An Analytical Model for Short-Lived TCP Flows in Heterogeneous Wireless Networks

Sameer Bataineh
Jordan University of Science and Technology, Irbid, Jordan
samir@just.edu.jo

Jamal Al-Karaki
Hashemite University, Jordan
jkaraki@hu.edu.jo

Haya Sammaneh
Jordan University of Science and Technology, Irbid, Jordan
hyhs_70@yahoo.com

ABSTRACT
The transmission control protocol (TCP) is widely used in wired and wireless networks. It provides reliable transport services between end-to-end hosts. Since TCP performance affects the overall network performance, several analytical models were proposed to describe the steady-state throughput of bulk transfer TCP flows (i.e., a flow with large amount of data to send, such as FTP transfers). However, most TCP flows in the Internet world are short-lived to see few losses and they cannot reach the steady-state, consequently their performance is determined by the startup effects such as the connection establishment and the slow start mechanisms. Surprisingly, all of the previous models did not investigate the heterogeneity of wireless networks. The heterogeneity is shown by different characteristics and different segment loss probability for various types of wireless networks such as IEEE 802.11 WLAN and 3G cellular network. Moreover, wireless TCP flows are much shorter than wired flows due to the time varying characteristics of wireless networks.

In this research, a recursive and analytical model is developed and used to determine the performance of TCP in heterogeneous wired-cum-wireless networks in terms of average completion time for the short-lived TCP flows is proposed. The proposed model focuses on heterogeneous wireless networks. The model tackles two different schemes, namely, End-to-End scheme that solve the wireless loss problems by the end hosts and Connection Division scheme that split the connection at base station aiming to distinguish different characteristics between different networks. The proposed model is based on the knowledge of average dropping probability, the average round trip time and the flow size in both wired and wireless links. The analytical model has been validated by means of simulations and using NS-2 simulator. The performance metric of TCP is the average completion time, which is the time that the source takes to successfully transfer a given amount of data in addition to the connection establishment time. It is shown clearly that the proposed model reflects the behavior of TCP; especially in computing the average completion time that increases as the session size (i.e. the number of data segments in the flow) increases. The simulation results are very much the same as the values obtained from the analytical model. The percentage difference for the average completion time (in Sec) is less than 3% and 6% for the End-to-End scheme and the Connection Division scheme, respectively. The model can be used by engineers to tune some of parameters to see its effect on the behaviour of TCP.

Key Words: analytical model, heterogeneity and short-lived TCP.
1. Introduction
Transmission Control Protocol (TCP) is the ubiquitous transport protocol used in the Internet world. The Transmission Control Protocol is a reliable, connection-oriented, full-duplex, byte-stream and end-to-end protocol that supports flow and congestion control [1]. Also, it is widely used to support applications like Telnet, and FTP [2]. TCP’s performance influences Internet traffic behavior. Hence, many models of TCP latency and throughput have been proposed, trying to capture its characteristics [6] [7]. In most of these models, the TCP performance (latency and throughput) is described based on the network parameters such as TCP round trip time and packet loss rate.

In today's world, more and more people are moving towards using mobile and wireless technology for communication. This technology is very much required for people on the move, and makes possible fast and easy installation in remote areas. Also, more and more people are using their mobile devices to access the Internet either for work or entertainment.

The recent works show that most TCP flows are very short-lived flows with average sizes of around 10 KB [7] [10].

In recent years, the modeling of the TCP behavior has received considerable attention and many analytical models have been proposed with the purpose of characterizing TCP performance [7] [8] [9] [18].

Since TCP was designed for wired networks and because TCP is still being the only protocol used in the Internet for reliable transfers, it fails to meet the requirements of wireless networks without reducing the performance of TCP. The standard TCP error recovery algorithms decrease the throughput in wireless links because TCP assumes all losses are caused by congestion, while in fact errors and packet losses in wireless networks can result from random bit errors, fading, shadowing, mobility, low bandwidth, handovers, channel losses, and link latency [15] [16].

Many of the assumptions made in the wired-domain networks are not valid in the wireless-domain networks. From the shared, open-air media, to the characteristics of the physical channels, to the radio signal propagation challenges, to supporting mobile devices, the transport control protocol (TCP) faces many challenges in responding to these emerging needs. Therefore, TCP flows in wireless networks are much shorter than those in wired networks, and their performance is dominated by startup effects such as three-way handshake connection establishment and slow start mechanisms. So, short-lived flows do not reach the steady-state. Therefore, the previous proposed analytical models cannot be used for short-lived flows.

Consequently, alternative analytical models for short-lived TCP flows were proposed in [6] [11]. All the previous models were focused on the TCP performance over wired networks. In [11], a recursive and analytical model for wired networks was proposed to determine the short-lived TCP performance in terms of connection setup time and completion time. On the other hand, Pack et al. [12] extended the previous recursive and analytical model in [11] and introduced a new model for short-lived TCP flows in wireless networks. The model did not take into account the heterogeneity of wireless networks that is demonstrated by different loss packet probability, different bandwidth and different round trip time.

