Modelling the Performance of TCP/ARQ over MIMO Rayleigh Fading Channels

Alberto Lopez Toledo
Department of Electrical Engineering
Columbia University, New York, NY.
Email: alberto,wangx@ee.columbia.edu

Xiaodong Wang

Ben Lu
Silicon Laboratories Inc.,
Austin, TX.

Abstract—In this paper we describe a general framework for the modelling of the TCP performance over MIMO wireless systems including parameters such as fading, space-time transmission schemes, multiple antenna size, modulation schemes, channel coding and ARQ. We apply the framework to analyze the performance, the optimal channel coding rate and the effect of Doppler on the TCP throughput over the BLAST MIMO system and the orthogonal space-time block coded (STBC) system. We use the network simulator ns-2 to demonstrate the accuracy of the proposed analytical framework in characterizing various parameters of the TCP performance. We apply our framework to study the behavior of a CBR video transmission over BLAST and STBC systems, by calculating the maximum throughput of the video system and the optimal coding rate under various modulation schemes. We show that the results obtained are very accurate when compared to simulations performed with the ns-2 network simulator, demonstrating the viability of the proposed modelling framework. As applications, we study the behavior of a CBR video transmission over BLAST and STBC systems, by calculating the maximum throughput of the video system and the optimal coding rate under various modulation schemes.

II. SYSTEM DESCRIPTIONS

Consider a MIMO system consisting of \( n_T \) transmit and \( n_R \) receive antennas, signalling through slow Rayleigh fading channels. The input-output relationship of this system is given by

\[
y(t) = \sqrt{\frac{\gamma}{n_T}}H(t)s(t) + n(t),
\]

where \( s_i(t) \in \mathcal{A} \) is the transmitted symbol from antenna \( i \), with \( \mathcal{A} \) being a constellation set with unit energy, i.e., \( E\{|s(t)|^2\} = 1 \), \( n(t) \) is the received ambient noise vector, \( n_i(t) \sim \mathcal{N}(0,\gamma) \); \( \gamma = \frac{P_c}{N_c} \) is the signal-to-noise ratio, and \( h_{ij}(t) \) denotes the channel gain between the \( j \)-th transmit and the \( i \)-th receive antennas at time \( t \), and it is a zero-mean complex Gaussian process with the Jakes’ correlation model [4].

For the upper layers, a typical TCP/IP/LL/RLP stack is used on the wireless link between the radio network controller (RNC) and the mobile host (MH). The RLP layer implements a type of truncated hybrid retransmission-repeat request (ARQ) type-I, that performs retransmissions, fragmentation and re-assembly. The TCP segment is correctly received when all the link layer frames are successfully received. In the case of a successful transmission a positive acknowledgement (ACK) is sent back to the transmitter over an error-free channel and the transmission of a new frame begins immediately. If, on the other hand, the frame is incorrectly received, a negative acknowledgement (NACK) is sent back to the transmitter, which will retransmit the frame in the next time block. If the maximum number of retransmissions is reached, the transmitter discards the frame silently (the upper layer, i.e., TCP, will eventually take care of the error). We assume the transmitter will always have a frame to transmit.
III. MODEL FOR MIMO WIRELESS LINKS

We consider two MIMO transmission schemes, namely the BLAST method [2], and the orthogonal space-time block coding (STBC) method [11]. We assume that an ideal channel coding is employed so that we can characterize the physical layer link loss behavior using the maximum information rate for BLAST and STBC systems. The basis of our analytical framework is as follows. By the Shannon’s channel coding theorem, a channel code rate \( R < C \), where \( C \) is the capacity of the channel. Considering the instantaneous mutual information of the channel at time \( t \) as a measure of its ‘instantaneous capacity’, we can say that if \( R < C \) the data at time \( t \) can be transmitted with an arbitrarily low probability of error. Then we will consider the channel to be in a ‘good’ state, in which errorless transmission is feasible. Conversely, if the rate of transmission at time \( t \) is greater than the ‘instantaneous capacity’, then we consider the channel to be in a ‘bad’ state, in which errorless transmission is not possible. For a MIMO system, a good channel state \( G \) corresponds to

\[
G \equiv \{ I(y(t); x(t) | H(t)) > R \},
\]

where \( I(y(t); x(t) | H(t)) \) is the instantaneous mutual information of the channel at time \( t \) given the realization of the channel \( H(t) \), and \( R \) is the information rate in bps/Hz. If (2) is not satisfied we consider the system to be in a bad state \( B \).

