**Noname manuscript No.**
(will be inserted by the editor)

# Packet Switching, AM Adjustment and Retry Mechanisms for Cross-Layer MIMO Link Design

**Erwin Anggadjaja · Ian McLoughlin**

Received: date / Accepted: date

**Abstract** The effects of sub-channel packet routing and adaptive modulation (AM) are explored in this paper for MIMO linking system under differing channel conditions. Simulations are derived from a single TCP protocol link and then extended into dual- and quad-link models. The aim is to exploit time-varying imbalances between logical sub-channels by different cross-layer routing and modulation strategies to improve overall goodput. Segmentation and reassembly (SAR) is employed with local error control in the form of ACK/NACK triggered retries with timeout. MIMO sub-channel characteristics are modelled in terms of frame-by-frame BER, channel delay, and data rate and used to explore the effect of channel BER imbalance. This paper demonstrates that even simple switching mechanisms can provide an overall performance improvement in a common real-world situation where sub-channels exhibit slightly unequal error rates for durations of one or more transmission frames. A BER-directed adaptive modulation switching scheme is then developed and evaluated, again showing potential performance gain for mismatched sub-channels.

**Keywords** Adaptive modulation · cross-layer design · MIMO sub-channels · Segmentation and Reassembly

E. Anggadjaja
School of Computer Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore, 639798
Tel.: +65 9242-1268
Fax: +65 6790-1585
E-mail: er0001ja@ntu.edu.sg

I. McLoughlin
School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui, 230027

## 1 Introduction

For future wireless systems, many challenges exist in fulfilling the demand for high data rates and good quality of service (QoS) requirements (such as latency and error rate). These are in addition to the ever-present need to maximize throughput/goodput performance and to provide efficient and effective spectral utilisation. Although there are numerous methods worthy of investigation to improve wireless network performance, as the title suggests, this paper assumes the use of TCP/IP networking – which implies a trade off between greater interoperability at a network level and the need to comply with such matters as the TCP packet structure, resend/timeout policy, congestion control measures and session establishment/disestablishment procedures. The use of TCP limits the degrees of design freedom as well as reduces the amount of comparative research literature. However TCP/IP is the pre-eminent general purpose network connectivity standard in use today. Although it is known to suffer from performance issues when applied over wireless links, much work has been done to alleviate these issues.

In terms of wireless communications, multiple-input multiple-output (MIMO) systems have become well-known for potentially delivering high data rate transmission without necessarily increasing bandwidth or transmit power. In fact, MIMO has become popular for its ability to provide higher spectral efficiencies and thus greater throughput or goodput to the end user. For this reason, it is also considered a foundational method for present and future wireless systems. This paper exploits the multi-channel nature of MIMO at the data link layer.

Another technique, adaptive modulation (AM) is emerging as a useful and practical method for managing wireless link error and bitrate trade-offs in deployed systems. This technique adapts modulation and transmission power according to channel propagation conditions, interference scenarios, and traffic or data rate requirements [32] [14] [39]. It usually involves a switching mechanism rather than a continuous adaptation, with the aim of enhancing the spectral efficiency of a wireless system while assuring QoS requirements.

### 1.1 Contribution

This paper assumes a situation where standard TCP/IP based traffic from a wired connection is to be conveyed over a wireless communications link. No change in underlying TCP/IP protocol is applied in this paper, instead, segmentation and reassembly (SAR) of TCP packets is performed at intermediary nodes which implement the wireless physical link [28]. This paper considers wireless links ranging from SISO (single input, single output) to full MIMO (multiple-input, multiple-output), and uses cross-layer design principles to propose simple but effective packet handling strategies that are able to exploit

A TCP-based MIMO-compatible technology called the Parallel Independent Link Model (PILM) will be developed based upon the requirement that the interface between data link and physical layers is expanded to match the multiple physical sub-channels operating in a MIMO system. Using this, we will explore the effect of packet switching mechanisms on for improving goodput in the light of diverse error rates between sub-channels.

This is based upon the observation briefly mentioned above that, in deployed multichannel links, it is rare that instantaneous BER/SNR of each channel is perfectly identical [28], even with co-located antennas. This was observed in [29] where, for a 2 × 1 TR-STBC (time reversal, space-time block coded) system with co-located polarized antennas, the distribution of BER between sub-channels is rarely identical. Most of the time there is a 'bad' channel as well as a 'good' channel. Not only will one or more channels be slightly better or worse than others (perhaps due to antenna placement or electronics tolerances), but all sub-channel BERs vary in complex fading patterns at a rate that normally significantly exceeds the packet length or retraining time [34] – except in the very unusual case when the system is operating continually under absolute worst-case conditions (even for mobile systems, the vast majority of traffic is not conveyed when a device is moving at maximum speed over the poorest channel that the system was designed to cope with).

The presented PILM model here leads naturally to a cross-layer design (CLD) solution, as we evaluate this in the aspects of packet re-routing mechanism and modulation switching for dual- and quad- link arrangements [4]. We note that CLD solutions are well known [36]; yet some caution is needed in the implementation, as shown by [20] and [41]. These techniques tend to be practical and pragmatic – relatively simple to implement and thus may be good choices for future systems, as shown in recent research [43] [18]. The models will be investigated in this paper through OMNeT++ simulation, specifically in looking for improvement in goodput for dual- and quad-PILM systems. Some interesting findings will also be discussed that may impact future system design.

### 1.2 Relevant work

Since much research has been conducted on exploring TCP performance over wireless networks, this paper will not discuss the protocol in detail – proposed solutions include [5] [22], lower-layer enhancements [2] [11] [26], CLD approaches [27] [6] [10] [7], as well as new mechanisms (similar to TCP) for providing the underlying transport protocol [37].

AM has also been investigated widely, with the purpose of matching transmission rates to varying channel conditions, for example in [1] [13] [9] [30] and [35], even for the realistic scenario where only outdated channel information is available [12], erroneous [33], or for MIMO orthogonal space–time block code (OSTBC) environments [17]. In relation to upper layers, AM operates at the physical layer independent of other layers, apart from the assumption of packet flow into and out of the physical layer. In fact, a CLD approach was analyzed by Liu et al. [24] who proposed an interaction of queuing at the data link with AM at the physical layer, this was then improved by Harsini and Lahouti [16] by considering traffic deadline constraints to analyze the joint effects of packet queuing and AM. More related studies regarding cross-layer design approaches have also been carried out, especially by combining AM with ARQ [23] [15] and the TCP layer [25].

Concerning TCP/IP and MIMO, there are many relevant studies. For example, [28] showed how data can be delivered over a MIMO link with the help of local link layer mitigation, without invoking TCP layer error control. In particular, the work also demonstrated that practical MIMO links with given average BER actually spend a substantial amount of time with individual sub-channels experiencing different instantaneous BERs. In [38], two different MIMO systems were proposed with the claim that, in regard to application performance, physical layer approaches do not necessarily improve TCP throughput. However there are trade-offs possible between system performance and SNR. A cross-layer scheme was proposed by Lin and Wong [21] which combined MIMO technology for different MAC protocols. Another work by Zhou et al. in [42] considered MIMO systems combined with AM and queuing mechanishm, which then improved by [19]. And recently, [40] analysed power variation related to adaptive modulation for space-time coded MIMO systems, while [31] proposed a framework of distributed MIMO ad-hoc networks for improving link capacity in regard to avoiding interferences among the nodes.

