The effect of the RTS/CTS handshake on TCP

Stylianos Papanastasiou, Mohamed Ould-Khaoua
Department of Computing Science
University of Glasgow
Glasgow G12 8QQ, UK
{stelios,mohamed}@dcs.gla.ac.uk

Vassilis Charissis
Digital Design Studio
University of Glasgow
Glasgow G41 5BW, UK
v.charissis@gsa.ac.uk

Abstract

Previous research enquiries into TCP performance in MANETs have largely assumed that the 802.11 RTS/CTS mechanism remains in full effect throughout the connection time. In this work we evaluate the effects on TCP goodput of utilising other modes of operation for the 802.11 handshake whilst remaining compliant with the original specification. To identify the source of the goodput discrepancies amongst the different modes of operation we introduce two new metrics to characterise spatial connection along a TCP connection and then use them to account for TCP goodput in string and mesh topologies when using different RTS/CTS strategies.

1 Introduction

Mobile Ad hoc Networks (MANETs) have come under intense research focus in concert with continuing development of proposed standards by the corresponding working group within the IETF, which aims to deliver on the MANETs' potential of ubiquitous connectivity and impromptu communication ability.

As identified in previous work [1, 3], the issue of spatial contention in such multihop wireless networks, which is aggravated by the existence of hidden terminals, is caused by the inability of the MAC mechanism to properly coordinate transmissions. Specifically, in the case of unoptimised TCP agents, too many segments may be injected into the pipe at any one time and the 802.11 MAC mechanism may be unable to handle those numerous elements competing for transmission time. Due to TCP's popularity and proved robustness in wired networks, previous research efforts have suggested modifications in various layers in the protocol stack in order to improve TCP performance in MANETs in view of such issues [1, 3].

Notably, although the RTS/CTS exchange of the 802.11 protocol has been identified as a source of spatial contention [5], the MAC mechanism configuration as used in the literature with respect to spatial contention issues [1–4], ignores optimisations possible with respect to the RTS/CTS handshake, whilst remaining compliant with the 802.11 specification. In particular, the RTS/CTS handshake may be partially activated according to frame size or omitted altogether; the tradeoff involves balancing the associated overhead savings with foregoing protection against hidden terminal effects [5].

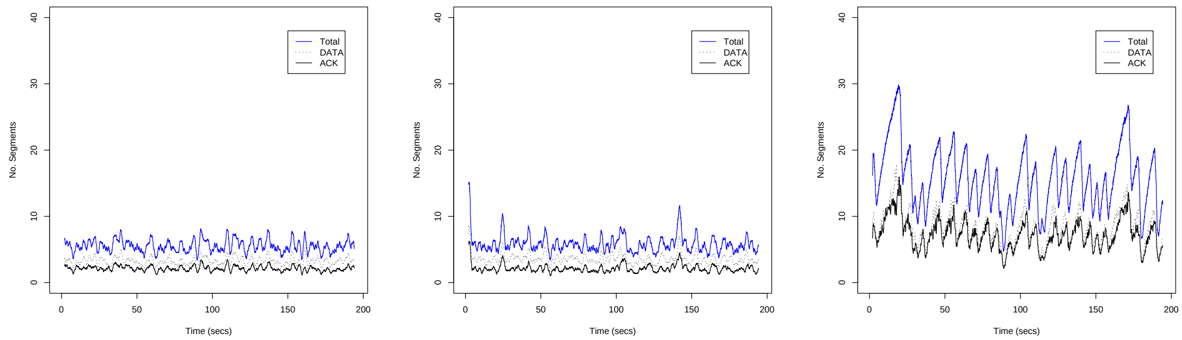
In this work, we consider the employment of different modes of the RTS/CTS handshake and measure their effect on TCP goodput. To this end, two new metrics are introduced to quantify the effects of spatial contention along an end-to-end TCP connection and identify the causes of the potential goodput difference. The results indicate that disabling the RTS/CTS mechanism may lead to significant goodput improvement in TCP, due to the decrease in spatial contention along the communications path.

