# Adaptive Intelligent Middleware Architecture for Mobile Real-time Communications

Sarogini Grace Pease, Iain Phillips and Lin Guan

*Abstract*—**Provision of instantaneous, mobile and dependable communications in military and safety-critical scenarios must overcome certain wireless network issues: lack of reliable existing infrastructure, immutability of mission-critical protocols and detrimental wireless dynamics with contributing factors including hidden transmitters and fading channels. Benchmarked approaches do not fully meet these challenges, due to reliance on addressing Quality of Service (QoS) at a layer-specific level rather than taking a system of systems approach. This paper presents an adaptive middleware methodology to provide timely MANET communications through predictive selection and dynamic contention reduction, without invasive protocol modification. This is done using ROAM, the proposed, novel Real-time Optimised Ad hoc Middleware based architecture. Extensive simulation results demonstrate the adaptability and scalability of the architecture as well as capability to bound maximum delay, jitter and packet loss in complex and dynamic MANETs.**

*Index Terms*—**Communication Networks, Mobile Ad hoc Networks, Military Communication, Real-time Systems, Vehicular Ad hoc Networks, Cross Layer Design, Middleware**

## I. Introduction

**M**ANETs are self-organising infrastructureless networks and MANET protocols work on a self-configuring basis to adaptively create network paths, without centralised management. This makes them ideal for media streaming and communications in military communication scenarios. However, shared channels and time-varying, complex topologies create non-deterministic layer-1 and 2 dynamics of high loss and variable end-to-end (E2E) delay [1], [2]. Movement, signal jamming and system failure also cause loss of network paths.

This paper therefore proposes and validates three novel contributions. (1) ROAM is a middleware entity to abstract information from the protocol stack using application programming interfaces (APIs). ROAM implements tuning optimisers that respond to dynamic network conditions. (2) A horizontal handoff optimiser responds to time-varying link quality to ensure optimal and most robust channel usage. (3) A distributed contention reduction optimiser reduces channel contention and related delay, in response to detection of the presence of a hidden terminal.

## II. Motivation

In wired networks, increasing capacity to match demand and predetermined resource management and provisioning are used

S. Pease is with the Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, Leicestershire, LE11 3TU UK e-mail: s.g.pease@lboro.ac.uk

I. Phillips and L. Guan are with the Department of Computer Science, Loughborough University, Loughborough, Leicestershire, LE11 3TU UK

to uphold fixed delivery deadlines and zero loss guarantees [3]. In dynamic networks, interference and contention for limited bandwidth mean absolute guarantees cannot be provided to real-time applications. Real-time processes use global physical completion time constraints, or deadlines, to manage their resources. Thus real-time applications are delay- and jitter-sensitive, in contrast to non-real-time (NRT) applications where resource management is not time or constraint driven.

Safety-criticality in a military scenario means that timely packet delivery can also influence both the usefulness of data and safety of the system. Safety-critical real-time wireless applications can operate within the remit of inelastic soft real-time [4] (ISRT) that tolerates intermittent loss, E2E delay and jitter within acceptable and guaranteed bounds. In this way a high level of deadline achievement can be stipulated, without the requirement for scheduling to a fixed arrival time.

Overloading a link results in QoS deterioration and congestion. Queueing and contention delay are the most significant components of E2E delay. The former is dependent on the relationship between packet arrival rate and service time at the link and the latter on error rates and recovery mechanism used. To avoid congestion collapse, delay-inducing methods such as exponential backoff with IEEE 802.11 CSMA collision avoidance (CSMA/CA, or virtual carrier sense) and TCP window reduction are used.

Sacrificing TCP's reliability and congestion control for bounded delay, timing sensitive applications therefore use UDP or Real-time Transport Protocol (RTP) at the transport layer. CSMA/CA is widely used to avoid overloading a busy channel and to avoid hidden terminals, by handshaking RTS/CTS packets. However, global signalling can reduce wireless network performance as a result of interference errors or even cause blocking of multiple nodes and congestion [5].

These and other benchmarked protocols address wireless delay requirements with active or permitted packet dropping. This is impractical when loss-recovery mechanisms are implemented at other layers and for ISRT support where timeliness cannot be at the expense of unbounded E2E loss. Taking a system of systems approach, cross-layer design tailors responses to layer-1 and 2 conditions and specific causes of packet losses and errors, improving application performance [6], [7], [8].