Although many other authors have explored this issue, to our knowledge, this is the first paper detailing a CLD solution using parallel sub-channel models that implement packet switching mechanisms to exploit diverse multi-channel conditions. In addition, although many authors have investigated AM, we believe that the evaluation of AM across parallel sub-channels with packet switching strategies is novel.

The remainder of this paper is organised as follows. Section 2 presents a TCP-based point-to-point transmission model, in relation to the concept of segmentation and reassembly (SAR) and error control mechanisms. We measure the performance in the light of goodput. In Section 3, we develop a dual-link model which implements proposed switching methods for diverse channel and retry scenarios. Section 4 will present further detail and discussion about the quad-PILM system before Section 5 incorporates adaptive modulation into the scheme, by discussing several restructured-AM scenarios. Finally, Section 6 will conclude the paper.

## 2 TCP-Based Point-to-point Transmission Model

The basic model employed in this paper is to use intermediary TCP/IP-aware wireless nodes to apply a SAR concept to TCP packets conveyed over a wireless channel. A fast ARQ-based sliding window error control mechanism, operating on the segmented data, reduces packet error rates (PER) visible to the TCP layer.

The model is simulated using the INET framework, an open-source network simulation package for the OMNeT++ environment. It contains models for both wired and wireless networking protocols, including TCP/ IP and 802.11 models. We adopt this for the baseline simulation throughout this paper.

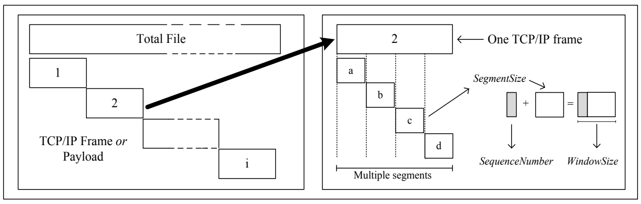


**Fig. 1** Concept of sequence number, segment size, and relationship to TCP/IP frame and original file.

### 2.1 Segmentation and reassembly mechanisms

We define two wired user nodes (A and B) with TCP connectivity which are to communicate using TCP. We then ‘insert’ two wireless transmission-nodes (X and Y) from our previous work [3]. Both X and Y connect to A and B by wired links respectively, and communicate between themselves wirelessly. The model complies with all of the well-known TCP/IP procedures, i.e. connection establishment and termination, flow and error control mechanisms, and also the concept of encapsulation and de-capsulation from the layers above and below. Node A transmits TCP/IP data to Node B through intermediate Nodes X and Y, which are responsible for 'hiding' the potentially high BER wireless link X–Y from nodes A and B.

Within the simulation, a large data block to be transferred from Node A to B is first split into a sequence of TCP payloads at Node A. If this sequence is then transmitted, and received by Node B without error, it will be reassembled into the received data block. During the transmission process, TCP payloads from source Node A travel to Node X where they are split into smaller segments (determined by the segment size parameter, $SS$). A one-byte sequence number is appended to each segment, before it is transmitted over air from Node X, and received by Node Y. This arrangement is shown diagramatically in Fig. 1.

We also add another error control mechanism, similar to the stop-and-wait ARQ protocol (just as TCP does) which is limited in terms of the number of retries only (note: we included re-ACKs in our previous work [3]). These retries are used in conjunction with a timeout: for each segment sent, the sender (Node X) will not send any further data until it receives an appropriate ACK signal for the previous segment. This was designed to ensure that segments are sent and received correctly, since such techniques can be found in implemented systems. More complexity and intelligence will be added to this underlying transport mechanism in subsequent sections.

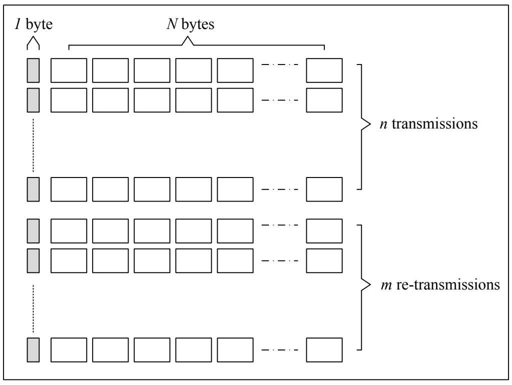


**Fig. 2** $n$ separate segment transmissions, each containing $N$ bytes of data, are made. Each segment includes a one byte sequence number. Due to errors, some segments need to be retransmitted, maybe more than once. So $m$ segment retransmissions will be necessary to successfully convey all data.

### 2.2 Evaluation methods

We define throughput and goodput as measurements used to evaluate the effects of adjusting and changing parameters (several adjustment types are possible in this system, to enable different trade-off dimensions). Throughput $TP$ is defined as the total number of bytes in a frame transmitted over the air and collected at the receiver per-transmission; these include retransmission frames and headers (refer to Fig. 2 for a definition of $m$, $n$ and $N$). Let $n + m$ be the total transmission in time $t_T$, calculated in megabits or kilobits or bits per second, throughput is given as:

$$TP = \frac{(n+m)(N+1)}{t_T} \tag{1}$$

By contrast, goodput is a measure of the amount of useful data received per unit time:

$$GP = \frac{nN}{t_T} \tag{2}$$

This is the application level throughput experienced by the user, i.e. it only considers the successfully received bytes of the payload at the receiver – excluding protocol overhead and retransmitted frames. Furthermore, if any one part (segment) of a TCP/IP frame is missing after all retries have been exhausted, then the entire TCP/IP frame is discarded – including the other segments that have been received correctly. None of the good data from the partial TCP/IP frame will be counted towards goodput.

### 2.3 Segment Size and Maximum Retransmission

A simulation model was then set up to explore the relationship of the parameters above, specifically in deciding the most effective segment size and maximum retransmission ($r_{MAX}$) to use for a particular data bandwidth, error rate and simulation conditions (which, in general, match [28]). The evaluation begins by varying parameter $SS$ under several different BER scenarios, according to the parameters listed in Table 1.

**Table 1** Single-link simulation parameters

| Attribute | Value |
|---|---|
| Transmitted file size | 30 MB |
| Max. TCP-payload retransmission | 12x |
| TCP flavour | TCP New Reno |
| Channel data rate | 256 kbps |
| BER | $10^{-4}$, $10^{-5}$, $10^{-6}$ |

Figs. 3 to 6 plot the achieved goodput for segment sizes of 64, 128, 256, and 384 bytes and BERs of $10^{-4}$, $10^{-5}$ and $10^{-6}$ (as listed in Table 1), when the maximum number of segment retransmissions is limited to 0, 2, 3, 4 and 5 times before an error is flagged.

