are enabled with Drop-Tail queue management.

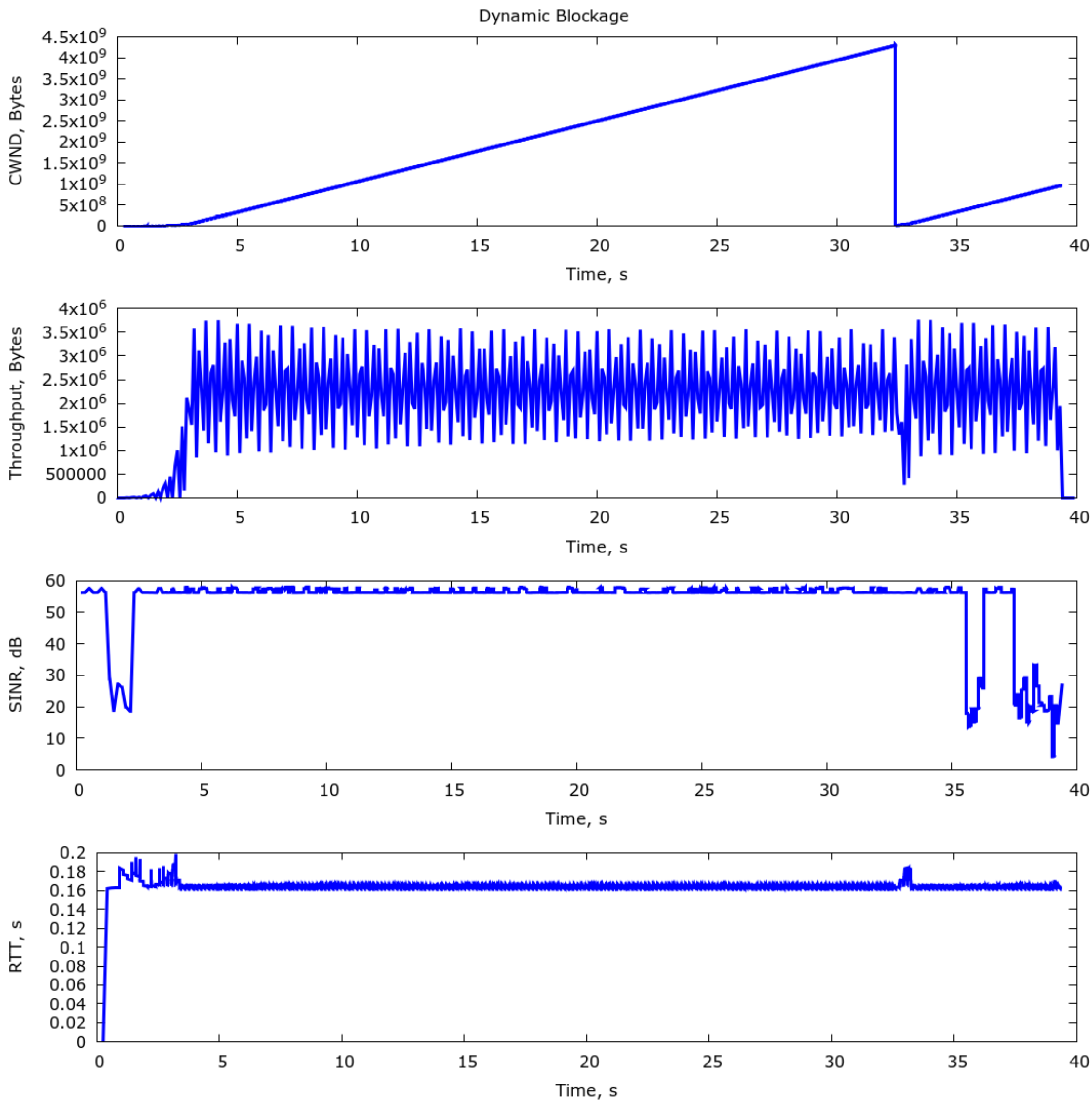

Figure 4.4 *TCP NewReno performance in dynamic blockage*

As you can see from Figure 4.4 TCP was able to withstand transitions from blocked to non-blocked states due to short periods of blockage. The most significant impact caused by proper RLC configuration, in our case it is RLC AM, to deliver packets over an unstable link with maximum bandwidth usage supported with HARQ with acknowledgment mode activated and flexible TTI settings. When error or packet loss is detected HARQ mechanism helps to keep smooth data flow by retransmitting corrupted bytes immediately to the TCP receiver.