Alternative congestion solutions can then prevent over-subscription of resources and related loss, backoff and retransmission, by adjusting transmission rate (TR) to individual link capacity. The authors in [9] performed this with lower layer signalling combined with resource reservation, demonstrating increased supported data rates. Merging MAC and routing with the use of virtual links, to avoid processing delays between
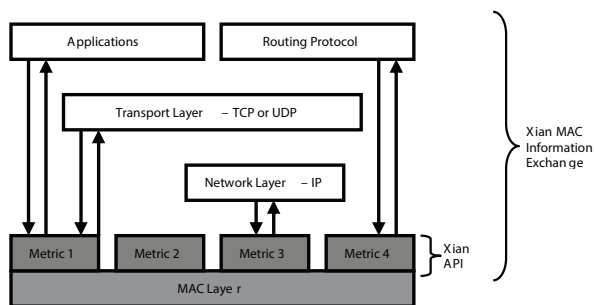
Fig. 1: Xian Architecture [12]

layers, has been implemented in [10]. The link layer was then responsible for selection of the next hop and re-encapsulation of packets. This resulted in a 7-10% throughput improvement and 50% reduction in processing time.

In [11] the MAC layer itself selected and prioritised paths based on link quality and route information. Shortcut paths and novel control messages were used to prevent transmission redundancy. In moving typical network functionality to the lower layers the modularity and re-usability of such an approach is low. Additionally, such approaches cannot provide bounded delay and jitter guarantees alone. This is because contention control can only take place once an E2E path has been selected as bandwidth and delay are dependent on channel quality and node capacity.

MANET routing protocols function on a distributed basis, letting each node select the next hop from among its neighbours, referring to QoS requirements and available resources. Reactive MANET routing (such as with DSR, AODV and OSPF) responds rapidly to topological change, lowering the impact of routing on delay. These protocols routinely select paths to remove loops and with minimum hop count. The latter is a metric that can be used instantaneously but that does not always select the most robust link or that with the least delay.

Alternatively, link metrics such as delay, Expected Transmission Count (ETX), Medium Time Metric or Worst Case Execution Time can be implemented. However, node mobility, link breakages [13] and the underlying processes of the MAC layer, such as repeated backoff, complicate and invalidate responses governed by these E2E metrics. Thus, probing instantaneous signal strength with dummy packets has been increasingly implemented. Using these to dynamically induce handoff between links of the same technology has been shown to reduce packet loss [14].

Withholding internal layer parameters from other layers facilitates fast development of interoperable systems and this paradigm can be preserved with parameter monitoring and tuning, managed by external cross-layer middleware [14], [15]. Current proposals have conceptualised but not fully validated non-intrusive middleware that uses API access to protocol data structures [16], [17], [18], [19], [20], [21].

The proof of concept for ECLAIR [16] has shown that TCP modifications and cross-layer API can be implemented, however they have not been validated together as a single approach. ECLAIR introduces tuning layers that enable
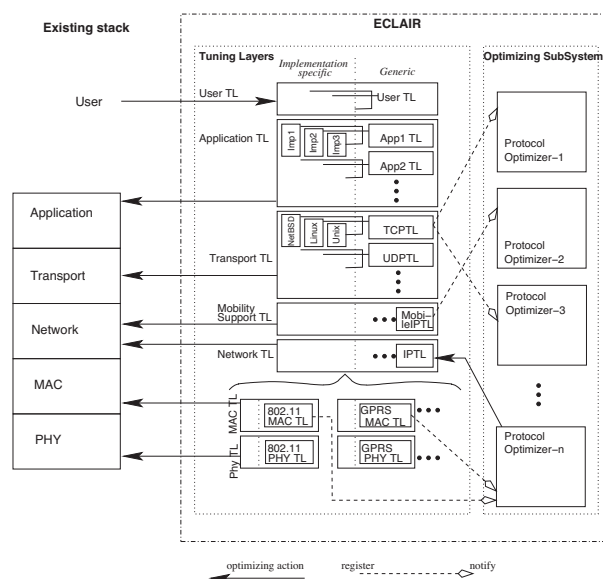


Fig. 2: The ECLAIR Architecture [24]

memory sharing with protocol data structures, to support and control communication between the middleware and protocols (Fig. 2). This architecture enables adaptable rapid prototyping and generic support to contemporary and legacy network protocols. Additionally, avoiding interruption of protocol processes ensures lower overheads and better packet-timeliness guarantees and, in optimising the stack from a single, external location, signalling loop errors are avoided.

The authors in [22] suggested a novel architecture that maintains global network status information and uses this to select and modify protocol parameters on a network-wide basis. The architecture proposes the use of CTS packets to piggyback channel quality estimates from the receiver, triggering optimisation along the path to the sender. However, such an approach would impact on E2E delay when processing cross-layer parameters in forwarding nodes [23], reducing the capability to provide bounded delay guarantees.

XIAN [18] is the first full Linux kernel and testbed implementation of an optimising middleware (Fig. 1). The authors demonstrated the capability of XIAN to broadcast ETX information to neighbouring nodes in order to improve routing decisions. However, the performance of the ETX metric has not been validated. Additionally, a requirement still exists to meet real-time requirements both through appropriate middleware design and optimising functionality.