As we are interested in Rayleigh fading channels, we generate the instances of \( H(t) \) according to the Jakes’ model [4]. As a result of the time-correlated Doppler fading, once the MIMO system has moved to a certain state \( G \) or \( B \), it may stay over a period of time. We use Monte-Carlo to simulate the state of the system for a long enough period of time, and estimate \( p \) and \( q \) by counting the events of the system going from state \( G \) to state \( B \) and vice versa. For a BLAST system [2] with \( n_T \) transmit and \( n_R \) receive antennas, the instantaneous mutual information between the output \( y(t) \) and input \( x(t) \) for a system like (1) with MPSK or MQAM constellations and uniformly distributed symbol probabilities is computed by \( I(y; x) = H(y) - H(y | x) \) [3], where \( I \) and \( H \) denote the instantaneous mutual information and entropy function respectively. It is known that \( H(y | x) = n_R \log(\pi e) \) for a Gaussian channel. The entropy of the received signal \( H(y) \), however, has to be computed by Monte Carlo as follows [3]

\[
I(y; x | H) = H(y | H) - H(y | x, H)
= -\varepsilon_n \log_2 \left[ \frac{1}{2^{n_T M_c n_R}} \sum_{A \in T} \exp \left( -\|y - \sqrt{\frac{pR}{n_T}} Hx\|^2 \right) \right]
\]

\[
- \varepsilon_n \log_2(\pi e),
\]

with \( R \equiv n_T M_c r \), where \( r \leq 1 \) is the coding rate, \( 2^{M_c} \) is the finite-constellation size (e.g. \( M_c = 4 \) for 16-QAM), \( y \) is defined in (1); the expectation \( \varepsilon \) is taken over the random noise \( n \) also defined in (1); and the summation \( \sum_{A} \) is over all possible values of \( x \in A \), in the total amount of \( 2^{M_c n_T} \) possibilities. If \( M_c n_T \) is too large the computation in (3) can be approximated using Monte Carlo as follows

\[
I(y; x | H) = -\varepsilon_n \log_2 \left[ \frac{1}{2^{n_T M_c n_R}} \sum_{A \in T} \exp \left( -\|y - \sqrt{\frac{pR}{n_T}} Hx\|^2 \right) \right]
\]

\[
- \varepsilon_n \log_2(\pi e),
\]

with \( R \equiv n_T M_c r \), where the expectation \( \varepsilon_n \) is taken over the symbols \( x \in A \). In computing (4), we carry out a large enough number of Monte Carlo runs with two loops: the outer loop generates complex Gaussian noise \( n \) according to (1), and the inner loop uniformly generates finite-constellation symbol \( x \). The average of all Monte Carlo runs result in an estimate of \( I(y; x | H) \).

On the other hand, the mutual information of a system employing orthogonal space-time block codes (STBC) [11] of rate \( r_c \) symbols per transmission with MPSK or MQAM constellations and uniformly distributed symbol probabilities is computed by:

\[
I(y; x | H) = -r_c \varepsilon_n \log_2 \left[ \frac{1}{2^{n_T M_c r_c}} \sum_{A \in T} \exp \left( -\|y - \sqrt{\frac{pR}{n_T}} Hx\|^2 \right) \right]
\]

\[
- r_c \log_2(\pi e),
\]

with \( R \equiv n_T M_c r_c r \), where \( y_t \) is the first element of vector \( y \) and \( x_1 \) is the first element of vector \( x \). Conversely, if \( M_c n_T \) is too large, (5) can be computed as

\[
I(y; x | H) = -r_c \varepsilon_n \log_2 \left[ \frac{1}{2^{n_T M_c r_c}} \sum_{A \in T} \exp \left( -\|y - \sqrt{\frac{pR}{n_T}} Hx\|^2 \right) \right]
\]

\[
- r_c \log_2(\pi e),
\]

with \( R \equiv m M_c r_c r \).

IV. TCP/ARQ PERFORMANCE MODEL

We use a two-state Markov model (i.e. Gilbert model) to model the physical layer transmission success/failure. Specifically, let \( S(t) \) be the state of the physical layer link corresponding to the \( t \)-th frame transmission. Then \( S(t) = G \) if the \( t \)-th transmission is successful and \( S(t) = B \) if the \( t \)-th transmission is erroneous. That is, the physical layer link is modelled by a two-state Markov chain with transition matrix \( Q = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix} \), where \( p \triangleq P[S(t) = G | S(t-1) = G] \) and \( q \triangleq P[S(t) = B | S(t-1) = B] \). The values \( p \) and \( q \), calculated in the previous
section, summarize the physical layer characteristics such as space-time transmission schemes, MIMO fading channels, modulation and channel coding, etc.