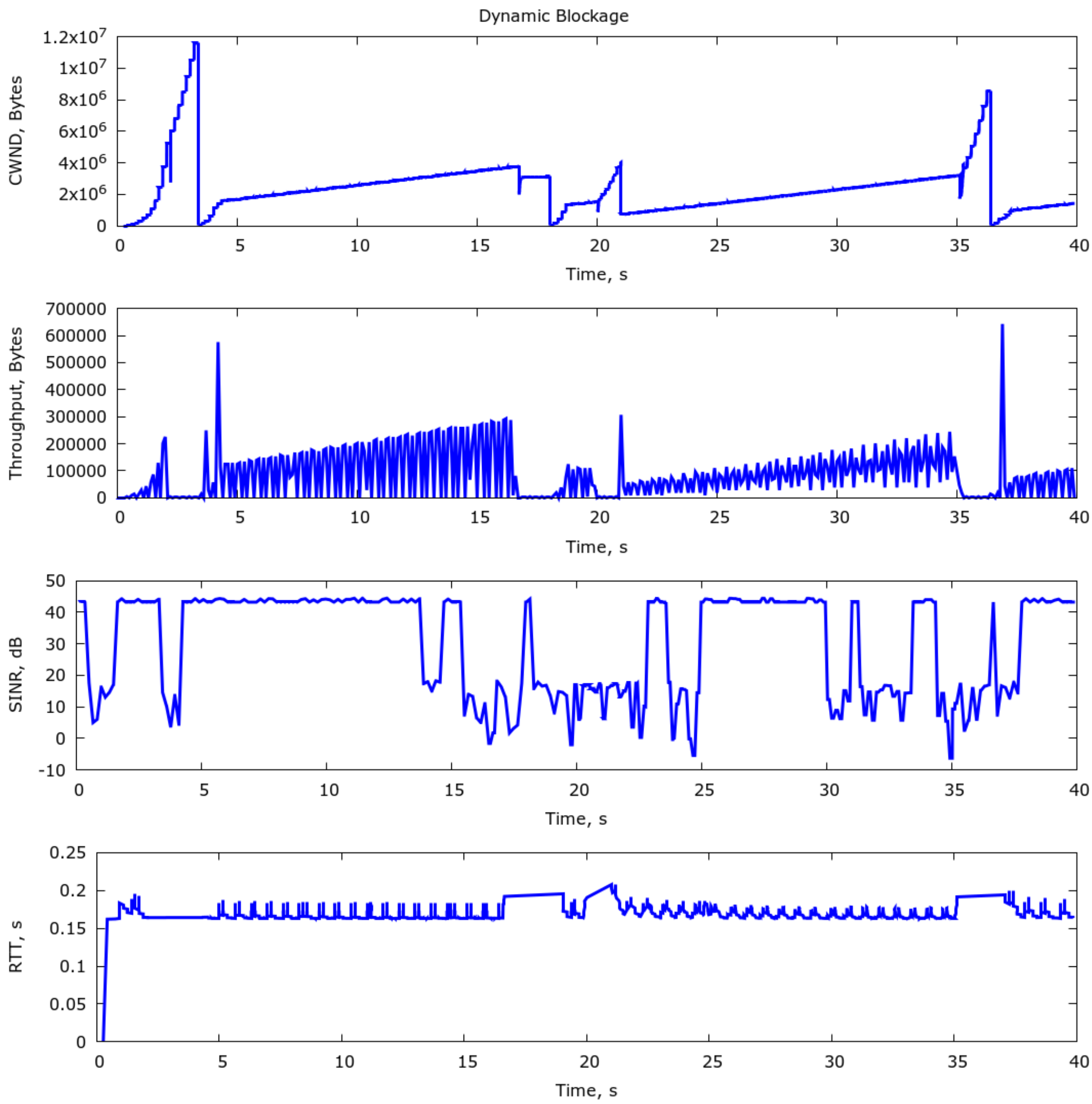

Figure 4.5 *TCP NewReno performance in dynamic blockage without RLC*

If we disable RLC, then TCP cannot do error correction through ARQ, that results in idle of the transmission and pushing data when buffers are filled up. What results in 82 Mbps throughput, shown in Figure 4.5 with the majority of available bandwidth unused even though CWND size continue to increase because no congestion or packet loss is detected.