Consequently we have combined a novel middleware architecture with selective protocol-stack interaction, empowered with optimising functionality to assure real-time applications that delay and jitter will be within acceptable and guaranteed bounds. ROAM has been designed to seamlessly support heterogeneous contemporary and immutable, safety-certified systems without imposing novel modifications on protocols or complex stack or interlayer interactions. ROAM uses generic API to abstract performance information held in protocol layer parameters. ROAM then uses API access above layer-2 to tune parameter data structures in an adaptive and scalable manner,

providing responsiveness to dynamic network conditions.

This system is characterised by improvement of the parameter abstraction functionality to access, but not exploit lower layer parameter data structures. This allows ROAM to maintain transparency and interoperability by avoiding firmware or hardware modification requirements. ROAM can support multiple tuning functions and two original optimisers are proposed for the management of link selection under node mobility and load control under hidden terminal contention conditions.

The first optimiser avoids sub-optimal link use when resource conditions are reduced, through accessing lower layer information and tuning the network and application layers. This optimiser also prevents the use of links that are not robust, due to high node speed or highly variable node mobility. Link selection without reference to channel conditions can result in localised increases in packet loss, jitter and delay as packets are repeatedly transmitted over fading links.

The Mobility Response optimiser collates information intercepted on neighbouring nodes to provide an early identification of link fading and institute rapid, controlled handoff. The need for complex parameter computation or exchanges is eliminated as the optimiser uses a relative comparison between optimal and sub-optimal paths. Executing concurrently with the stack, the optimiser does not pre-empt routing, selecting the optimal next hop only when this node appears within the routing table. Packets passing the MAC layer are transparently monitored to ensure that optimal links are successfully selected.

The second is a Distributed Contention Response optimiser that responds to detection of the presence of a hidden terminal. When hidden terminals contend for the same channel, each node will overhear the ACKs sent by the mutual forwarding node. A common ACK rate, combined with deteriorating performance at the MAC layer (increasing retransmissions, queue length and path delay; in spite of high link received signal strength) can be exploited by the middleware that also accesses information at the routing and application layers.

Rather than providing the continually changing response of approaches such as TCP, the ROAM optimiser incites a short term tuning of application settings in order to reduce pressure on the queue and link, and accordingly bound queueing and contention delays. The optimisation relies on minimal control packet exchange, does not require interaction of any of the protocols and requires no MAC or network layer cooperation.

MANET and real-time application performance varies, depending on factors occurring at layers 1 and 2 and the characteristics of the application. Thus the feasibility of the architecture has been validated in scenarios that demonstrate independence from the causes of network dynamics: application type and transmission setting and MANET conditions, such as number of competing sources, mobility and topology. Through compensation for changes in resource availability, ROAM provides better performance, in the form of bounded maximum delay and jitter and reduced packet loss guarantees, than can be provided by the unoptimised MANET protocol stack.

The rest of the paper is organised in the following way: Section III presents two performance scoping experiments

that have been conducted into the effects of node mobility and contention for a shared link on real-time performance. In Section IV, the cross-layer architecture designed specifically to support a safety critical protocol stack and used to meet the requirements of real-time applications in MANETs is described. Details of the middleware-implemented Mobility Response Optimiser, including the callback functionality, layer-specific API structure, and parameters optimised in response to a receding next hop, are presented in Section V. This is followed by a comprehensive validation of the optimiser under variation in traffic load, source number and encoding, topology and node speed. Section VI presents the structure and functionality of a Distributed Contention Response Optimiser, enabling a tuning response to the presence of hidden terminals. This section includes rigorous validation of the optimiser to demonstrate adaptability and scalability in complex and dynamic MANETs. Finally, Section VII presents our conclusions on the design, including discussion of limitations and future work.

## III. Background

MANETs are characterised by node mobility and wireless channel contention, that induce violation of real-time QoS guarantees. Neighbour and environmental interference, signal fading and attenuation trigger non-determinism in supported data-rates and link capacities. Thus ISRT provisioning must dynamically provide bounded delay, jitter and packet loss ratio (PLR). Our performance analysis, in this paper, is based on these three key metrics essential to the provision of real-time performance. Additionally, we investigate goodput, which is also necessary in determining the quality of the service provided to the application. This section analyses the impact of the aforementioned MANET characteristics on these performance metrics.

Based on a number of considerations, we chose to use network simulation for investigating both MANET protocol and cross-layer middleware performance as it supports the appraisal of a cross-layer approach without the overheads of a real-world implementation. The complexity of cross-layer design can be fully represented in a network simulator. While analytical models support performance analysis, protocol and middleware complexity and distributed device implementation would need to be simplified for investigation using such models [25]. However, investigation of sensitivity to parameters requires simulated models to be tested in a large number of scenarios. Therefore the proposed framework has been rigorously evaluated against varying application type and transmission setting, mobility and topology conditions in stochastically varying simulation runs.