**Probability of Packet Loss under ARQ:** To calculate the probability of loss of a packet, and the number of frames sent per packet we follow the method in [12] (other methods have been proposed, e.g., [1]). Given the maximum number of retransmissions $N$ on the link layer and the number of frames per packet $L$, define $p_i$ as the probability of the $i$-th link layer frame of the current packet being lost in this frame slot given the previous slot state was ‘bad’. Also define $q_i$ as the probability of the $i$-th link layer frame corresponding to the current packet being lost in this frame slot, including retransmissions, given that the previous slot state was ‘good’. Then we have $\tilde{p}_1$ and $\tilde{q}_1$ given by

$$\tilde{p}_1 = p (1 - q)^{N-1}, \quad \tilde{q}_1 = (1 - q)^N.$$  \hspace{1cm} (7)

By induction it is easy to calculate those probabilities for the $L$ frames that constitute a TCP packet as [12]

$$\tilde{p}_i = \tilde{p}_1 + (1 - \tilde{p}_1)\tilde{p}_{i-1}, \quad \tilde{q}_i = \tilde{q}_1 + (1 - \tilde{q}_1)\tilde{p}_{i-1},$$  \hspace{1cm} (8)

Denote $\pi_G$ and $\pi_B$ as the steady-state probabilities of $Q$. Then the probability of a packet loss in the current time slot is

$$P_e = \pi_G \tilde{p}_L + \pi_B \tilde{q}_L.$$  \hspace{1cm} (9)

**Average Number of Frames per Packet:** We now calculate the average number of frames transmitted per packet, taking into account retransmissions [12]. Let $n_i$ be the average number of actual frames sent for a packet consisting of $i$ link layer frames given that the first link frame found the channel in a ‘good’ state; and let $m_i^k$ be the average number of actual frames sent for a packet consisting of $i$ link layer frames, given that the first frame has already undergone $k$ ARQ retransmissions and the current channel state is (still) ‘bad’. Then, the average number of frames sent per packet is given by

$$F = \pi_G n_L + \pi_B m^0_L.$$  \hspace{1cm} (10)

In (10), $n_L$ and $m^0_L$ can be calculated recursively as follows:

the initialization step

$$n_1 = 1,$$

$$m_1^{N-1} = 1,$$

$$m_k = 1 + q \ n_1 + (1 - q) \ m_k^{k+1}, \quad k = N - 2, ..., 0,$$  \hspace{1cm} (11)

and the recursion steps for $i = 2, ..., L$

$$n_i = 1 + (1 - p) n_{i-1} + p \ m_{i-1}^0,$$

$$m_i^{N-1} = 1 + q \ n_{i-1} + (1 - q) \ m_{i-1}^0,$$

$$m_i^k = 1 + q \ n_i + (1 - q) \ m_i^{k+1}, \quad k = N - 2, ..., 0.$$  \hspace{1cm} (12)

**TCP Throughput:** Using the above results, the delay of a TCP packet will be directly proportional to the number of link layer transmissions needed. In the absence of buffer delay the round trip time for a TCP segment is [12]

$$RTT = 2 \cdot T_f + F \cdot T_w,$$  \hspace{1cm} (13)

where $T_f$ is the delay of the whole TCP segment through the internet, $T_w$ is the delay of a link layer frame through the wireless interface, and $F$ is the average number of link layer frames sent per TCP segment calculated in (10). We use the TCP Reno approximation in [7] to estimate the TCP throughput as

$$B_{TCP} \approx f(W_{max}, RTT, b, P_e, T_o),$$  \hspace{1cm} (14)

where $W_{max}$ is the maximum congestion window size, $b$ is the number of packets acknowledged by a received TCP ACK (usually 2), $T_o$ is the initial time-out for the TCP sender, and $RTT$ and $P_e$ are the round-trip time and the TCP loss probability calculated in (13) and (9) respectively.