Overall, it can be seen that significant improvement of goodput is possible through use of a retransmission scenario in most of the BER conditions (i.e. performance is better when $r_{MAX} > 1$). Focusing on BER = $10^{-4}$, we can see that the best minimum value of maximum retransmission is at $r_{MAX} = 4\times$, since for lower values $r_{MAX}$ will yield reduced goodput. We note this to be an improvement on previously published work that did not specifically include the effects of the TCP protocol [28].

In terms of segment size, we find the lowest goodput occurs with a larger segment size of $SS = 384$ bytes. On the other hand, $SS = 128$ bytes, again at BER $=10^{-4}$, achieves maximum goodput, which we note is very similar to the value for $SS = 256$ bytes. The difference in overhead between $SS = 128$ and $SS = 256$ bytes (0.45%) is negligible. Another consideration, since bigger segments will be more likely to produce errors, we pick the smaller $SS = 128$ bytes as an optimal size from those tested.

It is interesting to note that for worse BERs, the system is still able to achieve reasonable goodput, but at the cost of $r_{MAX} > 4$; which we consider to be lowering the system efficiency, hence the result is not plotted here.

To conclude this section, a baseline SISO model has been developed with exploring the effect of the SAR system (which differs from previously published results [28] which did not explicitly model TCP). And as the remaining simulations in this paper, using this baseline system, we will standardise $r_{MAX} = 4$ and fix the effective segment size to 128 bytes.

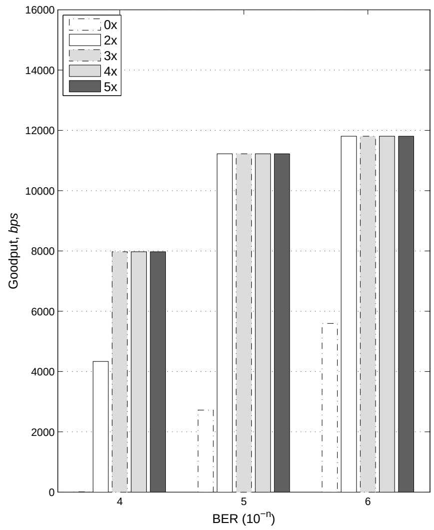


**Fig. 3** *Goodput* vs. $r_{MAX}$, with $SS = 64$.

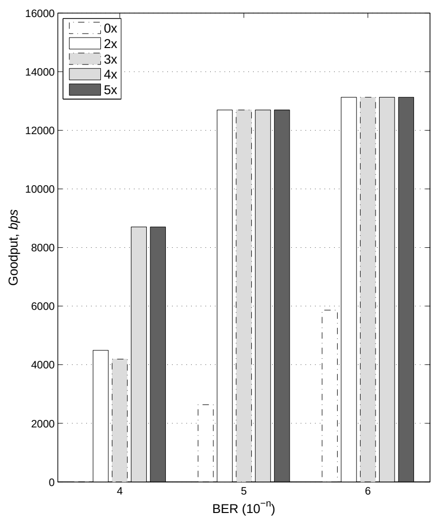


**Fig. 4** *Goodput* vs. $r_{MAX}$, with $SS = 128$.

## 3 Dual-link Model Implementing Switching for Different Channel Conditions

### 3.1 Model and Analysis

We now expand the model into a dual-antenna scenario instead of SISO. The nodes (X and Y) form a wireless communication system that has at least two physical sub-channels, with each sub-channel being a digital link with slightly differing characteristics (in this section we assume the existence of just two physical sub-channels, and that these are both made available to the data link layer, which is able to direct packets

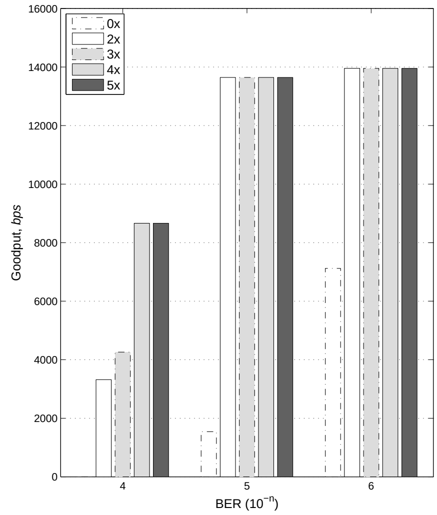


**Fig. 5** *Goodput* vs. $r_{MAX}$, with $SS = 256$.

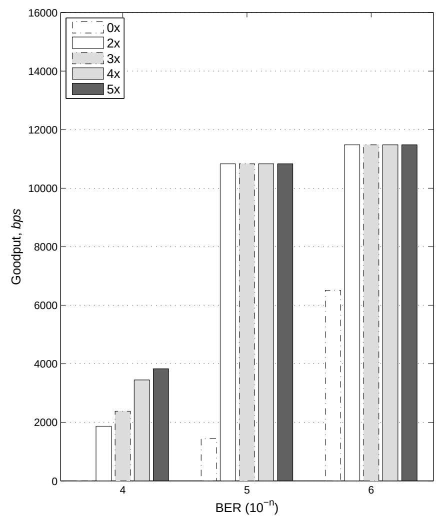


**Fig. 6** *Goodput* vs. $r_{MAX}$, with $SS = 384$.

into each sub-channel accordingly). Data transmission over each link will be subject to the time-varying and instantaneous BER and bandwidth experienced by that individual link[1].

In the scenario developed here, the dual-link model will switch packets down each sub-channel according

[1] While the overall system BER may be fixed at $10^{-4}$, each sub-channel will experience a BER that drifts up and down over time: the $10^{-4}$ figure refers only to the average error rate. For time periods that might range from a single segment transmission, up to (potentially) the lifetime of the deployed system, the BER over one sub-channel may be slightly lower, while the BER over the other sub-channel may be slightly higher.

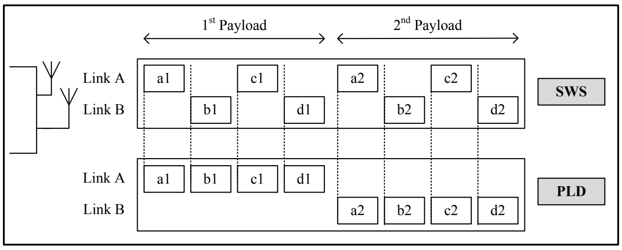


**Fig. 7** Switching methods illustrated with two TCP payloads each divided into four frames and transmitted on successive timeslots by over two physical links.

to experienced sub-channel BER. We propose and explore two control methods, named Segmented Window Switching (SWS), and Payload (PLD)-switching. Fig. 7 illustrates the difference between methods, showing two TCP data payloads being split into four segments and transmitted over two sub-channels. In SWS, segments are transmitted over available links in a round-robin fashion, whereas PLD uses higher-layer knowledge to force all parts of a TCP payload to be conveyed over the same link. The figure is a simplified diagram: in practice, with many TCP packets being transmitted, there would not be any idle slots. For all simulations, we use equal bandwidth links, but their instantaneous BER will be adjusted deliberately, to model the imbalance mentioned previously.

Real deployed systems would experience time-varying instantaneous BERs according to the statistical properties of the transmission over the particular wireless channel. Rather than model the temporal variations of one particular channel (which would limit its generality), we instead model the measurable effect of these time variations: i.e. the BER itself.

