TCP Performance over Wireless MIMO Channels with ARQ and Packet Combining

Alberto Lopez Toledo, *Student Member*, *IEEE*, and Xiaodong Wang, *Member*, *IEEE*

Abstract—Multiple-input multiple-output (MIMO) wireless communication systems that employ multiple transmit and receive antennas can provide very high-rate data transmissions without increase in bandwidth or transmit power. For this reason, MIMO technologies are considered as a key ingredient in the next generation wireless systems, where provision of reliable data services for TCP/IP applications such as wireless multimedia or Internet is of extreme importance. However, while the performance of TCP has been extensively studied over different wireless links, little attention has been paid to the impact of MIMO systems on TCP. This paper provides an investigation on the performance of modern TCP systems when used over wireless channels that employ MIMO technologies. In particular, we focus on two representative categories of MIMO systems, namely, the BLAST systems and the space-time block coding (STBC) systems, and how the ARQ and packet combining techniques impact on the overall TCP performance. We show that, from the TCP throughput standpoint, a more reliable channel may be preferred over a higher spectral efficient but less reliable channel, especially under low SNR conditions. We also study the effect of antenna correlation on the TCP throughput under various conditions.

Index Terms—TCP/IP, MIMO, BLAST, space-time block coding, ARQ, packet combining, antenna correlation.

1 Introduction

The use of multiple transmit and receive antennas in wireless communication systems together with the recently developed space-time coding and signal processing techniques has been shown to provide dramatic capacity increase over the traditional single-input single-output (SISO) channels, especially over rich scattered environments [13], [14], [26]. This potential gain in link throughput and network capacity makes such multiple-input multiple-output (MIMO) systems the ideal candidate as the core technology for the next generation broadband wireless communication systems. It is anticipated that these systems will play a major role in the development of the Internet by seamlessly integrating voice and data services. As the majority of Internet services, such as FTP, Web, or e-mail, are provided by TCP, it is essential for present and future wireless access to provide better support to TCP services in terms of reliability, throughput, and delay. The main objective of our work is to analyze the effect that these MIMO schemes have in TCP and if the increase of spectral efficiency that they provide is always beneficial in terms of TCP throughput for different SNR scenarios.

When TCP is used over wireless networks with typically high frame error rate (FER), the performance of TCP is severely affected [5], [8]. A common approach to improve its performance is the use of local automatic repeat-request (ARQ) mechanisms that prevent the TCP source from misinterpreting packet losses due to fading as network congestion by performing partial link layer recovery through a limited number of retransmissions. While such ARQ mechanisms effectively mitigate the impact of losses on TCP, they also introduce additional complexity into the system. The primary effect of such complexity is in the form of delay and rate variation due to the retransmissions [8]. This variation has a negative impact on TCP in the form of burstiness that may cause further losses and throughput reduction. Therefore, although ARQ mechanisms may improve TCP performance by reducing the observed FER, a solution in which the channel does not appear as highly variable is preferred from the TCP standpoint.

MIMO systems may serve this purpose, as they offer a flexible way of using the antenna diversity to trade off throughput for stability. The Bell-Labs layered space-time (BLAST) system [13], [14], [25] transmits different symbols from all transmitting antennas simultaneously and is aimed at high data-rate transmissions. On the other hand, the space-time coding (STC) systems [1], [32] exploit the transmission diversity by sending the same symbols from different transmit antennas, thus increasing the reliability at the expense of throughput. Depending on the quality of the channel, from the TCP perspective, it may be preferable to reduce the channel throughput while improving its FER than to activate a retransmission mechanism on a channel with high throughput but also high error rate. In this paper, we use the above reasoning to investigate the impact of the use of those two MIMO schemes on TCP systems to evaluate which of the schemes is preferable for TCP under different SNR conditions and show how the MIMO schemes interact when ARQ systems and combining is in place. We will conclude that, for low SNR scenarios, it is preferable to

- *The authors are with the Department of Electrical Engineering, Columbia University, 500 West 120th Street, New York, NY 10027.*
E-mail: {alberto, wangx}@ee.columbia.edu.

Manuscript received 19 Sept. 2003; revised 20 Mar. 2004; accepted 25 Sept. 2004; published online 16 Jan. 2006.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0152-0903.

1536-1233/06/$20.00 © 2006 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

use a more reliable system, such as STBC, to one with better spectral efficiency like BLAST, because TCP, even when ARQ and combining is used, cannot make use of the additional bit rate.

While the performance of TCP has been extensively studied over different wireless links [5], [10], [11], little research has been made on the behavior of TCP over MIMO systems. Stojanovic et al. [31] present a performance evaluation of TCP over a MIMO-based 2G broadband wireless access network using ARQ and adaptive modulation. Milani et al. [21] investigate the use of antenna selection on V-BLAST in order to increase TCP throughput. In [29], TCP throughput is evaluated over a STBC-based 802.11 system. However, no existing work compares the performance of TCP for MIMO schemes with different spectral efficiency in combination with ARQ and packet combining. Our work investigates the tradeoffs of spectral efficiency and retransmissions from a TCP cross-layer standpoint and further analyzes the effect of ARQ persistence and antenna correlation.

The remainder of this paper is organized as follows: In Section 2, we describe the two types of MIMO systems considered in this paper, namely, the BLAST system and the orthogonal space-time block coding (STBC) systems. In Section 3, the problem of TCP over wireless channels is analyzed, together with existing approaches to mitigate it. In Section 4, we discuss the local retransmission mechanisms with packet combining over MIMO channels. In particular, we analyze a pure NACK selective repeat hybrid ARQ type I [18] and weighted gain packet combining or Chase combining [9]. In Section 5, we describe the simulation setup. In Section 6, we present the simulation results and our analysis, including the effects of antenna correlation on TCP under various conditions. Section 7 concludes the paper by identifying the key variables that affect performance and, hence, constitute the basis for a cross-layer design.

2 MIMO Systems

This section discusses two representative MIMO techniques —BLAST and STBC. These two schemes cover a wide range of applications suitable for different future wireless systems. On one hand, the BLAST architecture (also known as spatial multiplexing) is targeted at the high-rate wireless systems, such as wireless LAN and wireless MAN, because of its high throughput. On the other hand, the STBC techniques achieve highly reliable data transmission in severe mobile wireless systems with scattering, shadowing, reflection, and diffraction, which makes them especially useful for 3G cellular with high mobility.

2.1 The BLAST System

In the BLAST architecture, a single data stream is split into n_T substreams, called layers, that are encoded separately and transmitted simultaneously from n_T transmit antennas. The signals received by the n_R receive antennas are processed to separate the streams and recover the original data.

The input-output signal relationship in a BLAST system is expressed as

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n_R} \end{bmatrix}}_{\boldsymbol{y}} = \sqrt{\frac{\rho}{n_T}} \underbrace{\begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,n_T} \\ h_{2,1} & h_{2,2} & \dots & h_{2,n_T} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n_R,1} & h_{n_R,2} & \dots & h_{n_R,n_T} \end{bmatrix}}_{\boldsymbol{H}} \underbrace{\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{n_T} \end{bmatrix}}_{\boldsymbol{s}} + \underbrace{\begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{n_R} \end{bmatrix}}_{\boldsymbol{n}}, \tag{1}$$

where $\boldsymbol{y} = [y_1, y_2, \dots, y_{n_R}]^T$ is the $(n_R \times 1)$ received symbol vector, $\boldsymbol{s} = [s_1, s_2, \dots, s_{n_T}]^T$ is the $(n_T \times 1)$ transmitted signal vector with $s_i \in \mathcal{A}$, where $\mathcal{A}$ is a finite constellation signal set with unit energy $(E\{|s_i|^2\} = 1)$, and $\boldsymbol{n}$ is the $(n_R \times 1)$ received noise vector with $n_i \sim \mathcal{N}_c(0, 1)$. The signal-to-noise ratio ρ is independent of the number of transmit antennas. The channel is represented by a $(n_R \times n_T)$ matrix $\boldsymbol{H}$, where h_{ij} represents the complex gain of the channel between the jth transmit antenna and the ith receive antenna. For the rest of the discussion, we will assume that the MIMO channel matrix $\boldsymbol{H}$ it is known at the receiver but not at the transmitter.[1]

The optimal BLAST detector is the maximum likelihood detector (ML) given by

$$\hat{\boldsymbol{s}}_{ML} = \arg \min_{\boldsymbol{s} \in \mathcal{A}^{n_T}} \left\| \boldsymbol{y} - \sqrt{\frac{\rho}{n_T}} \boldsymbol{H}\boldsymbol{s} \right\|^2, \tag{2}$$

which has a computational complexity $\mathcal{O}(|A|^{n_T})$ that grows exponentially with the number of transmit antennas n_T.

A lower complexity receiver is the MMSE detector with ordered interference cancellation. In this scheme, a symbol with the highest SNR is detected using a linear MMSE filter and then subtracted from the received signals. Such a procedure is repeated until all the transmitted symbols are detected as follows [6]:

```
$\bar{\boldsymbol{H}} = \boldsymbol{H}$
$\boldsymbol{r} = \boldsymbol{y}$
for $i = 1 : n_T$ do
    $\boldsymbol{\Omega} = \left(\frac{\rho}{n_T}\bar{\boldsymbol{H}}^H\bar{\boldsymbol{H}} + \boldsymbol{I}\right)^{-1}$    (MMSE criterion)
    $k_i = \arg\min\{\boldsymbol{\Omega}_{j,j}\}$    ($k_i$ is the current min $SNR$ symbol index)
    $\boldsymbol{w} = (\bar{\boldsymbol{H}}\boldsymbol{\Omega})(:, k_i)$    ($\boldsymbol{w}$ is the nulling vector)
    $z_{k_i} = \boldsymbol{w}^H\boldsymbol{r}$    (nulling operation)
    $\hat{s}_k = \mathcal{Q}_{\mathcal{A}}(z_{k_i})$
    $\boldsymbol{r} = \boldsymbol{r} - \sqrt{\frac{\rho}{n_R}}\boldsymbol{H}(:, k_i)\hat{s}_k$    (cancellation operation)
    $\bar{\boldsymbol{H}} =$ remove column $k_i$ from $\bar{\boldsymbol{H}}$
end for
```

2.2 Space-Time Block Coding

In the space-time coded (STC) MIMO systems, instead of transmitting different symbols, the same symbols are transmitted through different antennas to increase diversity. In STC, a group of M-PSK or M-QAM symbols are mapped after modulation into a space-time coding matrix,

[1] In practice, the receiver will need to estimate the channel matrix before detection [19].