To date, different types of wireless networks are found, such as IEEE 802.11 WLAN, ad hoc wireless networks and 3G cellular networks. They have different characteristics and different packet error behaviors. These different characteristics were not investigated in [12]. Therefore, analytical modeling of the short-lived TCP flows over heterogeneous wireless networks provides a good method for characterizing the TCP performance in terms of average completion time (i.e. the time that the source takes to successfully transfer a given amount of data
We propose a recursive and analytical model for heterogeneous wireless networks with integration of wired network based on the proposed model in [12], and we focus on the average completion time of the short-lived flow as TCP performance metric. We introduce this model for (1) End-to-End scheme that has been proposed in [3] [4] [5] and is used to improve the performance of TCP over wireless links. This scheme solves the wireless loss problems at the transport layers of the sender. (2) Connection Division scheme that breaks the TCP Connection into two connections at the base station. The purpose of this division is to distinguish between different characteristics of networks. The Connection Division scheme is different from the Split Connection scheme that has been proposed in [3] [4] [5] and is used to discriminate the corruption losses of wireless networks from the congestion losses of wired networks. (3) The Local Retransmission scheme in [3] [4] [5] that provides local reliability is not modeled because of the complexity of equations. The NS-2 simulator was used to validate our proposed model. The simulation values were compared with the model values for the two different schemes. Based on the obtained results we found that the proposed models reflect the behavior of TCP well, especially in computing the average completion time for the short-lived TCP flows at the initial stage: connection setup time (i.e. the time that the source takes to successfully complete the three way handshake algorithm), and slow start phase. Before communication can begin between two hosts, a connection must be established by employing a three-way handshake algorithm. TCP manages a retransmission timer which is started when a segment is transmitted. If the timer expires before the segment is acknowledged, then TCP retransmits the segment. The retransmission timeout value (RTO) is calculated dynamically based on measurements of the round trip time (RTT). The basic concept of the TCP congestion control algorithm is window based flow control. The window size determines the amount of data to be sent by the source. Its value will increase when all of the acknowledgements are properly received; otherwise, the window size may decrease. Besides the receiver’s advertised window (awnd) which is used to prevent the sender from overrunning the resources of the receiver, TCP’s congestion control introduced two new variables for the connection: the congestion window (cwnd) which is used to prevent the sender from sending more data than the network can accommodate and the slow start threshold (ssthresh). There are different versions of TCP implementations such as: TCP Tahoe and TCP Reno [14]. TCP Tahoe employs the slow start, the congestion avoidance, and the fast retransmit algorithms. When packet loss is detected, the sender retransmits the lost packet, reduces the congestion window to one and enters slow start. When the sender receives an ACK for the retransmitted packet, it will continue to send packets from the sequence number indicated in the ACK even if some of them are already sent. Thus, if any packet is lost in the meantime, it will be automatically retransmitted via this go-back procedure. In other words, TCP Tahoe will not have to wait for a timeout in the case where multiple packets are lost from the same window. However, frequent losses will force the TCP sender to operate in slow start phase for most of the time. TCP Reno added the fast recovery algorithm. This algorithm does not require slow start for every packet loss. Once the three duplicate ACKs are received, the TCP sender retransmits the lost packet, reduces its congestion window by half, and enters the fast recovery phase instead of the slow start algorithm. A packet loss can be detected either via a time-out mechanism or via duplicate ACKs.
In time-out mechanism each packet has a timer. If it expires, timeout occurs, and the packet is retransmitted. The value of the timer which is called RTO is calculated based on the measurements of the round trip time (RTT).

In duplicate ACKs, if a packet has been lost, the receiver keeps sending acknowledgements but does not modify the sequence number field in the ACK packets. When the sender observes several ACKs acknowledging the same packet, it concludes that a packet has been lost.

The purpose of slow start and congestion avoidance is to control the transmission rate in order to prevent congestion from occurring.

In slow start mechanism [14], when a connection is established, the value of congestion window (cwnd) is first set to 1 and after each received ACK the value is updated to cwnd = cwnd + 1. The exponential growth of cwnd continues until a packet loss is observed, causing the value of ssthresh to be updated to ssthresh = cwnd/2.

After the packet loss, the connection starts from slow start again with cwnd = 1, and the window is increased exponentially until it equals ssthresh. At this point, the connection goes to congestion avoidance phase [14] where the value of cwnd is increased with the pattern cwnd = cwnd + (1 / cwnd), implying linear instead of exponential growth. This linear increase will continue until a packet loss is detected.

Duplicate ACKs [14] are one way of detecting a lost packet. When receiving one duplicate ACK the sender can not distinguish whether the packet has been lost or it is out of order, but after receiving several duplicate ACKs it is reasonable to assume that a packet loss has occurred. The purpose of fast retransmit mechanism is to speed up the retransmission process by allowing the sender to retransmit a packet as soon as it has enough knowledge that a packet has been lost. This means that instead of waiting for the retransmit timer to expire, the sender can retransmit a packet immediately after receiving three duplicate ACKs.

In Tahoe TCP [14] the connection always goes to slow start after a packet loss. However, if the window size is large and packet losses are rare, it would be better for the connection to continue from the congestion avoidance phase, since it will take time to increase the window size from 1 to ssthresh. The purpose of the fast recovery algorithm in Reno TCP is to achieve this behavior. In a connection with fast retransmit, the source can use the flow of duplicate ACKs to clock the transmission of packets. When a lost packet is retransmitted, the values of ssthresh and cwnd will be set to ssthresh = cwnd/2 and cwnd = ssthresh, meaning that the connection will continue from the congestion avoidance phase and increases its window size linearly.

The rest of this paper is organized as follows: section 2 presents the previous work that is similar to our work or has in some other way dealt with similar problems, section 3 presents the proposed model, and the analysis of the analytical solution, section 5 demonstrates the results and gives a complete analysis of the generated plots. Finally, section 6 concludes the paper and highlights the major components of the proposed model.

1. Related Work

Models of TCP can be classified into three types according to the length of the TCP transfer. In case of short-lived transfer, TCP performance is strongly affected by the connection establishment and slow start phases. Models for long-lived transfer capture the steady-state performance of a bulk transfer TCP flow, which is dominated by the congestion avoidance phase. Lately, there are models for TCP transfers of arbitrary length.

Many studies have been done to model TCP performance in the steady-state in wired networks [7] [8] [18]. However, many recent researches have shown that most TCP flows are relatively short-lived [7] [10]. Since the
Internet is expanded to integrate wired and wireless networks and the wireless networks are characterized by random bit errors, fading, shadowing, mobility, low bandwidth, handovers, channel losses, and link latency. Compared to wired networks, the TCP flows in wireless networks are shorter than the flows in wired network.

An analytical model of the steady-state throughput was proposed in [7], which is a function of loss rate and round trip time (RTT). This model shows the behavior of fast retransmission mechanism and the effect of TCP's timeout mechanism on the throughput. Another TCP model was proposed in [8], this model is based on the Markovian model of a single TCP source. It is used to evaluate several performances such as queuing delay, throughput and packet loss of the TCP flows. Mathis, et al., [18] developed a model that predicts the steady state throughput of long-lived TCP transfers in the presence of light to moderate segment losses. The model considers the congestion avoidance phase and assumes no time-out losses. It is a function of maximum segment size, round trip time and dropping probability.