Simulation with ns2-MIRACLE provided an ideal option for analysis of these military based scenarios with high datarates and large distances between communicating vehicles [14]. In a real world implementation it is not often possible to develop a repeatable testbed experiment with inter-nodal distances of hundreds of metres or high node speeds. ns2-MIRACLE modularises ns-2 protocol classes into individual layers with more accurate lower-layer protocol functioning, such as multi-rate MAC, seen in most IEEE 802.11 WLAN

(a) Mobile Source: Packet Loss

(b) Mobile Source: E2E CBR Goodput

(c) Mobile Source: E2E Delay

(d) Mobile Source: Next Hop Selected

(e) Contending Sources: E2E CBR Goodput

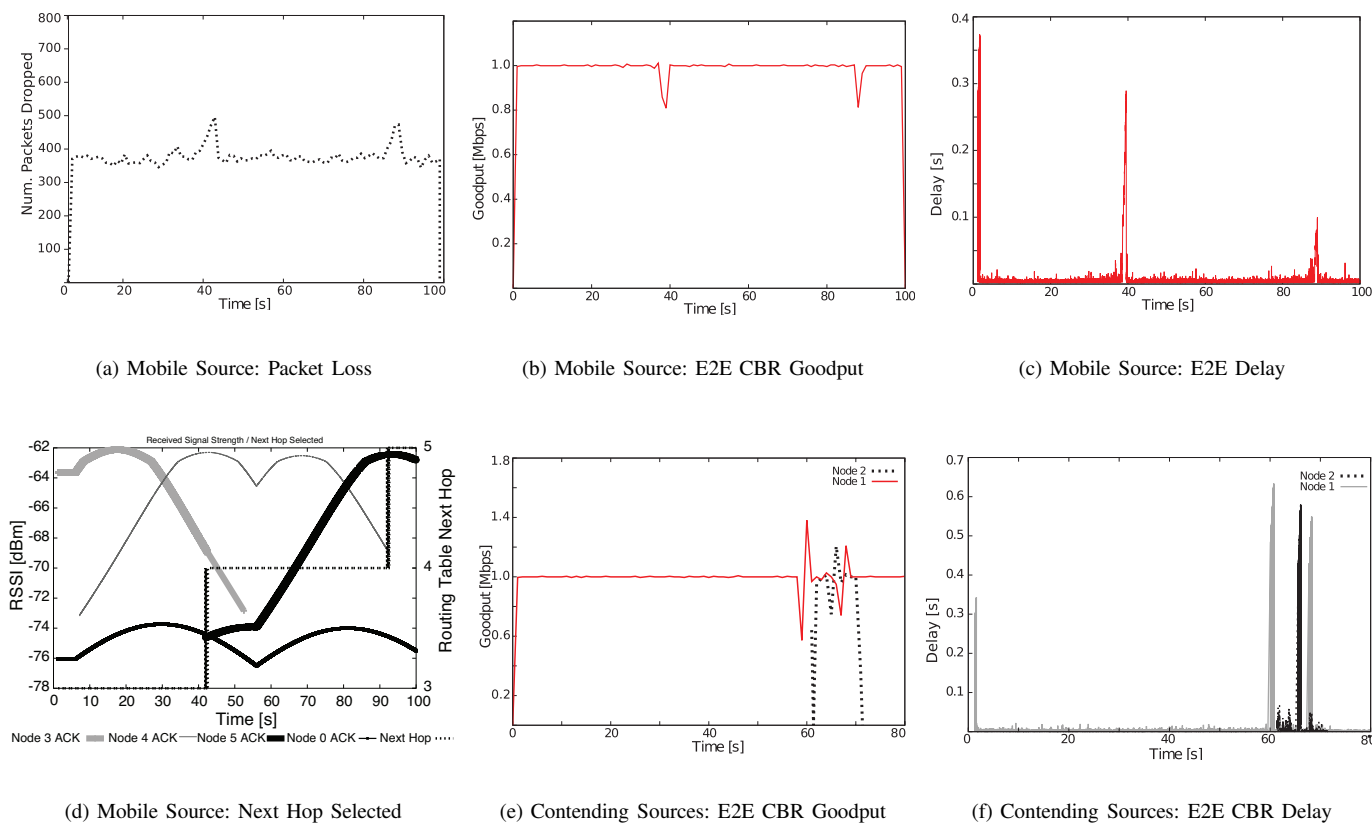(f) Contending Sources: E2E CBR Delay

Fig. 3: Mobile and Contending MANET Sources

cards. It extended ns-2 propagation models to include analytically derived interference bit-error models at layer-1 and packet error models at layer-2, as well as a more sophisticated two-ray ground propagation model [26]. As one of a small set of network simulators designed to simulate asynchronous middleware messaging, a repeatable appraisal of a complex cross-layer approach under the influence of scalable mobile communications could be conducted in this paper.

