TCP Performance Dynamics and Link-layer Adaptation based Optimization Methods for Wireless Networks

Jatinder Pal Singh, Student Member, IEEE, Nick Bambos, Member, IEEE, Ahmad Bahai, Member, IEEE, Bangnan Xu, Member, IEEE, and Gerd Zimmerman, Member, IEEE

Abstract—Almost a decade long research on the performance of TCP in wireless networks has resulted in a several proposals and solutions to the problem of throughput throttling. Over the years, end-to-end solutions [1], [2], [3], [4], [5], [6], split-connection schemes [7], [8] and adaptive link-layer solutions [1], [9] have been proposed, refined and evaluated for their relative merit [1]. Several of these measures have their share of drawbacks, and/or specify in applicability. With the continuing emergence of wireless technologies ever since the work on TCP performance over wireless began, smart link-layer mechanisms like adaptive modulation and coding, power control, Link Adaptation, Incremental Redundancy etc. have been evolving and getting deployed. Although some recent works [10], [11] have tried to use the link-layer adaptiveness to TCP’s benefit, these do not base their optimization on the dynamics of TCP and also do not suggest the application of their methods to real link-layer designs. In our work, we outline an optimization framework based on the congestion control dynamics of a bulk TCP flow, and demonstrate its application to networks which offer link-layer adaptive measures. The approach adopted can be summarized as follows; we first isolate and present recurrence patterns in TCP’s congestion control dynamics, that are useful for identifying its operation in a wireless network. We then overlay a generalization optimized approach which can be used for enhancing TCP throughput via link-layer-adaptiveness measures. Finally we demonstrate the performance benefits that can be achieved via application of the optimization approach to EGPRS and IEEE 802.11a. The main contributions of this paper are twofold: development of a generalized optimization framework that permits TCP performance optimization via adaptive link-layer mechanisms, and demonstration of measures to noticably enhance TCP throughput via power control and link adaptation in networks like EGPRS and IEEE 802.11a.

Index Terms—TCP, 3G Networks, EDGE, EGPRS, IEEE802.11a, Link Adaptation, Power Control, Dynamic Programming

I. INTRODUCTION

With the proliferation in use of wireless devices over past several years, there has a growing interest for access to the mobile Internet and web-based applications. As TCP carries most of the Internet traffic today, the integration of the mobile devices with the wired world necessitate its use as a transport layer protocol in wireless networks. The problem of TCP source throttling in a wireless scenario has been a research focus for a decade now. As TCP was originally designed to operate in wireline environments with highly reliable underlying channel conditions, it faces operational challenges in wireless scenario which is characterized by sporadic losses and disconnections. The primary cause of TCP’s malperformance is its perception of the wireless losses to be an indication of network congestion. The resulting congestion control and avoidance mechanisms that ensue, lead to low TCP throughput. While TCP SACK and NewReno are able to recover from segment losses relatively efficiently, these cannot still distinguish between congestion and wireless link losses.

The solutions that have been proposed over the past several years to enhance TCP performance in wireless networks include end-to-end schemes [1], [2], [3], [4], [5], [6], split connection approaches [7], [8] and TCP aware link layer protocols [1], [9]. Each has its share of advantages and drawbacks. The end-to-end schemes like the Explicit Loss Notification (ELN) preserve TCP’s semantics but require modifications to TCP. The infeasibility of Internet wide deployment of such changes poses a severe restriction to the practical utility of such solutions. The split connection approaches can employ a separate specialized protocol over the wireless hop and have the TCP connection terminated at the Base Station (BS) or an Access Point (AP). The approach is however marred by increased processing overheads, violation of end-to-end semantics of TCP acknowledgements, and slow, complicated handoffs.

Enhanced link-layer reliability [12], [13] has been investigated as a mechanism to improve TCP performance in wireless scenario. However link layer designs that are TCP unaware, cannot efficiently shield TCP from the wireless losses [12], and are also associated with increased rate and delay variability [14]. On the other hand, approaches on line of SNOOP protocol [1], [9] represent a TCP aware link-layer design. While SNOOP preserves the end-to-end semantics of TCP and does not require any changes to TCP implementation, it has its own share of limitations. It cannot be used for the case when TCP data and ACKs do not both traverse through the BS/AP. It also has overheads of SNOOP cache maintenance. During the interim period between handoffs, the BS/AP to which the handoff is occurring cannot snoop on any acknowledgements sent from the mobile host. Another
None of the aforementioned solutions encompass or utilize adaptive wireless system design features like FEC, transmission power, and existence of multiple transmission modes. There have been some recent efforts [10], [11] to examine adaptive link layer measures for TCP throughput optimization, where the authors adopt standard steady state TCP throughput expressions and make an attempt for optimization via adaptive FEC, ARQ and power control. The TCP window dynamics and congestion events are not considered in the work. We observe that the work on TCP throughput modeling via consideration of its dynamics [15], [16] and throughput optimization via adaptive system design has thus remained addressed orthogonally. We argue that for performance optimization, link-layer needs to be adaptive to the instantaneous dynamics of a TCP flow. In [17] we model TCP’s congestion avoidance dynamics and evaluate adaptive power control measures for throughput enhancement in a simplified scenario.

In this paper we explore the adaptiveness of link layer mechanisms in a wireless system for optimization of TCP throughput. We present a general optimization framework based on TCP dynamics and demonstrate its utility via adaptive power control, and link adaptation. Our approach does not require changes to TCP or any extensive implementations. We begin by analyzing in Section II, the performance dynamics of a bulk transfer TCP flow. Unlike several bulk transfer throughput evaluations which ignore TCP’s slow start phase [16], [18], we model both the slow start and congestion avoidance mechanisms. We outline an optimization framework in Section II based on TCP dynamics. We then demonstrate in Section III, TCP performance enhancement via proposed optimization measures. We start by adopting a simplified scenario and assess the impact of transmission power adaptation on TCP throughput. We then demonstrate the application of proposed optimization measures to EGPRS and WLANs.

II. Congestion Control Dynamics of TCP

We in this section discuss the dynamics of a steady state bulk-transfer TCP flow. The bulk-transfer originates from a TCP sources with a large amount of data to send. The protocol dynamics can be described via a) TCP window size evolution and b) congestion events. TCP congestion window designates the limit on maximum amount of data (or the number of segments) that can be transmitted without waiting for an acknowledgement (ACK). The receiver advertizes a similar limit on outstanding data based on its buffer limitations. At any time, a TCP sender can send as many unacknowledged segments as allowed by the minimum of congestion and receiver windows.

The slow start and congestion avoidance algorithms [19] determine the evolution of congestion window and are used by a TCP sender to control the amount of unacknowledged data being injected into the network. During the slow start phase, TCP increments congestion window by 1 segment for each ACK that acknowledges new data. This entails doubling of the window every Round Trip Time (RTT). The congestion avoidance phase on the other hand is marked by an increment of 1 in window size every RTT. The slow start phase governs dynamics until the window size reaches a threshold (called the slow start threshold) beyond which congestion avoidance takes over. The idea is to make TCP probe for network capacity by increasing the window size first aggressively and then cautiously. In case of a Timeout (TO) or Triple Duplicate (TD) loss indication [19], the value of the threshold is set to the minimum of 2 segments and half of congestion window. At all times TCP’s window size is limited by the receiver advertised window.

The TO and TD indications characterize the congestion (or loss) events of TCP. The TO indication occurs when the TCP sender is waiting for an ACK and the retransmission timer expires. TCP infers that the packet has been lost: it reduces window size to 1 segment, retransmits the packet and doubles the Retransmission Time Out (RTO) value. This retransmission procedure is repeated until the packet is ACKed. Subsequently, TCP window dynamics follow the slow start or congestion avoidance algorithms, depending on the values of threshold and congestion window size. The TD loss indication on the other hand is characterized by the arrival at the sender of three duplicate ACKs. A duplicate ACK is generated by the receiver in response to arrival of out-of-order segment and bears the sequence number of the next expected segment. TCP’s Fast Retransmit Algorithm uses the arrival of 3 duplicated ACKs (4 ACKs with same sequence number) as an indication that a segment has been lost. Following the TD indication, TCP performs the retransmission of what appears to be the missing segment, without waiting for the retransmission timer to expire.