The rest of the paper is organised as follows. The next section provides an enquiry into the effects on TCP of the different 802.11 RTS/CTS modes of operation in a particular string topology. Section 3 expands the enquiry to account for string topologies of variable length as well as multiple TCP agents. Section 4 further elaborates on the effect of different RTS/CTS handshake types in the case of multiple flows in a mesh topology. Finally, section 5 concludes the paper and offers suggestions for future work.

2 Measuring the effect of RTS/CTS on TCP

This section demonstrates with the aid of a simulation trace example the effect of disabling the RTS/CTS response for ACK segments only, or for both ACK and DATA segments. To this end, two new metrics on spatial contention are introduced and are, further, used in the rest of this paper.

In this case, a string topology of 5 nodes (4 hops) [1, 2] is considered. An FTP session is initiated at the beginning of the simulation between the two end-points of the string topology and continues for 120 secs at which point the sim-



(a) Full RTS/CTS (b) Partial RTS/CTS (c) No RTS/CTS

Figure 1. Segments in flight for different RTS/CTS strategies

ulation ends. The TCP agent used to carry the FTP traffic is NewReno with a packet size of 1460 bytes. The routing protocol used is AODV, the transceiver bandwidth is set to 2Mbps and the signal propagation model is Two-Ray Ground.

To *quantify* the effects on spatial contention of the RTS/CTS exchange two metrics are introduced. The first metric is the number of DATA frames dropped due to *repeated* failed MAC layer retransmissions. Note that the maximum number of MAC layer retries for a frame is set to 4 attempts as per the 802.11 specification.The payload of these frames is either a TCP DATA or ACK segment and so a series of repeated transmission failures leading to a drop is marked as either FAIL_{DATA} or FAIL_{ACK} respectively. The second metric is the number of failed RTS/CTS exchange *procedures*. It is worth mentioning that an RTS/CTS exchange is attempted several times by the MAC mechanism before it is marked as having failed. The required number of such attempts is 7 in the 802.11 specification. Such failed attempts are noted as $\text{FAIL}_{RTS/CTS}$ drops.

Further, the number of collisions is noted during the FTP transfer. Such collisions may be MAC frames containing TCP DATA, ACK or RTS/CTS payloads and so are marked COL_{DATA}, COL_{ACK} and $\text{COL}_{RTS/CTS}$. It should be specified that a high number of collisions indicate an *increasing degree* of spatial contention, whilst a high number of failed negotiations, either in TCP DATA or ACK transfers, denotes an *increasing inability* of the distributed MAC mechanism (the Distributed Coordination Function in 802.11 nomenclature) to effectively cope with spatial contention.

The simulation is run three times and with each iteration a different RTS/CTS strategy is employed. In the first round the RTS/CTS exchange is fully utilised in both TCP DATA and ACK segment exchanges. Subsequently, the RTS/CTS mechanism is only opted for "sufficiently large" TCP segments, i.e. only for DATA segments. Finally, in the third iteration, the RTS/CTS exchange is eliminated altogether. These three strategies are hereafter referred to as "Full RTS/CTS", "Partial RTS/CTS" and "No RTS/CTS" respectively.

Table 1. Frame collisions and drops for each RTS/CTS strategy

	Full RTS/CTS	**Partial RTS/CTS**	**No RTS/CTS**
$\textbf{COL}_{RTS/CTS}$	15047	8236 (-45%)	0 (-)
$\textbf{COL}_{DATA}$	79	240	3696
$\textbf{COL}_{ACK}$	67	6318	8197
$\textbf{COL}_{TOTAL}$	15193	14794	11893
$\textbf{FAIL}_{RTS/CTS}$	273	169 (-38%)	0 (-)
$\textbf{FAIL}_{DATA}$	167	169 (1%)	35 (-79%)
$\textbf{FAIL}_{ACK}$	106	81 (-23%)	310 (192%)
$\textbf{FAIL}_{TOTAL}$	546	419 (-23%)	345 (-36%)