allowing both the temporal diversity and spatial diversity to be exploited. A space-time block code is represented by

$$\mathcal{C}_{m,n_T} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n_T} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n_T} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,n_T} \end{bmatrix}, \tag{3}$$

where the rows represent the n_T transmit antennas and the columns represent the number of time slots that the block takes to be transmitted (here, the block size is m slots). In this paper, we focus on the 4×4 antenna configuration, i.e., $n_T = n_R = 4$. In what follows, we consider three STBC systems with rate 1/2, 1, and 2, respectively. For four transmitter antennas, the half-rate orthogonal code employs a $\boldsymbol{C}_{8,4}$ transmission matrix, transmitting four symbols in eight transmissions. The received signal at antenna i over the eight transmissions is given by

$$\begin{bmatrix} y_{i,1} \\ y_{i,2} \\ y_{i,3} \\ y_{i,4} \\ y_{i,5} \\ y_{i,6} \\ y_{i,7} \\ y_{i,8} \end{bmatrix} = \sqrt{\frac{\rho}{4}} \underbrace{\begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ -s_2 & s_1 & -s_4 & s_3 \\ -s_3 & s_4 & s_1 & -s_2 \\ -s_4 & -s_3 & s_2 & s_1 \\ s_1^* & s_2^* & s_3^* & s_4^* \\ -s_2^* & s_1^* & -s_4^* & s_3^* \\ -s_3^* & s_4^* & s_1^* & -s_2^* \\ -s_4^* & -s_3^* & s_2^* & s_1^* \end{bmatrix}}_{\boldsymbol{C}_{8,4}} \begin{bmatrix} h_{i,1} \\ h_{i,2} \\ h_{i,3} \\ h_{i,4} \end{bmatrix} + \begin{bmatrix} n_{i,1} \\ n_{i,2} \\ n_{i,3} \\ n_{i,4} \\ n_{i,5} \\ n_{i,6} \\ n_{i,7} \\ n_{i,8} \end{bmatrix},$$
$$i = 1, 2, \ldots, n_R. \tag{4}$$

Note that (4) can be rewritten as

$$\underbrace{\begin{bmatrix} y_{i,1} \\ y_{i,2} \\ y_{i,3} \\ y_{i,4} \\ y_{i,5}^* \\ y_{i,6}^* \\ y_{i,7}^* \\ y_{i,8}^* \end{bmatrix}}_{\boldsymbol{y}_i} = \sqrt{\frac{\rho}{4}} \underbrace{\begin{bmatrix} h_{i,1} & h_{i,2} & h_{i,3} & h_{i,4} \\ -h_{i,2} & h_{i,1} & -h_{i,4} & h_{i,3} \\ -h_{i,3} & h_{i,4} & h_{i,1} & -h_{i,2} \\ -h_{i,4} & -h_{i,4} & h_{i,2} & h_{i,1} \\ -h_{i,1}^* & -h_{i,2}^* & -h_{i,3}^* & -h_{i,4}^* \\ -h_{i,2}^* & h_{i,1}^* & -h_{i,4}^* & h_{i,3}^* \\ h_{i,3}^* & -h_{i,4}^* & -h_{i,1}^* & h_{i,2}^* \\ h_{i,4}^* & h_{i,3}^* & -h_{i,2}^* & -h_{i,1}^* \end{bmatrix}}_{\bar{\boldsymbol{H}}_i} \underbrace{\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}}_{\boldsymbol{s}_i} + \underbrace{\begin{bmatrix} n_{i,1} \\ n_{i,2} \\ n_{i,3} \\ n_{i,4} \\ n_{i,5}^* \\ n_{i,6}^* \\ n_{i,7}^* \\ n_{i,8}^* \end{bmatrix}}_{\boldsymbol{n}_i},$$
$$i = 1, 2, \ldots, n_R. \tag{5}$$

The matrix $\bar{\boldsymbol{H}}_i$ is orthogonal, i.e., $\bar{\boldsymbol{H}}_i^H \bar{\boldsymbol{H}}_i = \sum_{k=1}^4 |h_{i,k}|^2 \boldsymbol{I}_4$. Hence, at the receiver, the symbols are detected by a simple linear detector $\hat{\boldsymbol{s}} = \mathcal{Q}_A(\boldsymbol{z})$, where

$$\boldsymbol{z} = \sum_{i=1}^{n_R} \bar{\boldsymbol{H}}_i^H \boldsymbol{y}_i. \tag{6}$$

A rate-1 orthogonal code does not exist for four transmit antennas [32]. However a rate-1 quasi-orthogonal scheme [22] is given by the $\boldsymbol{C}_{4,4}$ transmission matrix in (7). The received signal at the ith receive antenna for the four transmissions is

$$\begin{bmatrix} y_{i,1} \\ y_{i,2} \\ y_{i,3} \\ y_{i,4} \end{bmatrix} = \sqrt{\frac{\rho}{4}} \underbrace{\begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ s_2^* & -s_1^* & s_4^* & -s_3^* \\ s_3 & -s_4 & -s_1 & s_2 \\ -s_4 & -s_3 & s_2 & s_1 \end{bmatrix}}_{\boldsymbol{C}_{4,4}} \begin{bmatrix} h_{i,1} \\ h_{i,2} \\ h_{i,3} \\ h_{i,4} \end{bmatrix} + \begin{bmatrix} n_{i,1} \\ n_{i,2} \\ n_{i,3} \\ n_{i,4} \end{bmatrix},$$
$$i = 1, 2, \ldots, n_R, \tag{7}$$

that can be rewritten as

$$\underbrace{\begin{bmatrix} y_{i,1} \\ y_{i,2}^* \\ y_{i,3} \\ y_{i,4}^* \end{bmatrix}}_{\boldsymbol{y}_i} = \sqrt{\frac{\rho}{4}} \underbrace{\begin{bmatrix} h_{i,1} & h_{i,2} & h_{i,3} & h_{i,4} \\ -h_{i,2}^* & h_{i,1}^* & -h_{i,4}^* & h_{i,3}^* \\ h_{i,3} & -h_{i,4} & -h_{i,1} & h_{i,2} \\ h_{i,4}^* & h_{i,3}^* & -h_{i,2}^* & -h_{i,1}^* \end{bmatrix}}_{\bar{\boldsymbol{H}}_i} \underbrace{\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}}_{\boldsymbol{s}_i} + \underbrace{\begin{bmatrix} n_{i,1} \\ n_{i,2} \\ n_{i,3} \\ n_{i,4} \end{bmatrix}}_{\boldsymbol{n}_i},$$
$$i = 1, 2, \ldots, n_R. \tag{8}$$

The decision statistic at the receiver antenna is given by [17]

$$\boldsymbol{z}_i = \bar{\boldsymbol{H}}_i^H \boldsymbol{y}_i = \sqrt{\frac{\rho}{4}} \bar{\boldsymbol{H}}_i^H \bar{\boldsymbol{H}}_i \boldsymbol{s} + \bar{\boldsymbol{H}}_i^H \boldsymbol{n}_i = \sqrt{\frac{\rho}{4}} \boldsymbol{\Omega}_i \boldsymbol{s} + \boldsymbol{w}_i, \quad i = 1, 2, \ldots, n_R, \tag{9}$$

with

$$\bar{\boldsymbol{\Omega}}_i = \begin{bmatrix} \gamma_i & 0 & \alpha_i & 0 \\ 0 & \gamma_i & 0 & -\alpha_i \\ -\alpha_i & 0 & \gamma_i & 0 \\ 0 & \alpha_i & 0 & \gamma_i \end{bmatrix}, \tag{10}$$

$$\gamma_i = \sum_{j=1}^4 |h_{i,j}|^2, \qquad \alpha_i = 2j\Im(h_{i,1}^* h_{i,3} + h_{i,4}^* h_{i,2}), \tag{11}$$

and $\boldsymbol{w}_i \sim \mathcal{N}_c(0, \bar{\boldsymbol{\Omega}})$. We can group the statistics in (9) to form two 2 × 2 BLAST systems defined by

$$\begin{bmatrix} z_{i,1} \\ z_{i,3} \end{bmatrix} = \sqrt{\frac{\rho}{2}} \begin{bmatrix} \gamma_i & \alpha_i \\ -\alpha_i & \gamma_i \end{bmatrix} \begin{bmatrix} s_1 \\ s_3 \end{bmatrix} + \begin{bmatrix} w_{i,1} \\ w_{i,3} \end{bmatrix}, \tag{12}$$

and

$$\begin{bmatrix} z_{i,4} \\ z_{i,2} \end{bmatrix} = \sqrt{\frac{\rho}{2}} \begin{bmatrix} \gamma_i & \alpha_i \\ -\alpha_i & \gamma_i \end{bmatrix} \begin{bmatrix} s_4 \\ s_2 \end{bmatrix} + \begin{bmatrix} w_{i,4} \\ w_{i,2} \end{bmatrix}, \qquad i = 1, 2, \ldots, n_R, \tag{13}$$

The systems (12) and (13) can be decoded using the ML detector in (2) or the MMSE detector with ordered interference cancellation described in Section 2.1.

Finally, we consider a rate-2 system by combining STBC and BLAST [35]. For a symbol set $\boldsymbol{s} = [s_1, s_2, s_3, s_4]^T$, two antennas can be used to transmit $\bar{\boldsymbol{s}}_1 = [s_1, s_2]^T$ and the other two antennas to transmit $\bar{\boldsymbol{s}}_2 = [s_3, s_4]^T$, both using the rate-1 Alamouti code [1] as follows:

$$\begin{aligned} \begin{bmatrix} y_{i,1} \\ y_{i,2} \end{bmatrix} &= \sqrt{\frac{\rho}{2}} \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix} \begin{bmatrix} h_{i,1} \\ h_{i,2} \end{bmatrix} \\ &\quad + \sqrt{\frac{\rho}{2}} \begin{bmatrix} s_3 & s_4 \\ -s_4^* & s_3^* \end{bmatrix} \begin{bmatrix} h_{i,3} \\ h_{i,4} \end{bmatrix} + \begin{bmatrix} w_{i,1} \\ w_{i,2} \end{bmatrix}, \\ &\quad i = 1, 2, \ldots, n_R, \end{aligned} \tag{14}$$

The received signal at the ith receive antenna after the two separate transmissions is given by [17]

$$\begin{aligned} \underbrace{\begin{bmatrix} y_{i,1} \\ y_{i,2}^* \end{bmatrix}}_{\boldsymbol{y}_i} &= \sqrt{\frac{\rho}{2}} \underbrace{\begin{bmatrix} h_{i,1} & h_{i,2} \\ h_{i,2}^* & -h_{i,1}^* \end{bmatrix}}_{\bar{\boldsymbol{H}}_{i,1}} \underbrace{\begin{bmatrix} s_1 \\ s_2 \end{bmatrix}}_{\bar{\boldsymbol{s}}_1} \\ &\quad + \sqrt{\frac{\rho}{2}} \underbrace{\begin{bmatrix} h_{i,3} & h_{i,4} \\ h_{i,4}^* & -h_{i,3}^* \end{bmatrix}}_{\bar{\boldsymbol{H}}_{i,2}} \underbrace{\begin{bmatrix} s_3 \\ s_4 \end{bmatrix}}_{\bar{\boldsymbol{s}}_2} + \begin{bmatrix} w_{i,1} \\ w_{i,2}^* \end{bmatrix}, \\ i &= 1, 2, \ldots, n_R. \end{aligned} \tag{15}$$

This can be expressed as a BLAST system for n_R receive antennas of the form

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n_R} \end{bmatrix}}_{\boldsymbol{y}} = \sqrt{\frac{\rho}{2}} \underbrace{\begin{bmatrix} \bar{\boldsymbol{H}}_{1,1} & \bar{\boldsymbol{H}}_{1,2} \\ \bar{\boldsymbol{H}}_{2,1} & \bar{\boldsymbol{H}}_{2,2} \\ \vdots & \vdots \\ \bar{\boldsymbol{H}}_{n_R,1} & \bar{\boldsymbol{H}}_{n_R,2} \end{bmatrix}}_{\bar{\boldsymbol{H}}} \underbrace{\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}}_{\boldsymbol{s}} + \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n_R} \end{bmatrix}}_{\boldsymbol{w}}. \tag{16}$$

Again, the received signal in (16) can be detected using the ML detector in (2) or the MMSE detector with ordered cancellation described in Section 2.1.