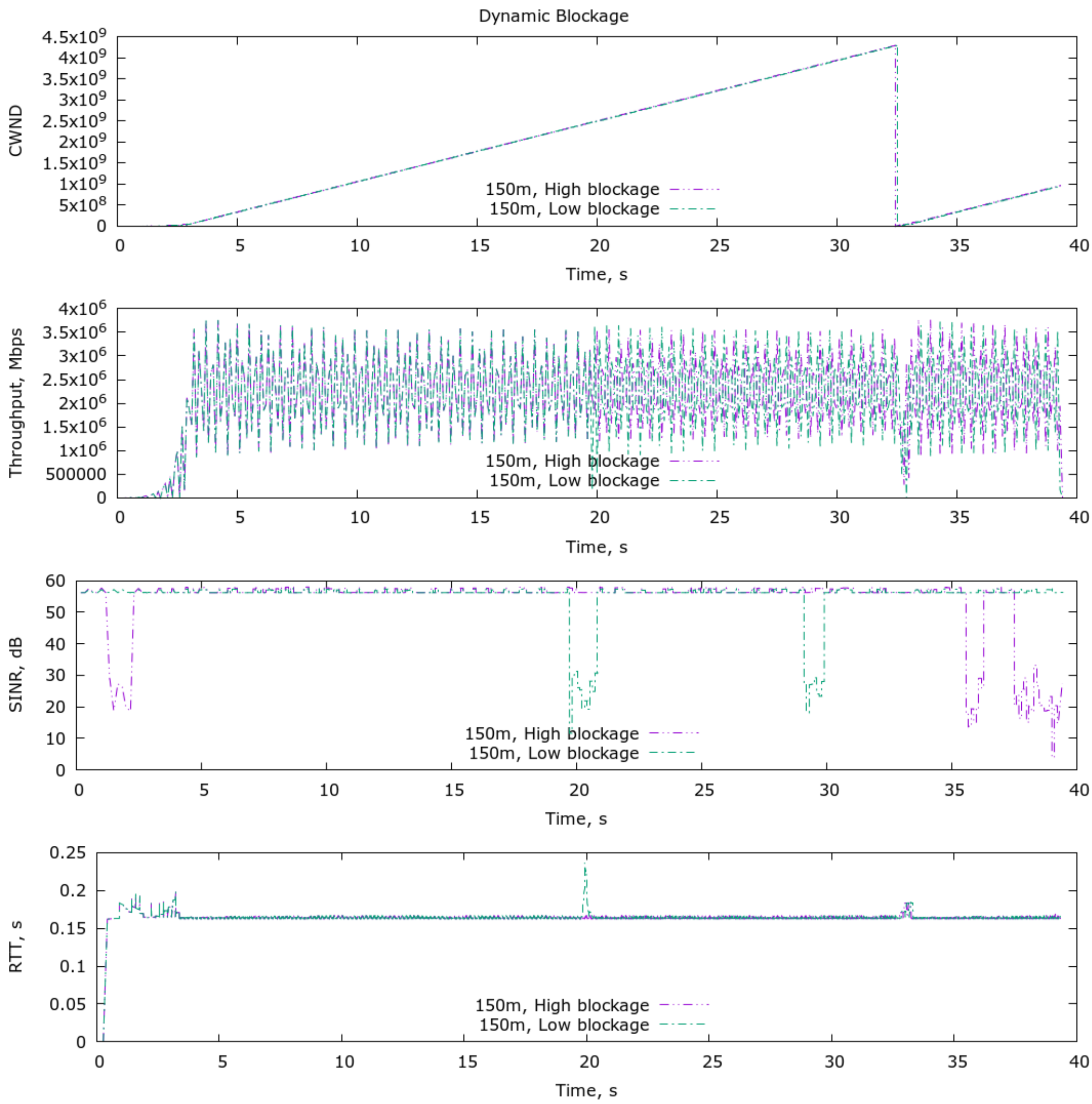

Figure 4.6 *Comparison of the TCP NewReno characteristics on distance of 150 meters*

From Figure 4.6 we can see that TCP NewReno can withstand blockage events with HARQ and RLC AM enabled. High blockage scenario shows that TCP is not affected by short time blockage (at 36^{th} second of a simulation run) and during slow throughput stages when it started to increase CWND size (at 2^{nd} second). There were two drops of the speed: 1) when CWND size reached maximum and packet loss happened (at 35^{th} second) and 2) when longtime blockage caused errors in the link (at 37^{th} second). During low blockage, scenario run past behavior confirmed, at 19^{th} second the first blockage happened that lead to throughput drop, but CWND continue to advance because buffer could handle short time break. Average throughput of these scenarios is 2103 Mbps and 2097 Mbps, correspondingly. Thus, TCP NewReno perform good in such dynamic environment, short blockage periods (less than 1 second) doesn't affect throughput, but longer blockage (more than 1 second) can cause drop of the packets and throughput, however, big buffer size can help in this cases, what will increase tolerated blockage period of time.