**V. Numerical Results**

Fig. 1(a) shows the values of $p$ and $q$ as a function of the information rate $R$, for $2 \times 2$ BLAST and STBC (rate 3/4) and a normalized Doppler fading $f_d T = 0.5$ and $E_s/N_0 = 20$ dB. It is clear that the BLAST system has a larger capacity, as it is able to transmit at a higher information rate without errors. Also we can see that the STBC system is more reliable, because the parameter $p$ draws a steeper curve than the BLAST case. Interestingly we can see the effect of mobility...
in Fig. 1(b). Note that the points where the $p$ and $q$ values cross for $f_dT = 0.01$ is much lower in probability (around 2%, compared to the 50% in the $f_dT = 0.5$ case), i.e., for the same SNR the fading are less frequent but larger for $f_dT = 0.01$. For TCP with ARQ, this implies that the effect of the mobility is positive: at $f_dT = 0.01$, although the capacity of the channel does not change compared to $f_dT = 0.5$, and the probability of going to the ‘bad’ state $p$ is lower, the probability of getting back to the ‘good’ state $q$ is only 1%. This implies that when the probability of error blocks starts to increase, the average length of the burst errors is much larger for $f_dT = 0.01$ than for $f_dT = 0.5$. This, as we will see, makes the fading correlation (i.e., high mobility) very attractive for FEC and ARQ schemes that will require less retransmissions to approach the capacity curve. Fig. 1(c) shows the $p$ and $q$ values for the three systems when $f_dT = 0.5$ and $SNR = 2dB$. As expected the rates at which the transmission is possible without errors is much lower.

**Comparison with ns-2 simulations:** Fig. 2 compares the results obtained for the TCP throughput calculated in (18) and the simulations for a $4 \times 4$ BLAST system with $\frac{E_s}{N_0} = 30$ dB. For the simulations we modified RLC module in the GPRS implementation by Richa Jain at IITB (India) to implement a link layer retransmission mechanism for the ns-2 simulator [10]. For the physical layer we generated a sequence of states according to the model in (2). We consider $T$ to be large enough to accommodate a link layer frame, and the transmissions to be synchronized at the beginning of the frame time. For every frame to be transmitted the link controller checks the state of the channel and discards the frame if the state is ‘bad’, and transmit the frame if the state is ‘good’. Fig. 2 shows that our framework is indeed very accurate, as the simulations match almost perfectly the analytical results.

**Optimal information rate that maximizes the TCP throughput:** Fig. 3 shows the information rate that maximizes TCP throughput in a $4 \times 4$ BLAST and STBC (rate 3/4) for $f_dT = 0.01$. The dashed lines denote the ergodic capacities of BLAST and STBC, and the other lines show the rate at which TCP operates at maximum throughput for different maximum number of retransmissions. It is seen that the optimal rate for

**Effect of Doppler on optimal TCP throughput:** Fig. 4 shows the maximum TCP throughput for BLAST and STBC for both $f_dT = 0.5$ and $f_dT = 0.01$ and 10 retransmissions. As expected, the values for the TCP throughput are substantially better in the case of high mobility ($f_dT = 0.5$), and only when the SNR reaches 6 dB the TCP throughput for $f_dT = 0.01$ begins to grow significatively. Our framework effectively captures the effect that mobility has on TCP: while the capacity of a MIMO system does not change with the Doppler, the effects of the time correlation and the deep fadings on TCP are significant.

**Optimal channel coding rate with finite constellations:** Fig. 5 shows the optimal coding rates that maximizes the TCP
throughput for QPSK and 16QAM with a maximum ARQ persistence of 10 retransmissions. Note that QPSK needs less coding than the 16QAM, especially for low SNR. Also it is interesting to note that, for optimal operation, the transmission under $f_dT = 0.5$ needs less coding protection.

VI. APPLICATIONS OF THE FRAMEWORK

Analysis of TCP Buffer Occupancy: Assume each TCP packet is formed by $L$ link layer frames that arrive in bursts with a Bernoulli distribution. Also assume there is a queue at the base station for link layer frames that can accommodate up to $K$ TCP packets, i.e., $LK$ link layer frames. At the beginning of time slot $i$, let $Q_i$ denote the number of frames in the buffer and $S_i$ the state of the channel with $S_i = 1$ if the channel is ‘good’ and $S_i = 0$ if the channel is ‘bad’. Then \{$(Q_i, S_i), i \geq 0$\} is a two-dimension discrete-time Markov process. This model has been extensively used in the literature [5], [6]. A TCP packet is lost due to a packet overflow if any of the constituting link layer frames cannot be accommodated in the buffer (i.e., buffer is full). Let $P_{enter}$ be the probability of a TCP packet entering the queue, then the probability of a TCP packet arrival is [5]

$$P_e = B \cdot T \cdot P_{enter},$$

where $B$ is the TCP throughput, and $T$ is the time slot corresponding to the duration time of ‘good’ and ‘bad’ states of the Gilbert model. If the channel is bad there can be no departures and if the channel is good, there will be one departure with probability one if there is at least one frame in the buffer.