Most TCP versions today implement the above discussed congestion control algorithms. TCP Reno in addition employs Fast Recovery [20] that enables it to recover from single packet loss in a window without a timeout. TCP New Reno [21] can recover from multiple segment losses in a window via partial acknowledgements. TCP SACK can counter multiple segment loss as well, via selective acknowledgements. It can employ congestion control algorithms similar to Reno, or can utilize information from its SACK options [22].

We in our work model the slow start and congestion avoidance algorithms with maximum window size limitation and consider both TD and TO congestion events. Most of the Reno implementations today have been rendered obsolete by New Reno/SACK deployment. Hence we assume that TCP is able to recover from multiple segment losses in a window in the event of TD loss indication. A TCP flavor may implement its own fast recovery process: the one assisted by SACK options (TCP SACK) or partial ACK mechanisms (TCP NewReno); we do not specifically model any particular one in our work. Further, we do not consider TCP’s delayed ack mechanism for reasons of clarity of presentation of our work, and also since it does not bring out any insights relevant to the scope of this paper. However the benefits of delayed ACK measures can be easily applied to the proposals in this work.

We observe that any given trace of TCP’s window size evolution comprises of certain atomic patterns which can be isolated and represented as in figs. 1 and 2. These window size
variation patterns are termed as cycles. Two kinds of cycles are identified: one beginning in congestion avoidance phase (fig. 1) and the other beginning in slow start phase (fig. 2). We term them as CA begin and SS begin cycles respectively. Each of these cycles can either end due to TD loss indication (figs. 1(a) and 2(a)) or a Timeout (figs. 1(b) and 2(b)).

In the next section we will discuss optimization of TCP performance based on the dissection of TCP window evolution into cycles.

III. OPTIMIZATION FRAMEWORK

We now present the operational scenario of TCP and delineate a methodology to relate transport layer dynamics to wireless channel conditions and network congestion. Fig. 4 pictures a network scenario for traversal of TCP flow through wired infrastructure and the wireless hop. An instance of operation could be a mobile station communicating with a remote server (or vice versa), with wireless connectivity between the mobile host and AP/BS.

TCP segments are transmitted over the wireless channel with suitable header encapsulation by the lower layers. A segment may be transmitted via a variable number of physical layer frames. Error detection and error correction mechanism (FEC, ARQ, etc.), play a role in ascertaining the successful delivery of a TCP segment. In addition, measures like Link Adaptation (LA) and Incremental Redundancy (IR) [23], could be employed networks like EDGE and WLANs (e.g. IEEE 802.11a).

We characterize the progression of SS begin and CA begin cycles in terms of rounds. A round [16] involves back to back transmission of a window of TCP segments. It has a duration equal to the larger of window transmission time, and the interval between commencement of window transmission
and arrival of an ACK for a segment in the window. The duration of a round can vary depending on the window size, the variation in transmission time of the frames on the wireless channel and also the RTT in wireline domain.

An instance of window size evolution with rounds is shown for a CA begin cycle in Fig. 5. The window size increases by one if all the segments transmitted in a round are ACKed. In the event of a segment loss in round \( R \), the TCP cycle can end with a TD or a TO indication. When three or more of the packets following the lost packet are successful, duplicate ACKs for the lost packet are generated and a TD loss indication occurs. In that case, the cycle terminates at round \( R + 1 \). We assume that TCP is able to recover from single or multiple losses in the window by using, for instance, SACK options, and we do not specifically model the recovery process. In the event when less than three segments make it successfully to the receiver, TCP times out. On each subsequent attempt to transmit the unACKed segment, the timeout duration doubles. The cycle finally ends at round \( K \) when the lost segment is retransmitted successfully. An SS begin cycle can be explained likewise in terms of its window size evolution in rounds.

We next discuss the optimization framework for TCP dynamics based on dissection of TCP operation into cycles. Fig. 3 outlines the framework. The dynamics of a round in slow start or congestion avoidance phase are shown in Fig. 3(a).

In a round, cost is incurred for transmission of window of segments. The cost may be attributed to transmission power, transmission time, etc., and is represented by \( C(r) \). The total cost to be incurred (called cost to go) starting from round \( r \) till the end of the cycle is represented by \( J(r) \). In case a segment in the round is lost, additional cost \( C_{\text{Loss}}(r) \) is incurred. If there is TD loss indication, \( C_{\text{Loss}}(r) \) is simply evaluated as the negative of the throughput achieved during the cycle (with the throughput evaluated as the number of packets transmitted during the cycle divided by the duration of the cycle). The cycle ends at round \( r + 1 \) with the completion of transmission of window of segments following the segment for which TD loss indication occurred (e.g. round R+1 in fig. 5).

In the event that the loss indication is a timeout, \( C_{\text{Loss}}(r) \) is simply evaluated as the negative of the throughput achieved during the cycle. The cost of the timeout phase is given by \( T(r + 2, r) \) where \( T(k, r) \) is the cost to go from round \( k \) given that the timeout occurred due loss indication of a segment in round \( r \). Note that in the TD loss indication case, round \( r + 1 \) involves completion of transmission of segment window following the lost segment. During round \( k \) of timeout phase, the cost corresponding to the transmission of segment window is termed as \( C_{\text{seg}}(k) \). In case the segment transmitted during round \( k \) is successfully ACKed, the timeout phase and the cycle ends with a termination cost \( C_{\text{TO}} \) which is the negative of the throughput achieved during the cycle.

The optimization approach for the discussed framework is motivated by Dynamic Programming (DP) [24] principles and involves minimization of cost-to-go \( J(r) \). We will reuse the cost symbols in Fig. 3 and formulate detailed optimization equations. We let vector \( \gamma \) represent the channel condition vector corresponding to the wireless state encountered by the segments in a round and \( c \) represent the corresponding congestion vector. The dimension of \( \gamma \) depends on the scenario for which optimization is being attempted. For instance, when the wireless channel decorrelates over transmission duration of a physical data block, then the dimension of \( \gamma \) is the number of blocks required to transmit the TCP segments.

Fig. 3. Evolution of a Slow Start/Congestion Avoidance round (left) and Timeout phase (right) for SS begin and CA begin cycles.

Fig. 5. TCP segment transmission dynamics in a CA begin cycle. The cycle ends with TD indication at round \( R + 1 \) in case three of the dotted segments are successful, or else the cycle ends in Timeout phase at round \( K \).
in a round. The congestion vector \( c \) has a dimension equal to the number of segments in a round. The quantities \( \gamma \) and \( c \) together ascertain the success of delivery of TCP segments in a round. We assume a reliable TCP ACK delivery in our work. The channel and congestion state vectors have probability distribution functions denoted by \( f_\gamma(\gamma) \) and \( f_c(c) \) respectively. The success probability vector of segments in the round is denoted by \( s \). The success probability of a round is a function of \( s \), and can be expressed as \( S_{\text{round}}(s) \). The cost \( C \) of round \( r \) depends on the success probability of the segments, the wireless channel state vector, the congestion state vector and the round number.