3 TCP Behavior on Wireless Links

The design of current Internet protocols did not account for wireless architectures. The available links, although far less reliable than current wired technologies, were supposed to behave reasonably well and have low bit error rates. With this in mind, TCP was carefully designed to fairly manage congestion situations where resources are scarce, but it was not designed to take the variable characteristics of the wireless links into account. TCP error control behaves faulty in a fading situation because it always identifies a loss as a result of a congestion situation and not due to bursty errors in the link.[2] The same problem appears in mobile users when the mobile node performs a handoff, in which it is disconnected temporarily from the network and, hence, suffering losses.

TCP detects a loss by the use of timeouts and on the reception of duplicated acknowledgements (dupacks). When a loss occurs, TCP reacts by reducing the sending rate to adapt to the available bandwidth left from the competing flows in the congested node (i.e., buffer overflow in the weakest link). This reaction is based on the assumption that a buffer overflow implies that the buffers have grown (and are full) in the congested node(s). TCP uses a sliding window scheme for rate control. The size of the window indicates the amount of packets that can be sent without the need of an acknowledgement. The control of the window size follows the so-called additive-increase multiplicative-decrease (AIMD) model: When no losses occur, the window size is slowly incremented until the maximum capacity is reached. When a severe loss occurs, the window size is then drastically reduced to rapidly ease the congestion on the path. The size of the window is advertised to the peer TCP layer to achieve the end-to-end rate control.

In the following two sections, we emphasize the aspects of current TCP versions in use, as it is our assumption that a cross-layer design should account for the fact that there is a great TCP base and that it will take some time to replace. In the last section, we give a brief overview of more current state-of-the art proposals. For an extensive discussion of these issues and the current state of TCP research, see [34], [15], [36].

3.1 TCP Congestion Window Control

TCP has different phases for congestion control that determine the behavior of the congestion window on the sender side [30]. In the *slow start* phase, TCP probes the available bandwidth on the link, observing the rate at which the other end of the communication acknowledges the packets. TCP transmits all the segments on the congestion window and waits for the acknowledgements. For each ACK received, the windows size is incremented by one until the congestion avoidance threshold (ssthresh) is reached, at which moment the congestion avoidance phase is activated. Note the exponential behavior, as the congestion windows size (cwnd) is doubled for each round. The *congestion avoidance* phase is activated when congestion is encountered. In this phase, cwnd is increased linearly at a rate of one segment per window size. With *fast retransmit*, a dupack is caused by an out-of-order segment received on the other end. Normally, this is interpreted as a loss. However, this behavior does not take into account the fact that packets on the Internet do not travel at the same speed. To avoid it, fast retransmit allows a certain number of dupack to be received before activating the slow start phase. When a number of dupacks is received, TCP assumes that the particular segment has been lost. It halves both ssthresh and cwnd and immediately retransmit the segment without waiting for a timeout. Finally, with *fast recovery*, if a loss is sensed during fast retransmit due to dupacks, the sender knows that at least one segment (out-of-order) was just received at the other end. Because this is an indication of moderate congestion, fast recovery tries to avoid the slow start from being triggered. For every subsequent dupack, the congestion window is incremented by one (as another segment has been buffered at the receiver) and, upon the reception of a new ACK, cwnd is set to the ssthresh stored when fast retransmit started, which is equivalent to getting back to the congestion avoidance phase.

When fast retransmit/recovery are used (usually together), the slow start phase is only triggered when a timeout occurs. On the other side, a problem of the fast retransmission/recovery is that, in the event of multiple losses in the same window, a new ACK received may acknowledge only part of the missing segments, in which case TCP should not leave fast recovery. A way of solving this problem is to use a form of selective acknowledgement in TCP [20], giving more information to the sender about the missing packets. With the extra information, the sender can retransmit more than one missing packet per round and

[2] From [30]: "The assumption of the algorithm is that packet loss caused by damage is very small (much less than 1 percent), therefore the loss of a packet signals congestion somewhere in the network between the source and destination."

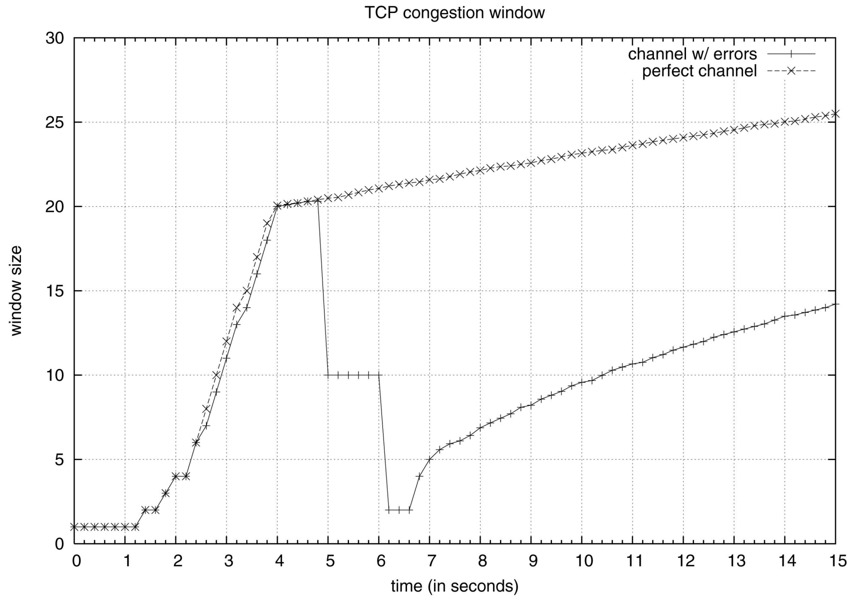


Fig. 1. TCP congestion window behavior under different bit error rate links.

it might, for example, give more preference to missing segments than to a new data [12].

3.2 TCP on Wireless Links

Fig. 1 shows a typical behavior of TCP in a wireless link compared to a perfect link.[3] At the beginning of the communication ($t = 1 - 4$ sec.), TCP is in the slow-start mode. TCP probes the network and increments the window size for every successfully received segment. At $t = 4$ sec., TCP enters the congestion avoidance phase. This is not caused by a loss, but because the congestion avoidance threshold (20 is the default value in the ns-2 simulator) has been reached. This congestion avoidance phase shows a linear growth of the window until $t = 4.8$ sec., when the dupacks are received. At this point, the window size and ssthresh are halved to 10 segments. The channel is in a typical deep fade and packet loss is severe. Because no further dupacks are received, the window does not grow or shrink and, finally, at $t = 6$ sec., TCP times out. Slow start is invoked and the the whole process begins with the exponential slow start, but, at this time, the congestion avoidance threshold is five segments. The congestion avoidance begins at $t = 7$ sec., this time without further losses.

When a TCP with such characteristics is used in a wireless environment, deep fading causes several packet losses in a short time. As we have seen, the congestion avoidance phase will be invoked and the TCP rate is reduced. However, in a wireless channel, a deep fade rarely means a long term reduction in available bandwidth. The momentary losses are not caused by congestion and the TCP measures result in a unnecessary reduction in end-to-end throughput. In Fig. 1, the losses occurred due to a fading that lasted less than 2 seconds. After $t = 7$ sec., TCP has all the link throughput available. However, TCP assumed that congestion was in place and erroneously estimated that ssthresh for that situation was five segments. The window size will take several seconds to reach its optimal value and a significant percentage of throughput is wasted.

As we see, even the most basic principle of efficiency evaluation from the original TCP design needs to be revisited for wireless channels. Plain average channel throughput does not necessarily yield an equivalent TCP throughput. From the user standpoint, a new wireless technology offering high throughput but poor BER does not make a difference, as the observed throughput will heavily depend on the behavior of TCP over those channels.

3.3 Existing Approaches

It seems evident that the major problem of TCP when it is used over wireless channels is the lack of an appropriate error control mechanism [34]. TCP is unable to determine if an out-of sequence segment is due to a loss, a congestion problem, or a reduction in real bandwidth. The problem is exacerbated when the error pattern is varying, such as the one observed in a wireless channel. To overcome these problems, several approaches have been proposed in the literature [34], basically of three types:

- *Changes on TCP implementation*. As examples, *TCP New Reno* [12] introduces the concept of partial acknowledgements where the received ACK does not acknowledge in flight segments, avoiding misinterpreting reordering as losses, and *TCP Santa Cruz* [23] introduces new algorithms for a better estimation of the round trip time. In general, the changes in TCP implementation require the change of the existing TCP base, so they are long term solutions.
- *TCP connections splitting*. This method separates the TCP connection in two parts: the correspondent to the wired part of the path and the correspondent to the wireless part. It is usually implemented either in a new TCP implementation, like *I-TCP* [4], or

[3] The configuration is identical to the one described in Section 5, with BLAST MMSE and SNR=10dB.

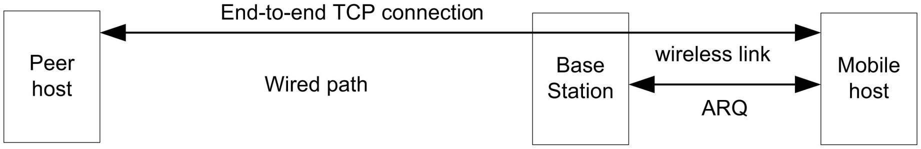


Fig. 2. Local retransmission architecture.

transparently by using a proxy as in *WTCP* [28]. The proxy, located on the base station, snoops for TCP packets and takes actions, like dropping duplicated ACKs so the connection in the wireless path can recover faster, increasing the throughput.

- *Local retransmission*. The most common way to hide losses from TCP is to use a local retransmission mechanism just below the IP level in the wireless link, as shown in Fig. 2. These link layer protocols use the available time of the generous TCP time-out values to retransmit the lost frames. The key, again, is to avoid the slow start phase to take place. This approach is already present in wireless standards, such as the 3G1X Radio Link Protocol (RLP) [33] and UMTS Radio Link Control (RLC) [24]. The local retransmission protocol also fragments the TCP segments because the FER of the channel heavily depends on the frame size. The reduction of the size of the frames reduces the observed FER but also increases overhead.

A further benefit from the local retransmission mechanism is the fact that it is transparent to higher layers, in particular, to TCP. This, together with its standardization, makes it the most common method for enhancing TCP over wireless channels.

4 ARQ WITH PACKET COMBINING FOR MIMO

The local retransmission mechanism is usually located on the Data Link Control layer (DLC) and implements a form of ARQ error detection [18]. The basic ARQ protocols work as follows: When a frame is received, it is first checked for errors. If the frame contains errors, it is discarded and a retransmission is requested if the sender is known. Upon timeout, the sender typically retransmits the frames not acknowledged and, depending on the scheme, the receiver may also request the expected frames not received yet.

Packet combining can be employed in conjunction with ARQ: Instead of discarding the old packets that contain errors, the soft decision statistics obtained for every ARQ retransmission are coherently combined symbol by symbol, resulting in a gain of effective SNR [9]. We next discuss the packet combining for the MIMO systems described in Section 2.