Thus our simulations use Monte Carlo methods to identify system performance at a particular state of imbalance. We devise a method of artificially adjusting the BER experienced by each channel separately. In other words, we deliberately force one link to be slightly 'bad' and another to be slightly 'good' during the simulation, but ensure that their average BER remains as specified and that the total number of errors experienced is consistent with this average[2].

First we define an average BER target:

$$BER_{target} = \frac{1}{B_T} \tag{3}$$

[2] Streams of information transmitted by a physical layer may be characterised by their average BER, but in fact they vary instantaneously in time while one is better the other may be worse. This is the real-world experience of various commercial and research-based wireless systems [28]. As MIMO coding can take advantage of this variability, the techniques discussed in this paper can similarly exploit it.

**Table 2** Dual-link $B_T$ distribution

| | $m=0$ | $m=1$ | ..... | $m=9$ |
|---|---|---|---|---|
| Link A | 1000 | 1100 | ..... | 1900 |
| Link B | 1000 | 900 | ..... | 100 |

This means, for every $B_T$ bits in $L$ communication channel, we will have $L$ error bits for every $LB_T$ bits. We then introduce a BER control variable as $\delta_{(m)} = m(0.1B_T)$ to unbalance the links. This changes BER experienced by one particular link relative to the other link, with a channel difference coefficient, $m = 0..9$. For two channels A and B, we have:

$$BER_A(m) = \frac{1}{B_T + \delta_{(m)}} \tag{4}$$

and

$$BER_B(m) = \frac{1}{B_T - \delta_{(m)}} \tag{5}$$

So, for an example of $B_T = 1000$, (4) and (5), may give us the BER distribution values for two channels as given in Table 2. It should be clear that the systems average BER (and total number of errors per unit time) in each scenario remain constant. The simulation was run with the parameters as established previously to evaluate goodput at different degrees of channel BER imbalance.

The effect can be seen clearly in Fig. 8 for both SWS and PLD methods for different average BERs. The extreme left of the plot, where there is no channel difference, simply shows the goodput available for an average BERs of $10^{-3}$, $10^{-4}$, and $10^{-5}$ respectively. Moving to the right, as one channel degrades and another improves correspondingly, this has a negative effect on goodput even though BER remains constant throughout.

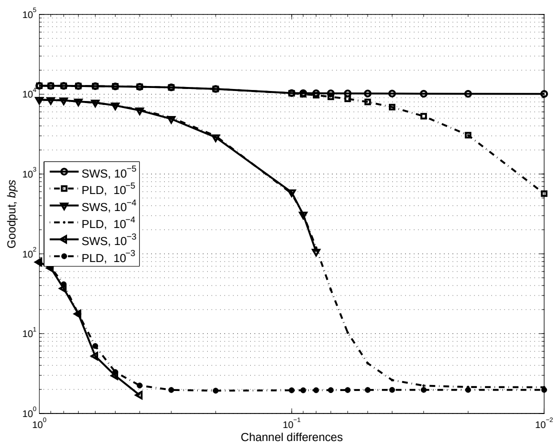


**Fig. 8** Goodput over dual-link model with different switching methods and channel BER imbalance.

At BERs of $10^{-3}$, it can be seen that as channel difference increases, SWS starts to fail early when the maximum simulation time limit is reached (i.e. the plotted line ends). By contrast, PLD does not give up as easily. At a more moderate BER of $10^{-4}$, SWS performs slightly better, but still PLD is best. However for improved BER conditions of $10^{-5}$, SWS maintains a steady goodput while PLD gradually decreases its performance. This can be explained by noting that SWS averages out error rates, whereas PLD allows half of the TCP packets to enjoy a reduced error rate, and half to enjoy an improved error rate. The consequence on goodput of an error in one TCP packet is severe, and should be avoided if at all possible. With poor average BER, SWS effectively corrupts almost all TCP packets (yet PLD allows half to pass); whereas with good average BERs, PLD concentrates errors on alternate TCP packets (yet SWS spreads the errors more evenly).

To conclude this section, we can see that for the given scenario, PLD may be better for lower average BER in a dual link arrangement. This is generally consistent with findings in [28], although this is the first time that the technique has been validated by taking into account the full TCP/IP protocol. Most interestingly, we can see clearly that the 'best' arrangement for payload switching will change depending upon the overall average BER. For practical wireless linking systems, these findings indicate that it may be advantageous to incorporate different payload switching mechanisms that are selected based upon current operating parameters.

### 3.2 Dual-link Retries Scenario

To investigate further we evaluate two different retry rules to handle segments received in error (or not received). This is shown diagrammatically in Fig 9.

The first rule forces retries to be re-transmitted over a different link (RDL) to the one on which they experienced an error. In other words, instead of re-sending a segment over the same sub-channel as before (i.e. the one which exhibited the error which caused the re-transmit), the segment will instead be transmitted over another sub-channel. Since this always means that when a packet is sent twice (i.e. resent once), it will have tried both the better and the worst sub-channels, so its errors are spread and overall its error rate is guaranteed to match the average BER.

The second rule utilised a random link (RRL) to re-transmit erroneous segments. This may involve either re-using the previous sub-channel link or using the other one, and basically allows the system to retransmit over whichever sub-channel is next available. TCP data will thus experience a small but definite BER imbalance depending upon which sub-channel it it using (especially so for the PLD scheme). By contrast to both of these methods, the original default retry mechanism behaviour was to always retransmit erroneous segments over the same link that had failed previously. Again, the remaining simulation parameters are those defined previously.

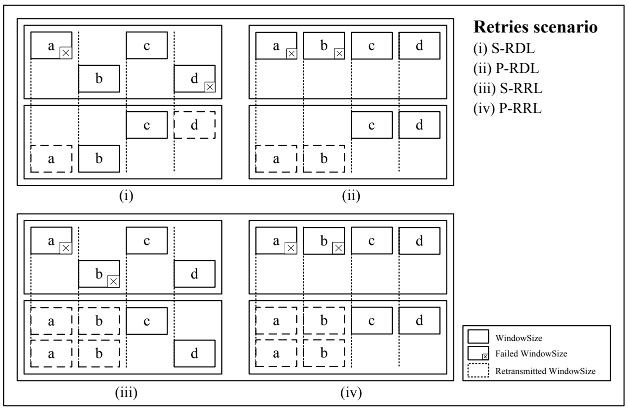


**Fig. 9** Two retry scenarios for the dual-link transmission model for both SWS and PLD switching mechanisms.

The graph in Fig. 10 presents the goodput results of all methods with respect to increasing channel difference behaviour. Though difficult to examine since the condition was for good BER (and thus few errors), we can see that at low channel differences, performance of all schemes is identical. However as channel difference increases, the first retry scenario for both switching methods (S-RDL and P-RDL) prevails slightly compared to the other combinations. Since BER is good, there is only a very slight performance improvement for both RDL (retransmit over the other link) and RRL (retransmit over random link) over the default behaviour as channel difference increases.