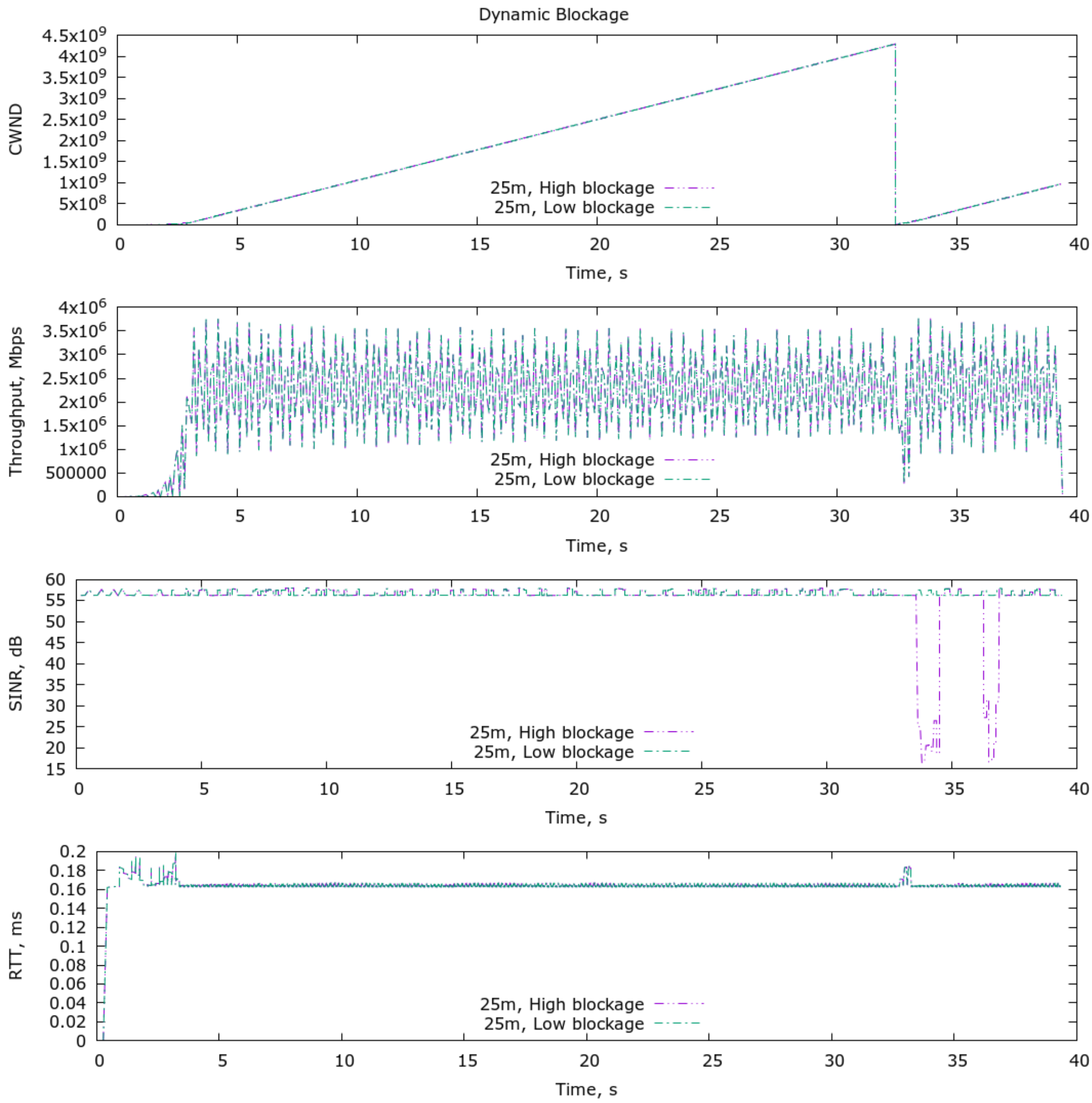

Figure 4.7 *Comparison of the TCP NewReno characteristics on distance of 25 meters*

If Tx is placed closer to Rx, in this case, 25 meters away, the probability of blockage decreases drastically and become almost zero for the low-intensity scenario. It is shown in Figure 4.7, no blockage happened during the first 40 seconds and only drop of the throughput caused by CWND size reaching maximum level. Two short time blockage happened during high blockage scenario, which didn't break the connection and retransmission mechanism handled all the errors due to sufficient buffer size.

Blockage probability during the simulation runs are summarized in Figure 4.8 where shown that blockage probability is increasing faster with distance than the intensity of the blockers. Geometrical nature of the model can explain it. Size of the blockage area increases with the distance of UE from eNB.