### A. Ad Hoc QoS: Mobility

The impact of node mobility on real-time QoS was investigated with a single CBR source, node 1 (N1) that orbited an association of five nodes and a receiver, node 0 (N0). Fig. 3(c) shows E2E delay from N1 to the receiver, N0. Reactive ad hoc routing has a high initial setup cost, as the E2E path is flooded with RREQ packets and RREPs are returned on the best path from the receiver.

Fig. 3(d) demonstrates the received signal strength (RSSI) at N1 from each of the forwarding INs and the receiver, and the next hop selected by N1. The mobility pattern meant that a new next hop was available approximately every 50s as N1 orbited the MANET. While RSSI from N4 was higher than N3 at approximately 30s, N1 changed to the more robust path after 40s and switched between N3 and N4.

The IEEE 802.11 auto-fallback, or multirate mechanism steps down MAC TR if noise on a channel increases. As a result, goodput dropped as the path to N3 began to fade, packet

errors increased and routing repeatedly selected the old path (fig. 3(b)). Poor path selection created E2E delays similar to those of initial path setup and peak loss occurred due to use of the incoherent link.

The results highlight the time lags between robust path identification and use, and the impact of oblivious path discovery mechanisms on network performance.

### B. Ad Hoc QoS: Contention

A worst case situation of channel contention, the hidden MANET terminal, was investigated. Two CBR sources, N1 and N2 travelled towards the receiver (N0) and hidden terminals between 60-70s. Fig. 3(e) shows that with a low, 1Mbps traffic rate, N1 goodput dropped by up to 50% during this period. Packets are increasingly buffered when channel quality is poor and MAC rate increases when the channel is free. Goodput bursts then appear as the queue is allowed to drain. The impact on E2E delay was greater than with path setup and maintenance.

This section demonstrates some transmission setting and military MANET configuration scenarios that result in rapid increases in packet loss and E2E delay. Investigation has highlighted that a mobility response and control of contention are promising approaches to the provision of bounded PLR, E2E delay and jitter: requirements of inelastic soft real-time applications.
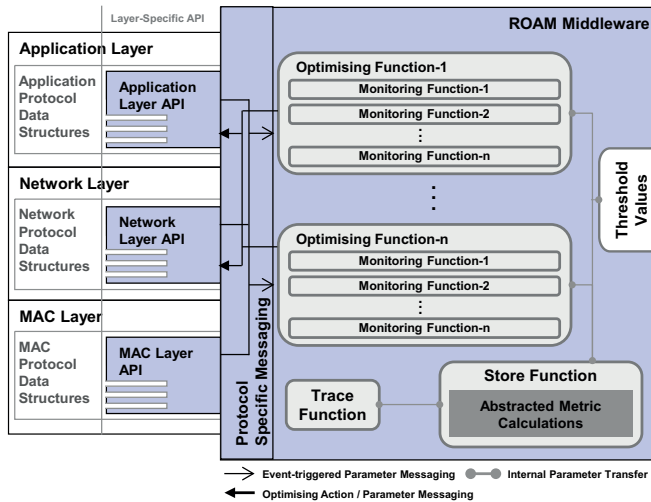
Fig. 4: ROAM Architecture Overview

## IV. ROAM ARCHITECTURE

Fig. 4 gives an overview of the proposed Real-time Optimised Ad hoc Middleware (ROAM) architecture, consisting of the middleware, multiple APIs and associated cross-layer messages. It is designed to support safety-certified real-time protocols through minimal imposition on and concurrent execution with the protocol stack.

Instead of modifying protocols, ROAM abstracts and returns parameters to layer-specific APIs, to which protocol data structures can read and write. The APIs can be exported to any MANET protocol. This does increase the complexity of the protocol stack as parameters are frequently abstracted by the API. Abstraction is concurrent with execution of the protocol stack, thus providing performance improvement while allowing protocols to function as intended, without the need to modify radio firmware or hardware. This is important in contexts where such components are safety certified and supports extensibility. ROAM, instead, overhears lower layer information to improve performance. This makes the solution reusable and transparent as the stack and middleware function independently.

The rest of the architecture is generic: messaging, storage and trace functions are not dependent on interaction with any particular protocol or layer. Specific parameters accessed by ROAM are stipulated by the API function calls. When these reach a predefined threshold or their value changes, this value is abstracted across the interface and passed to the optimisers (detailed in Sections VI and V).

ROAM implements a local, distributed approach, avoiding the overheads associated with inter-nodal messaging and executing concurrent to the network stack. Unlike previous cross-layer approaches such as ECLAIR [16] or the Control Middleware Plane [17], the middleware and optimisers specifically support MANET protocols and real-time QoS.