However as BER worsens, as plotted in Fig. 11, we see that besides S-RDL, P-RDL also performs slightly better as channel difference increases. The reason is that P-RDL, in good BER links, will preserve the transmission in that link ensuring a complete and good, TCP frame delivery. By contrast, S-RDL will retransmit and pass the segmented frame to the other link (round robin fashion), thus slightly reducing goodput – since the retransmitted segment is considered to have failed, therefore this has an effect on the goodput calculation.

From these results, we conclude that the RDL method is advantageous for systems with poorer BER. Overall, the P-RDL re-routing mechanism performs best.

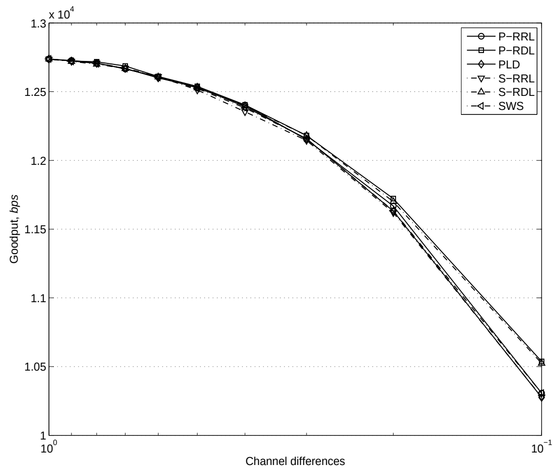


**Fig. 10** Comparison of dual-link performance methods with respect to $m$, for BER = $10^{-5}$.

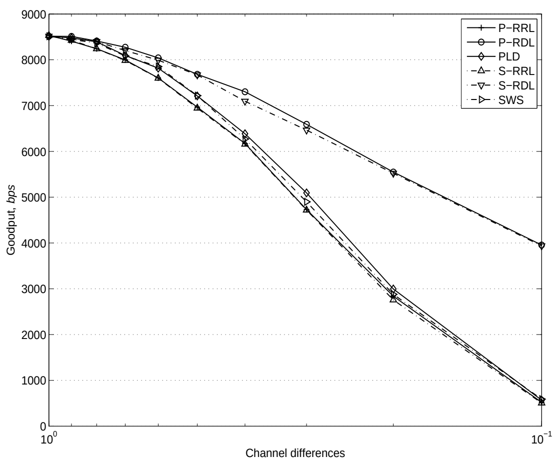


**Fig. 11** Comparison of dual-link performance methods with respect to $m$, for BER = $10^{-4}$.

## 4 Quad-Parallel Independent Link Model

We now extend the simulation to a more interesting model named the Quad-Parallel Independent Link Model (Quad-PILM). The methodology, packetization methods and retry strategies remain the same, along with the previous simulated conditions, except this time we explore performance over four parallel links, and that the channel degredation is now more one-sided.

Fig. 12 illustrates the switching and retransmission scenarios for Quad-PILM. Similar to the behaviour described in Section **??**, for S-RDL in subfigure (i) and P-RDL in subfigure (ii), when a segment fails, it will be re-sent to another one of three remaining links. Subfigures (iii) and (iv) involve resending over a random link, which could be the same link that just failed (although the probability of this happening is obviously lower than it is for the dual link system).

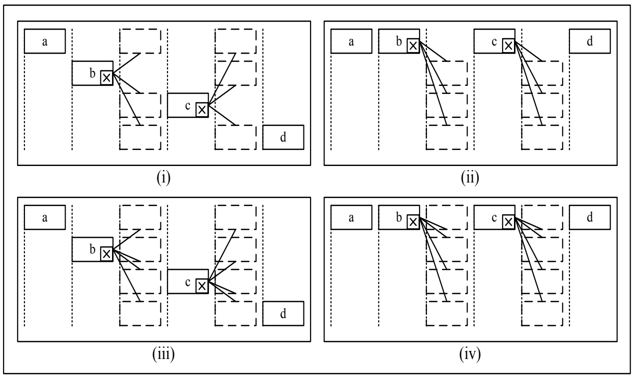


**Fig. 12** Quad-link switching methods and retries scenarios showing (i) S-RDL, (ii) P-RDL, (iii) S-RRL, and (iv) P-RRL.

Instead of a symmetrical channel difference between two links, we apply a 1–3 comparison with only one link (Link A) degrading, while the other three links progressively become better at 1/3 of the rate. Again, we maintain overall BER in the system.

Refering to (4) and (5), the simulation runs with the BER distribution for Link A, given as:

$$BER_A(n) = \frac{1}{nB_T} \tag{6}$$

and for the other three links will be :

$$BER_{B,C,D}(n) = \frac{L-1}{LB_T - BER_A(n)} \tag{7}$$

with $n$ indicating the channel difference distribution for quad-link model. The BER distribution can be seen in Table 3.

We again examine performance to highlight the difference due to changing switching method in PILM for poor BER conditions. Fig. 13, plotting goodput against channel difference distribution, clearly shows that for Quad-PILM, methods S-RDL and S-RRL both finally outperform the original SWS as the value exceeds 1. In this case, channel difference needs to be explored in both a negative and a positive direction because it is no longer symmetrical as was the case for dual-PILM.

Between the improvements of the switching methods, we found that when the system delivers the retry frames into a random link (RRL), it performs slightly

**Table 3** Quad-link $B_T$ distribution

| | $n = 0.8$ | $n = 0.9$ | $n = 1$ | ..... |
|---|---|---|---|---|
| Link A | 800 | 900 | 1000 | ..... |
| Link B | 1067 | 1033 | 1000 | ..... |
| Link C | 1067 | 1033 | 1000 | ..... |
| Link D | 1067 | 1033 | 1000 | ..... |

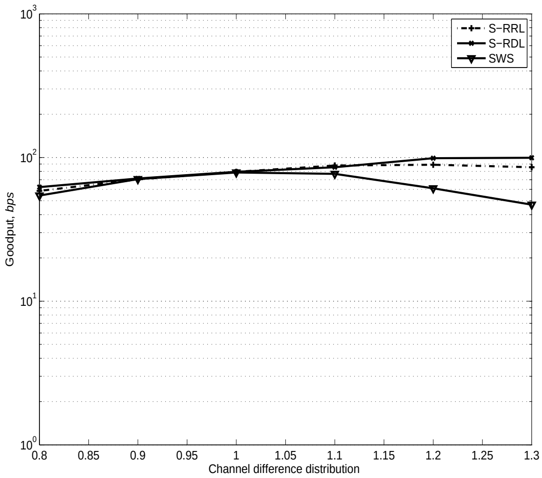


**Fig. 13** Quad-PILM: SWS-switching of all retry mechanisms.

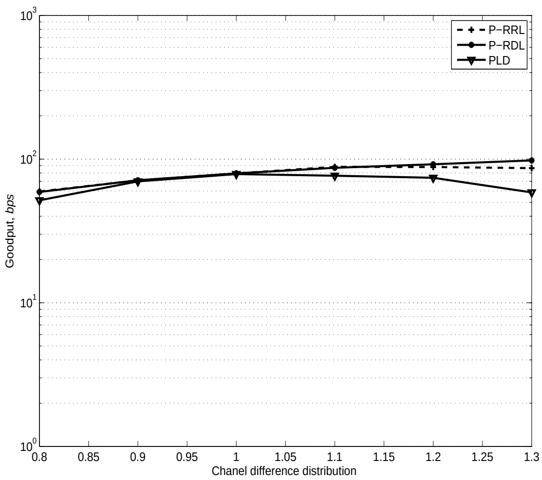


**Fig. 14** Quad-PILM: PLD-switching of all retry mechanisms.

worse than when retries are delivered into any three other links apart from the erroneous one (RDL). This can be explained as being due to the latter not re-transmitting the frame into the current failed link; while in RRL, there is always a chance the frame is delivered again over the failed link.