The key to the optimization process is to ascertain the segment success probability vectors \( s \) for all given \( \gamma \) and \( c \). The set of these optimal vectors would result in lowest cost to go averaged over the channel state and congestion distribution. \( J(r) \) can thus be represented as the cost-to-go minimized over all possible success probability vectors.

\[
J(r) = \int_{\gamma} \int_{c} \min \left[ C_t(r, s, \gamma, c) \right] f_\gamma(\gamma)f_c(c) \, d\gamma \, dc
\]

where the cost-to-go for given \( \gamma \) and \( c \) is

\[
C_t(r, s, \gamma, c) = C(r, s, \gamma, c) + S_{\text{round}}(s) J(r+1) + (1 - S_{\text{round}}(s)) C_{\text{Loss}}(r, s)
\]

The terminal cost \( C_{\text{Loss}}(r, s) \) incurred on the loss of one or more segments in round \( r \), is expressed in (2). The vectors \( \gamma', c' \) and \( s' \) respectively denote the channel state, congestion vector, and success probabilities of the segments in the rounds in round \( r+1 \) for the case where there is TD loss in round \( r \). \( p_{TD}(r, s, s') \) represents the probability of a TD loss indication, and can be evaluated as the probability of three or more segments being successful amongst the ones in round \( r+1 \) and those transmitted following the first lost segment in round \( r \). We will discuss the formulation of \( p_{TD}(r, s, s') \) in subsequent sections. \( C_L(r, s', \gamma, c) \) represents the cost associated with transmission of segments in round \( r+1 \). Round \( r+1 \) is the terminal round for a TD loss indication. \( C_B(r) \), the terminal cost for the TD loss case, is modeled as the negative of throughput attained during the cycle.

\[
C_B(r) = \lambda \left( - \frac{\sum_{i=1}^{r} W_i + n_i}{\sum_{i=1}^{r} D_i} - \rho(W_r) \right)
\]

where \( D_i \) represents the duration of round \( i \), \( n_i \) is the number of segments transmitted in the terminal round \( r+1 \) of TD loss case, \( W_i \) represents the window size during round \( i \), and \( \lambda \) is the scaling factor between the transmission and throughput costs. The term \(-\rho(W_r)\) where \( \rho \) is an increasing function of \( r \) is introduced to the influence the evolution of the current cycle to favor high throughput in the subsequent cycle. For instance, when a congestion avoidance phase ends in a TD loss indication with terminal round window of \( W_r \), the next cycle would have a initial window size of \( W_r/2 \). To favor a higher initial window size for the next cycle, we make the cost for current cycle have a deduction which is an increasing function of \( W_r \). By virtue of the cost deduction, termination of a cycle with a larger window size would be preferred, since the overall objective is to minimize the cost of a cycle.

The event that loss indication in a cycle is a timeout is represented by probability \( 1 - p_{TD}(r) \) in (2). In accordance with the dynamics represented in fig. 3(b), the corresponding cost-to-go from round \( k \) of the timeout phase, given that the TO indication occurred in round \( r \), is given by

\[
T(k, r) = \int_{c} \int_{\gamma} \min \left[ C_{\text{seg}}(r, s, \gamma, c) + s C_{\text{TO}}(k, r) \right]
\]

\[
+ (1 - s) T(k+1, r) \right] f_c(c) \, d\gamma \, dc
\]

where \( s \) is the success probability of the transmitted segment, and \( C_{\text{TO}} \) represents the terminal cost for the cycle ending in a timeout.

\[
C_{\text{TO}}(k, r) = \lambda \left( - \frac{\sum_{i=1}^{r} W_i + n_i + (k-r)}{\sum_{i=1}^{r+1} D_i + f(k) T_0} - \rho(W_r) \right)
\]

where \( T_0 \) is the timeout value and \( f(k) \) denotes the timeout sequence given by

\[
f(k) = \begin{cases} 
2^{k-i} - 1, & k-i \leq 7 \\
127 + 64(k-i-7), & k-i \geq 8
\end{cases}
\]

The optimization framework discussed in this section can be applied to both CA\begin{symbol}BEGIN\end{symbol} and SS\begin{symbol}BEGIN\end{symbol} cycles via the associated window evolution. Given the initial size, the window size during the rounds of SS\begin{symbol}BEGIN\end{symbol} cycle can be ascertained (a linear increase until a loss indication). Similarly with a given initial window size and slow start threshold, the window size can be determined for CA\begin{symbol}BEGIN\end{symbol} cycle as a function of round number. The solution to optimization equations yields the cost minimizing segment success probability vector for all channel state and congestion vectors and for every round in a cycle. There is a set of CA\begin{symbol}BEGIN\end{symbol} cycles for which optimization is done. This set contains cycles beginning with different initial window sizes and SS\begin{symbol}BEGIN\end{symbol} cycles for which optimization is done, comprises of all cycles beginning with different initial window sizes and slow start thresholds.

IV. TCP THROUGHPUT OPTIMIZATION VIA ADAPTIVE LINK-LAYER TECHNIQUES

Based on the framework presented in the previous section, we now demonstrate TCP’s performance enhancements via
adaptive link layer measures. We show that by suitably modeling the cost of transmission, measures like power control and link adaptation can be utilized for throughput optimization. To discuss the intuition behind the approach, we begin by adopting a simple evaluation model and assess the merits of power adaptation for efficient TCP dynamics. We then move on to investigate the benefits of optimization for EGPRS and IEEE802.11a networks.

A. Adaptive Power Control

The success probability of TCP segments in a wireless network can be regulated by adapting the transmission power of physical layer frames on the wireless channel. This enables a TCP dynamics adaptive power control for enhanced throughput. Power control [25] is desirable in wireless networks for several reasons including limiting the interference that a wireless link generates in a multi-user environment, and conserving energy for battery power limited mobile devices. We will see how controlling transmission according to TCP dynamics results in significant throughput enhancement over conventional power control methods.

1) Evaluation Model: For present evaluation we neglect the time spent by TCP in slow start phase and and incorporate only CAbegin cycles in TCP’s window evolution. While many works including [16], [18] have adopted this assumption, we will relax it in subsequent subsections. We attribute the loss of TCP segments to wireless channel errors and the losses due to congestion are neglected. Furthermore, we adopt a simple channel and MAC mode. The constant length TCP segments are assumed to be encapsulated in single data frames which are transmitted at a suitable power level. Adaptive modulation, error correction mechanisms like FEC, and ARQ are disregarded here. The wireless channel is modeled as AWGN Rayleigh flat-fading channel and the modulation is taken to fixed as BPSK. The channel is assumed to decorrelate over transmission of successive frames, and hence the frames undergo independent fading. The wireless link is assumed to be high-bandwidth so that it does not present a bottleneck to the realizable throughput. The duration of each round is assumed to be fixed and taken as average RTT. While a high-bandwidth wireless link will comply with this assumption adopted in popular modeling approaches [16], we will relax it after this section and evaluate round duration accurately. The Retransmission timeout To is taken to be fixed, and the updates are considered only in further subsections.

The channel state vector $\gamma$ is represented by Signal to Noise Ratio (SNR) values experienced on the wireless channel by frames encapsulating TCP segments. A TCP segment is in error if any of the bits of the encapsulating frame is in error. Furthermore we assume that error detection mechanisms are capable of identifying bit errors incurred during transmission. Neglecting the headers encapsulating a TCP segment being neglected, the segment error probability $p$ can be expressed as

$$p(\gamma) = 1 - [1 - p_b(\gamma)]^N;$$

where $\gamma$ is the SNR during the frame transmission duration and $N$ is the frame length in bits. For an AWGN channel and BPSK modulation, the bit error probability is

$$p_b(\gamma) = Q(\sqrt{2\gamma})$$

where $Q$ denotes the Q-function and the SNR $\gamma$ follows the Rayleigh exponential distribution with mean $\bar{\gamma}$.

$$f_\gamma(\gamma) = \frac{1}{\bar{\gamma}} \exp\left(-\frac{\gamma}{\bar{\gamma}}\right), \gamma \geq 0$$

We model the cost of transmission (Section III) of TCP segments in a round as the aggregate transmission power for the frames encapsulating the segments in the round. Greater the transmission power, more the associated cost because of higher generated interference and battery power drain. As fixed modulation and constant frame size entail a constant transmission time, the power cost for a segment can also be translated to transmission energy cost via a scalar constant. Hence maximizing throughput with constrained power also implies achieving the same objective with constrained energy.

With a reasonably simplified model and the approach above, we will demonstrate the intuition behind TCP optimization measures presented in section III and also investigate the impact of power control measures on TCP throughput.

2) Optimization Framework: Adopting the assumptions discussed in the previous subsection, (1) reduces to the simplified following form (9) where $\gamma_1, \gamma_2, \ldots, \gamma_W$ are the components of SNR vector $\gamma$ for round $r$. The cost of a round is modeled in terms of the transmission power $P_{\text{seg}}$ of the segments in the round.