Figure 1 depicts a 101-running average of the number of segments in flight throughout the simulation for the three different strategies. The graphs depict the number of DATA and ACK segments existing along the path through the simulation time and also shows their combined (aggregate) presence. A visual inspection of the figures reveals that disabling RTS/CTS altogether (Figure 1(c)) results in the TCP agent being able to maintain more segments in the pipe at any one time in both its receiving and sending aspects, i.e. both for DATA and ACK segments. In this particular case, on average, 15.74 segments exist in the pipe at any one time using the "No RTS/CTS" strategy which is significantly higher (by 183.76% and 167.4%) than the averages of 5.548 and 5.886 segments achieved by the "Full RTS/CTS" and "Partial RTS/CTS" strategies, respectively. The complete numerical set of averages for all three strate-

Table 2. Average segments in flight for each RTS/CTS strategy

	Full RTS/CTS	Partial RTS/CTS	No RTS/CTS
Mean$_{TCP}$	3.278	3.602 (9.8%)	8.337 (154.3%)
Mean$_{ACK}$	2.148	2.156 (0.3%)	7.313 (240.4%)
Mean$_{TOTAL}$	5.548	5.886 (6%)	15.74 (183.7%)

gies, categorised by type (DATA or ACK or both) is shown in Table 2. In this case, as for the rest of this section, table entries may be accompanied (where applicable) with a number in parenthesis denoting the numerical difference (percentage-wise) between the value examined for that particular strategy against the value achieved under the "Full RTS/CTS" strategy.

Overall, the "No RTS/CTS" strategy allows the MAC mechanism to be more efficient in coordinating the transmissions of a higher number of outstanding TCP segments. Table 1 contains the number of collisions and overall transmission failures for each RTS/CTS strategy. Note that no segment drops were recorded due to buffer overflows in the forwarding nodes.

The goodput results for each strategy indicate that the "Partial" and "no RTS/CTS" methods outperform the "Full RTS/CTS" technique by 6% and 19.8% respectively. Both the "Partial" and the "No RTS/CTS" strategies suffer from fewer frame collisions overall, but incur greater TCP and ACK frame collisions than the "Full RTS/CTS" method. Similarly, a greater number of RTS/CTS failed negotiations occur in the case of the "Full RTS/CTS" method as compared to the "Partial RTS/CTS" strategy ("No RTS/CTS" does not employ this exchange and thus records no such failures). This provides some indication of the overhead added spatial contention provided by the RTS/CTS exchange; collisions among segments increase and the MAC mechanism is unable to effectively coordinate transmissions (as denoted by the average number of TCP segments maintained in the pipe).

It is evident that the increased number of RTS/CTS transmissions leads to several RTS/CTS collisions which have a detrimental effect on goodput. An indication of this is the number of failed TCP and ACK transmissions for the "Full RTS/CTS" method. The number of TCP DATA and ACK segment drops cannot be solely attributed to MAC frame drops containing DATA or ACK payloads. The number of collisions of those are too few (79 and 67 segments respectively) to account for the number of frame drops (167 for TCP and 106 for ACK payloads). Hence the increased number of RTS/CTS exchange failures as compared to the other methods (273 in the case of "Full RTS/CTS" as opposed to 169 for the "Partial RTS/CTS" method) as well as the number of RTS/CTS collisions (45% greater than the "Partial" strategy) can largely account for the discrepancy in goodput (where the Full RTS/CTS method incurs performance hit of 6 and 19.8% compared to the Partial and No RTS/CTS techniques respectively). Note that an RTS/CTS exchange failure (that is 7 consecutive failed attempts) results in a TCP DATA or ACK segment drop corresponding to that transmission process. This fact explains why there are 273 segment drops recorded in the case of the "Full RTS/CTS" method and only 146 DATA and ACK bearing MAC frame collisions - many of the drops would be explained in terms of RTS/CTS exchange failure.