Cardwell, et al., [6] expanded the steady-state TCP model proposed in [7] and developed a new model for the latency of TCP transfers of arbitrary length. They observed that the performance of TCP flows is dependent on the initial stage and is affected by the startup effects such as connection setup time and the slow start mechanism. The extended model characterized the TCP connection establishment and the data transfer latency as a function of data transfer size, round trip time (RTT) and the packet loss rate. The proposed model performed well and characterized the TCP flows for different packet loss conditions when compared with simulation results. On the other hand, the analytical model for estimating the latency and the steady-state throughput of arbitrary length transfers of TCP Tahoe, Reno and SACK was proposed in [9]. It is a function of round trip time and number of data segments. The model computes the latency when there is no loss, a single loss, and multiple losses.

These proposed models were validated using simulations and TCP traces collected from the Internet world, and were focused on the steady-state throughput of the bulk transfer TCP flows.

Recently the analytical rate control (ARC) scheme was proposed for wireless applications [19]. The model describes the TCP throughput. Subsequently, the ability of the wireless receiver to distinguish wireless packet losses is used to drive the correct behavior of the rate control algorithm, which reacts only to congestion events.

Another model that characterizes TCP window evolution as a set of differential equations can be found in [20]. The analytical model to compute the average completion time necessary for the transfer of a file of a given size was proposed in [21]. The model describes the behavior of the multi bottleneck network when there is a mixture of TCP Tahoe and TCP NewReno connections.

A mathematical method was proposed in [22] to derive the throughput of a TCP connection over heterogeneous networks. The mechanism does not need modifications to the sender TCP or the base station. It uses the ACK-splitting method to prevent the throughput degradation caused by packet losses due to the high bit error rate of wireless links. The ACK-splitting method is a method to increase the congestion window size of the sender-side TCP quickly than usual by sending multiple Acknowledgement (ACK) packets when a data packet arrives at the receiver.

On the other hand, Mellia, et al., [11] proposed a recursive and analytical model for short-lived TCP flows to investigate TCP performance in terms of connection establishment time and completion time for the short flows in wired networks. This model is based on the knowledge of the average dropping probability, the average
round trip time (RTT), retransmission time-out and the flow size. The proposed model introduced good results when compared with simulation results.

All the previous models were done for modeling the performance of TCP in wired links. Since TCP was designed for wired networks, TCP suffers from poor performance in wireless links because it cannot distinguish congestion losses from packet losses. To overcome these drawbacks and to improve the performance of TCP over wireless links, many schemes have been proposed in [3] [4] [5]: End-to-End scheme, where the loss recovery is done by the sender, Split Connection scheme which breaks the connection into wired and wireless connections and Local Retransmission scheme that provides local reliability.

Pack et al. [12] expanded the wired TCP model proposed in [11] to a new recursive and analytical model for short-lived TCP flows in wireless networks to predict the TCP performance in terms of completion time at the initial stage. It is based on the knowledge of the average dropping probability, the average round trip time (RTT), retransmission time-out and the flow size. The model did not investigate the heterogeneity of wireless networks and considered a constant packet loss probability. However, heterogeneity is an important issue in wireless networks and has a significant impact on TCP performance. The proposed model introduced good results when compared with simulation results for the three wireless schemes: End-to-End scheme, Split Connection scheme and Local Retransmission scheme.

Although, our model adopts the criteria used in the model of [11] and [12], our model is more general and more practical as it is closer to the real systems. The major differences are in the following aspects: first, it investigates the heterogeneity of wireless networks. Second, it uses multi-wireless networks each one connected with non homogeneous wired link (i.e. speed of link).

3. System Modeling and Analytical Solution

3.1 Model Objectives

The main goal of our proposed model is to estimate the average completion time (i.e. the time that the source takes to successfully transfer a given amount of data) for short-lived TCP flow in heterogeneous wireless networks integrated with wired networks.

A transfer is considered successful when the source receives the acknowledgment ACK for the last segment. This average completion time consists of two parts:

1. The connection establishment time: the time it takes to successfully complete the three way handshake algorithm [2].
2. The time to complete transferring the data segment.

3.2 Assumptions

Recent works have shown that the average size of most short-lived flows is around 10 KB [7] [10]. Assuming a unidirectional for data transfer with maximum transfer unit (MTU) of 1.5 KB, this will result in 10 KB / 1.5 KB = up to 8 segments maximum data transfer, which covers most of short-lived TCP flows. We consider the slow start algorithm implemented in [13]. Since all major TCP proposals such as Reno and New Reno implement the same slow start and the loss recovery algorithms.

We assume that the sender maintains only one timer for all in-flight segments [2] [13]. The timer is associated with the segment with the lowest sequence number that has not been acknowledged yet. When an ACK segment is received, the timer is reset and reassigned to the subsequent segment. Because of the small size of TCP flows, many of them never exit the slow start phase and the probability that the loss packet will trigger the fast recovery algorithm is very low. This is because the window size will be
very small and the sender will not be able to send enough segments that generate the three duplicate acknowledgments that will trigger the fast recovery algorithm. Therefore, most losses will trigger the retransmission timeout (RTO) that will cause the TCP to reenter the slow start phase [12].

3.3 Model Variables

Figure 1 shows our topology and several variables that are needed in heterogeneous wireless networks for short-lived TCP flows. The proposed model must deal with two separate wired links as shown in Figure 1, in order to resolve the mismatch since the links are not homogeneous (i.e. determining the performance of TCP is not the purpose of this separation), to achieve the heterogeneity and to discriminate between different characteristics in different wireless networks, to make the system more practical such as finding a mobile host from a fixed host, to achieve the simplicity and tractability when splitting the connection from a fixed host to a mobile host at base station.

3.4 The Proposed Wireless TCP Model

3.4.1 End-to-End Scheme

Phase 1: Connection Establishment Time, C_{setup}. Derivative and Analysis:

In End-to-End scheme, both wired and wireless losses are solved at the transport layer of the sender. In the connection establishment time as shown in Figure 2, the TCP sender transmits one SYN segment and waits for a SYN-ACK segment. When the sender receives this segment, the sender acknowledges the SYN-ACK segment and then starts the data transmission.