## V. MOBILITY RESPONSE OPTIMISER

Ad hoc routing protocols tend to select long, sub-optimal links to find the shortest E2E path. The mobility-response

functionality of ROAM (fig. 5) learns which is the fading link to avoid automatic selection. The optimiser registers with API to enable signalling of RSSI, rapid routing table changes and retransmissions, to detect fading and unstable links.

ROAM manages the replacement of the next hop in the routing table data structure. Therefore, while the approach is scalable to multiple protocols, the conditional logic of the implementation is specific to MANET protocol data structures. For example, conditional statements denote that transmitted and received routing control packets are for path setup or maintenance.

The optimiser (fig. 5) uses three monitoring functions that call OnEvent() tuning callback functions. The APIs use multiple callback functions to abstract parameters; for example get_packet_size() abstracts packet size (PS) when this value changes.

**Monitor_Receive_Power()** runs continually to monitor the current next hop (CNH), to which the MAC layer is sending frames, and neighbours with a higher RSSI. If the CNH RSSI fades below a threshold and was previously added to a list of INs to avoid, this is cleared as it is assumed that the node is out of range and a new next hop has been selected.

**Monitor_Next_Hop()** is called if a neighbouring node has a high RSSI and enables the following callback functions:

- *OnEvent1()*: monitors queue length and rapid changes in routing table CNH.
- *OnEvent2()*: observes relative RSSI between the CNH and an approaching neighbour and tunes temporary blacklisting of a fading CNH.
- *OnEvent3()*: temporarily pauses the application, to prevent restoration of the low-RSSI CNH on the basis of received ACKs or link layer frames.
- *OnEvent4()*: ensures new link RSSI is more than the previous low-RSSI CNH.

**Monitor_Very_Low_Coherence_Links**: monitors excess of retransmission limit and rate of change of control packet RSSI. Nodes moving at high relative velocity are assumed
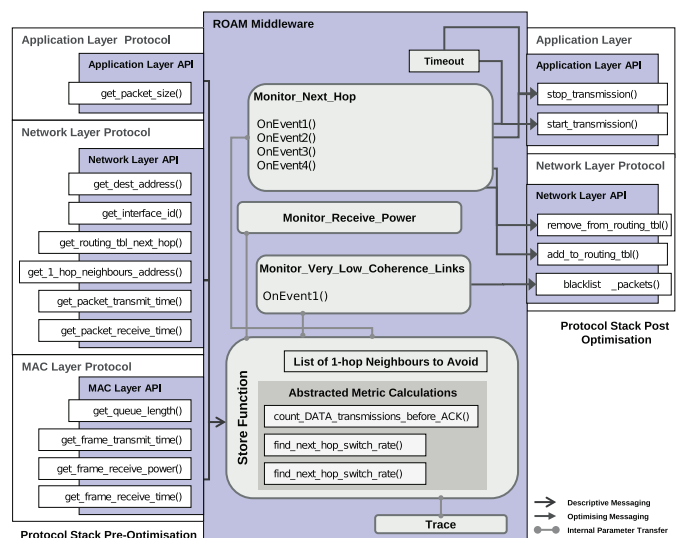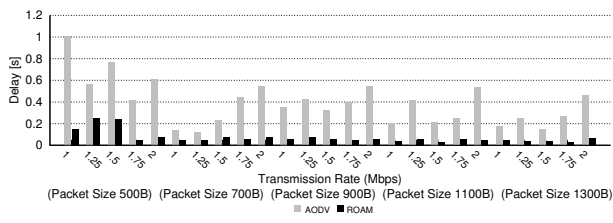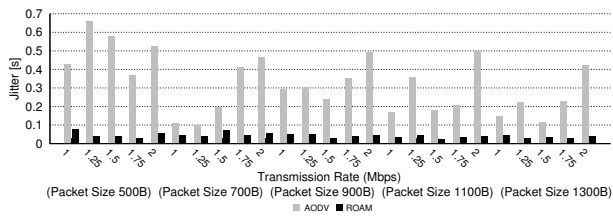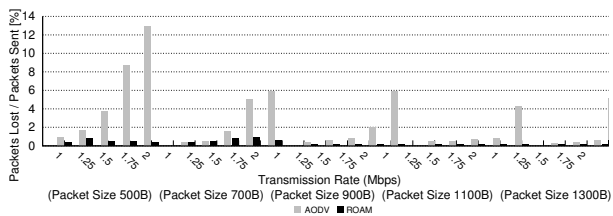


Fig. 5: ROAM with Mobility Response Optimiser

(a) Maximum Delay ($N1 \rightarrow N0$)



(b) Maximum Jitter ($N1 \rightarrow N0$)



(c) E2E PLR ($N1 \rightarrow N0$)

Fig. 6: Homogeneous Traffic Load



(a) Goodput (AODV)



(b) Goodput (ROAM)



(c) Delay (AODV)



(d) Delay (ROAM)
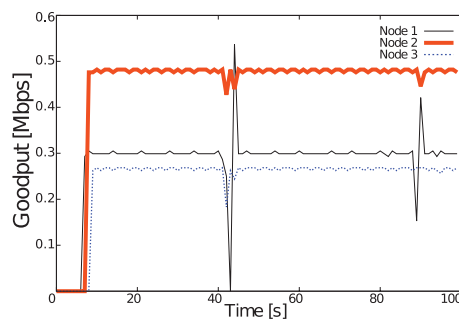
Fig. 7: Heterogeneous Traffic Performance

to maintain their speed. *OnEvent1()* tunes the temporary blacklisting of nodes with these attributes.