This trend also applies to the payload-switching methods (see Fig 14). When the distribution value > 1, the original PLD is the worst compared to all of the improved PLDs (P-RDL and P-RRL). The RDL scenario, as expected, will prevent the re-transmitting frame from being delivered into the same susceptible link, and thus will slightly surpass the RRL scenario.

Based upon the results and discussion above, we again conclude that the PLD method with the RDL

**Table 4** BER value for different QAM modulation at SNR = 15dB, from [8].

| QAM | Data rate (Kbps) | BER |
|---|---|---|
| 128 | 256 | $1{,}0.10^{-2}$ |
| 64 | 128 | $8{,}0.10^{-4}$ |
| 32 | 64 | $1{,}5.10^{-4}$ |
| 16 | 32 | $1{,}0.10^{-7}$ |

retry scenario is the best approach for the given conditions, and it will thus be adopted for the final scenario of this paper.

## 5 P-RDL based scenario for Quad-Link Model with Adaptive Modulation

### 5.1 RDLQ-PILM with no-AM scenario

In this final simulation, we address another novel simulation scheme: once that combines the P-RDL scenario in quad-link TCP-based point-to-point nodes, with an adaptive modulation concept.

This simulation is based on the different BER values experienced by various QAM modulations over a fixed link. The exact values are obviously system dependent, but for comparison purposes, we use BER values reported by Cho and Yoon [8] as shown in Table 4. We also assign an appropriate channel data rate for each QAM setting. In this system, instead of keeping the transmission rate constant, a switching mechanism is applied on a per-link basis, to attempt to better match a link to prevailing BER conditions. Thus, we allow a channel that experiences persistent errors (i.e. likely to be the worst channel) to reduce its modulation based upon some packet error criteria (i.e. again, it is cross-layer driven). When the adjustment is made, the BER and data rate for that link will change accordingly. For example, halving the data rate for each downwards step in modulation, and doubling the data rate for each upward step (although note that in the simulation graphs reproduced here only downward steps occurred since the system starts at the highest QAM level of 128 and then reduces this in response to error).

For benchmarking purposes, we first plot a normal result for each QAM modulation, without any AM adjustment in Fig. 15 for different values of $r_{MAX}$ (parameters reported in Section 2). As expected, some goodput improvement is seen in Fig. 15 for bigger retry values, indicating that given more chance to re-send the frame, there is a higher probability that it will be received correctly, thus improving goodput. This again validates the underlying simulation model now adapted to utilise AM.

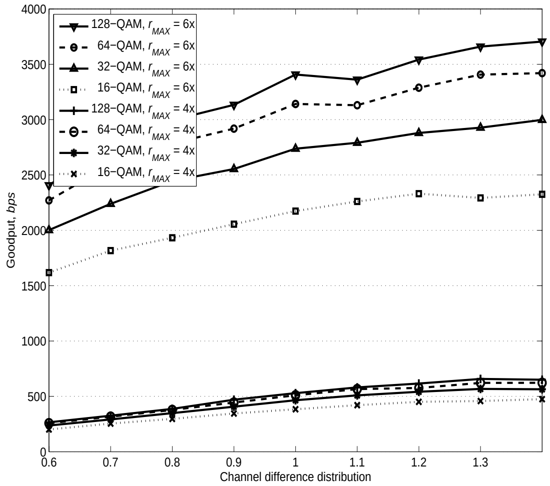


**Fig. 15** Quad-PILM with no-AM applied, for $r_{MAX} = 4$ and 6.

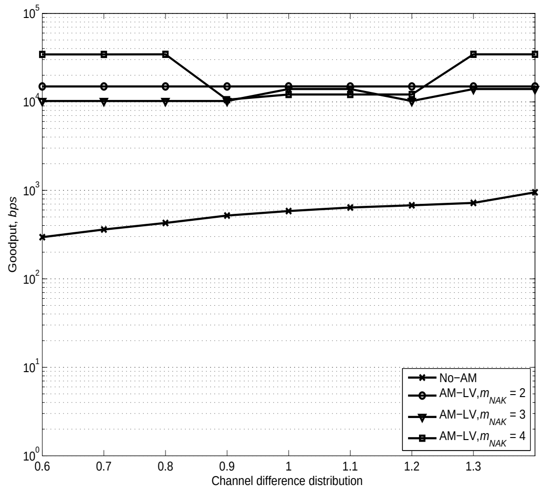


**Fig. 17** No-AM scenario vs. AM-LV with varying $m_{NAK}$.

## 5.2 AM-scenario with link value parameter

Two parameters are used in this scenario. First, a link value parameter ($LV$) is added to track the performance of each link individually in terms of PER. For a successfully transmitted segment over a particular link, we increase the value, and decrease it for an unsuccessful one. Another parameter called $NAK$ tracks the total number of retries at the sender (from the receiver's perspective, it is the total number of unsuccessfully received segments). When this value reaches a threshold $m_{NAK}$, we drop the bandwidth (data rate) by one step, and change the BER accordingly. This $NAK$ parameter will thus be used as an indicator of when to perform an AM adjustment.

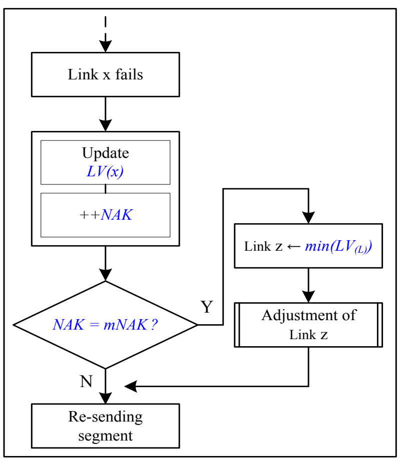


**Fig. 16** AM scenario with $NAK$ and $LV$; $m_{NAK}$ is counted for each link at node base.

The method – we name this AM-LV – can be seen in the flowchart of Fig. 16. Once the sender becomes aware that one of the links has a problem (un-ACKed segment), it will increment both $LV$ and $NAK$ values for that link. As transmission continues, if the $NAK$ value reaches the link's $m_{NAK}$ limit, then the system will inspect the $LV$ value for each link to find out which is the worst one. At this point, the system will decide which link has the worst link value, and subsequently the AM adjustment will be performed on that problematic link. Whilst this means that occasionally an incorrect link will be adjusted, it is a simple, portable and pragmatic method of tracking adjustments.