Derivation of Connection Establishment Time, C_{setup}:

In this section, we will derive the connection establishment time depending on the variable distribution as shown in Figure 3 and using the geometric distribution.

\[ C_{setup} = R_1 + R' + T_{s1} (p_{s-total} / (1 - 2 \cdot p_{s-total})) \]

We derive \( p_{s-total} \) according to the properties of probabilities as shown in Figure 4.

\[ p_{s-total} = p_{s1} + \prod_{i=1}^{N} (p_{s2}(i) + p_{s3}(i) - p_{s2}(i) \cdot p_{s3}(i)) - p_{s1} \prod_{i=1}^{N} (p_{s2}(i) + p_{s3}(i) - p_{s2}(i) \cdot p_{s3}(i)) \]

\( R' \) is the average round trip time in wired-wireless links.

\[ R' = \text{average round trip time in wired-wireless links} \]

During the connection establishment time as shown in Figure 2, if the segment is lost in wired or wireless links, the sender will retransmit the SYN segment after retransmission timeout (T_{s1}) and double the SYN timeout value (T_{s1} = 2 \cdot T_{s1}). The term (R_{s1}+R') is the average round trip time when there is no segment loss in the entire network, the index (i) refers to the number of SYN segments dropped, the index (j) refers to the number of back-offs of the retransmission timer (T_{s1}), \( p_{s-total} \) is the overall dropping probability for SYN segment in End-to-End scheme and (N) is the number of wireless networks.

Phase 2: Data Completion Time C (m, w), Derivation and Analysis:

In the following, we will derive the general form for average completion time by using a recursive manner. Let C (m, w) be the average completion time required to successfully send (m) data segments with an initial congestion window of size (w), let C (m, 1) be the average completion time of a flow of size (m) segments with an initial congestion window of size one, and let (N) be the number of wireless networks.

\[ C (m, w) = C (m, m) \text{ for } w \geq m, \text{ because we need only a congestion window of size (m) to send the (m) segments.} \]

\[ C (m, 1) = C (1, 1) + C (m-1, 2), \text{ since after the TCP source receives the ACK for the first segment, it increases its congestion window and transmits the remaining (m-1) segments using an initial congestion window of size two.} \]

We will use up to eight segments as a maximum data transfer as discussed in Section 3.2 to derive the average completion time C (m, w) which is defined by a general form as
shown below:

$R_t$: the average round trip time in wired link.

$T$: the retransmission timeout for the data segment in wired link.

$p_{total}$ = the overall dropping probability for the data segment in End-to-End scheme.

$q_{total} = 1 - p_{total}$ = the overall success probability for the data segment in End-to-End scheme.

$R'$ = the average round trip time in wireless links = $(1/ N) \sum_{(i =1 to N)} [R_2 (i) + R_3(i)]$

\[
\begin{align*}
C(1,1) + C(m, m-1, 2) \\
, 8 \geq m \geq 1, w =1
\end{align*}
\]

\[
C_{setup} = C(1, 1),
\]

\[
C(m, m), m = 1, w > 0
\]

\[
C(m, w) = \begin{cases} 
q_{total} (R_1 + R' + C (m-w, 2w)) \\
p_{total} (T_1 + R_1 + R' + C (1, 1)) \\
+ q_{total} p_{total} (T_1 + C (1, 1)) + q_{total} p_{total} q_{total} (T_1 + C (1, 1)) \\
+ q_{total} p_{total} q_{total} (T_1 + C (2, 1)) + p_{total} q_{total} (T_1 + C (3, 1)) , w \geq m \\
, 8 \geq m \geq 2, w = 2
\end{cases}
\]

\[
C(m, w) = \begin{cases} 
q_{total} (R_1 + R') + 2q_{total} \\
p_{total} (T_1 + R_1 + R' + C (1, 1)) \\
+ q_{total} q_{total} (T_1 + C (1, 1)) + q_{total} p_{total} q_{total} (T_1 + C (1, 1)) + q_{total} p_{total} q_{total} (T_1 + C (2, 1)) + p_{total} q_{total} (T_1 + C (3, 1)) , m = 3, w = 3
\end{cases}
\]

\[
q_{total} (R_1 + R') + 3q_{total} p_{total} (T_1 + R_1 + R' + C (1, 1)) + q_{total} p_{total} q_{total} (T_1 + R_1 + R' + C (1, 1)) + q_{total} p_{total} q_{total} (T_1 + R_1 + R' + C (3, 1)) + 3p_{total} q_{total} (T_1 + C (3, 1)) + 3q_{total} p_{total} q_{total} (T_1 + R_1 + R' + C (2, 1)) +
\]

Analysis of $C(2, 2), C(3, 2), C(4, 2), C(5, 2), C(6, 2), C(7, 2), C(3, 3), C(4, 4), and C(5, 4)$:

Case one: the following terms denote that there is no segment loss, $q_{total} (R_1 + R')$, $q_{total} (R_1 + R' + C(1, 1))$, $q_{total} (R_1 + R' + C(2, 2))$, $q_{total} (R_1 + R' + C(3, 3))$, $q_{total} (R_1 + R' + C(4, 4))$, and $q_{total} (R_1 + R' + C(5, 4))$ in $C(2, 2), C(3, 2), C(4, 2), C(5, 2), C(6, 2)$ and $C(7, 2)$, respectively.

Upon receiving the ACK (After $R_1+R'$), the sender doubles its congestion window and sends the remaining segments. The probability for this case is $q_{total}^2$.

Case two: the following terms denote that the first segment is successfully transmitted while the second segment is lost, $q_{total} p_{total} (T_1 + R_1 + R' + C(1, 1))$, $q_{total} p_{total} (T_1 + R_1 + R' + C(2, 2))$, $q_{total} p_{total} (T_1 + R_1 + R' + C(3, 2))$, $q_{total} p_{total} (T_1 + R_1 + R' + C(4, 2))$, $q_{total} p_{total} (T_1 + R_1 + R' + C(5, 2))$, and $q_{total} p_{total} (T_1 + R_1 + R' + C(6, 2))$ in $C(2, 2), C(3, 2), C(4, 2), C(5, 2), C(6, 2)$ and $C(7, 2)$, respectively.