This optimiser is to ensure that maximum delay, jitter and loss are not associated with mobility-induced link selection. These metrics were investigated with ROAM versus routing managed by the reactive ad hoc protocol, AODV. ROAM has been validated for independence from MANET configuration and application and traffic settings that create detrimental network dynamics.
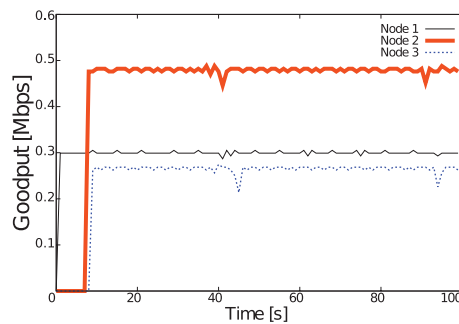
### A. Mobility Response with Traffic Load

The mobility response optimiser has firstly been validated with TR and PS variation between 1-2Mbps and 500-1300B. Means were taken from 10 stochastically varying simulation runs.
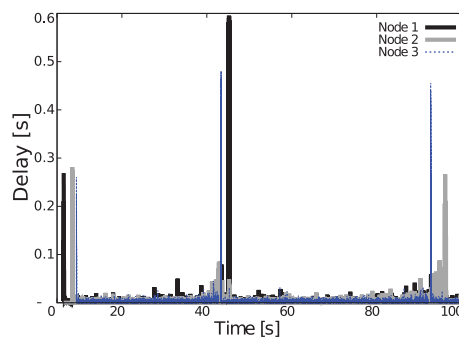
IEEE 802.11 transmits packets at a specific rate, tuned by auto-fallback and with smaller PS, more frames are transmitted under the same rate. Random backoff and packet collisions can occur more frequently, while link fading during backoff hinders MAC layer detection of channel state. Ad hoc routing reliance on link layer information for path maintenance then leads to preferential selection of longer, established hops. ROAM reduced transmissions subject to interference, by avoiding these links, reducing packet drops in all scenarios.
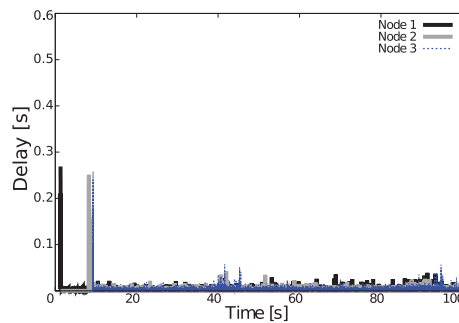
(a) Next Hop Selected by N1 (AODV)