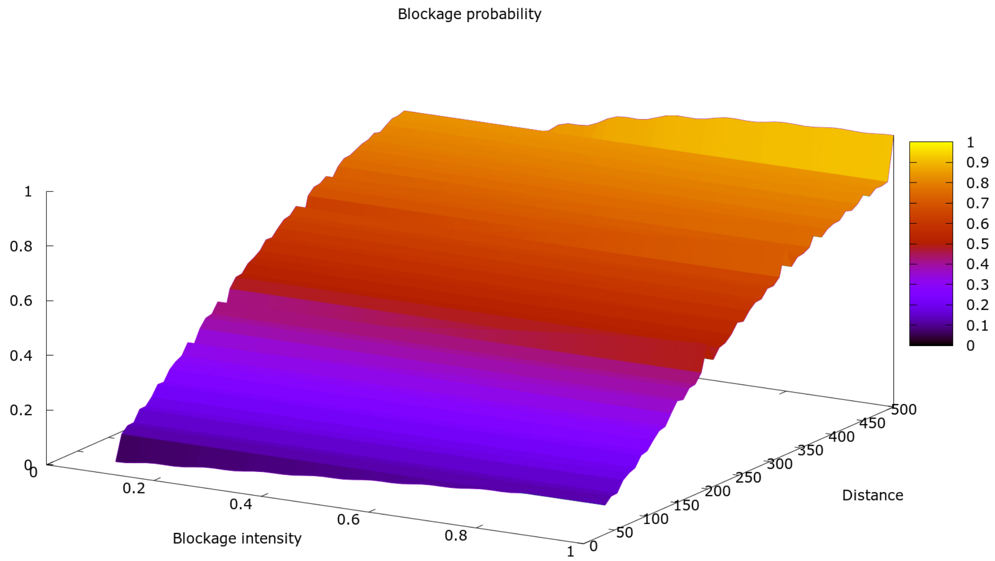

Figure 4.8 *Probability of blockage depending on distance and intensity of the blockers*

In addition to performance evaluation of the single TCP implementation, performance of different TCPs was analyzed during the work. TCP NewReno, TCP Yeah, and TCP Cubic were taken as the most widespread implementations of TCP. All of them differ in congestion control algorithms, error and packet loss handling, resulting in different CWND behavior and response to blockage based errors and packet losses in the network. Comparison of TCPs is shown in Figure 4.9 where the distance of UE from eNB equal to 150 meters with high intensity of the blockers ($\lambda = 1$), blockers movement speed is 1 m/s, the topology of the network is the same as shown in Figure 4.3.