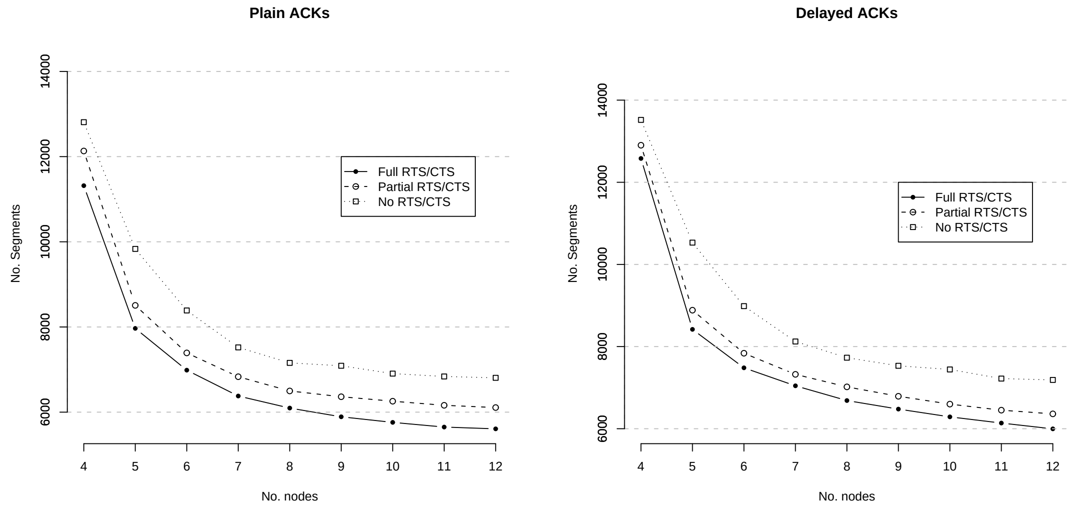
In summary, this section has demonstrated that in the special case of the 5-node string topology, the increased spatial contention due to the RTS/CTS exchange results in a goodput penalty for the TCP agent. Also, spatial contention has been quantified in this special case, in terms of frame types (RTS/CTS, TCP DATA and ACK frames). It has been shown that an increase in collisions during the RTS/CTS exchange leads to increased segment drops and a decrease in achieved goodput. This example has also shown that a more conservative approach in the generation of RTS/CTS segments, through the "Partial RTS/CTS" method, results in goodput improvement.

3 Evaluation on string topologies

This section expands on the scope of the previous examination to include a variety of string topologies and determine the goodput merit of the "Partial" and "No RTS/CTS" techniques. As in the previous setup, an FTP connection is set up between the endpoints of the string topology and runs throughout the simulation time (set to 300s) before results are collected. The string topologies considered are of size n, with $4 \leq n \leq 12$. For this enquiry TCP agents with both plain and delayed ACK responses have been evaluated. The metrics noted are achieved goodput, frame collisions and frame drops.

3.1 Goodput

Figure 2 shows the goodput results of a plain TCP receiver on topologies of increasing hop-count for each of the RTS/CTS strategies examined here. The achieved goodput regardless of the ACK strategy used decreases as the hop-count (number of nodes in the string topology) increases, hence altering the ACK strategy does not alter that TCP behavioural characteristic as noted in the case of the "Full RTS/CTS" mechanism in previous research work [1, 3]. However, notably, goodput increases both when the "Partial RTS/CTS" and "No RTS/CTS" techniques are in use. In the case of the "Partial RTS/CTS" mechanism the increase is in the range of 6-9% for plain ACKs (Figure 1(a)) and 3-6%



(a) Normal TCP Receiver (b) Delayed ACK TCP Receiver

Figure 2. Goodput against number of nodes in string topologies for a single TCP connection

for delayed ACKs (Figure 1(b)). The improvement in the case of disabling the RTS/CTS exchange is high and ranges within 17-23% for plain ACKs and 15-22% for delayed ACKs. As ACK-thinning techniques are employed, the effectiveness of both the "Partial" and "No RTS/CTS" strategies diminishes, most notably for the "Partial" RTS/CTS technique. This is because ACK-thinning alleviates some of the spatial contention and there is less scope for improvement by this MAC layer modification.