The optimization framework allows us to determine the cost-minimizing success probability vector in a round for every given $\gamma$, and ascertain the transmission power required for transmission of segements. However there may be practical limitations to adopting the optimization guidelines on the fly. When transmission of segments in a round commences, we may only have an estimate of SNR for the current frame transmission, and it may be difficult to predict the SNRs for all frame transmission in the current round. However the selection of target success probability vector can be done only by selecting the optimal success probability for a given $\gamma$. To overcome the problem, we modify the formulation to yield a causal cost model. We first select the targeted packet success probability to be the same for all segments in the round. Then $s$ simplifies to a single-element optimization parameter $s$. Not assuming apriori information about the SNR experienced by the segments in the round, we replace the power cost of each segment in (9) by power consumption averaged over the channel state distribution, $f_\gamma(p(s(\gamma))f_\gamma(\gamma))d\gamma$. We also assume that for the case when there is a loss indication in a round, the target success probability of the following round (e.g. round $R+1$ in fig. 5) is the same as that of the current round. Then, the cost formulation $J(r)$, with no apriori knowledge of $\gamma$, is given by (10). Accordingly, (2) reduces to (11).

We next discuss the evaluation of $p_{TD}(r,s)$, the probability of a TD loss indication conditioned on the event that a segment is lost in round $r$. The probability of one of more segments in a round being lost is $(1 - s^{W_r})$, where $s$ is the target success probability for segments in a round. The probability that the first segment lost in a round is the $i^{th}$ one is given by $s^{i-1}s$.
\[ J(r) = \int_{\gamma} \min_s \left[ \sum_{j=1}^{W_r} P_{\text{seg}}(s_j, \gamma_j) + \prod_{j=1}^{W_r} s_j J(r+1) + (1 - \prod_{j=1}^{W_r} s_j) C_{\text{Loss}}(r, s) \right] f_{\gamma}(\gamma) d\gamma \] (9)

\[ J(r) = \min_s \left[ W_r \int_{\gamma} P_{\text{seg}}(s, \gamma)f_{\gamma}(\gamma)d\gamma + s^{W_r} J(r+1) + (1 - s^{W_r}) C_{\text{Loss}}(r, s) \right] \] (10)

\[ C_{\text{Loss}}(r, s) = p_{TD}(r, s) \left[ n_t \int_{\gamma'} P_{\text{seg}}(s, \gamma')f_{\gamma'}(\gamma')d\gamma' + C_B(r) \right] + \left[ 1 - p_{TD}(r, s) \right] T(r+2, r) \] (11)

\[ p_{TD}(r, s) = \frac{1}{(1 - s^{W_r})} \sum_{i=1}^{W_r} \left( \begin{array}{c} W_r - 1 \\ i \end{array} \right) s^i (1 - s)^{W_r - 1 - i} I_{\{W_r > 3\}} \] (12)

\[ = \sum_{i=3}^{W_r-1} \left( \begin{array}{c} W_r - 1 \\ i \end{array} \right) s^i (1 - s)^{W_r - 1 - i} I_{\{W_r > 3\}} \] (13)

\[ T(k, r) = \min_s \left[ \int_{\gamma} P_{\text{seg}}(s, \gamma)f_{\gamma}(\gamma)d\gamma + sC_{TO}(k, r) + (1 - s)rT(k+1, r) \right] \] (14)

The formulation for \( p_{TD}(r, s) \) is then given by (12) where \( I_{\{W_r > 3\}} \) is the indicator function which assumes a value of 1 when \( W_r > 3 \) and is 0 otherwise. On simplification, \( p_{TD}(r, s) \) reduces to (13). The timeout cost from (3) can be expressed as (14).

3) Performance assessment: Based on the discussed assumptions and methodology, we simulate bulk transfer of TCP segments to assess TCP throughput with adaptive power control mechanisms. Bulk transfer implies that the sender is saturated and has data to send all times.

As a TCP segment is transmitted, its success probability is ascertained by the state of the wireless hop of the TCP connection and transmission power level on the link. The SNR experienced by a frame encapsulating TCP packet is evaluated based on exponential Rayleigh fading distribution and the channel model presented before. The transmission power levels are selected depending on the employed power adaptation scheme.

The approach for evaluating TCP throughput with adaptive power control is discussed next. (10) and (14) are first converted to finite-period. Round numbers \( R_t \) and \( K_t \), defined to be the final rounds for evaluation of optimal solutions to (10) and (14), are selected as 200 each. The selection is based on taking into consideration that in any realistic environment it is highly likely that a TCP cycle ends before the round number value of 200. Our simulations too justify this selection. (10) and (14), can then be solved together with (11), (13), (2), (4) and (5) for optimizing power control. We choose the function \( \rho(W_r) \) in (2) and (4) to be \( \frac{W_r}{2} \) and approximate \( n_t \) as \( \frac{W_r}{2} \) or \( W_r \) depending on whether the round \( r \) incurring a segment loss belongs to slow start or congestion avoidance phase.

The following procedure is then performed for the SSbegin cycle and for a range of values of the cost ratio \( \lambda \). A look up table \( s_{opt} \) comprising of target success probability \( s_{opt}(r, W_{init}) \) for every round number \( r \) and initial window size \( W_{init} \) is generated. For a given round and initial window, the value \( s \) that minimizes the integrands in (10) (and likewise (14)) is the success probability stored in the look up table. \( W_{init} \) ascertains the evolution of window size in a cycle for determining the terminal costs (2) and (4), and can take values from 1 to the maximum window size, \( W_{max} \).

For simulation run, SNR value is drawn every round from a Rayleigh distribution. The target segment success probability for the round and initial window size with which the cycle began is then retrieved from the look-up table. With the SNR value and the target success probability, the transmission power for frames in that round is determined. Throughput is evaluated as the fraction of number of segments transmitted during the simulation run and the duration of the run. Several runs are performed to obtain an average throughput value. With increasing values of \( \lambda \), the priority given to throughput in the dynamic programming formulation increases relative to the power cost. Hence increasing throughput values are obtained, but at the cost of higher average power.

The throughput variation based on DP solutions is evaluated with a segment size 1500bytes, RTT 250ms, To 3s and the maximum window size \( W_{max} \) as 48. The performance is compared with that with Truncated Channel Inversion (TCI) [26] power control policy with a cut-off threshold \( \gamma_c \) of 5dB. The power adaptation for TCI is given by

\[ \frac{P(\gamma)}{P} = \begin{cases} 2 & \gamma \geq \gamma_c \\ 0 & \gamma < \gamma_c \end{cases} \] (15)

where \( P \) represents the average transmission power and is set to the average power of dynamic programming based power control for a given average SNR. The constant received
SNR, $\sigma$, subject to the constraint,
\[ \int_{\gamma}^{\infty} \frac{P(\gamma)}{P} d\gamma = 1 \quad (16) \]
is given by
\[ \sigma = \frac{1}{\int_{\gamma_c}^{\infty} f_\gamma(\gamma) d\gamma} \quad (17) \]

Fig. 6 shows the TCP throughput performance with Dynamic Programming (DP) based power adaptation, TCI and constant power. The average power for all the power adaptation measures is the same for a given average SNR. As can be seen, the DP based solutions result in significant enhancement in TCP throughput. For instance at 20 dB the optimized throughput is almost 10 folds as compared to the one with TCI.

4) Operational analysis of throughput enhancement: We now discuss some insights into the ability of DP based optimization in enhancing TCP performance. Fig. 7 plots the target success probability ($s_{opt}(r,W_{init})$) for segments in a round and different initial window sizes for a cycle. The target success probability function is evaluated as solution to (10) and (14), for an average SNR of 18.5 dB. The different curves correspond to different initial window sizes at which a TCP cycle begins. It can be seen that TCP dynamics aware optimization sets a higher target success probability for early rounds of a cycle.