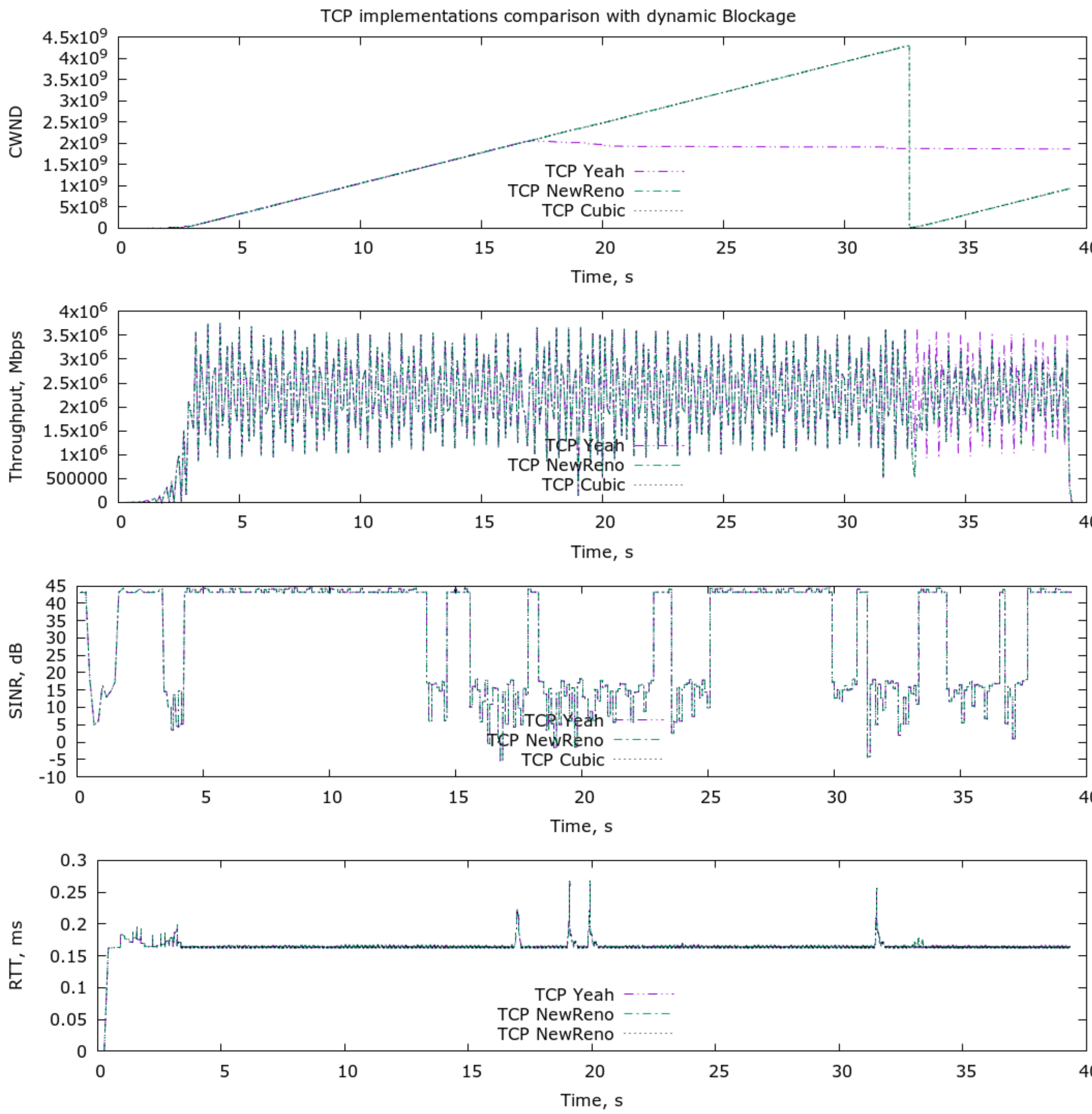

Figure 4.9 *TCP NewReno, TCP Yeah, TCP Cubic performance with dynamic blockage*

From Figure 4.9 we can see that in this scenario TCP NewReno and TCP Cubic perform the same due to similar congestion control algorithms, but TCP Yeah perform slightly better because it did not overshoot CWND limits. It can be seen from the throughput figure that TCP Yeah do not have the drop in throughput at 33^{rd} second. As result throughput of TCP NewReno is 2090.68 Mbps, TCP Cubic is 2090.34 Mbps, and TCP Yeah is 2103.98 Mbps. This slight drop affected average throughput, it is shown that communications over LTE mmWave link are capable of providing very high throughput, which limited by legacy configurations standards such as MTU, MSS, buffer size and TCP congestion control algorithms. Each of these parameters will increase data rates and lead to increased TCP throughput and overall performance of the network.

5. CONCLUSIONS

TCP performance modeling in 5G mmWave systems model created in NS3 with 3gppMmWave channel module where Dynamic Blockage scenario added. Developing or choosing proper TCP congestion control and delay handling is very important in the high bandwidth mmWave systems due to high frequencies and sensitivity of the channel to a broad variety of affecting factors such as attenuation, delay, outage, number of users, etc.

In this thesis work, we evaluated the TCP performance in mmWave systems with dynamic blockage. Real TCP implementations are used, and the model showed a close match with empirical and analytical results of mmWave characteristics of the link. Drastically decreasing performance of TCP is present in the system over the mmWave link, caused by the transaction from line-of-sight to non-line-of-sight and back. A drop of performance can be reduced by usage of delay or latency tolerant TCP implementations. Throughput is affected by choice of TCP implementation, the configuration of the mmWave link and other intermediate connections, such as sizes of buffers, algorithms of acknowledgment processing, MTU, MSS, frequency, etc. It is shown that buffer size is severely limiting resulting throughput because default buffer size cannot handle the rate of packets arrival caused by large TCP CWND size, what leads to series of packet drops. Packet drop triggers TCP fast retransmit state.

Even if the specification of 3GPP mmWave propagation model show maximum distance more than 5 kilometers [27], simulation run of the created model shows that channel in the blocked state all the time on distances more than 500 meters due even with very low blocker intensity. Blockage leads to extra losses of received signal power that causes an outage of the signal and connection break. This problem can be solved by a higher gain of the antennas or more accurate beam direction. It points to the fact that TCP requires direct visibility of the transmitter and non-blocked channel to get best possible throughput.

TCP performance in mmWave might be as good as in Wi-Fi or wired networks if required configurations are done on all layers of the communication. The most significant impact and therefore the most critical parameters are buffer sizes of transmitter and receiver, segment sizes of the transmitted packets, and conditions of the channel itself, such as LoS/nLoS, blocked/non-blocked states, what require appropriate TCP implementation choice or even development of new one to satisfy increasing demand of the next generation 5G networks over mmWave links. Combinations of those parameters can affect either positively or negatively on overall TCP performance. 5G is a heterogeneous network that combines different types of communication interfaces and technologies. Transmission of the data over such diverse network require a lot of optimizations of each and every part of the network as well as overall adaptation of the data flow.

Different TCP implementations usage showed that in the next-generation 5G networks with mmWave bands will require selection of appropriate TCP implementation or even development of new one to fulfill requirements of higher frequency usage and utilize all possible bandwidth. TCP performance evaluation showed over 30 percent of the channel capacity is not utilized. It can be utilized via better algorithms of TCP congestion control and flow control or shared with other protocols and services.

TCP connection will share bandwidth with other protocols and data flows that causes additional fluctuations in protocol behavior. TCP multi-path and multi-UT can be added and tested as an improvement of the model. In addition to TCP connections, other connections can be tested and evaluated with given model. mmWave provides very high throughput with proper configurations and settings that wired connection can become the bottleneck of total throughput of the network.