First, consider the BLAST system (1) with ML detection (2). Suppose that the symbol vector $\boldsymbol{s}$ is transmitted by the ARQ protocol L times. Then, we have

$$\boldsymbol{y}(l) = \sqrt{\frac{\rho}{n_T}}\boldsymbol{H}(l)\boldsymbol{s} + \boldsymbol{n}(l), \quad l = 1, 2, \ldots, L, \tag{17}$$

where $\boldsymbol{y}(l)$, $\boldsymbol{H}(l)$, and $\boldsymbol{n}(l)$ are the received signal, the MIMO channel value, and the receiver noise corresponding to the $l\text{th}$ retransmission, respectively. Then, the ML decision rule based on the L received signals is given by

$$\hat{\boldsymbol{s}}_{ML} = \arg \min_{\boldsymbol{s}\in\mathcal{A}^{n_T}} \sum_{l=1}^{L} \left\| \boldsymbol{y}(l) - \sqrt{\frac{\rho}{n_T}}\boldsymbol{H}(l)\boldsymbol{s} \right\|^2. \tag{18}$$

On the other hand, when the MMSE detection with ordered interference cancellation is employed, we denote the decision statistic corresponding to the $i\text{th}$ symbol s_i and the $l\text{th}$ transmission as $z_i(l)$ (line 7 of the algorithm in Section 2.1). Then, the combined decision statistic is given by $z_i = \sum_{l=1}^{L} \omega_i(l) z_i(l)$.

Two packet combining schemes are in order. In equal gain combining, we simply set $\omega_i(l) = 1$ for all $l = 1, 2, \ldots, L$. In a maximal ratio combining, on the other hand, the combining weight $\omega_i(l)$ is proportional to the signal-to-noise ratio, i.e.,

$$\omega_i(l) = \frac{1}{\{\boldsymbol{\Omega}_{k_i,k_i}\}}, \tag{19}$$

where $\boldsymbol{\Omega}$ and k_i are specified by lines 4 and 5 of the MMSE algorithm.

Now, we turn to the space-time coding schemes discussed in Section 2.2. For the half-rate code $\boldsymbol{C}_{8,4}$, we denote the decision statistics vector given by (6) and corresponding to the $l\text{th}$ retransmission as $\boldsymbol{z}(l)$. Then, the combined decision statistic vector is given by $\boldsymbol{z} = \sum_{l=1}^{L} \omega(l)\boldsymbol{z}(l)$, and the combining weight $\omega(l)$ for the $l\text{th}$ retransmission is given by

$$\omega(l) = \sum_{i=1}^{n_R} \sum_{k=1}^{4} \left| h_{i,k}(l) \right|^2, \quad l = 1, 2, \ldots, L. \tag{20}$$

For the rate-1 code $\boldsymbol{C}_{4,4}$, the decision statistics per antenna obtained in (12) and (13) define two different 2×2 BLAST systems, and the combining can be performed separately for each as in the BLAST scheme described in (19). Similarly, the rate-2 code defined by the equivalent BLAST system in (16) can be combined following the scheme in (19).

5 SIMULATION SETUP

In this section, we combine the different elements described in the previous sections. We want to measure the TCP throughput when operating over a wireless system that employs different MIMO and ARQ techniques. For this study, we modified RLC module in the GPRS implementation by Richa Jain at IITB (India) to implement a link layer retransmission mechanism based in the IS-99 RLP for the ns-2 simulator [27]. The physical layer MIMO modules were implemented in MATLAB as described in Section 2 and embedded in ns-2 as external functions for the characterization of the links.

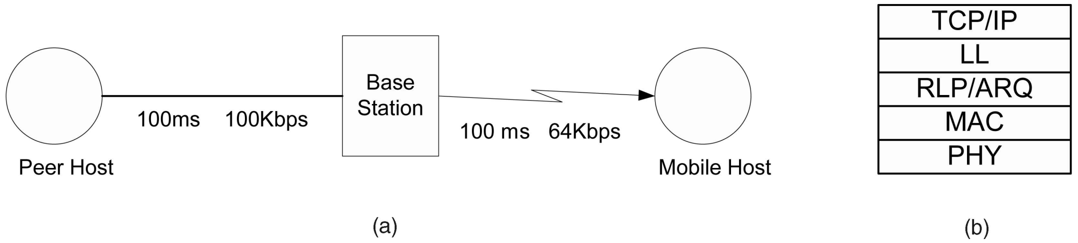


Fig. 3. Network scenario used in our simulations. (a) Network scenario. (b) Stack.

5.1 Network Scenario

We consider the scenario depicted in Fig. 3a, in which a large data file is transferred via File Transfer Protocol (FTP) from a fixed node to a mobile host. The fixed links have a delay of 100ms representing (more than) one noncongested hop.

A typical TCP/IP/LL/RLP stack is used on the wireless link between the Radio Network Controller (RNC) and the Mobile Host (MH) (Fig. 3b). We do not consider the multiuser scenario in which the medium is shared and a complex MAC protocol is needed, as we are interested only in the asymptotic performance of end-to-end TCP connections and not in the multiuser interaction. We do not consider other intermediate transport layers, such as Point-to-Point Protocol (PPP), as they usually have fixed sizes that generate a constant overhead over the total performance.

The TCP implementation used is Reno with selective acknowledgement. The size of the data segment is 536 bytes, which is the standard MSS for the RLP in the IS-99 implementation [33], and the LL maximum frame size is 1,500 bytes.

The RLP layer implements a pure NACK selective acknowledgment hybrid ARQ type I protocol that performs retransmissions, fragmentation, and reassembly. The RLP frame size is 30 bytes, so, typically, a TCP segment will need 20 RLP frames to be completely transmitted, taking into account the RLP overhead. The selective repeat ARQ protocol requires buffering both in the sender and in the receiver. Moreover, the receiver has a timeout for every missing frame. The retransmission timeout accounts for buffering and segmentation delays, and it is typically set to the time needed for sending four RLP frames. A loss is detected when a nonconsecutive RLP frame is received or a timeout for a frame occurs. In case of loss, a NACK for the missing frame is sent back to the sender, which proceeds to a retransmission. This process continues until the correct frame is received or a maximum number of timeout expiration n per frame is reached (n ranges from three to 10 retransmissions). If, after the n attempts, an RLP frame is still missing, the RLP layer does not pass any of the fragments to the link layer and discards them silently (the upper layers will eventually handle the loss). RLP also periodically sends ACK packets to free buffers from the sender.

5.2 Link Layer Retransmission Mechanism

The RLP retransmission algorithm is explained next. Let s denote the sequence number of the packet just arrived. RLP keeps one counter for the sender *nextseq* that accounts for the next packet to be sent. The sender algorithm simply keeps sending the frames received from the upper layer and retransmits the segments requested by the receiver's NACKs.

The receiver maintains two counters: $expected$ is the next frame that is expected to be received and $needed$ is the next missing frame needed (the minimum sequence number of the missing frames). The receiver algorithm also has to account for the combining of frames. For the combining to be effective, certain fields of the frames need to be heavily protected to avoid corruption, particularly the sequence number. Otherwise, the receiver would be unable to tell with which frame the newly received frame is combined. We assume that a strong forward-error-correction (FEC) code is applied to the RLP headers so the sequence information and the packet type can always be recovered[4] or at least the error can be detected with a high probability. This is a reasonable assumption considering that the RLP headers are small (24 bits in IS-99). If the sequence number cannot be recovered, the received frame would be silently discarded without being combined. The receiver algorithm, based on [10], is detailed next without considering the maximum number of retransmissions:

```
while receive frame $f_s$ do
    if $f_s$ from the upper layer then
        send($f_s$).
    else if $s < needed$ then
        discard frame. (Frame is duplicated)
    else if $s = needed$ then
        receive_buffer($s$) $\leftarrow$ combine(receive_buffer($s$),$f_s$).
        if receive_buffer($s$) is corrupt then
            send NACK($s$)
        else
            $needed \leftarrow$ next frame needed with lowest sequence number.
            Pass all frames up to $s$ to upper layer.
        end if
    else if $needed < s < expected$ then
        receive_buffer($s$) $\leftarrow$ combine(receive_buffer($s$),$f_s$).
    else if $s = needed = expected$ then
        receive_buffer($s$) $\leftarrow$ combine(receive_buffer($s$),$f_s$).
        $next = next + 1$
        if receive_buffer($s$) is corrupt then
            send NACK($s$)
```

[4] Note that this implies that RLP NACKs and ACKs are always correctly received, as all information in the control packets is carried in the headers.

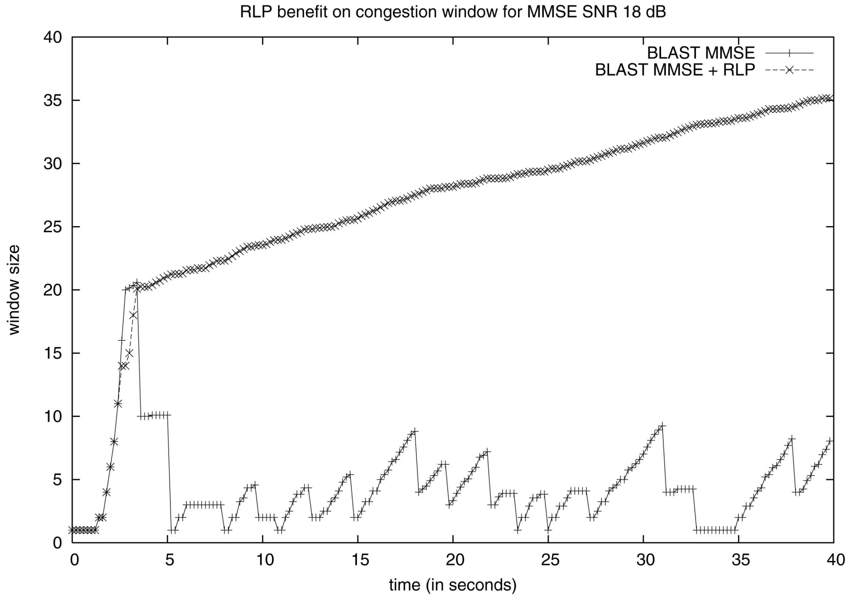


Fig. 4. RLP effect on TCP window for BLAST with MMSE receiver.

```
            else
                $needed \leftarrow$ next frame needed with lowest sequence number.
                Pass all frames up to $s$ to upper layer.
            end if
        else if ($s = expected \neq needed$) or $s > next$ then
            receive_buffer($s$) $\leftarrow$ combine(receive_buffer($s$),$f_s$).
            $next = s + 1$
            if receive_buffer($s$) is corrupt then
                send NACK($s$)
            end if
        end if
    end while
```

The effect of the ARQ algorithm in TCP is to flatten the channel, making it appear as less variable and, more importantly, hiding the losses. Fig. 4 shows the windows size of TCP over a MIMO channel with BLAST MMSE receiver with SNR 18dB. Note that there are no losses when RLP is used, but the delay increment due to the retransmissions prevents the window size to have an optimal growth, producing the "ripple" effect.

5.3 Physical Layer

We consider MIMO systems with $n_T = 4$ transmit and $n_R = 4$ receive antennas signaling over a quasi-static flat-fading channel with quadrature phase shift keying (QPSK) modulation in a rich-scattering indoor wireless environment. Therefore, the BLAST system has a spectral efficiency of 8 bits/sec/Hz. The STBC systems have different spectral efficiencies depending on its rate: The half rate orthogonal code has a spectral efficiency of 1 bit/sec/Hz, the rate-1 quasi-orthogonal code has a spectral efficiency of 2 bits/sec/Hz, and the rate-2 group Alamouti scheme has a spectral efficiency of 4 bits/sec/Hz. To account for that difference in spectral efficiency, the wireless link is 64kbps for the BLAST schemes, 32kbps for the rate-2 STBC, 16kbps for the rate-1 STBC code, and 8kbps for the half-rate STBC code.