To understand the rationale behind protection of early rounds, consider the case when a CA begins TCP cycle starts with a small initial window size. TCP begins to probe network capacity by increasing the window size every round. If a segment loss occurs in one of early rounds, the window size is further reduced (to half via TD or to one via TO indication) and the throughput takes an adverse hit. On the other hand, a segment loss later on in a cycle when window size is high, is not that detrimental to TCP throughput. Thus a segment loss early on in a cycle should be prevented. Such an early round protection mechanism can be seen in Fig. 7. The protection is more prolonged for smaller initial window sizes and is relatively less for higher initial window sizes. For example the target success probability for a cycle starting with initial window size is close to 1 until a round number 16, but has a high value only until round 4 when the initial window size is 25.

Fig. 8 plots the average transmission power per segment noted for simulation scenario with average SNR of 18.5dB. The initial rounds can be seen to be protected by higher transmission power. However the TCI and constant power schemes do not distinguish between segments belonging to different rounds while deciding the transmission power. From the throughput performance plot in Fig. 6, we see that the absence of such intelligence in a MAC power adaptation scheme can severely affect the throughput attained at the transport layer. Fig. 9 plots the average window size during different rounds for average SNR of 18.5dB. Higher window size can be seen to be more likely for DP based power adaptation, and hence a higher attained throughput.

To conclude, the DP based power adaptation considers the dynamics of TCP while adapting the transmission power: a feature absent in TCI and constant power policy. The earlier rounds are crucial for reaching a higher window size, and hence losses in these rounds need to be prevented via a higher target success probability.

B. Link Adaptation for enhanced TCP throughput over EG-PRS

The Global System for Mobile (GSM) communications has been evolving towards a 3rd generation mobile cellular system with the standardization of technologies like the General Packet Radio Services (GPRS) [27] and the Enhanced Data...
Rates for GSM Evolution (EDGE) [28]. EGPRS [23] is the part of EDGE technology targeted towards the enhancement of GPRS, the the packet data component of GSM. While the fundamental physical layer specifications of GSM (TDMA bursts, channel spacing, etc.) are retained in EGPRS, enhancements like multi-level modulation, better code granularity, and incremental redundancy are introduced to increase the link level throughput over that in GPRS. In this section we generalize the several assumptions made in assessment of TCP optimization framework via power control in a simplified scenario, and evaluate link adaptation measures leading to TCP throughput enhancement in EGPRS. We begin by providing a brief introduction to EGPRS data transmission mechanisms.

1) Data Transmission methods in EGPRS: EGPRS employs nine coding schemes five of which (MCS 5 to MCS 9) are 8-PSK modulated and the remaining use GMSK modulation. Each of the coding schemes has different puncturing patterns. MCS 1,2,5 and 6 have two classes of puncturing patterns (P1 and P2), while the remaining coding schemes have three classes, namely P1, P2, and P3. As a Packet Data Unit (PDU) from the Logical Link Control (LLC) layer is broken into 20 ms Radio Link Control (RLC) data blocks, the amount of data fitted into the block is decided by the coding scheme. The process of matching the coding scheme to the prevailing radio link conditions is called Link Adaptation (LA). Together with Incremental Redundancy (IR), LA constitutes the main component of Link Quality Control (LQC) [23] for EGPRS. IR involves the combination of retransmitted RLC data blocks with the previous transmission attempts of the block. For example consider that MCS-9 is used to transmit a data block. The block will be transmitted using the puncturing scheme P1. If the received block is erroneous, the retransmission will be done using the puncturing scheme P2. The receiver does not discard the previously transmitted block but uses it for joint decoding with the current retransmission, thereby resulting in a lower effective code rate.

The throughput performance of EGPRS coding schemes has been studied in several works including [23], [29]. The throughput for a TU ideal frequency hopping and interference-limited scenario is presented for different EGPRS coding schemes, as a function of $C/I_c$ (ratio of carrier power to co-channel interference) in [23]. The plots, which have been parameterized in [30], are reproduced in fig. 10. The throughput in these plots is evaluated by assessing the Block Error Rate (BLER) as a function of $C/I_c$ and using the relation

$$Throughput = [1 - BLER_{C/I_c}] R_{MCS-X}$$

where $R_{MCS-X}$ is the user data rate for the coding scheme MCS-X. Note that the evaluation of BLER for obtaining throughput plots for different coding schemes can be done by noting the number of transmissions required for successful delivery of a data block at a given $C/I_c$ and without IR recombination. The IR recombination is shown to reduce the this BLER [23]. The probability that a data block transmitted via MCS-1 to 9 modes is in error, can thus be obtained by noting the throughput for a given $C/I_c$ from Fig. 10 and using the relation (18).

The data bits per block for each of the transmission modes (MCS-1 to 9) and the associated data rates, modulation schemes and coding rates [32] are presented in Table I.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Rate</th>
<th>$R_{1+2}$</th>
<th>$R_{1+2+3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCS-1</td>
<td>176 bits</td>
<td>8.8 kbps</td>
<td>0.5</td>
</tr>
<tr>
<td>MCS-2</td>
<td>224 bits</td>
<td>11.2 kbps</td>
<td>0.46</td>
</tr>
<tr>
<td>MCS-3</td>
<td>296 bits</td>
<td>14.8 kbps</td>
<td>0.38</td>
</tr>
<tr>
<td>MCS-4</td>
<td>352 bits</td>
<td>17.6 kbps</td>
<td>0.49</td>
</tr>
<tr>
<td>MCS-5</td>
<td>448 bits</td>
<td>22.4 kbps</td>
<td>0.37</td>
</tr>
<tr>
<td>MCS-6</td>
<td>592 bits</td>
<td>29.6 kbps</td>
<td>0.5</td>
</tr>
<tr>
<td>MCS-7</td>
<td>696 bits</td>
<td>44.8 kbps</td>
<td>0.55</td>
</tr>
<tr>
<td>MCS-8</td>
<td>1088 bits</td>
<td>59.2 kbps</td>
<td>0.66</td>
</tr>
<tr>
<td>MCS-9</td>
<td>1184 bits</td>
<td>59.2 kbps</td>
<td>0.53</td>
</tr>
</tbody>
</table>

**TABLE I**

Parameters for MCS-1 to 9. $R_{1+2}$ and $R_{1+2+3}$ denote effective code rates on IR recombination of a block with one and two previous transmissions respectively.

We note that the selection of coding schemes as per prevailing radio conditions alone may not be optimal for TCP traffic. As we will demonstrate, coding scheme selection as per TCP dynamics and prevailing radio conditions, results in better transport layer throughput.

2) Optimization Framework and evaluation model: We model the cost of transmission of a TCP segment as the time required to transmit the segment via the RLC data blocks. We consider a slow frequency hopping scenario where the
transmission frequency is updated every 20ms TDMA burst. The block error process is taken to decorrelate over different bursts as TCP data is sent in a time slot belonging to a burst.

The congestion in the network, c in (1), (2) and (3) is modeled as a constant TCP segment loss process. We assume that the wireless link adaptation process does not have information pertaining to the network congestion, as would be the case in a real scenario. The optimization is hence not performed over congestion characteristics. The framework as presented in sec. IV-A.2 then applies, with the cost \( P_{\text{seg}}(s, \gamma) \) changed to \( T_{\text{seg}}(s, \gamma) \). \( T_{\text{seg}}(s, \gamma) \) represents the total time over the air during which bursts containing the data corresponding to a TCP segment are transmitted, with \( \gamma \) denoting the \( C/I_c \) of the channel. With these representations, (10), (11), (13) and (14) apply to this scenario. We will discuss in the next subsection, the choice of target success probability vector \( s \), and the selection of modes of transmission for EGPRS blocks based on DP solutions.

EGPRS block transmission mechanisms, TCP dynamics and network congestion effects are considered for the flow of TCP data in the network. A TCP segment is transmitted over variable number of EGPRS blocks. The modes for transmission of each of these blocks are selected depending on the link adaptation policy employed. For TCP dynamics aware link-adaptation, the mode selection will be described in the following subsection. The TCP Timeout period \( T_o \) is no longer assumed to be constant but updated based on RTT observations, as described in [33]. The TCP segment size is taken to be 512 bytes. Results will be presented for different values of round trip delay \( D \) which includes the delay experienced by the segment in the Internet and also the wired part of GPRS network, but excludes the wireless link transmission time. Delay due to transmission on the wireless link is variable and depends on the transmission modes selected for the blocks containing TCP segment data.