BIBLIOGRAPHY

[1] Cisco. Cisco visual networking index: forecast and methodology, 2016-2021. URL: `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html`.

[2] T. Nitsche, C. Cordeiro, A. Flores, et al. Ieee 802.11 ad: directional 60 ghz communication for multi-gigabit-per-second wi-fi. *IEEE Communications Magazine*, 52(12):132–141, 2014.

[3] D. Solomitckii, V. Semkin, R. Naderpour, A. Ometov, and S. Andreev. Comparative evaluation of radio propagation properties at 15 GHz and 60 GHz frequencies. In *Proc. of 9th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 91–95. IEEE, 2017.

[4] Y. Niu, Y. Li, D. Jin, L. Su, and A. Vasilakos. A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges. *Wireless Networks*, 21(8):2657–2676, 2015.

[5] D. Moltchanov, A. Ometov, S. Andreev, and Y. Koucheryavy. An Upper Bound on Capacity of 5G mmWave Cellular with Multi-Connectivity Capabilities. *Electronics Letters*, 2018.

[6] R. Dunaytsev. TCP Performance Evaluation over Wired and Wired-cum-Wireless Networks. *Publication 873*, 2010.

[7] A. Ometov, S. Bezzateev, J. Kannisto, et al. Facilitating the delegation of use for private devices in the era of the internet of wearable things. *IEEE Internet of Things Journal*, 4(4):843–854, 2017.

[8] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Elsevier*, 29(7):1645–1660, 2013. DOI: `https://doi.org/10.1016/j.future.2013.01.010`.

[9] R. Kovalchukov, A. Samuylov, D. Moltchanov, A. Ometov, Andreev, et al. Modeling three-dimensional interference and sir in highly directional mmwave communications. In *Proc. of IEEE Global Communications Conference*, pages 1–7. IEEE, 2017.

[10] T. E. Bogale and L. B. Le. Massive MIMO and mmWave for 5G wireless HetNet: Potential benefits and challenges. *IEEE Vehicular Technology Magazine*, 11(1):64–75, 2016.

[11] M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi. Mobility Management for TCP in mmWave Networks. In *Proc. of the 1st ACM Workshop on Millimeter-Wave Networks and Sensing Systems 2017*, pages 11–16. ACM, 2017.

[12] N. González-Prelcic, R. Méndez-Rial, and R. Heath. Radar aided beam alignment in mmwave V2I communications supporting antenna diversity. In *Proc. of Information Theory and Applications Workshop (ITA)*, pages 1–7. IEEE, 2016.

[13] IEEE 802.15 WPAN Millimeter Wave Alternative PHY Task Group 3c (TG3c). URL: `http://www.ieee802.org/15/pub/TG3c.html.`.

[14] Y Niu et al. A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges. *Wireless Networks*, 21(8):2657–2676, 2015.

[15] A. Ometov, E. Sopin, I. Gudkova, S. Andreev, Y. Gaidamaka, and Y. Koucheryavy. Modeling Unreliable Operation of mmWave-Based Data Sessions in Mission-Critical PPDR Services. *IEEE Access*, 5:20536–20544, 2017.

[16] R Hu and Y Qian. Heterogeneous cellular networks. *Wireless Networks*, 21(8), 2013.

[17] H Mehrpouyan, M Matthaiou, R Wang, et al. Hybrid millimeter-wave systems: a novel paradigm for hetnets. *IEEE Communications Magazine*, 53(1):216–221, 2015.

[18] J Qiao, X Shen, et al. Enabling device-to-device communications in millimeter-wave 5G cellular networks. *IEEE Communications Magazine*, 53(1):209–215, 2015.

[19] A. Orsino, A. Ometov, G. Fodor, D. Moltchanov, et al. Effects of heterogeneous mobility on D2D-and drone-assisted mission-critical MTC in 5G. *IEEE Communications Magazine*, 55(2):79–87, 2017.

[20] Z Pi and F Khan. An introduction to millimeter-wave mobile broadband systems. *IEEE Communications Magazine*, 49(6):101–107, 2011.

[21] W Roh, J Seol, J Park, et al. Millimeter-wave beamforming as an enabling technology for 5G cellular communications: theoretical feasibility and prototype results. *IEEE Communications Magazine*, 52(2):106–113, 2014.

[22] C Anderson and T Rappaport. In-building wideband partition loss measurements at 2.5 and 60 GHz[J]. IEEE Transactions on Wireless Communications. *IEEE Communications Magazine*, 3(3):922–928, 2016.

[23] A Alejos, M Sanchez, and I Cuinas. Measurement and analysis of propagation mechanisms at 40 ghz: viability of site shielding forced by obstacles. *IEEE Transactions on Vehicular Technology*, 57(6):3369–3380, 2008.