3.2 Collisions

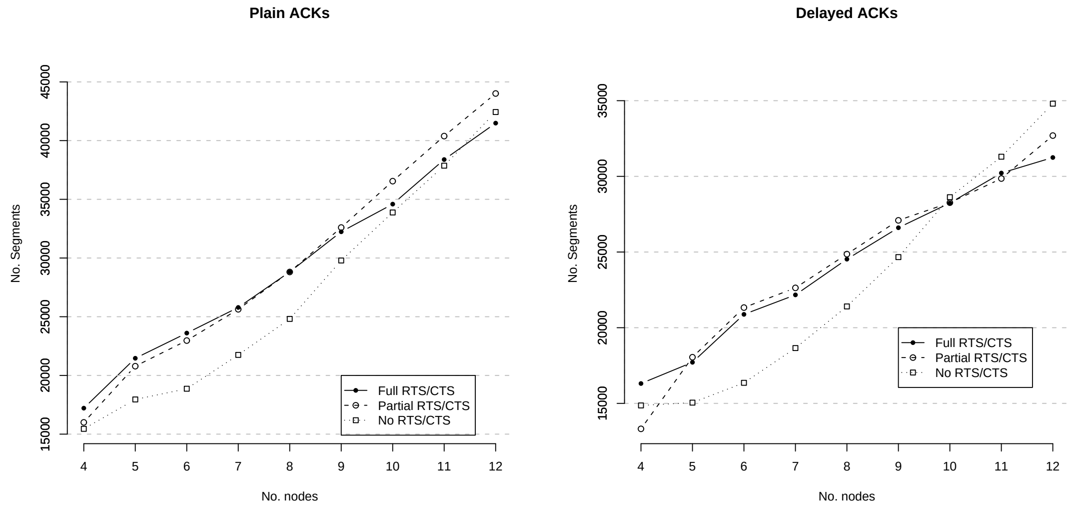
Figure 3 presents the number of total collisions recorded for each strategy in the same scenarios. In all the RTS/CTS techniques as the number of nodes in the string increases so does spatial contention (indicated by the increasing number of frame collisions). This fact is reflected on the declining goodput as the string topology length increases (Figure 2).

We have also noted the results for the above scenario in the case of multiple TCP connections among the end-points. In this case, the *aggregate* goodput is considered. and the subsequent results verify the observations made above in the case of 2 and 3 TCP connections. It is worthy of mention that the goodput advantage of both the "Partial" and "No RTS/CTS" methods against the "Full RTS/CTS" exchange strategy remains consistent as more connections are employed on string topologies of the length used here. In particular, Table 3 presents the goodput performance improvement noted by employing the two RTS/CTS strategies against the full RTS/CTS exchange for the two types of ACK response methods (plain and delayed ACKs).

Table 3. Range of goodput difference for each RTS/CTS strategy vs "Full RTS/CTS" for a variable number of TCP flows

ACK strategy	Range of goodput difference against "Full RTS/CTS"			
	2 TCP con.		3 TCP con.	
	PARTIAL	NO	PARTIAL	NO
Plain	7-8%	19-23%	6-7%	20-27%
Delayed	4-6%	19-22%	4-6%	19-24%
	4 TCP con.		5 TCP con.	
	PARTIAL	NO	PARTIAL	NO
Plain	5-8%	20-27%	6-8%	20-28%
Delayed	4-7%	20-28%	4-6%	21-27%
	6 TCP connections			
	PARTIAL		NO	
Plain	6-8%		21-28%	
Delayed	4-6%		22-28%	

For a single connection, generally, disabling RTS/CTS decreases the number of collisions compared to the other two strategies, but as the hop count increases, this trend does not hold consistently across methods, notably for the delayed ACKs technique for 9, 10 and 11 hops (Figure 3(b)) and the plain ACKs for 11 hops (Figure 3(a)). The overall results are presented in Figure 3. For more than one connection, disabling RTS/CTS consistently reduces the number of collisions throughout different hop counts (for 2 and 3 TCP connections). It can therefore be deduced that in these cases disabling the exchange leads to reduction in *spatial contention*, which is in turn reflected in the goodput results.