We model the wireless channel variations as a Markov Chain with its two states good and bad represented by \( C/I_c \) uniformly distributed in the intervals \([0,15] dB \) and \([16,30] dB \) respectively. The time spent in these states is exponentially distributed with respective rates \( r_g \) and \( r_b \). The discrete time representation of the channel is shown in fig. 11, with the transition probabilities \( p_{g,b} \) and \( p_{b,g} \) related to the transition rates as \( \frac{\gamma}{\gamma + r_g} \) and \( \frac{r_b}{\gamma + r_b} \). The same model has been used for the work in [31].

3) Dynamic Programming Solutions and Performance Assessment: Methodology similar to the one highlighted in Sec. IV-A.3 is adopted for obtaining dynamic programming solutions. However, we now generalize the TCP dynamics to have both SSbegin and CAbegin cycles. The duration of rounds in this case is no longer assumed to be constant and can vary with the data transmission time on the wireless link: an instance being the case when the link bandwidth is a bottleneck as compared to the delays aside from the wireless link.

The evaluation of cost minimizing success probability \( s \) (in adapted (10) (14)) of TCP segments in a round for a given \( C/I_c \) can be done by scanning a range of possible values of \( s \) and finding the value for which optimization equations are satisfied. To decrease the computational complexity we restrict the set of values that are scanned. This set \( S_{\text{set}} \) is evaluated in terms of the average segment error probability by assuming identical modes to be employed for transmission of all blocks corresponding to a TCP segment. Note that this assumption is restricted only to the evaluation of \( S_{\text{set}} \), and not carried over to actual block transmission or analysis via simulations.

\[
S_{\text{set}} = \{ 1 - \overline{SER}(1), ..., 1 - \overline{SER}(9) \} \quad (19)
\]

where \( \overline{SER}(m) \) is the average error probability for transmission of segment via blocks of mode-\( m \).

\[
\overline{SER}(m) = BLER_{RTX}(m)^{n_{Blocks}(m)} \quad (20)
\]

\( n_{Blocks}(m) \) denotes the number of blocks of mode-\( m \) required to transmit a TCP segment and \( BLER_{RTX}(m) \) is the average block error rate for mode-\( m \) with ARQ retransmissions, and is given by

\[
BLER_{RTX}(\gamma, m) = \prod_{r=1}^{\text{RTL}} \overline{BLER}_r(m) \quad (21)
\]

\( \overline{BLER}_r(m) \) in the above equation represents the average (over channel state) error probability of a block when transmitted via mode-\( m \). The subscript \( i \) denotes the transmission attempt and varies from 1 for first transmission to the retransmission limit \( \text{RTL} \) for the last one (the number RTL includes all transmission of a data block, including the first one). Note that the transmission attempts yield different block error probabilities because of IR combination. We will discuss their evaluation in next subsection.

With the reduced success probability set evaluated as described above, dynamic programming solutions are obtained for target segment success probability. For the CAbegin cycle, target success probability is a function of round number and initial window size, i.e., \( s^CA_{\text{opt}}(r, W_{\text{init}}) \). This constitutes the look-up table \( S^CA_{\text{opt}} \). The target success probability for SSbegin cycle is in addition a function of the slow start threshold \( s_{\text{thresh}} \) at the beginning of the cycle, and can be expressed as \( s^CA_{\text{opt}}(r, W_{\text{init}}, s_{\text{thresh}}) \); there values constitute the look-up table \( S^{SS}_{\text{opt}} \). The look up tables are used to ascertain the transmission modes for blocks of a TCP segment.

We next discuss the mode selection procedure for transmission of a TCP segment data. Let the transmission round be \( r \) and the estimated \( C/I_c \) for current burst transmission be \( \gamma \). Then mode \( m_b \) which minimizes the cost of deviation from the target success probability and transmission time is selected for transmission.
\[ m_{\text{opt}} = \min_{m \in \{1, \ldots, 9\}} [(s_{\text{target}}(m) - s_{\text{mode}}(m, \gamma))^+ + \beta(t_{\text{target}}(m) - t_{\text{mode}}(m, \gamma))^+] \tag{22} \]

where \( \beta \) is the cost ratio, and the function \((x)^+\) represents \( \max\{0, x\} \). \( s_{\text{target}}(m) \) and \( t_{\text{target}}(m) \) represent for mode-\( m \), the target success probability and target transmission time required by the optimization methodology for transmission. \( s_{\text{mode}}(m, \gamma) \) and \( t_{\text{mode}}(m, \gamma) \) on the other hand represent the estimate of success probability and transmission time that can be offered by mode-\( m \) for an \( C/I_c \) of \( \gamma \). The target success probability for mode selection is taken to be

\[ s_{\text{target}}(m) = \left[ s_{\text{opt}} \right]_{\text{numBlock}(m)} \]

with \( s_{\text{opt}} \) being the target segment success probability obtained by DP solutions. \( s_{\text{opt}} \) is drawn from the appropriate look-up tables for SSbegin or CAbegin cycles depending on the cycle to which the transmission belongs. Note that the estimation of target block error probability from segment error probability is done by considering identical mode blocks to constitute transmission of the segment. This is in line with the evaluation of \( S_{\text{set}} \). The actual transmission of a TCP segment, however, is not restricted by any such assumptions and occurs via blocks of different modes ascertained to be optimal for current round and \( C/I_c \).

The block success probability \( s_{\text{mode}}(m) \) in (22) is ascertained as

\[ s_{\text{mode}}(m) = [1 - \text{BLER}_{RTX}(\gamma, m)] \]

where \( \text{BLER}_{RTX}(\gamma, m) \) is the estimate of block error rate offered by mode-\( m \) with ARQ retransmissions. Having information pertaining only to the current block transmission \((C/I_c \) values for subsequent attempts to transmit the block can’t be estimated), \( \text{BLER}_{RTX}(\gamma, m) \) can be estimated as

\[ \text{BLER}_{RTX}(\gamma, m) = \text{BLER}_1(\gamma, m) \prod_{r=2}^{RTL} \text{BLER}_r(m) \tag{23} \]

The target time \( t_{\text{target}}(m) \) in (22) is taken as

\[ t_{\text{target}}(m) = \frac{RTT}{W_c \text{numBlock}(m)} \tag{24} \]

where RTT is the average round trip time of a TCP segment in the network, and \( W_c \) is the TCP window size. Finally, the transmission time estimate for a mode-\( m \) block transmission is taken to be \( t_{\text{mode}}(m) = T_{\text{block}}(\gamma, m) \), where \( T_{\text{block}}(\gamma, m) \) is given as

\[ T_{\text{block}}(\gamma, m) = [1 - \text{BLER}_1(\gamma, m)] T_{\text{burst}} + \text{BLER}_1(\gamma, m) \sum_{r=2}^{RTL} (\text{BLER}_r(m))^r \tag{25} \]

\( T_{\text{burst}} \) in above equation is 20ms, the duration of EGPRS radio burst.

With the above methodology, the mode selection policy (22) enables the selection of mode for transmission of blocks comprising of TCP segment data in round \( r \) and with \( C/I_c \) estimate for block transmission as \( \gamma \).

4) The Early Round Protection (ERP) Scheme: The early round protection scheme is based on the observation that earlier rounds in a TCP cycle are crucial for attaining higher throughput and need a transmission bias (e.g., higher transmission power, robust coding) to provide a greater success rate to the segments. This is so because if there is a segment loss before the bandwidth-delay product of the network is filled by TCP data, the TCP window size is reduced, and hence the throughput takes a hit. For the case when the maximum TCP window size permits transmission of more data than what is required to fill the bandwidth-delay product of the network, then the round adaptive transmission of data blocks need be done up to a point when this limit is reached. This applies to EGPRS based networks where wireless link data rates are not as high as for instance in case WLANs. As per the ERP scheme, adaptive mode selection is done for EGPRS on the basis of measures highlighted in the previous section. After the growth of window size beyond a limit \( W_{\text{erp}} \), ordinary LA/IR adaptation procedures for data transmission take over.