### *5.2.1 Discussion*

The results of this scenario are shown in Fig. 17. We note a significant goodput improvement for all $m_{NAK}$ values compared to a no-AM scenario, which shows that AM-LV works better under the tested conditions (i.e. for all $m_{NAK}$ values) ; with the adjustment scheme being well utilised.

For $m_{NAK} = 2$, a slight goodput improvement can be noted over $m_{NAK} = 3$. This is because the AM-LV system is working as a very early 'filter' to force the system to perform the adjustment before significant link retries occur. As a consequence, when any of the links experience error, the adjustment happens as soon as the $NAK$ value reaches 2, increasing the overall chance that segments arrive safely at the receiver. However, we found that this may be too premature to force an adjustment (it is acting upon insufficient information to make a correct choice in many cases), although significant goodput is achieved in the simulation. This does validate the premise that very simple AM adjustment triggered by segment errors over different links can provide an improvement, even when this is at the cost of temporarily reducing the aggregate maximum bandwidth.

When $m_{NAK}$ is set to 4, it can be seen that for moderate channel differences (in between 0.9 and 1.2) the probability of adjustment is lower, giving just a fair goodput. This is as expected since the channels have similar BER and little switching adjustment is needed.

It is appealing to notice that for more extreme channel differences, i.e. when the channel difference distribution is under 0.9 and bigger than 1.2, the goodput improves significantly compared to moderate channel differences, which indicates that the AM-LV scenario works perfectly. This is because, when the channel difference is large, adjustments will happen more often, thus increasing the probability of frames being received correctly at the receiver. A noticeable improvement in total goodput is shown in the graph, showing a step change in performance due to a thresholding effect on the experienced PER – and this indicates that the system is well tuned. The system as presented here has demonstrated that AM adjustment using a cross-layer methodology (i.e. counting PER) can improve performance.

Looking closely at the different $m_{NAK}$ values plot, we see more evidence that for large $m_{NAK}$, the system performs well using the AM-LV scheme. Bigger $m_{NAK}$ values obviously give the system an opportunity to do more error recovery before adjustments happen. Clearly, we can see this on the most extreme channel distribution (top lines), shown in Fig 18 which plots goodput against $m_{NAK}$ for several channel difference values. Note that the recoverer will not affect the goodput much compared to halving the data rate.

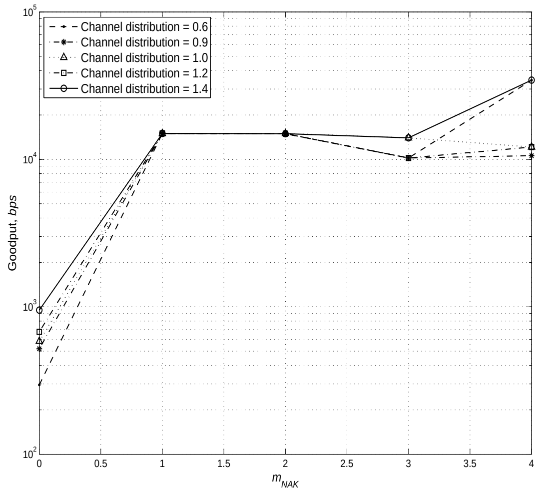


**Fig. 18** AM-LV scheme with different $m_{NAK}$ value.

## 6 Conclusion

This paper has investigated a cross-layer design solution for switching packets over different sub-channels for future MIMO wireless network systems. A simulation model was constructed from a foundation of a straightforward SISO link, and then extended through a dual link to a more advanced quad-link model – combined with packet re-routing mechanisms. We made use of goodput as our primary performance indicator for investigating various conditions such as delay, data rate, and bit error rate.

For dual- and quad-link arrangements, two switching mechanisms were proposed; segmented window switching (SWS) and payload (PLD), and evaluated. PLD was found to maintain performance as channel difference increases, compared to SWS. Two different retry scenarios were also implemented, in which retries are sent on a different link (RDL) and sent over a random link (RRL). For the scenario tested, RDL prevails compared to RRL. Combining both techniques above, we conclude that the payload-RDL (P-RDL) arrangement is the best for the poor BER scenarios explored in this paper.

Finally, we present an payload RDL-based quad-parallel independent transmission link model (Quad-PILM) combining the segment switching and retry mechanisms with an adaptive modulation scheme operating on individual sub-channel links. Several adjustment were carried on and show improvement of the model for both moderate and extreme channel differences. Significant goodput was achieved for those quad links using these pragmatic and low complexity methods, showing the system performs well with AM adjustment. Based on the finding, we conclude that our proposed AM model may also be beneficial when implemented in the multiple antenna systems deployment.

The application of the models described in this paper to a real world system would require an identification of the actual degree of BER imbalance exhibited in the real system on a per-segment basis. This can be easily measured in a real system, but also obtained from a mathematical analysis of any particular system (for example by evaluating outage probability).