(a) Normal TCP receiver (b) Delayed ACK TCP receiver

Figure 3. Number of collisions against number of nodes in string topologies for a single TCP connection

3.3 Drops

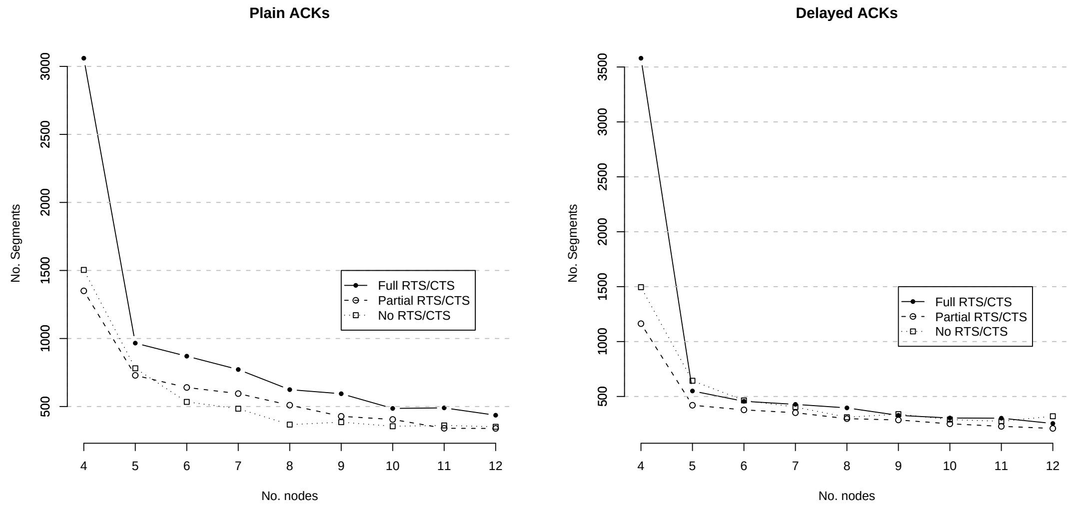
The last metric appraised for this section is the number of drops registered due to *repeated failed transmissions*. Figure 4 depicts the recorded number of total drops (including both TCP DATA and ACK segments) for a single TCP connection for the two ACK strategies. Both disabling RTS/CTS for ACK segments ("Partial RTS/CTS" method) and disabling the exchange completely ("No RTS/CTS") result in fewer drops than the "Full RTS/CTS" method by a margin of 16-55% for the former and 19-50% for the latter when no ACK optimisations are employed (Figure 4(a)). Notably, as ACK optimisation methods are utilised, the number of recorded drops decreases as the ability of the MAC layer to cope with spatial contention improves. When delayed ACKs are used, as shown in Figure 4(b) there are notably less drops in the case of employing "Partial RTS/CTS" compared to the "Full" method, but disabling RTS/CTS altogether leads to higher segment drops in many instances (string topologies of 5,6,9 and 12 nodes in Figure 4(b)). This discrepancy reveals that not all "final" drops have an equal impact on goodput, i.e. certain segment drops are more damaging to goodput than others (note that in all cases, the goodput record for the "Full RTS/CTS" technique is less than the one recorded for the other methods).

The above observation may be explained once the *nature* of TCP segment loss is examined. In the case of a 5-node string topology in Figure 4(b) a total of 643 segment losses are noted for the "No RTS/CTS" strategy and 550 for "Full RTS/CTS" method. It would therefore seem that the latter handles spatial contention better than the former; an observation not reflected in goodput as the "Full RTS/CTS" technique transfers, in total, 25% less segments. A breakdown of these losses reveals that most losses in the case of "No RTS/CTS" are ACK segments (608 out of 643) whilst the "Full RTS/CTS" records only 74 such losses. Further, the "No RTS/CTS" method experiences 35 TCP DATA drops as opposed to 201 for "Full RTS/CTS".