To demonstrate the merits of ERP scheme, simulations are performed for two separate sets of cases: a simplified one with no IR and the other with approximate IR. The cost ratio \( \beta \) in (22) is found to give optimal or close to optimal throughput for a value of 0.1 when the unit of time is seconds. The parameter \( \lambda \) in (4) and (2), which represents the ratio between the cost associated with TCP segment transmission and throughput attained, is varied to find optimal TCP throughput. The EGPRS Radio Link Control and block transmission mechanisms are executed in the simulations. The TCP Timeout period \( To \) is initialized to 3s and is updated based on RTT observations, as described in [33]. \( W_{\text{erp}} \) is approximated as the ratio of average RTT of the TCP segments and the segment transmission time. The segment transmission time is estimated as the average over \( \gamma \) and mode-\( m \) of \( n\text{Blocks}(j) T_{\text{block}}(\gamma, m) \), where \( T_{\text{block}}(\gamma, m) \) is given by (25).

For the evaluation case with no IR, the BLER does not depend on the transmission attempt number since there is no combining. The BLER subscripts in (23) and (25) designating the transmission number can hence be dropped. \( \text{BLER}(\gamma, m) \) can be evaluated using Fig. 10 and relation (18).

Figs. 12 and 13 plot TCP throughput versus the transition probability \( p_{\text{bg}} \) for \( \text{RTL}=2 \) and 3, \( D=500 \) ms and 1000 ms, and no Internet congestion losses. The corresponding plots for an Internet congestion loss of \( p_{\text{int}} = 5\% \) are presented in Figs. 14 and 15. In all cases, DP based power adaptation can be seen to yield substantial performance enhancement over LA which selects transmission mode based on the maximum throughput for a given \( C/I_c \). For low vaues of \( p_{\text{bg}} \) representing the case when the channel tend to stay bad, TCP dynamics based optimization measures can be seen to result in several folds throughput increase. Noticeable improvement in throughput can be observed for good channel conditions as well.

We demonstrate ERP merits for another scenario with approximate IR. The performance information on IR for some
MCS schemes is tabulated in [23], [34]. However, complete set of data is not available in literature, and can be obtained only by systems simulations. To obviate the limitation, We adopt measures to approximate the IR mechanisms. The goal is to be able to ascertain the error rate for successive retransmission attempts for data blocks. Upon every block retransmission the code rate of an MCS scheme is reduced due to IR combining with previous transmissions. This code rate reduction for second and third transmission is noted in Table. I. To obtain the block error rate for retransmissions, we determine the MCS of the same modulation as the transmitted block which has its code rate R1 closest to the code rate of retransmitted data block upon IR combination. We then approximate the BLER of the retransmitted block to be the same as that of this MCS. For example $R_{1+2}$ for MCS-9 (0.5) is the closest to $R_{1}$ of MCS-6 (0.49). Since these schemes have the same modulation (8PSK), we approximate BLER for first IR combination of a retransmitted MCS-9 block as that of MCS-6. BLER for MCS-6 can be ascertained using fig. ?? and relation (18). This approach enables us to approximate IR performance for high code rate schemes. For low code rate schemes it has been observed that IR is anyway not very beneficial [23].

Figs. 16 and 17 plot TCP throughput versus the transition probability $p_{b,g}$ for approximate $P_1+P_2$ and $P_1+P_2+P_3$ (denoting combination of blocks transmitted via puncturing schemes 1, 2 and 3) combining, with D=500 ms and 1000ms, and no Internet congestion losses. Note that for the modes which offer only two puncturing schemes, the third transmission attempt is decoded for P1 puncturing. Throughput for a congestion loss of $p_{int} = 5\%$ and D=1000ms, is presented for $P_1+P_2$ recombination and for RTL=2 with no recombination, in Fig. 18. The corresponding results for $P_1+P_2+P_3$ recombination and for RTL=3 and no recombination are shown in Fig. 19. As in the no-IR case, DP based power adaptation can be seen to yield multiple folds throughput enhancement for unfavorable channel conditions, and in general $50 - 100\%$ performance enhancement over LA which selects transmission mode based on the maximum throughput for a given $C/I_c$.

C. Link Adaptation in IEEE 802.11a

The IEEE 802.11 is a widely prevalent WLAN standard with several compliant products in use. 802.11a is a high speed
IEEE 802.11a employs an OFDM PHY with 52 subcarriers, 48 of which are used to carry data. The data rates, modulation and code rate pertaining to the 8 transmission modes are shown in Table ??tab:802.11amodes). The quantity $N_{DBPS}$ represents data bytes per OFDM symbol.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Data Rate</th>
<th>Modulation</th>
<th>Code Rate</th>
<th>$N_{DBPS}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BPSK</td>
<td>6Mbps</td>
<td>1/2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>BPSK</td>
<td>9Mbps</td>
<td>3/4</td>
<td>4.5</td>
</tr>
<tr>
<td>3</td>
<td>QPSK</td>
<td>12Mbps</td>
<td>1/2</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>QPSK</td>
<td>18Mbps</td>
<td>3/4</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>16-QAM</td>
<td>24Mbps</td>
<td>1/2</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>16-QAM</td>
<td>36Mbps</td>
<td>3/4</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>64-QAM</td>
<td>48Mbps</td>
<td>2/3</td>
<td>24</td>
</tr>
<tr>
<td>8</td>
<td>64-QAM</td>
<td>54Mbps</td>
<td>3/4</td>
<td>27</td>
</tr>
</tbody>
</table>

The transmission time of an $l$ byte data payload using PHY mode $m$ is shown in [31] to be

$$T_{\text{data}}(l, m) = 20\mu s + \left[ \frac{30.75 + l}{N_{DBPS}(m)} \right] A\mu s$$ (26)

The data frames can be transmitted at the any of the supported rates. However the Ack frames can be transmitted only at one of the rates in the BSS rate set: \{6Mbps, 12Mbps, 24Mbps\}. Also an Ack frame is transmitted at the highest rate in BSS basic rate set that is less than or equal to the rate of data frame it is acknowledging. The transmission duration of an Ack frame using mode $m'$ is given by [31]

$$T_{\text{ack}}(m') = 20\mu s + \left[ \frac{16.75}{N_{DBPS}(m')} \right] A\mu s$$ (27)

Before the transmission of a frame, a sender waits for a random backoff interval. The backoff interval which is in units of Slot time, has a uniform distribution in [0,CW], where CW, the contention window has minimum and maximum permissible vaules of $aCW_{min}$ and $aCW_{max}$. In case a transmission is unsuccessful, CW is updated to [2(CW+1)-1]. When the transmission is successful CW value is reset to $aCW_{min}$. The average backoff interval before the $i$th transmission attempt of a frame, can be evaluated as

$$\bar{T}_{\text{backoff}}(i) = \min\left[2^{i-1}(aCW_{min} + 1) - 1, aCW_{max}\right] \frac{t\text{SlotTime}}{2}$$ (28)

with values $t\text{SlotTime}$, $aCW_{min}$ and $aCW_{max}$ for 802.11a PHY as 9µs, 15 and 1023 respectively.

When a frame transmission fails, the transmitting station has to wait for an EIFS interval (if the frame is lost) or an Ack timeout interval (if Ack for the frame is lost). An EIFS interval is equal to the sum of SIFS interval, DIFS interval and Ack transmission time at most robust 6Mbps. An Ack timeout is equal to a SIFS time plus an Ack transmission time plus a slot time. The SIFS and DIFS times are 16µs and 34µs respectively. $T_{\text{wait}}$, the average wait time before
transmission attempt, is then given by (28) with the frame error rate \( FER(\gamma, m) \) denoting the probability that an MPDU transmitted via mode \( m \) at SNR \( \gamma \) is in error.