Note that the independent quasi-static channel assumption is an ideal case for TCP. In the presence of a slow varying fading channel, the combination of ARQ and MIMO further benefits TCP. While the diversity provided by ARQ can effectively suppress the short fades of a fast fading channel, the diversity provided by MIMO greatly helps to reduce the probability of fades in slow varying channels that otherwise would not be completely removed by ARQ. Also note that it may seem unfair to use the same modulation (QPSK) for both BLAST and STBC systems, as the latter are able to use higher constellations due to their inherent reliability. However, the objective of our analysis is to study the tradeoff between high throughput systems (BLAST) and reliable systems (STBC) and the impact of both schemes in TCP. By increasing the constellation size in STBC, we would be able to increase its throughput at the cost of higher error rate and, hence, lose its main difference with BLAST. Moreover, we will show that the spectral efficiency or the BER cannot be taken alone as performance metrics from the user point of view, but the TCP throughput itself determines the goodput that the user will experience.

6 Results

The performance measurement is the end-to-end throughput of TCP during a 35-second FTP transmission. We want to measure the *effective* bandwidth observed by the user, which, as we saw, does not necessarily have a direct relationship with the available bandwidth of the link. Note also that not all the successfully received RLP frames account for received TCP data. Some TCP fragments may be duplicated and sent in two different successfully received RLP frames, or part of the successfully received fragments of a TCP segment may be discarded if any of them is later considered as nonrecoverable (i.e., it reached the maximum number of retransmissions). In that case, all the successfully received fragments of that TCP segment will be discarded and will not count as effective TCP throughput.

6.1 TCP Throughput without ARQ

The TCP throughput when the ARQ retransmission mechanism of the RLP layer is not activated is shown in Fig. 6. As all TCP frames are of the same size, we measure the throughput in number of segments successfully received and passed to the upper layers. Although the ARQ protocol is not active, the TCP segments are still fragmented at the RNC. This is a reasonable policy as the probability of frame error increases with the frame size. Note also that the overhead with and without ARQ is the same. As the FER increases with the frame length, a typical TCP segment without fragmentation would observe very high error rates.

The effect of the difference on spectral efficiency for the different channels on the overall TCP performance is noticeable. We have also added the results for a SISO MMSE system for comparison purposes. It is clear the For a SNR of 21dB and above the BLAST MMSE channel is preferable because the low FER observed. However, the quality of the BLAST channel drops significantly in the 15-20dB range in favor of the more reliable STBC channels. As described in Section 2, the increment in the STBC ratio is obtained by sacrificing the orthogonality of the code matrix, which increases the effective throughput of the channel but also has an impact on its reliability. The most reliable channel is the half rate STBC, which allows TCP to have the maximum available throughput with a SNR as low as 2dB. Above SNR 7db, STBC rate-1 and the STBC rate-2 offer similar TCP throughput, but the higher spectral efficiency of the STBC rate-2 receiver makes it preferable in terms of TCP throughput.

6.2 TCP Throughput with ARQ

Fig. 7 and Fig. 8 show the TCP throughput when ARQ is used for 10 and three maximum retransmissions, respectively. It is clear that the persistence of RLP retransmissions is beneficial for the overall TCP throughput despite the increase in round trip time, at least when combining is not used. However, the difference between three and 10 retransmissions is small, so it is important to note that increasing the persistence beyond certain limits may not be beneficial, especially for uncorrelated channels [3]. The drop in BLAST MMSE occurs in the SNR range of 12-20dB, meaning a consistent 4-5dB gain. In the STBC systems, the drop in throughput is smoother when ARQ is activated. The reliability lead of the half-rate system is still clear for lower SNR values and, above 5dB, the rate-2 STBC system achieves the best results.

6.3 TCP Throughput with ARQ and Combining

The effect of combining is shown in Fig. 9 and Fig. 10. The RLP persistence is showed for 10 and three retransmissions, respectively. Several observations are in order. First, by using combining, the TCP performance is improved in all systems. However, the gain is clearly superior in the BLAST MMSE receiver, in which the gain ranges from 1dB in 16dB SNR to a significant 8dB gain for channel conditions below 12dB of SNR. The throughput difference for STBC systems is, however, negligible above 6dB and the SNR gain ranges 1-2dB for channel conditions below 6dB of SNR. Second, the performance of a the rate-1 STBC outperforms BLAST MMSE in the 0-14dB SNR range while the rate-2 STBC is preferable for SNR values below 16dB.

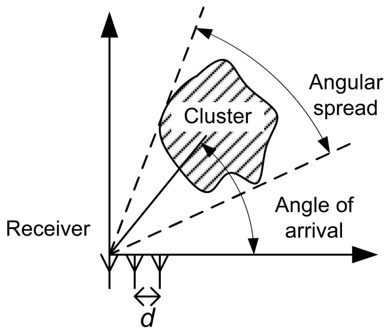


Fig. 5. Antenna correlation model.

Note that, for a normal range of operation, with SNR in the range of 15-25dB, the BLAST MMSE system outperforms the rest of the receivers. As an important observation, the ARQ with packet combining does significantly increase the throughput observed when STBC systems are used, unlike the BLAST MMSE receiver.

Finally, it is important to show that the RLP persistence for low SNR values when using combining is not always beneficial. As we can see in Fig. 9 and Fig. 10, the TCP throughput for 10 and three retransmissions is similar, and even slightly worse, in some SNR regions. This indicates that going beyond three retransmissions does not provide further benefit to the TCP throughput, but might even be counterproductive. Further increasing the number of retransmissions only introduces more congestion in the path and reduces available bandwidth, provoking TCP to timeout and reducing the throughput.

6.4 Effects of Channel Correlation on TCP Performance

So far, we have considered spatial uncorrelated MIMO channels, i.e., the channel matrix $\boldsymbol{H}$ contains i.i.d. elements. It is known that such an assumption is reasonable if enough separation is provided among the antennas [16]. However, this separation may not be feasible in small devices such as cellular phones or PDAs, especially when n_T and n_R are large. The effect of MIMO channel correlation on the physical layer performance, such as capacity and BER, is discussed in [6]. We are interested, however, in the effect of the correlation on the effectiveness of the RLP layer and how the persistence values obtained for the uncorrelated case is affected.

Assuming no line of sight between transmit and receiver antennas and assuming the signals encounter a cluster of scatters on their way to the receiver, the signals reaching the receive antennas can be modeled with the following three parameters (Fig. 5): 1) The *distance* d

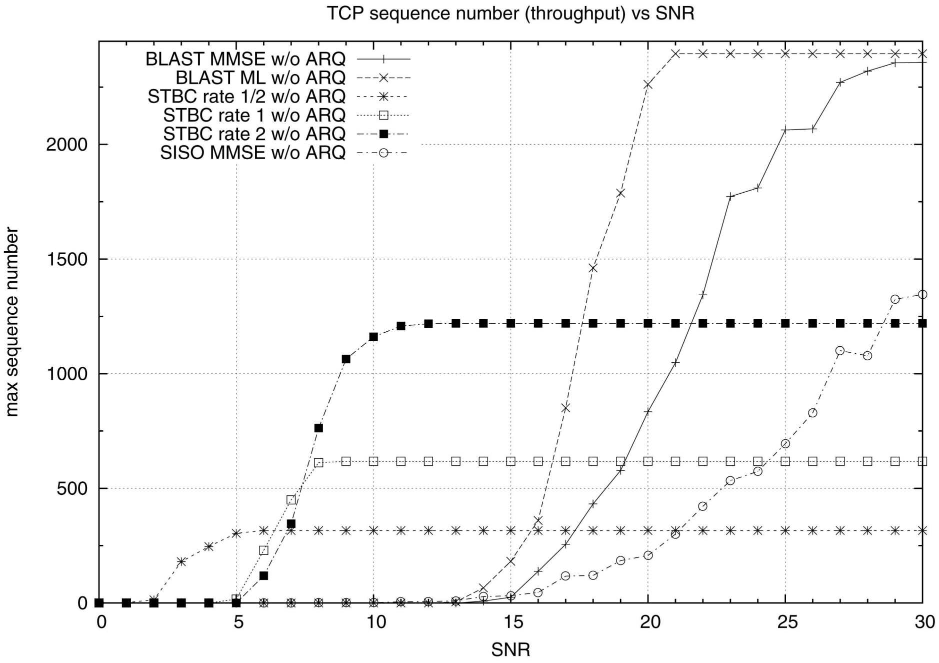


Fig. 6. TCP throughput versus SNR without ARQ.

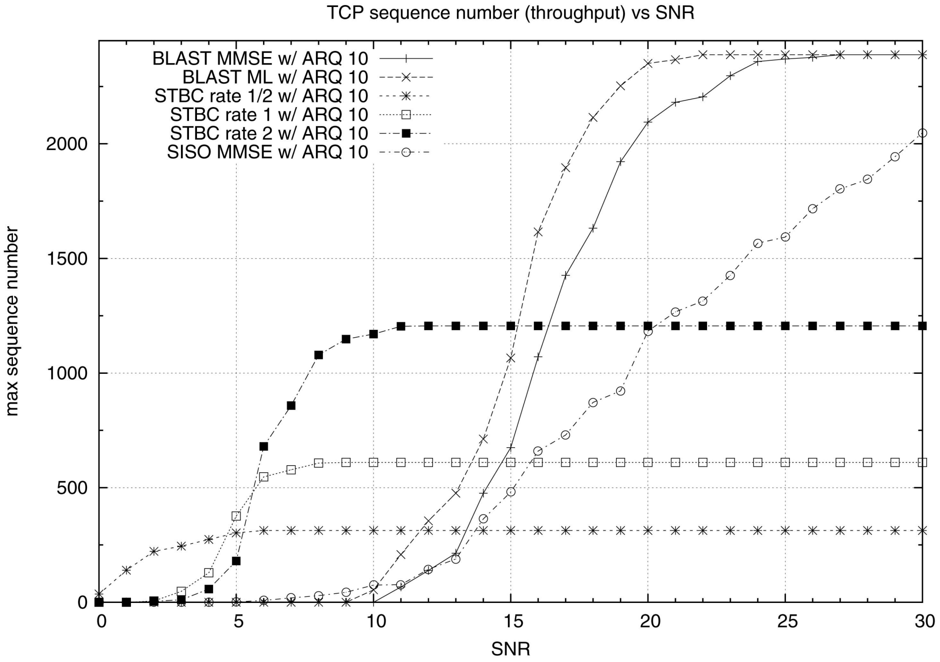


Fig. 7. TCP throughput with ARQ and without combining for 10 retransmissions.

between antennas measured in wavelengths (λ). 2) The *angular spread* δ_o^R of the arrival incident waves (δ_o^T for transmit antennas). If δ_o^R is large, the signals appear uncorrelated, like in the case of an urban environment where the scatterers are big buildings close to the receiver. If δ_o^R is small, the signals appear correlated. That would be the case of a rural environment in which the scatterers are small and located far away from the receiver. c) The *mean angle* $\bar{\phi}_o^R$ of the arrival incident waves ($\bar{\phi}_o^T$ for the transmit antennas), which indicates the orientation of the antennas with respect to the direction of the incident waves.