Upon receiving the ACK for the first segment (After $R_1+R'$), the sender increases its congestion window. At the same time, the retransmission timer is rescheduled ($T_1$). Upon expiration of the retransmission timer ($T_1$), the congestion window is reset to one segment. But, because TCP uses cumulative acknowledgments and uses only one retransmission timer, the sender retransmits
the lost and the remaining segments at the same time as a burst of two segments (i.e. the possible loss of the third segment can be detected only when the ACK of the second segment is received, so the retransmission of the second segment and of the third segment must happen at the same time). The probability for this case is $q_{\text{total}} p_{\text{total}}$.

Case three: the following terms denote that the first segment is lost while the second segment is successfully transmitted, $p_{\text{total}} q_{\text{total}} (T_1 + C (1, 1))$, $p_{\text{total}} q_{\text{total}} (T_1 + C (2, 1))$, $p_{\text{total}} q_{\text{total}} (T_1 + C (3, 1))$, $p_{\text{total}} q_{\text{total}} (T_1 + C (4, 1))$, $p_{\text{total}} q_{\text{total}} (T_1 + C (5, 1))$ and $p_{\text{total}} q_{\text{total}} (T_1 + C (6, 1))$ in $C(2, 2)$, $C(3, 2)$, $C(4, 2)$, $C(5, 2)$, $C(6, 2)$ and $C(7, 2)$, respectively.

After retransmission timeout ($T_1$) for the first segment, the congestion window is reset to one segment and the sender retransmits the lost and the remaining segments. The probability for this case is $p_{\text{total}} q_{\text{total}}$.

Case four: the following terms denote that the two segments are lost, $p_{\text{total}}^2 q_{\text{total}} (T_1 + C (2, 1))$, $p_{\text{total}}^2 q_{\text{total}} (T_1 + C (3, 1))$, $p_{\text{total}}^2 q_{\text{total}} (T_1 + C (4, 1))$, $p_{\text{total}}^2 q_{\text{total}} (T_1 + C (5, 1))$, $p_{\text{total}}^2 q_{\text{total}} (T_1 + C (6, 1))$ and $p_{\text{total}}^2 q_{\text{total}} (T_1 + C (7, 1))$ in $C(2, 2)$, $C(3, 2)$, $C(4, 2)$, $C(5, 2)$, $C(6, 2)$ and $C(7, 2)$, respectively.

After the retransmission timer expires ($T_1$), the congestion window is reset to one and the sender resends all segments again. The probability for this case is $p_{\text{total}}^2$.

Similarly to the previous cases, we can derive equations $C(3, 3)$, $C(4, 4)$ and $C(5, 4)$ that have $2^x.2^x$ combinations, where $x$ is the minimum $(m, w)$.

In case three in $C(4, 4)$ and $C(5, 4)$, we found that after three duplicate acknowledgements, the fast recovery algorithm will be triggered, and the sender will retransmit the segment immediately without waiting for an RTO.

### 3.4.2 Connection Division Scheme

#### Phase 1: Connection Establishment Time, $C_{\text{setup}}$

**Derivation and Analysis:**

In Connection Division scheme we divide a TCP connection at the base station into two separate TCP connections: wired and wireless TCP connections as shown in Figure 5. The main goal of Connection Division scheme is to distinguish between different characteristics of networks. Whereas, the Split Connection scheme is used to discriminate the congestion losses in wired networks from the corruption losses in wireless networks by splitting the connection at base station in to two connections: wired and wireless connections. We deal with wired-wireless link as End-to-End scheme.

**Derivation of Connection Establishment Time, $C_{\text{setup}}$:**

In this section, we will derive the connection establishment time depending on the variable distribution as shown in Figure 3 and using the geometric distribution.

$$C_{\text{setup}} = C_{\text{setup-wired}} + C_{\text{setup-wireless}} = [R_1 + (1 + p_{s1}) \sum_{i=1}^{\infty} p_{s1}^i] + [R' + (1-p_{s-total}) \sum_{i=1}^{\infty} (p_{s-total})^i]$$

$$= (3R_1/2 + T_{s1} (p_{s1} / (1-2p_{s1}))) + R' + T_{s2} (p_{s-total} / (1-2p_{s-total}))$$

We derive $p_{s-total}$ according to the properties of probabilities as shown in Figure 4.

$$p_{s-total} = \prod_{i=1}^{N} (p_{s2} (i) + p_{s3} (i) - p_{s2} (i) p_{s3} (i))$$

$$R' = (1/N) \sum_{i=1}^{N} [R_2 (i) + R_3 (i)]$$

$$T_{s2} = \text{Maximum} \{T_{s2} (i), \text{worst case}\}$$

The base station sends the data to the destination node only after receiving all the data from the fixed host. The connection is started in wired-wireless links only after it is completed in the wired link, so an additional time $(R_1/2)$, which is equal to half of the round trip time in the wired link is required. During the connection establishment time as shown in Figure 5, if the segment is lost in wired link, the sender will retransmit the SYN segment after retransmission timeout ($T_{s1}$) and double the SYN timeout value ($T_{s1} = 2*T_{s1}$). The terms $R_1$, $R'$ is the average round trip time when there is no segment loss in the wired and wired-wireless links, the index $(i)$ refers to the number of SYN segments dropped, the index $(j)$ refers to the number of back-offs of the retransmission
timer \((T_{s1})\), \((p_{s1})\) is the dropping probability for the SYN segment in wired link \((p_{s-\text{total}})\) is the overall dropping probability for SYN segment in wired-wireless links and \((T_{s2})\) is the retransmission timer for the wired-wireless link.

**Phase 2: Data Completion Time \(C(m, w)\).**

**Derivation and Analysis.**

In the following, we will derive the general form for average completion time by using a recursive manner. Let \(C(m, w) = C_{\text{wired}}(m, w) + C_{\text{wireless}}(m, w)\) be the average completion time required to successfully send \((m)\) data segments with an initial congestion window of size \((w)\) in wired and wired-wireless links, respectively.