Let $\pi$ be the stationary distribution of the queue. Then the average probability of a TCP packet drop in the queue is given by $P_{drop} = \sum_{Q_i}^{LK} (1 - (\pi_{0,0} + \pi_{1,0}))$ i.e., the probability of finding the buffer with more than $L$ slots free, in which case one of the frames would be dropped and the TCP packet will be lost. The average buffer occupancy is given by $E\{Q\} = \sum_{Q_i=0}^{LK} Q_i \cdot (\pi_{0,0} + \pi_{1,0})$ and the average length of the queue for a TCP packet is $E\{Q_{TCP}\} = \sum_{Q_i=0}^{LK} \left[ \frac{Q_i}{T} \right] (\pi_{0,0} + \pi_{1,0})$ [5]. Applying Little’s law we can obtain the average waiting time in the buffer

$$T_{buffer} = \frac{E\{Q_{TCP}\}}{B(1 - P_{drop})}.$$  

(20)

The expressions for the TCP packet loss probability and RTT are as follows

$$P_e = P_e + P_{drop} - P_e \cdot P_{drop}$$  

(21)

$$RTT_q = 2 \cdot T_f + F \cdot T_w + T_{buffer},$$  

(22)

where $P_e$ is the probability of TCP loss including the maximum number of retransmissions (9), $P_{drop}$ is the probability of a TCP packet finding the buffer full, $F$ is given by (10) and $T_{buffer}$ is the delay at the buffer. The final expression for the TCP throughput is obtained by substituting the values of (21) and (22) in (18).

Figs. 6 show the behavior of a 20 packets buffer for $4 \times 4$ BLAST and STBC (rate 3/4) systems, at $E_b/N_0 = 10$ dB and normalized Doppler fading of $f_dT = 0.01$ and $f_dT = 0.5$ respectively. The maximum number of ARQ retransmissions in all cases is 10. The highly correlated fading ($f_dT = 0.01$) have a strong impact on TCP, as it requires more retransmissions, and the queues start to fill quickly. At $f_dT = 0.5$ the queues start to fill at a higher information rate and more abruptly, usually when the optimal capacity of the channel for TCP is reached. In general, BLAST has a higher capacity as the buffer saturates at a higher information rate. Also we can see that STBC is more tolerant to the Doppler, as even for $f_dT = 0.01$ the buffer starts to fill in closer to capacity relative to BLAST.

Analysis of CBR Video Transmission: We consider CBR video source with a fixed maximum delay $D$ and frame error rate $P_v$, where the video frames are split into $L$ link layer frames. We also consider a buffered ARQ system on the wireless side similar to the one presented in Section IV. Then, the maximum number of retransmissions allowed by the system $N_{video}$ is calculated as

$$T = T_f + F_{video} \cdot (T_w + T_{buffer}) \leq D,$$  

(23)

where $F_{video}$ is the average number of transmissions per frame under the delay constraint, $T_f$ is the TCP packet delay in the fixed link, $T_w$ is the delay of the frame in the wireless link and $T_{buffer}$ is the delay at the buffer.
greater than the channel capacity for applications that tolerate errors and delay. Also, the mobility does not significantly affect the performance because the system is allowed a large delay that absorbs the fading with retransmissions, except for low SNRs, in which the $f_d T = 0.5$ performs better. The video conference application tolerates a 1% error rate which moves the optimal rate closer to capacity. Finally, the optimal rate for the real-time application is below capacity in order to maintain the zero error rate. In this case, the low delay requirement prevents ARQ from recovering most of the long burst errors, which makes the $f_d T = 0.5$ to have better performance.

VII. CONCLUSIONS

We have presented a simple modelling framework for TCP over coded multi-antenna wireless systems. We showed that the type of application plays a crucial role in the optimization of a wireless system (for example, increasing the channel bandwidth efficiency is not always a good strategy). By direct framework application, we showed that the more uncorrelated the channel (higher doppler) the more the TCP/ARQ system can benefit from larger buffers without performance penalty. Finally, we showed that the application requirements affect the optimal rate of transmissions (for error and delay-tolerant video applications, increasing the transmission rate beyond capacity is a good strategy). Parameters such as the delay tolerance or the TCP AIMD feedback scheme drive the system performance, so systems should take these into account and accommodate the information rate at which the physical layer is transmitting to the real system demands.

REFERENCES