In the receiver, the angle of incident of the signal from the cluster is $\phi_o^R = \bar{\phi}_o^R + \hat{\phi}_o^R$ with $\hat{\phi}_o^R \sim \mathcal{N}_c(0, \sigma_{\phi_o^R})$, and $\sigma_{\phi_o^R} \sim \delta_o^R$. Let us denote $\boldsymbol{H}_k$ as the $k\text{th}$ column of the channel matrix $\boldsymbol{H}$, such as $\boldsymbol{H} = [\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_{n_T}]$. Then, the correlation matrix for the $k\text{th}$ column is given by $\boldsymbol{R}_r = E\{\boldsymbol{H}_k \boldsymbol{H}_k^T\}$, and it is independent of k, i.e., the correlation statistics do not depend on the transmit antenna considered [7]. As shown in [2], for small angular spread, the correlation matrix for the receiver can be approximated as

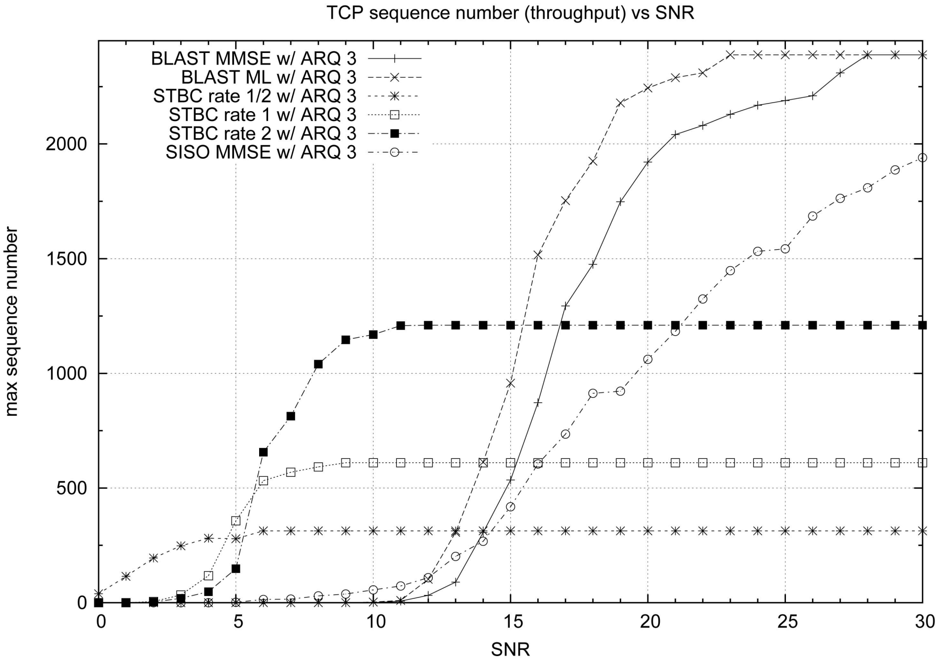


Fig. 8. TCP throughput with ARQ and without combining for three retransmissions.

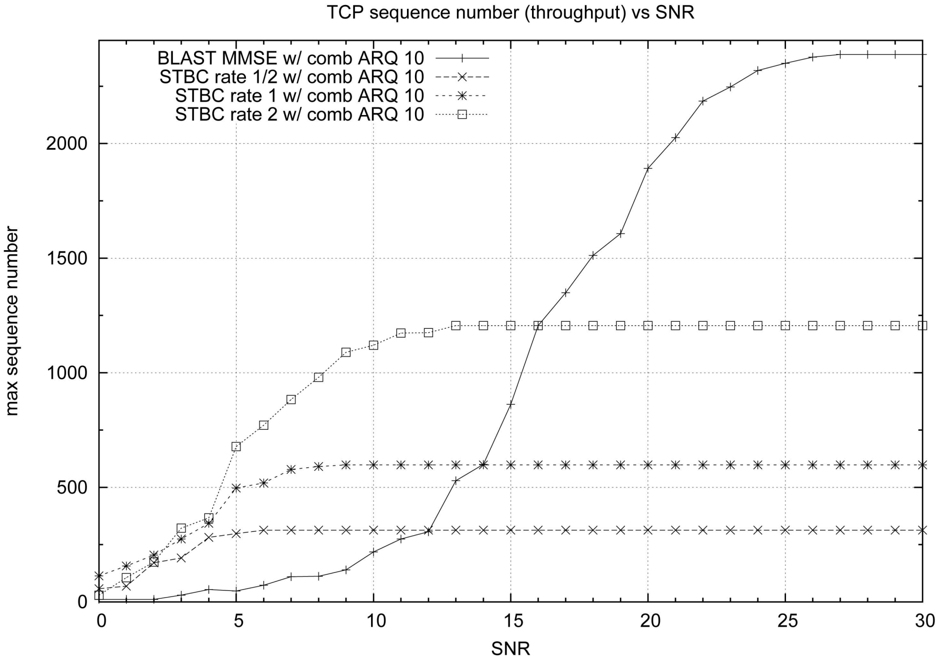


Fig. 9. TCP throughput with ARQ and combining for 10 max retransmissions.

$$[\boldsymbol{R}_r]_{i,j} \approxeq e^{-\jmath 2\pi(j-i)d\cos(\bar{\phi}_o^R)} e^{\frac{1}{2}[2\pi(j-i)d\sin(\bar{\phi}_o^R)\delta_o^T]}, \quad i,j = 1,2,\ldots,n_R, \tag{21}$$

assuming equal antenna spacing d and normalized signal power $|\beta| = 1$. A similar expression for (21) is obtained for the correlation matrix of the transmit antennas $\boldsymbol{R}_t$ considering $\bar{\phi}_o^T$ and $\sigma_{\phi_o^T}$ as mean angular spread and angular spread, respectively. Assuming correlation at both the transmitter and the receiver, the MIMO channel response matrix can be expressed as $\boldsymbol{H} = \boldsymbol{R}_r^{1/2}\boldsymbol{H}_w\boldsymbol{R}_t^{1/2}$, where $\boldsymbol{H}_w$ is an $n_T \times n_R$ matrix containing i.i.d. $\mathcal{N}_c(0,1)$ random variables; $\boldsymbol{R}_r$ and $\boldsymbol{R}_t$ represent the $(n_R \times n_R)$ and $(n_T \times n_T)$ covariance matrices defined in (21) that induce the receive and transmit correlations, respectively.

Following the correlation model previously described, we consider two different correlation scenarios detailed in [6]. The first one is an urban environment with high population of scatters and with a medium-high angular spread for transmitter and receiver. This scenario provides a low correlation of antennas. The second is a rural area where scatters are rare and located far from both the sender and receiver, producing a low angular spread and, hence,

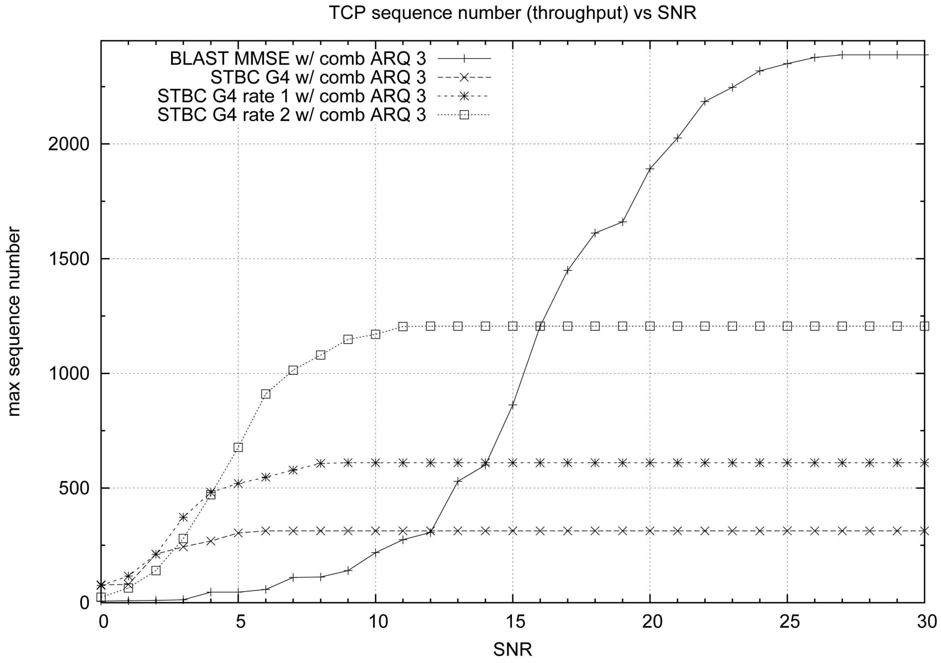


Fig. 10. TCP throughput with ARQ and combining for three max retransmissions.

big correlation among the antennas. In all cases, the antennas are spaced at distance $d = 0.5\lambda$. The exact parameters are shown in Table 1.

6.4.1 TCP Throughput without ARQ

Fig. 11 shows the effect of the urban correlation in the TCP throughput when ARQ is not used. The negative impact in the TCP performance is noticeable, mainly in the less reliable BLAST schemes. The BLAST MMSE receiver does not provide any TCP throughput for the 0-30dB SNR range. The ML observes a similar effect up to 23dB SNR. The STBC receivers, however, behave better than the BLAST systems, but suffer a significat reduction of SNR gain. The rate-2 STBC system reaches its peak on TCP throughput at 19dB SNR, which means a 7dB loss with respect to the uncorrelated case obtained in Fig. 6. The half-rate and rate-1 STBC systems suffer a similar 6-9dB drop on their performance when the channels are correlated according to the urban model.

The rural correlation is higher compared to the urban correlation and, so, the expected result is a reduction in the TCP throughput, as Fig. 12 shows. The effects are severe for all systems, especially the BLAST systems that are not able to transmit a single TCP segment in the 0-30dB SNR range. The STBC systems observe a performance drop equivalent to 12dB SNR for the rate-2 receiver and around 10dB for the other two STBC receivers.

As expected, the performance of MIMO channels in correlated conditions is worse than the uncorrelated case, and it has a significant impact on the TCP performance. In general, the urban rich-scattering environment is better suited for MIMO channels than the rural. Also, note that the effect is severe for the less reliable BLAST receiver than for the STBC systems. It is clear from this result that the higher spectral efficiency of the BLAST scheme does not translate into a higher observable throughput from the point of view of the application.

TABLE 1
Values of the Correlation Parameters for the Urban and Rural Scenarios

Scenario	δ_o^T	ϕ_o^T	δ_o^R	ϕ_o^R	d
Urban	7^o	84^o	7^o	60^o	0.5λ
Rural	2^o	84^o	2^o	60^o	0.5λ

6.4.2 TCP Throughput with ARQ

Fig. 13 shows the TCP performance in an urban correlated scenario when the maximum number of ARQ retransmissions is three. As expected, the ARQ mechanism improves the TCP throughput compared to the case when no ARQ is used (Section 6.4.1). The most noticeable improvement occurs in the BLAST systems. The ML receiver suffers a SNR gain greater than 6dB and, in the case of the MMSE receiver, the gain is at least 15dB for moderate to low SNRs. It is evident that the ARQ persistence in the case of correlation results in an dramatic decrease in the observed error rate, mitigating the effects of the correlation. The benefit for the STBC receivers, on the other side, is minor. As expected, the benefit obtained by the ARQ is inversely proportional to the reliability of the MIMO scheme. In that sense, the STBC rate-2 receiver observes the largest gain in SNR, around 5-6dB, while the other STBC schemes improve by a lower margin, approximately 2-3dB.