## References

1. Alouini, M., & Goldsmith, A. J. (2000). Adaptive modulation over Nakagami fading channels. *IEEE Personal*

*Communications*, 13, 119–143.
2. Anantharaman V., Park S.-J., Sundaresan K., & Sivakumar R. (2004). TCP performance over mobile ad hoc networks: a quantitative study. *Journal of Wireless Communications and Mobile Computing*, 4, 203–222.
3. Anggadjaja, E., & McLoughlin, I. (2010). Point-to-point OMNeT++ based simulation of reliable transmission using realistic segmentation and reassembly with error control. In *2nd Int. Conf. Advances in Computing, Control and Telecommunication Technologies (ACT)*, 125–128.
4. Anggadjaja, E., McLoughlin, I., & Premkumar, A. B. (2013). TCP-based Multi Parallel Links Exploiting Packet Re-routing Mechanisms in Diverse Channel Condition. In *19th European Wireless Conference (EW 2013)*, Apr., to be published.
5. Chandran, K., Ragbunathan, S., & Venkatesan, S., Prakash, R. (1998). A feedback based scheme for improving TCP performance in ad-hoc wireless networks. In *Proc. 18th Int. Conf. Distributed Computing Syst.*, 472–479.
6. Chen, D., Ji, H., & Leung, V.C.M. (2012). Distributed Best-Relay Selection for Improving TCP Performance Over Cognitive Radio Networks: A Cross-Layer Design Approach. *IEEE Journal on Selected Areas in Communications*, 30(2), 315–322.
7. Cheng, R-S., Deng, D-J., Huang, Y-M., Huang, L., & Chao, H-C. (2009). Cross-layer TCP with bitmap error recovery scheme in wireless ad hoc networks. *Telecommunication Systems*, 44(1-2), 69–78.
8. Cho, K., & Yoon, D. (2002). On the general BER expression of one- and two-dimensional amplitude modulations. *IEEE Transactions on Communications*, 50(7), 1074–1080.
9. Choi, B., & Hanzo, L. (2003). Optimum mode-switching-assisted constant-power single- and multicarrier adaptive modulation. *IEEE Transactions on Vehicular Technology*, 52(3), 536– 560.
10. Deng, D-J., Cheng, R-S., Chang, H-J., Lin, H-T., & Chang, R-S. (2009). A cross-layer congestion and contention window control scheme for TCP performance improvement in wireless LANs. *Telecommunication Systems*, 42(1-2), 17–27.
11. Fu, Z., Luo, H., Zerfos P., Lu S. Zhang L., & Gerla M. (2005). The impact of multihop wireless channel on TCP performance. *IEEE Transactions on Mobile Computing*, 4(2), 209–221.
12. Goeckel, D. (1999). Adaptive coding for time-varying channels using outdated fading estimates. *IEEE Transactions on Communications*, 47(6), 844–855.
13. Goldsmith, A., & Chua, S. (1998). Adaptive coded modulation for fading channels. *IEEE Transactions on Communications*, 46(5), 595–602.
14. Goldsmith, A., Chua, S-.G. (1997). Variable-rate variable-power MQAM for fading channels. *IEEE Transactions on Communications*, 45(10), 1218–1230.
15. Gonzalez, C., Szczecinski, L., & Assa, S. (2004). Throughput maximization of ARQ transmission protocol employing adaptive modulation and coding. In *Telecommunications and Networking - ICT 2004, ser. Lecture Notes in Computer Science, J. de Souza, P. Dini, and P. Lorenz, Eds. Springer Berlin / Heidelberg* 3124, 33–44.
16. Harsini, J., & Lahouti, F. (2007). Queuing with adaptive modulation over MIMO wireless links for deadline constrained traffic: Cross-layer analysis and design. In *Proc IEEE Int. Conf. Communications (ICC 07)*, 235–240.
17. Huang, J., & Signell, S. (2009). On performance of adaptive modulation in MIMO systems using orthogonal space time block codes. *IEEE Transactions on Vehicular Technology*, 58(8), 4238–4247.
18. Huang, Y., Yang, X., Yang, S., Yu, W., & Fu, X. (2011). A cross-layer approach handling link asymmetry for wireless mesh access networks. *IEEE Transactions on Vehicular Technology*, 60(3), 1045–1058.
19. Jafari, A., & Mohammadi, A. (2009). A cross layer approach based on queuing and adaptive modulation for MIMO systems. *Telecommunication Systems*, 42(1-2), 85–96.
20. Kawadia, V., & Kumar, P. (2005). A cautionary perspective on cross-layer design. *IEEE Wireless Communications*, 12(1), 3–11.
21. Lin, Y., & Wong, V. (2009). Cross-layer design of MIMO-enabled WLANs with network utility maximization. *IEEE Transactions on Vehicular Technology*, 58(5), 2443–2456.
22. Liu, J., & Singh, S. (2001). ATCP: TCP for mobile ad hoc networks. *IEEE Journal on Selected Areas In Communications*, 19(7), 1300–1315.
23. Liu, Q., Zhou, S., & Giannakis, G. (2004). Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links. *IEEE Transactions on Wireless Communications*, 3(5), 1746–1755.
24. Liu, Q., Zhou, S., & Giannakis, G. (2005). Queuing with adaptive modulation and coding over wireless links: cross-layer analysis and design. *IEEE Transactions on Wireless Communications*, 4(3), 1142–1153.
25. Liu, Q., Zhou, S., & Giannakis, G. (2004). TCP performance in wireless access with adaptive modulation and coding. In *Proc. IEEE Int. Conf. Communications*, 7, 3989–3993.
26. Lohier, S., Doudane, Y.G., & Pujolle, G. (2007). Cross-Layer Loss Differentiation Algorithms to improve TCP Performance in WLANs. *Telecommunication System*, 36(1-3), 61–72.
27. Luo, C., Yu, F.R., Ji, H., & Leung, V.C.M. (2010). Cross-Layer Design for TCP Performance Improvement in Cognitive Radio Networks. *IEEE Transactions on Vehicular Technology*, 59(5), 2485–2495.
28. McLoughlin, I., & Sirisena, H. (2012). TCP/IP link layer error mitigation for MIMO wireless links. *Telecommunication Systems*, 50(3), 137–148.
29. Ming, J.L., & McLoughlin, I. (2006). Performance Investigation and Implementation of a Real-Time Adaptive MIMO-DFE System. In *10th IEEE Singapore International Conference on Communication Systems*, 1–5.
30. Ong, L.T., Shikh-Bahaei, M., & Chambers, J. (2008). Variable rate and variable power MQAM system based on bayesian bit error rate and channel estimation techniques. *IEEE Transactions on Communications*, 56(2), 177–182.
31. Park, J., & Lee, S. (2012). Distributed MIMO ad-hoc networks: Link scheduling, power allocation, and cooperative beamforming. *IEEE Transactions on Vehicular Technology*, 61(6), 2586–2598.
32. Pottie, G. (1995). System design choices in personal communications. *IEEE Personal Communications*, 2(5), 50–67.
33. Prakash, S., & McLoughlin, I. (2011). Analysis of adaptive modulation with antenna selection under channel prediction errors. In *Proc. Int. Conf. Comms. and Signal Process.*

34. Prakash, S., & McLoughlin, I. (2011). Effects of channel prediction for transmit antenna selection with maximal-ratio combining in Rayleigh fading. *IEEE Transactions on Vehicular Technology*, 60(6), 2555–2568.
35. Seong, T.C., & Goldsmith, A. (2001). Degrees of freedom in adaptive modulation: A unified view. *IEEE Transactions on Communications*, 49(9), 1561–1571.
36. Shakkottai, S., Rappaport, T., & Karlsson, P. (2003) Cross-layer design for wireless networks. *IEEE Communications Magazine*, 41(10), 74–80.
37. Sundaresan, K., Anantharaman, V., Hsieh, H.-Y., & Sivakumar, A. (2005). ATP: A reliable transport protocol for ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(6), 588–603.
38. Toledo, A., & Wang, X. (2006). TCP performance over wireless MIMO channels with ARQ and packet combining. *IEEE Transactions on Mobile Computing*, 5(3), 208–223.
39. Webb, W., & Steele, R. (1995). Variable rate QAM for mobile radio. *IEEE Transactions on Communications*, 43 (7), pp. 2223–2230.
40. Yu, X., B., Leung, S. H., Kuang, Q., & Yin, X. (2012). Performance analysis of variable-power adaptive modulation with outdated feedback for space-time coded MIMO systems. *IEEE Transactions on Vehicular Technology*, 61(6), 2613–2624.
41. Zhang, L., Snac, P., Lochin, E., & Diaz, M. (2013). A cross-layer architecture to improve mobile host rate performance and to solve unfairness problem in WLANs. *Telecommunication Systems*, 1–14.
42. Zhou, Z., Vucetic, B., Dohler, M., & Li, Y. (2005). MIMO systems with adaptive modulation. *IEEE Transactions on Vehicular Technology*, 54(5), 1828–1842.
43. Zhu, L., Yu, F., Ning B., & Tang, T. (2011). Cross-layer design for video transmissions in metro passenger information systems. *IEEE Transactions on Vehicular Technology*, 60(3), 1171–1181.