Intuitively, TCP DATA losses have a greater impact on goodput than ACK losses. Due to their cumulative nature, an ACK loss may be inconsequential if a subsequent ACK is received in time, i.e. before an RTO timeout is registered. For such an effect to occur, the average congestion window (cwnd) has to be sufficiently large so that several segments in the pipe would trigger ACK responses, some of which might be lost, but some of which would be received *in time so as not to trigger an RTO*. In the case of "No RTS/CTS" such a condition exists as the average value of cwnd is noted at 6 segments. Hence, it is the *nature* of segment loss which affects goodput in this case in tandem with the *amount* of segment loss. This statement holds true in all the other cases where the discrepancy occurs (6, 9, 12 node-string topologies).

Further, it should be noted that the relevant segment drop results for 2 and 3 TCP connections follow the same trend as those for a single connection, with the added note that the inverse relationship between total number of drops and total achieved goodput holds in every case. Due to lack of space, the relevant graphs are omitted here.

In conclusion, the results indicate a substantial improvement in goodput in using either technique in all cases, with disabling the RTS/CTS exchange having the greatest im-



(a) Normal TCP receiver

(b) Delayed ACK TCP receiver

Figure 4. Number of drops due to consecutive failed transmissions against number of nodes in string topologies for a single TCP connection

pact in every case. The next section examines the RTS/CTS handling methods with respect to intra-flow interference in a mesh topology as used previously in the literature [2, 3].

4 Evaluation on a mesh topology

The mesh topology has been commonly used in literature to examine spatial contention and its effect on TCP when multiple interfering flows are present [2, 3]. As compared to the previous section, this setting offers insight into the interaction of TCP with the MAC mechanisms in the case of moderate *inter-flow* interference. For the purposes of the following evaluation and discussion we consider a simulation setup involving a $5x5$ (25-node) mesh topology, where the horizontal and vertical distance of successive nodes is set to 200m; such a setup mirrors that of previous work [2]. Overall, two simulation scenarios are considered. First, three horizontal flows are active (one at every other row of the mesh) and with each flow starting at the first and ending at the last node of its respective row. This configuration offers *inter-flow spatial interference* alone; the flows do not share any common path but still interfere with each other due to the discrepancy between the interference and transmission ranges of their transceivers. The second simulation scenario involves three additional vertical flows (one at every other column in the mesh), with each flow starting at the first and ending at the last node of its respective column. This setup allows for both *buffer space* and *spatial* sharing between the flows as each flow shares its source and destination with another (thus flows interact and interfere with each other).

Table 4. Goodput results for mesh topology with 3 and 6 TCP flows

ACK Strategy	Goodput Achieved with 3 TCP Flows		
	Full RTS/CTS	Partial RTS/CTS	No RTS/CTS
Plain ACKs	15953	16683(5%)	18798(18%)
Delayed ACKs	16115	1676(4%)	20968(30%)
	Goodput Achieved with 6 TCP Flows		
Plain ACKs	16884	17879(6%)	23185(37%)
Delayed ACKs	16676	17171(3%)	21822(30%)

4.1 Goodput

Table 4 contains the goodput results for the plain and delayed ACK TCP agents for the different RTS/CTS methods, when 3 TCP flows are used. The values in parenthesis next to the numerical values for the “Partial” and “No RTS/CTS” methods indicate the performance improvement compared to the “Full RTS/CTS” method. For all three ACK strategies using an alternate strategy to the “Full RTS/CTS” exchange results in substantial improvement in goodput. As in the case of the string topologies, the “No RTS/CTS” method yields greater goodput improvement (18-30%) compared to the “Partial RTS/CTS” method (4-5%). In the case of 6 TCP flows, as shown also in Table 4, the same observation holds true. It can be deduced that in both cases, i.e. whether spatial or buffer contention at the forwarding nodes is evident or otherwise, the alternative RTS/CTS strategies are beneficial goodput-wise compared to the “Full RTS/CTS” paradigm.