6.4.3 TCP Throughput with ARQ and Packet Combining

Finally, we investigate the effect that the packet combining produces on the TCP throughput of the systems like the ones

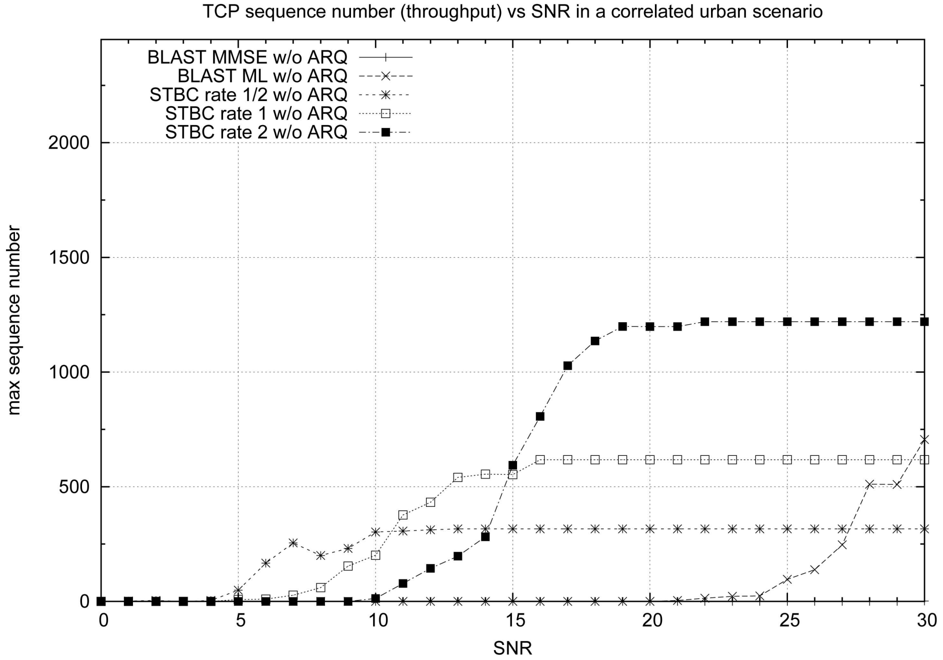


Fig. 11. TCP throughput without ARQ and with correlated antennas in an urban scenario.

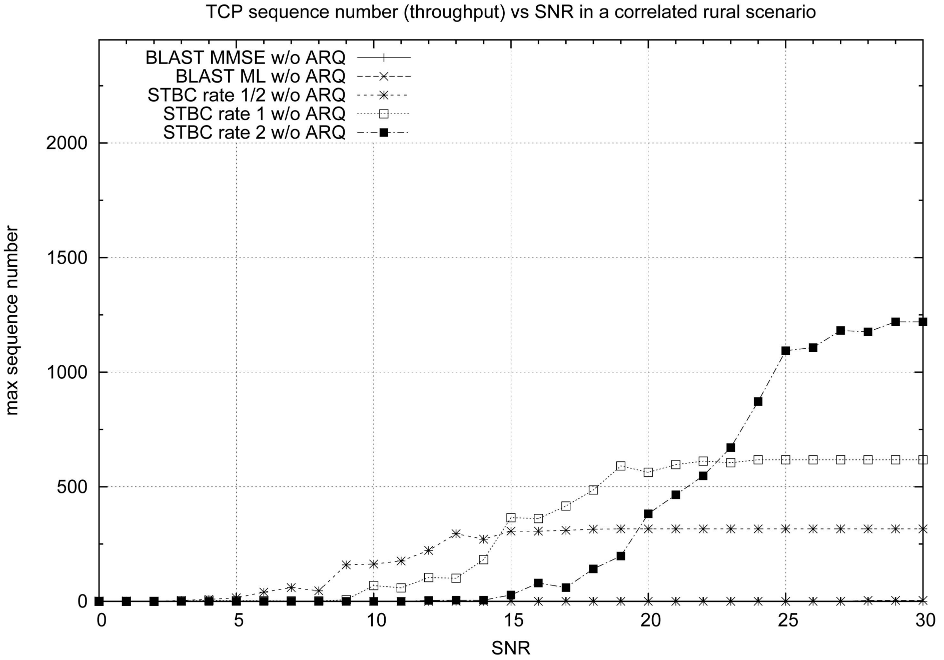


Fig. 12. TCP throughput without ARQ and with correlated antennas in an rural scenario.

in Section 6.4.2 that already implement an ARQ mechanism. The objective is, first, to evaluate the benefit of the packet combining in correlated scenarios and, second, to estimate the maximum ARQ persistence for those systems.

Fig. 14 shows the effect of the packet combining on the systems without combining showed in Fig. 13. Unlike the uncorrelated case, the benefit obtained through combining is minimal for STBC systems and larger for BLAST systems. The hostile MIMO correlated channels allow the retransmission mechanism just little room for improvement, and the TCP end-to-end delay begins causing an overall negative impact. It is interesting to note, however, that the combining effectively improves the TCP throughput. As in the previous section, the less reliable BLAST systems take more advantage than the STBC systems.

Fig. 15 shows the results for rural correlation and a maximum of 10 retransmissions. The results are similar to the urban correlation: The BLAST receiver improves and the STBC receivers hardly observe any changes for either combining or ARQ persistence.

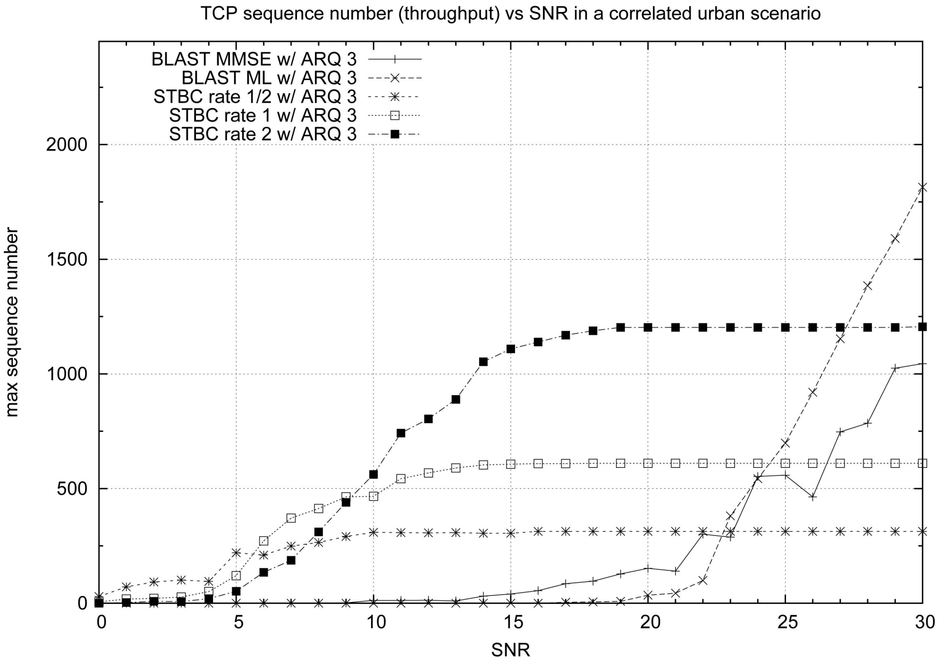


Fig. 13. TCP throughput with ARQ and correlated antennas for the urban scenario. The maximum number of retransmissions is three.

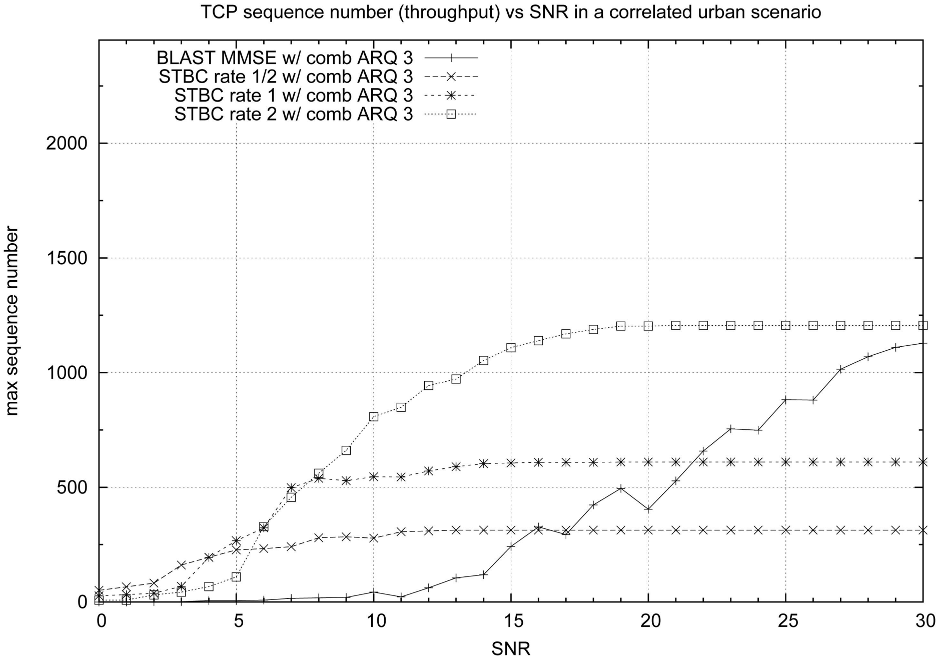


Fig. 14. TCP throughput with combining ARQ and correlated antennas for the urban scenario. The maximum number of retransmissions is three.

7 Conclusions

In this paper, we investigated the effect of MIMO channels in modern TCP systems. In particular, we considered the TCP throughput performance over two different MIMO systems: the BLAST system and three space-time block coding systems with different rates. We showed that TCP can benefit from the better reliability of the STBC systems up to a SNR of 20dB. However, at higher SNR, the BLAST system outperforms the STBC systems.

We also showed the benefit obtained when using link level retransmissions mechanisms that implement hybrid ARQ type I with packet combining. The results obtained show that the packet combining method significantly improves the performance of the MMSE BLAST receiver (more than 10dB some times). From a cross-layer design perspective, it shows that space-time coding can be used instead of an ARQ protocol to improve TCP performance under poor channel conditions (SNR before 12dB), but when channel conditions improve, a switch to the BLAST scheme with ARQ is preferred.