With the framework outlined above, we will evaluate link adaptation measures for 802.11a through a similar approach in the previous section, and demonstrate enhancement in TCP performance. We make the same assumptions as before to evaluate the target success probability set \( S_{\text{set}}(\gamma) \) for a given SNR \( \gamma \). Assuming that a TCP segment is encapsulated in a single MPDU, the segment error rate \( SER(\gamma, m) \) equals the MPDU Error Rate with MAC ARQ retransmissions. The MPDU error rate is a function of SNR and transmission mode, and is abbreviated as \( FER_{\text{RTX}}(\gamma, m) \), the Frame Error Rate. As before \( FER_{\text{RTX}}(\gamma, m) \) for the success probability set evaluation is taken to be

\[
FER_{\text{RTX}}(\gamma, m) = 1 - FER(\gamma, m) \prod_{i=2}^{RTL-1} FER_i(\gamma, m) \tag{34}
\]

where \( FER(\gamma, m) \) represents the error probabilities of an MPDU when transmitted via mode \( m \), \( S_{\text{set}}(\gamma) \) is then taken as \( \{1 - SER(\gamma, 1), ..., 1 - SER(\gamma, 9)\} \). The elements of transmission time set \( T_{\text{set}}(\gamma) \), equal the frame transmission times \( T_{\text{frame}}(\gamma, m) \) given by (30). The expression is presented as the sum of three terms representing respectively the cases of frame transmission success on the first attempt, success on a retransmission attempt, and failure through all attempts. The time for success on the first attempt is given by (31). It incorporates the \( td\text{IFS}t\text{ime} \) wait, the backoff and payload-1 frame transmission time by the transmitting station. It further includes the \( ts\text{IFS}t\text{ime} \) wait time and ACK transmission time by the receiving station. The small air propagation delay has been neglected in these computations [31]. The frame transmission time for \( n^{th} \) transmission attempt is given by (32). It includes the time \( T_{\text{wait}}(\gamma, m) \) that a transmitting station has to wait before each transmission attempt. In the event that none of the transmission attempts is successful, the time spent for frame transmission is given by \( T_{\text{fail}}(\gamma, m) \).

The mode selection policy given by (??) is adopted for the present scenario with the target frame success probability \( s_{\text{target}}(m) \) given by \( s_{\text{opt}}(r, \gamma) \). As before \( s_{\text{opt}}(r, \gamma) \) represents the optimal success probability for TCP segments given by DP solutions as a function of round number and SNR. The success probability for the transmission modes is given by

\[
s_{\text{mode}}(m) = 1 - FER_{\text{RTX}}(\gamma, m)
\]

The target time \( t_{\text{target}}(m) \) in the same as given by (24) and the transmission time of a mode-\( m \) frame is given by \( t_{\text{mode}}(m) = T_{\text{frame}}(\gamma, m) \).

With that we complete discussing the methodology for link-adaptation for enhancing TCP throughput. For demonstrating the benefits of TCP dynamics based power adaptation, we select a TCP payload size in a segment to be 2000 bytes and use the MAC goodput plots in [31] for MSDU of 2000 bytes. We assume that an MSDU payload is transmitted via a single MPDU. The goodput for a given SNR then represents the data payload successfully sent per unit time in MAC MPDUs. From the plot for a given mode \( m \), the ratio of goodput at a given SNR and the maximum goodput, represents the success probability of an MPDU at that SNR. The maximum goodput for a mode \( m \) represents the rate \( R(m) \) that TCP is offered by that mode. These rates for 2000 bytes MPDU size are approximately \{5.7, 7, 11, 15.5, 20, 27.5, 34, 37\} Mbps.

1) Performance Appraisal: Optimization methodology similar to section IV-B.3 is adopted for obtaining power adaptation measures. Simulation are performed with a TCP segment size of 2000 bytes. When a new TCP segment is to be transmitted, the transmission mode is selected based on the mode selection policy presented in the previous subsection. If the MPDU encapsulating TCP the segment is not successful, the subsequent retransmission modes are selected by ascertaining the mode which offers the maximum goodput (or the minimum frame error rate) at a given SNR.

The throughput achieved via the policy above is compared with that of pure goodput based link adaptation whereby the modes for every MPDU transmission are selected based on maximum goodput they offer.

Figs. 20 and 21 plot TCP throughput versus the transition probability \( p_{\text{int}} \) for \( D=50 \text{ ms and 100 ms} \), and no Internet congestion losses. Fig. 22 shows the performance plot for Internet loss of \( p_{\text{int}} = 5\% \) and RTL of 3, and similar performance pattern is observed for higher RTLs.

It can be seen for all scenarios that link adaptation based on TCP dynamics yields a higher throughput than MPDU link adaptation based on maximum MAC goodput.
\[
T_{\text{wait}}(\gamma, m) = FER(\gamma, m) \left[ t_{\text{SIFS time}} + T_{\text{ack}}(m') + t_{\text{Slot time}} \right] + [1 - FER(\gamma, m)] \left[ t_{\text{SIFS time}} + T_{\text{ack}}(m') + t_{\text{SIFS time}} + T_{\text{ack}}(1) + d_{\text{DIFS time}} \right]; \tag{28}
\]

\[
T_{\text{frame}}(\gamma, m) = [1 - FER(\gamma, m)] T_{\text{success}}(\gamma, m) + FER(\gamma, m) \sum_{n=2}^{RTL} \left( FER(m) \right)^{n-1} FER(m) T_{\text{RTX}}(\gamma, m, n) + FER(\gamma, m) \left[ FER(\gamma, m) \right]^{RTL-1} T_{\text{fail}}(\gamma, m); \tag{30}
\]

\[
T_{\text{success}}(\gamma, m) = d_{\text{DIFS time}} + \bar{T}_{\text{bkooff}}(1) + T_{\text{data}}(m) + t_{\text{SIFS time}} + T_{\text{ack}}(m') \tag{31}
\]

\[
T_{\text{RTX}}(\gamma, m, n) = d_{\text{DIFS time}} + [T_{\text{wait}}(\gamma, m) + (n - 1)\bar{T}_{\text{wait}}(m)] + \sum_{i=1}^{n} T_{\text{bkooff}}(i) + n T_{\text{data}}(m) + t_{\text{SIFS time}} + \bar{T}_{\text{ack}}(m); \tag{32}
\]

\[
T_{\text{fail}}(\gamma, m) = d_{\text{DIFS time}} + [T_{\text{wait}}(\gamma, m) + (RTL - 1)\bar{T}_{\text{wait}}(m)] + \sum_{i=1}^{n} \bar{T}_{\text{bkooff}}(i) + n T_{\text{data}}(l, m) \tag{33}
\]

Fig. 22. TCP throughput for RTL=3, and $p_{\text{int}} = 5\%$

V. CONCLUSION

We address the problem of malperformance of TCP in wireless networks by introducing an optimization framework based on the congestion control dynamics of a bulk transfer TCP flow. We observe atomic patterns called cycles in the dynamics and overlay a throughput maximization strategy that enables utilization of adaptive link-layer measures to enhance TCP throughput in current and futuristic wireless technologies. We discuss the essence behind the optimization approach which is to protect selective rounds of TCP by ensuring a high segment success probability. The high success probability can be obtained via link layer measures like robust modulation and coding, greater transmission power, etc. The farther a round from filling the bandwidth-delay product of the network, greater the protection attained by it. We demonstrate the intuition behind the approach by presenting TCP throughput enhancement via adaptive power control in a simplified scenario. We then apply the framework to EGPRS and show that choosing the transmission modes according to TCP dynamics results in up to $50 - 100\%$ throughput over link adaptation which selects mode based on maximum MAC throughput. We also demonstrate similar performance improvement for IEEE 802.11a PHY and show that the TCP dynamics aware selection of transmission modes yields enhanced throughput over policies which try to maximize the MAC goodput.

ACKNOWLEDGMENT

We would like to thank Yan Li for his help with dynamic programming concepts and some simulation. We are also grateful to Gerd Zimmerman and Gerhard Kadel for their useful insights that motivated this work.

REFERENCES