4.2 Collisions

In the case of 3 TCP flows, it is noteworthy that the number of total collisions decreases for both the “Partial” and “No RTS/CTS” strategies, with the latter registering a more noteworthy decrease (12-35% as opposed to 1-3%). These improvements mirror the improvement noted in the case of multiple TCP connections in string topologies as discussed in the previous section. The equivalent results for the cross-traffic pattern of 6 TCP flows demonstrate show a similar trend (2-3% decrease for “Partial” vs 17-23% for “No RTS/CTS”). Overall, the reduction in collisions indicate that spatial contention is reduced in this case, particularly when the RTS/CTS mechanism is disabled.

4.3 Drops

As indicated previously, the number of drops due to repeated failed transmissions is a useful indicator of the ability of the MAC mechanism to deal with spatial contention, i.e. to effectively coordinate transmissions. For 3 TCP flows the number of such recorded drops is substantially decreased for both the RTS/CTS strategies. The percentage of reduction in drops is consistently high regardless of the ACK strategy used, with an exception in the case of delayed ACKs, where “No RTS/CTS” registers only a 6% decrease. However, it should be noted that in that case the vast majority of drops (1251 out of 1325) are frames bearing an ACK payload, which has a smaller impact in goodput than the loss of TCP DATA segments as shown in the previous section. Note that for the other two RTS/CTS techniques, that is the “Full” and “Partial RTS/CTS” methods, in the case of the delayed ACKs strategy, DATA-bearing frame losses severely dominate ACK-bearing ones. This fact is reflected in the recorded goodput for each method. When 6 TCP flows are employed the reduction in drops noted when using the RTS/CTS handshake alternatives as opposed to “Full RTS/CTS” is consistent across ACK strategies (13-14% for “Partial RTS/CTS” and 56-59% for “Full RTS/CTS”).

In summary, in the case of the mesh topology examined here, employing either the “Partial” or “No RTS/CTS” strategies has a positive impact in goodput as either method helps *alleviate* and *handle* spatial contention, compared to the “Full RTS/CTS” handshake. It has been shown that this observation remains valid regardless of the ACK strategy employed.

5 Conclusion

This work has investigated the impact of the 802.11-compliant RTS/CTS on TCP agents using plain and delayed ACKs. To facilitate the investigation two new metrics have been introduced, measuring both the amount of spatial contention on TCP data exchanges and the ability of the MAC mechanism to effectively coordinate transmissions (maximise spatial reuse). Using these metrics and with the aid of simulation trace analysis, alternative RTS/CTS functions (other than the full RTS/CTS exchange) have been shown to improve on spatial reuse in both string and mesh topologies.

For the future we plan to expand on the evaluation to include other ACK response strategies as suggested by recent MANET research work. Further we aim to include appraisal of the effectiveness of different RTS/CTS strategies on dynamic topologies using more diverse signal propagation models.

References

[1] K. Chen, Y. Xue, S. H. Shah, and K. Nahrstedt. Understanding bandwidth-delay product in mobile ad hoc networks. *Computer Communications*, 27(10):923–934, June 2004.

[2] R. de Oliveira and T. Braun. A Dynamic Adaptive Acknowledgment Strategy for TCP over Multihop Wireless Networks. In *Proceedings of Twenty-Fourth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, volume 3, March 2005.

[3] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla. The impact of multihop wireless channel on TCP throughput and loss. In *Proceedings of Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, volume 3, pages 1744–1753, March 2003.

[4] S. Kurkowski, T. Camp, and M. Colagrosso. Manet simulation studies: the incredibles. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(4):50–61, 2005.

[5] K. Xu, M. Gerla, and S. Bae. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks? In *Global Telecommunications Conference, 2002. GLOBECOM ’02*, volume 1, pages 72–76, November 2002.