In addition, we observed that when the MIMO channels are correlated, either with a low correlation, such as the

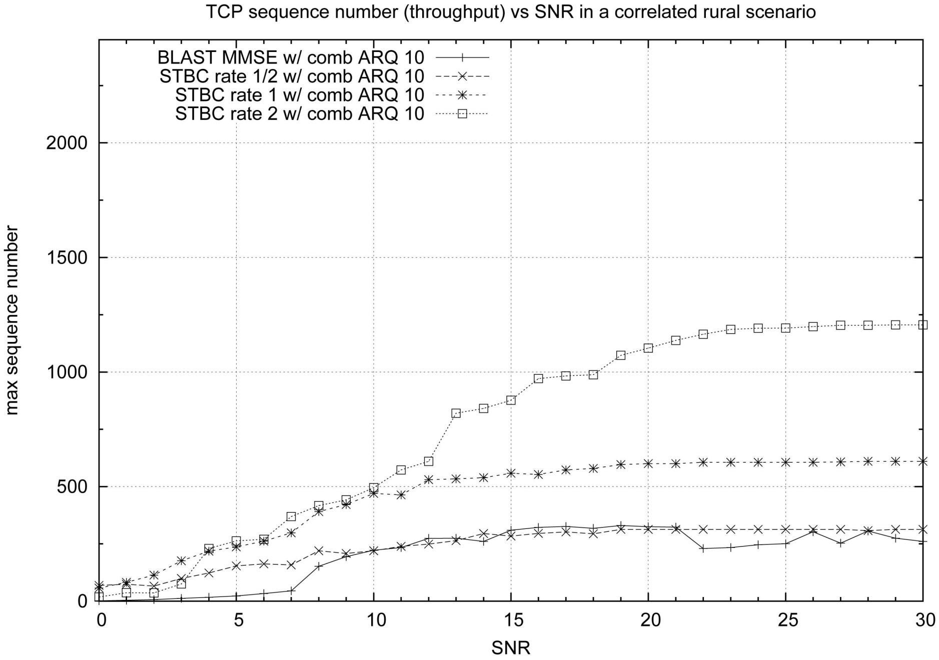


Fig. 15. TCP throughput with combining ARQ and correlated antennas for the rural scenario. The maximum number of retransmissions is 10.

urban scenario, or when the correlation is more severe, such as in the rural scenario, the more reliable STBC systems are always preferable for channel conditions below SNR 30dB. The ARQ retransmission mechanisms, together with the packet combining, significantly improve the performance of TCP under correlated channels for BLAST systems, offering a minor improvement for the more reliable STBC systems. In general, the more reliable the system and the more correlated (hostile) the channel, the less improvement the ARQ will provide and, in some cases, increasing the persistence has a negative impact in form of TCP round-trip delay (and, hence, a reduction in TCP throughput).

As a major point, our investigation shows that, when regarding application performance, the common physical-layer approach of just increasing spectral efficiency does not necessary result in an increment of the TCP throughput. TCP is not designed for wireless channels that show varying throughput/BER, and the effect is exacerbated by the channel correlation encountered in certain systems. It is clear from our results that a cross-layer design should take this into account and tradeoff spectral efficiency for more reliable channels, primarily under low SNR conditions. This is true even when employing modern ARQ and combining techniques, as they are not completely able to improve the TCP throughput in hostile situations of low SNR and moderate/high correlation.

Acknowledgments

This work was supported in part by the US National Science Foundation (NSF) under grant CCR-0207550 and in part by the US Office of Naval Research (ONR) under grant N00014-03-1-0039. Alberto Lopez Toledo is supported by a La Caixa Fellowship.

References

[1] S.M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications," *IEEE J. Selected Areas in Comm.*, vol. 16, no. 8, pp. 1451-1458, Oct. 1998.

[2] D. Asztly, "On Antenna Arrays in Mobile Communication Systems: Fast Fading and GSM Base Station Receiver Algorithms," Technical Report IR-S3-SB-9611, Royal Inst. of Technology, Stockholm, Sweden, 1996.

[3] Y. Bai, A.T. Ogielski, and G. Wu, "Interactions of TCP and Radio Link ARQ Protocol," *Proc. IEEE Vehicular Technology Conf.*, 1999.

[4] A. Bakre and B.R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," *Proc. Int'l Conf. Distributed Computing Systems,* 1995.

[5] H. Balakrishnan, V.N. Padmanabham, S. Seshan, and R.H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *Proc. ACM SIGCOMM '96,* Aug. 1996.

[6] I. Berenguer and X. Wang, "Space-Time Coding and Signal Processing for MIMO Communications," *J. Computer Science and Technology*, vol. 18, no. 6, pp. 689-702, 2003.

[7] H. Bolcskei, D. Gesbert, and A.J. Paulraj, "On the Capacity of OFDM-Based Multi-Antenna Systems," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP),* June 2000.

[8] M.C. Chan and R. Ramjee, "TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation," *Proc. ACM Mobicom '02,* Sept. 2002.

[9] D. Chase, "Code Combining—A Maximum Likelihood Decoding Approach for Combining an Arbitrary Number of Noisy Packets," *IEEE Trans. Comm.*, vol. 33, no. 5, pp. 385-393, May 1985.

[10] A. Chockalingam and G. Bao, "Performance of TCP/RLP Protocol Stack on Correlated Rayleigh Fading DS-CDMA Links," *IEEE Trans. Vehicular Technology,* vol. 49, no. 1, pp. 28-33, Jan. 2000.

[11] E. Cianca, M. Ruggieri, and R. Prasad, "Improving TCP/IP Performance over CDMA Wireless Links: A Physical Layer Approach," *Proc. IEEE Int'l Symp. Personal Indoor and Mobile Radio Comm. (PIMRC),* Sept. 2001.

[12] S. Floyd and T. Henderson, "The New-Reno Modification to TCP's Fast Recovery Algorithm," *Internet Eng. Task Force, Request for Comments (RFC) 2582,* Apr. 1999.

[13] G.J. Foschini, "Layed Space-Time Architecture for Wireless Communication in a Fading Environment when Using Multi-Element Antennas," *Bell Labs. Technical J.,* vol. 1, no. 2, pp. 41-59, 1996.

[14] G.J. Foschini and M.J. Gans, "On the Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas," *Wireless Personal Comm.*, vol. 6, no. 3, pp. 311-335, 1998.

[15] A. Gurov and R. Ludwig, "Responding to Spurious Timeouts in TCP," *Proc. IEEE INFOCOM,* Mar. 2003.

[16] W.C.Y. Lee, *Mobile Communications Design Fundamentals,* second ed. Wiley, 1992.

[17] C. Li and X. Wang, "Performance Comparisons of MIMO Techniques with Application to WCDMA Systems," *EURASIP J. Applied Signal Processing,* to appear.

[18] S. Lin, D. Costello, and M. Miller, "Automatic-Repeat-Request Error-Control Schemes," *IEEE Comm. Magazine,* vol. 22, no. 12, pp. 5-17, Dec. 1984.

[19] T.L. Marzetta, "BLAST Training: Estimation Channel Characteristics for High-Capacity Space-Time Wireless," *Proc. Ann. Allerton Conf. Comm., Computer Control,* Sept. 1999.

[20] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgement Options," *Internet Eng. Task Force, Request for Comments (RFC) 2018,* Apr. 1996.

[21] A. Milani, V. Tralli, and M. Zorzi, "On the Use of Per-Antenna Rate and Power Adaptation in V-BLAST Systems for Protocol Performance Improvement," *Proc. IEEE Vehicular Technology Conf. (VTC),* Sept. 2002.

[22] C.B. Papadias and G.J. Foschini, "A Space-Time Coding Approach for Systems Employing Four Transmit Antennas," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP),* May 2001.

[23] C. Parsa and J.J. Garcia-Luna-Aceves, "Improving TCP Congestion Control over Internets with Heterogeneous Transmission Media," *Proc. Seventh IEEE Int'l Conf. Network Protocols (ICNP)* 1999.

[24] Third Generation Partnership Project, RLC Protocol Specification (3G TS 25. 322), 1999.

[25] G.D. Golden, P.W. Wolniansky, G.J. Roschini, and R.A. Valenzuela, "V-BLAST: An Architecture for Realizing Very High Data Rates over the Tich-Scattering Wireless Channel," *Proc. 1998 Int'l Symp. Signals, Systems, and Electronics (ISSSE'98),* Sept. 1998.

[26] D.S. Shiu, G.J. Foschini, M.J. Gans, and J.M. Kahn, "Fading Correlation and Its Effect on the Capacity of Multielement Antenna Systems," *IEEE Trans. Comm.,* vol. 48, no. 3, pp. 502-513, Mar. 2000.

[27] Network Simulator 2, http://www.isi.edu/nsnam/ns, 2004

[28] P. Sinha, T. Nandagopal, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks," *Wireless Networks,* vol. 8, no. 2-3, pp. 301-316, 2002.

[29] A. Stamoulis and N. Al-Dhahir, "Impact of Space-Time Block Codes on 802.11 Network Throughput," *IEEE Trans. Wireless Comm.,* vol. 2, no. 5, pp. 1029-1039, Sept. 2003.

[30] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," *Internet Eng. Task Force, Request for Comments (RFC) 2001,* Jan. 1997.

[31] I. Stojanovic, M. Airy, D. Gesbert, and H. Saran, "Performance of TCP/IP over Next Generation Broadband Wireless Access Networks," S. Dixit and R. Prasad, eds., *Wireless IP and Building the Mobile Internet,* Artech House Inc., 2003.

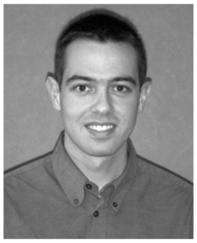
[32] V. Tarokh, H. Jafarkhami, and A.R. Calderbank, "Space-Time Block Codes from Orthogonal Designs," *IEEE Trans. Information Theory,* vol. 45, no. 5, pp. 1456-1467, July 1999.

[33] TIA/EIA IS-707-A-2.10, Data Service Options for Spread Spectrum Systems: Radio Link Protocol Type 3, Jan. 2000.

[34] V. Tsaoussidis and I. Matta, "Open Issues on TCP for Mobile Computing," *J. Wireless Comm. and Mobile Computing,* vol. 2, no. 1, pp. 3-20, Feb. 2002.

[35] Y. Xin, Z. Liu, and G.B. Giannakis, "High-Rate Layered Space-Time Coding Based on Linear Constellation Precoding," *Proc. Wireless Comm. and Networking Conf. (WCNC),* Mar. 2002.

[36] G. Xylomenos, G.C. Polyzos, P. Mahonen, and M. Saaranen, "TCP Performance Issues over Wireless Links," *IEEE Comm. Magazine,* vol. 39, no. 4, pp. 52-58, 2001.



Alberto Lopez Toledo graduated in computer engineering (with highest honors) from the University of Murcia (UMU), Murcia, Spain, in 1999 and received the DAE in computer science from the same university in 2002. He received the MS degree in electrical engineering from Columbia University, New York, in 2003, where he is currently a PhD candidate in electrical engineering. From September 1999 to August 2002, he was with the Department of Telematic Systems Engineering at the Technical University of Madrid (UPM), Madrid, Spain. His research interests are in the area of wireless networking and cross-layer design. He received Spain's National Academic Excellence Award, the Edwin Howard Armstrong Memorial Award, and the La Caixa fellowship. He is a student member of the IEEE.

Xiaodong Wang received the BS degree in electrical engineering and applied mathematics (with highest honors) from Shanghai Jiao Tong University, Shanghai, China, in 1992, the MS degree in electrical and computer engineering from Purdue University in 1995, and the PhD degree in electrical engineering from Princeton University in 1998. From July 1998 to December 2001, he was an assistant professor with the Department of Electrical Engineering, Texas A&M University. In January 2002, he joined the faculty of the Department of Electrical Engineering, Columbia University. Dr. Wang's research interests fall in the general areas of computing, signal processing, and communications. He has worked in the areas of digital communications, digital signal processing, parallel and distributed computing, nanoelectronics, and bioinformatics, and has published extensively in these areas. Among his publications is a recent book entitled *Wireless Communication Systems: Advanced Techniques for Signal Reception*, published by Prentice Hall, Upper Saddle River, in 2003. His current research interests include wireless communications, Monte Carlo-based statistical signal processing, and genomic signal processing. Dr. Wang received the 1999 US National Science Foundation CAREER Award and the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award. He currently serves as an associate editor for the *IEEE Transactions on Communications*, the *IEEE Transactions on Wireless Communications*, the *IEEE Transactions on Signal Processing*, and the *IEEE Transactions on Information Theory*. He is a member of the IEEE.