We compute the average completion time for wired-wireless links exactly as it was computed in End-to-End scheme. Let \(C(m, 1)\) is the average completion time of a flow of size \((m)\) segments with an initial congestion window of size one. \(C(m, 1) = C(1, 1) + C(m-1, 2)\) as discussed in End-to-End scheme.

We will use up to eight segments as a maximum data transfer as discuss in section 3.2 to derive the average completion time \(C(m, w)\) which is defined by a general form as shown below: \(C(m, w)\) equal:

\[
\begin{align*}
C_{\text{wired}}(1, 1) + C_{\text{wired}}(m-1, 2) \\
+ C_{\text{wireless}}(1, 1) + C_{\text{wireless}}(m-1, 2) \\
, 8 \geq m > 1, w = 1 \\
\text{setup} = C_{\text{wired}}(1, 1) + C_{\text{wireless}}(1, 1) \\
, m = 1, w > 0 \\
C_{\text{wired}}(m, m) + C_{\text{wireless}}(m, m) \\
, w \geq m
\end{align*}
\]

\[
\begin{align*}
C_{\text{wired}}(m, w) + C_{\text{wireless}}(m, w) \\
q_1^2 (R_1 + C_{\text{wired}}(m-w, 2w)) + q_1 p_1 (T_1 + R_1 + C_{\text{wireless}}(m-1, 2)) + p_1 q_1 (T_1 + C_{\text{wired}}(m-1, 1)) + p_1^2 (T_1 + C_{\text{wireless}}(m, 1)) +
\end{align*}
\]

\[
\begin{align*}
q_1^2 \text{total} (R_1 + C_{\text{wireless}}(m-w, 2w)) + q_1^r \text{total} p_1^r (T_2 + R_1 + C_{\text{wired}}(m-1, 2)) + p_1^q \text{total} q_1^q (T_2 + C_{\text{wireless}}(m-1, 1)) + p_1^2 \text{total} (T_2 + C_{\text{wireless}}(m, 1)) \\

, 8 \geq m > 2, w = 2
\end{align*}
\]

\[
\begin{align*}
q_1^3 R_1 + 2 q_1^3 p_1 (T_1 + R_1 + C_{\text{wired}}(1, 1)) + p_1 q_1^2 (T_1 + C_{\text{wireless}}(1, 1)) + q_1 p_1^2 (T_1 + R_1 + C_{\text{wired}}(2, 1)) + 2 p_1^q q_1 (T_1 + C_{\text{wireless}}(2, 1)) + p_1^3 (T_1 + C_{\text{wireless}}(3, 1)) + q_1^3 \text{total} R_1 + 2 q_1^2 \text{total} p_1^r \text{total} (T_2 + R_1 + C_{\text{wireless}}(1, 1)) + p_1^q \text{total} q_1^q \text{total} (T_2 + C_{\text{wireless}}(2, 1)) + 2 p_1^2 \text{total} (T_2 + C_{\text{wireless}}(2, 1)) + p_1^3 \text{total} (T_2 + C_{\text{wireless}}(3, 1)) \\

, m = 3, w = 2
\end{align*}
\]

\[
\begin{align*}
q_1^4 R_1 + 3 q_1^3 p_1 (T_1 + R_1 + C_{\text{wireless}}(1, 1)) + p_1 q_1^2 (R_1 + C_{\text{wireless}}(1, 1)) + q_1^3 (T_1 + R_1 + C_{\text{wireless}}(2, 1)) + 3 q_1^2 p_1 (T_1 + C_{\text{wireless}}(2, 1)) + 3 p_1^q q_1 (T_1 + C_{\text{wireless}}(2, 1)) + p_1^3 (T_1 + C_{\text{wireless}}(4, 1)) + q_1^3 \text{total} R_1 + 3 q_1^2 \text{total} p_1^r \text{total} (T_2 + R_1 + C_{\text{wireless}}(3, 1)) + 3 p_1^q \text{total} q_1^q \text{total} (T_2 + C_{\text{wireless}}(3, 1)) + 3 q_1^2 \text{total} p_1^r \text{total} (T_2 + C_{\text{wireless}}(3, 1)) + 3 p_1^q \text{total} q_1^q \text{total} (T_2 + C_{\text{wireless}}(4, 1)) + p_1^4 \text{total} (T_2 + C_{\text{wireless}}(4, 1)) \\

, m = 4, w = 4
\end{align*}
\]

\[
\begin{align*}
q_1^4 (R_1 + C_{\text{wireless}}(1, 1)) + 2 q_1^3 p_1 (T_1 + R_1 + C_{\text{wireless}}(2, 1)) + 2 q_1^2 q_1^3 (R_1 + C_{\text{wireless}}(2, 1)) + q_1^3 (T_1 + R_1 + C_{\text{wireless}}(4, 2)) + 3 p_1^q q_1 (T_1 + C_{\text{wireless}}(4, 1)) + 3 q_1^2 p_1^q (T_1 + R_1 + C_{\text{wireless}}(3, 2)) + 3 p_1^q q_1^2 (T_1 + C_{\text{wireless}}(3, 1)) + p_1^4 (T_1 + C_{\text{wireless}}(5, 1)) + q_1^4 \text{total} (R_1 + C_{\text{wireless}}(1, 1)) +
\end{align*}
\]

\[
\begin{align*}
2 q_1^3 \text{total} p_1^r \text{total} (T_2 + R_1 + C_{\text{wireless}}(2, 2)) + 2 p_1^q \text{total} q_1^q \text{total} (R_1 + C_{\text{wireless}}(2, 1)) + q_1^3 \text{total} p_1^r \text{total} (T_2 + R_1 + C_{\text{wireless}}(4, 2)) + 3 p_1^q \text{total} q_1^q \text{total} (T_2 + C_{\text{wireless}}(4, 1)) + 3 q_1^2 \text{total} p_1^q \text{total} (T_2 + R_1 + C_{\text{wireless}}(3, 2)) + 3 p_1^q \text{total} q_1^q \text{total} (T_2 + C_{\text{wireless}}(3, 1)) + p_1^4 \text{total} (T_2 + C_{\text{wireless}}(5, 1)) \\