[24] L Wei, Y Qian, et al. Key elements to enable millimeter wave communications for 5G wireless systems. *IEEE Wireless Communications*, 21(6):136–143, 2014.

[25] M Akdeniz, Y Liu, et al. Millimeter wave channel modeling and cellular capacity evaluation. *IEEE Journal on Selected Areas in Communications*, 32:1164–1179, 2014.

[26] M. Gapeyenko, A. Samuylov, M. Gerasimenko, D. Moltchanov, et al. On the temporal effects of mobile blockers in urban millimeter-wave cellular scenarios. *IEEE Transactions on Vehicular Technology*, 66(11):10124–10138, 2017. DOI: `https://doi.org/10.1109/TVT.2017.2754543`.

[27] 3GPP. Channel model for frequency spectrum above 6 ghz (release 14). *3GPP TR 38.900*, V2.0.0:1–7, 2016.

[28] A Ghoshet. 5G channel model for bands up to 100 GHz. *5GCM white paper*, 2015.

[29] V Cerf and R Kahn. A protocol for packet network intercommunication, ieee transactions on communication. 22(5):637–648, 1974.

[30] R Zakon. Hobbes' internet timeline. *IETF*, RFC 2235, 1997.

[31] V Cerf, Y Dalal, and C Sunshine. Specification of internet transmission control program. *IETF*, RFC 675, 1974.

[32] V Cerf. Specification of internet transmission control program tcp (version 2). URL: `http://chris/DIGITAL_www.cs.utexas.edu/users/ARCHIVE/TCPIP/IEN5.pdf`.

[33] J Postel. Dod standard transmission control protocol. *IETF*, RFC 761, 1980.

[34] J Postel. Transmission control protocol. *IETF*, RFC 793, 1981.

[35] J Postel. Internet protocol. *IETF*, RFC 791, 1981.

[36] I.-T. R. X.200. Information technology - open systems interconnection - basic reference model: the basic model, 1994.

[37] J Nagle. Congestion Control in IP/TCP Internetworks. *IETF*, RFC 896, 1984.

[38] V Jacobson. Congestion control in ip/tcp internetworks. *ACM SIGCOMM Computer Communication*, Review 25(1):157–187, 1988.

[39] W Stevens. Tcp slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. *IETF*, RFC 2001, 1997.

[40] H Schulze and K Mochalski. Internet study 2008/2009. URL: `http://www.ipoque.com/study/ipoque-Internet-Study-08-09.pdf`.

[41] Internet2. Netflow: weekly reports. URL: `http://netflow.internet2.edu/weekly`.

[42] CAIDA. CAIDA's traffic analysis research: Analyzing UDP usage in Internet traffic. URL: `http://www.caida.org/research/traffic-analysis/tcpudpratio`.

[43] C Barakat, E Altman, and W Dabbous. On tcp performance in a heterogeneous network: a survey. *IEEE Communications Magazine*, 38(1):40–46, 2000.

[44] H Inamura and G Montenegro. TCP over Second (2.5G) and Third (3G) Generation Wireless Networks. *IETF*, RFC 3481, 2003.

[45] Y Tian, K Xu, and N Ansari. Tcp in wireless environments: problems and solutions. *IEEE Communications Magazine*, 43(3):27–32, 2005.

[46] J Postel. Tcp maximum segment size and related topics. *IETF*, RFC 879, 1983.

[47] V Jacobson, R Braden, and D Borman. Tcp extensions for high performance. *IETF*, RFC 1323, 1992.

[48] M Allman, V Paxson, and W Stevens. Tcp congestion control. *IETF*, RFC 2581, 1999.

[49] S Floyd, T Henderson, and A Gurtov. The NewReno Modification to TCP's Fast Recovery Algorithm. *IETF*, RFC 3782, 2004.

[50] D Clark. Window and acknowledgment strategy in tcp. *IETF*, RFC 813, 1982.

[51] S Floyd, J Mahdavi, M Mathis, and M Podolsky. An Extension to the Selective Acknowledgement (SACK) Option for TCP. *IETF*, RFC 2883, 2000.

[52] M Allman, S Floyd, and C Partridge. Increasing tcp's initial window. *IETF*, RFC 3390, 2002.

[53] D Cox and H Miller. The theory of stochastic processes. *Hoboken, NJ, USA: Wiley*, 1965.

[54] M. Zhang, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi. ns-3 Implementation of the 3GPP MIMO Channel Model for Frequency Spectrum above 6 GHz. *Information Theory*, 2017. DOI: `https://arxiv.org/abs/1702.04822`.

[55] G. T. 38.900. Study on channel model for frequency spectrum above 6 ghz. *Technical Report*, version 14.2.0(Release 14), 2017.

[56] M. Mezzavilla, M. Zhang, et al. Mmwave ns-3 module. URL: `https://github.com/nyuwireless-unipd/ns3-mmwave`.