, m = 5, w = 4
\end{align*}
\]
R₁: the average round trip time in wired link
T₁: the retransmission timeout for the data segment in wired link
p₁: the dropping probability in wired link
q₁: the success probability in wired link
p' total = the overall dropping probability for the data segment in wired-wireless link.
q' total = the overall success probability for the data segment in wired-wireless link.

\[ p'_{total} = \prod_{i=1}^{N} \left( p_2(i) + p_3(i) - p_2(i) p_3(i) \right) \]
\[ q'_{total} = 1 - p'_{total} \]

R': the average round trip time in wired-wireless links
\[
R' = \left( \frac{1}{N} \right) \sum_{i=1}^{N} \left( R_2(i) + R_3(i) \right)
\]
T₂: the retransmission timeout for the data segment in the second connection at base station = Maximum \{T₂(i), worst case\}.

The average completion time for data transmission in the Connection Division scheme is the sum of the completion times in the wired link and wired-wireless links. Each completion time is calculated separately using the same recursive model as in End-to-End scheme.

The analysis of these equations is the same as discussed before for the End-to-End scheme, the main difference is in the using of variables in wired link and wired-wireless links.

3.4.3 Local Retransmission Scheme

It was found that deriving the equations for the Local Retransmission scheme would be very complex. Since the dropping probabilities in the wired and wired-wireless links are mutually independent, we should consider all cases that could occur. Each segment may be transmitted successfully or lost in either the wired link or the wired-wireless links.

So when we try to find the C (m, w), we find that the number of possible cases in wired link and wired-wireless links are equal to \(2^x.2^x.2^x\), where \(x = \text{minimum} \{m, w\}\), which result in complex combinations.

For example, to find the simplest equation, C (2, 2), we need \(2^2.2^2.2^2 = 64\) combinations as shown in Table 2.1. To find completion time C (3, 3), we need \(2^3.2^3.2^3 = 512\) combinations.

We did not derive the analytical model for the Local Retransmission scheme, because of the complexity of combinations.

4. Performance Evaluations and Results

4.1 Simulation Methodology

NS-2 [17] was used as a simulation tool to validate our proposed model and to evaluate its performance in terms of average completion time for short-lived TCP flow. For simulations we used the network topology shown in Figure 6. All values used in our simulations are the same as those used previously in other analytical models in [11] [12]. The number of wireless networks (N) is a simulation parameter that varies. The bandwidth and round-trip delay of the wired links are 10 Mbps and 20 ms, respectively. On the other hand, the wireless link is assumed to follow the specification of IEEE 802.11 with omni antenna, the bandwidth, round-trip delay are 1 Mbps and 20 ms, respectively. Initial retransmission timers (RTO) in wired and wireless links are set to 1500 and 500 ms, respectively. The dropping probabilities in all wired links are 0.1 %. Whereas the dropping probabilities in wireless links are different from one wireless network to another in order to achieve the heterogeneity which is due to the differences in their characteristics such as low bandwidth, random bit errors, fading, shadowing, channel losses, and link latency. In terms of packet dropping, the uniform distribution is used.

Our simulations are based on the TCP-Reno, which is the most popular implementation in the Internet world. In all simulations, TCP-Reno sender (Fixed Host) transfers data segments to the receivers (Mobile Host). In the following sections, we discuss the results obtained for End-to-End scheme and Connection Division scheme.
4.2 Results and Performance Analysis

4.2.1 End-to-End Scheme
In this section, we run the simulations using the same simulation methodology mentioned in section 4.1. We used two, three and six wireless networks in these simulations. The dropping probabilities for these wireless networks are 1%, 2%, 6%, 4%, 6% and 2% respectively.

Figures 7, 8, and 9 show that the average completion time increases as the session size (i.e. the number of data segments in the flow) increases. From the above plots (7, 8, and 9), it is observed that the NS-2 simulation and the model result obtained from equations for the End-to-End scheme is relatively close to each other as shown in Table 1. Also in these figures, the behavior of TCP in computing the average completion time for N ≥ 6 seemed to be slightly increasing line which holds for all values of the parameters of the model.

Table 1: Difference between Simulation Values and Model Values for End-to-End Scheme

<table>
<thead>
<tr>
<th>Number of wireless networks (N)</th>
<th>Average differences for the average completion time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10.073</td>
</tr>
<tr>
<td>3</td>
<td>6.327</td>
</tr>
<tr>
<td>6</td>
<td>7.553</td>
</tr>
</tbody>
</table>

These plots give a conclusion that the percentage of difference between simulation and model value for average completion time (in Sec) is less than 3%.

4.2.2 Connection Division Scheme
In this section we run the simulations using the same simulation methodology mentioned in section 4.1. The difference here is that we have two different connections; one from fixed host to the base station and the other from base station to the mobile host. We used two, three and six wireless networks in these simulations. The dropping probabilities for these wireless networks are 1%, 2%, 6%, 4%, 6% and 2% respectively.

Figures 10, 11, and 12, show that the average completion time increases as the session size (in segment) increases. From the above plots (10, 11, and 12), it is observed that the NS-2 simulation and the model result obtained from equations for the Connection Division scheme is relatively close to each other as shown in Table 2.

Table 2: Difference between Simulation Values and Model Values for Connection Division Scheme

<table>
<thead>
<tr>
<th>Number of wireless networks (N)</th>
<th>Average differences for the average completion time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>20.702</td>
</tr>
<tr>
<td>3</td>
<td>23.75</td>
</tr>
<tr>
<td>6</td>
<td>25.36</td>
</tr>
</tbody>
</table>

These plots give a conclusion that the percentage of difference between simulation and model value for average completion time (in Sec) is less than 6%.

A comparison of the results shows that the difference between the simulation and the model values of the Connection Division scheme is larger than that of the End-to-End scheme. This is because the proposed model for Connection Division scheme calculates the total average completion time as a simple sum of the average completion times in wired and wireless links. Also, the purpose of Connection Division scheme is different from the Split Connection scheme. The Connection Division scheme is used to distinguish between different characteristics of networks. The connection is started in wired-wireless links only after it is completed in the wired link. This will result in addition time (R/2) as shown in Figure 5. Whereas, the Split Connection scheme is used to discriminate the congestion losses in wired networks from the corruption losses in wireless networks.
5. Conclusions
The Transmission Control Protocol (TCP) is widely used in the Internet world and several analytical models were proposed to improve the performance of TCP in terms of throughput. However, these models cannot be used for short-lived TCP flow which is considered as the most important characteristic in the Internet.
In this paper, we proposed a recursive and analytical model for short-lived TCP flows and we focused on heterogeneous wireless networks that have different packet dropping probability. We proposed this model for (1) End-to-End scheme, (2) Connection Division scheme. The Local Retransmission scheme was not modeled because of the complexity of equations.
The average completion time in the proposed wireless TCP model for heterogeneous wireless networks is calculated to determine the TCP performance using the average round trip time, retransmission timeout, flow size and the segment dropping probability.
We used the NS-2 simulator to verify the accuracy of our model and we compared the simulated result for heterogeneous wireless topology with the calculated values that will be obtained from our proposed model.
Based on the obtained results, we found that the proposed model for the two wireless schemes reflects TCP behavior well, specifically the session completion time at the initial stage (i.e. slow start).
The results obtained show that the simulation and the model values are relatively close to each other with the percentage of difference for the average completion time (in Sec) being less than 3% and 6% for the End-to-End scheme and the Connection Division scheme, respectively.

6. References
<table>
<thead>
<tr>
<th>Variable Wired1</th>
<th>Its Meaning</th>
<th>Variable Wired2</th>
<th>Its Meaning</th>
<th>Variable Wireless</th>
<th>Its Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>Average round trip time</td>
<td>$R_2(i)$</td>
<td>Average round trip time</td>
<td>$R_3(i)$</td>
<td>Average round trip time</td>
</tr>
<tr>
<td>$P_{s1}$</td>
<td>Synchronization segment (SYN) dropping probability</td>
<td>$P_{s2}(i)$</td>
<td>Synchronization segment (SYN) dropping probability</td>
<td>$P_{s3}(i)$</td>
<td>Synchronization segment (SYN) dropping probability</td>
</tr>
<tr>
<td>$T_{s1}$</td>
<td>Retransmission Time Out (RTO) for SYN segment</td>
<td>$T_{s2}(i)$</td>
<td>Retransmission Time Out (RTO) for SYN segment</td>
<td>$T_{s3}(i)$</td>
<td>Retransmission Time Out (RTO) for SYN segment</td>
</tr>
<tr>
<td>$P_1$</td>
<td>Data segment dropping probability</td>
<td>$P_2(i)$</td>
<td>Data segment dropping probability</td>
<td>$P_3(i)$</td>
<td>Data segment dropping probability</td>
</tr>
<tr>
<td>$q_1$</td>
<td>Data segment success probability</td>
<td>$q_2(i)$</td>
<td>Data segment success probability</td>
<td>$q_3(i)$</td>
<td>Data segment success probability ($q_3 = 1 - p_3$)</td>
</tr>
<tr>
<td>$T_1$</td>
<td>Estimated RTO for data segment</td>
<td>$T_2(i)$</td>
<td>Estimated RTO for data segment</td>
<td>$T_3(i)$</td>
<td>Estimated RTO for data segment</td>
</tr>
</tbody>
</table>

Figure 1: The Network Topology Used in the Model
**Figure 2:** Connection Establishment in End-to-End Scheme

- **Fixed Host (FH):** PC Desktop
- **Base Station:** \( T_{s1} \cdot P_{s1} \cdot R_{s1} \)
- **Subsystems:** \( P_{s1}(i) \), \( R_{s1}(i) \)
- **Mobile Host (MH):** Laptop
- **AP:** \( P_{s2}(i) \), \( R_{s2}(i) \)

Variables:
- \( P_{s1} = \text{probability of fail for synchronization (SYN) packet} \)
- \( R_{s1} = \text{round trip time} \)
- \( R' = \text{Average round trip time from Base Station to MH} \)
- \( P_{s_{\text{total}}} = \text{probability of fail for synchronization (SYN) packet from Base Station to MH} \)
- \( P_{s_{\text{total}}} = \text{probability of fail for synchronization (SYN) packet from FH to MH} \)
- \( T_{s1} = \text{retransmission time out for synchronization (SYN) packet} \)
- \( P_{1} = \text{probability of fail for data segment} \)
- \( P'_{\text{total}} = \text{probability of fail for data segment from Base Station to MH} \)
- \( P_{\text{total}} = \text{probability of fail for data segment from FH to MH} \)

**Figure 3:** Variables Distribution

- Probability of Success in SubSystem 1 = \((1 - P_{11}) \cdot (1 - P_{12}) = 1 - (P_{11} + P_{12} - (P_{11} \cdot P_{12})) \)
- Probability of Fail in SubSystem 1 = \((P_{11} + P_{12} - (P_{11} \cdot P_{12})) \)
- Probability of Fail in SubSystem 2 = \((P_{21} + P_{22} - (P_{21} \cdot P_{22})) \)
- Probability of Fail in whole system = \(P + \prod_{i=1}^{N} (P_{i1} + P_{i2} - (P_{i1} \cdot P_{i2})) - (P \cdot \prod_{i=1}^{N} (P_{i1} + P_{i2} - (P_{i1} \cdot P_{i2}))) \), where \( N \) is the number of SubSystems

**Figure 4:** Derivation of Fail Probability
Figure 5: Connection Establishment in Connection Division Scheme

Figure 6: Network Topology and Simulation Scenario
Figure 7: Average Completion Time in End-to-End Scheme with N = 2 (N: Number of Wireless Networks)

Figure 8: Average Completion Time in End-to-End Scheme with N = 3

Figure 9: Average Completion Time in End-to-End Scheme with N = 6
Figure 10: Average Completion Time in Connection Division Scheme with N = 2

Figure 11: Average Completion Time in Connection Division Scheme with N = 3

Figure 12: Average Completion Time in Connection Division Scheme with N = 6