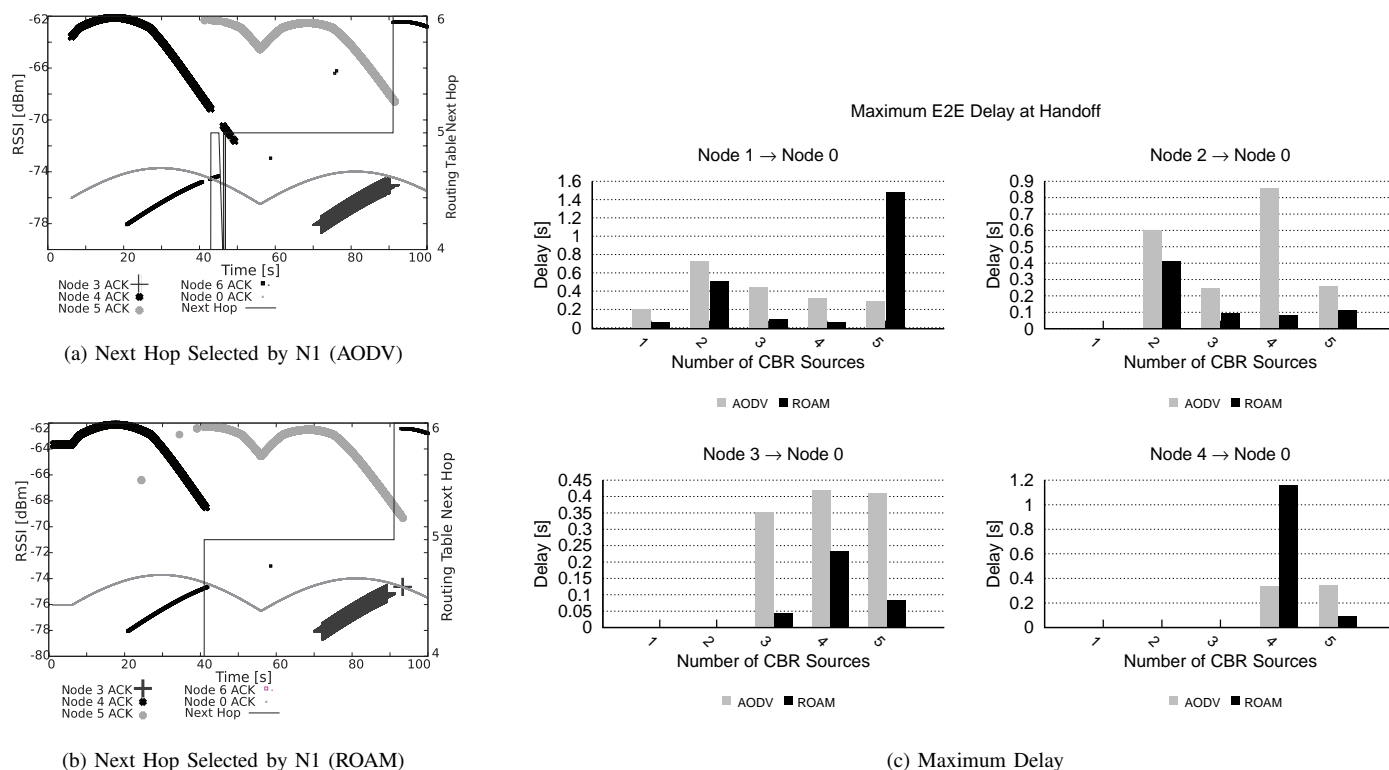(b) Next Hop Selected by N1 (ROAM)



(c) Maximum Delay

Fig. 8: Performance with Multiple CBR Sources

Fig. 6(c) shows overall PLR was reduced, but to a greater degree at higher PSs.

ROAM monitors and identifies robust links to ensure that the routing protocol selects these rather than switching intermittently to sub-optimal links, where collisions and routing errors can increase due to destructive signal fading. Then increased noise and auto-fallback countermeasures result in a queueing backlog, with packets being enqueued at a higher rate than they are dequeued.

This paper does not address the path setup cost of ad hoc routing: when delay peaks as packets are buffered pending E2E path setup. Therefore, maximum delay values were measured following this initial period. Figs. 6(a)-(b) demonstrate that under all traffic settings delay and jitter were bounded to below 0.3 s and to below 0.1s when the PS was larger than 500B.

Notably, with AODV, periods of degraded performance and increased delay surrounding handoff were longer at higher TR. Increasing traffic pressure on the IFQ induces regular buffer overflow when fading link use is causing a decline in link service rates and increasing error recovery requirements. With small packets sent at high TR, the queueing backlog exceeds buffer provisioning causing packets to be dropped. Loss of routing packets adds to the problem.

A network with mixed flows was also used to validate ROAM, given that future military networks will also require high performance under heterogeneous conditions. Channel dynamics become more complex, as usage differs between nodes and packet delays in enqueueing, dequeueing and transmission change rapidly and abruptly. In this example N1-N3

transmitted PS of 800B, 1000B and 600B at TR of 0.3Mbps, 0.6Mbps, 0.2Mbps.

Handoff for each CBR source occurred every 50s. Figs. 7(a)-(b) indicate that with AODV, handoff related degradation in goodput would lead to connection loss and E2E delays of up to 0.6s. When the ROAM optimiser was implemented over AODV, nominal degradation resulted. ROAM was capable of meeting real-time requirements by providing bounded maximum delay and jitter during handoff periods.

With multiple nodes in range, increased interference and channel busy periods force transmitters to repeatedly backoff and negotiate wireless channel access before transmitting. ROAM demonstrates stateless optimisation capabilities that are independent of particular flow settings and not reliant on continuous conditions across the network. Management of rapid handoff reduces maximum E2E delay, ensuring the peak is associated with path setup rather than link handoff, as seen in figs. 7(c)-(d).

### B. Mobility Response with Sources and Encoding

Multiple applications in military communications scenarios will be considered to be high priority. ROAM has been tested with multiple CBR and VoIP flows as it is expected that bounded delay and loss guarantees are provided to concurrent real-time streams. With an increasing number of mobile transmitters present, the peripheral topology and traffic configuration of the MANET changes dynamically as these act as forwarding nodes and compete for channel access on E2E paths.

(a) Next Hop Selected by N1 (AODV)



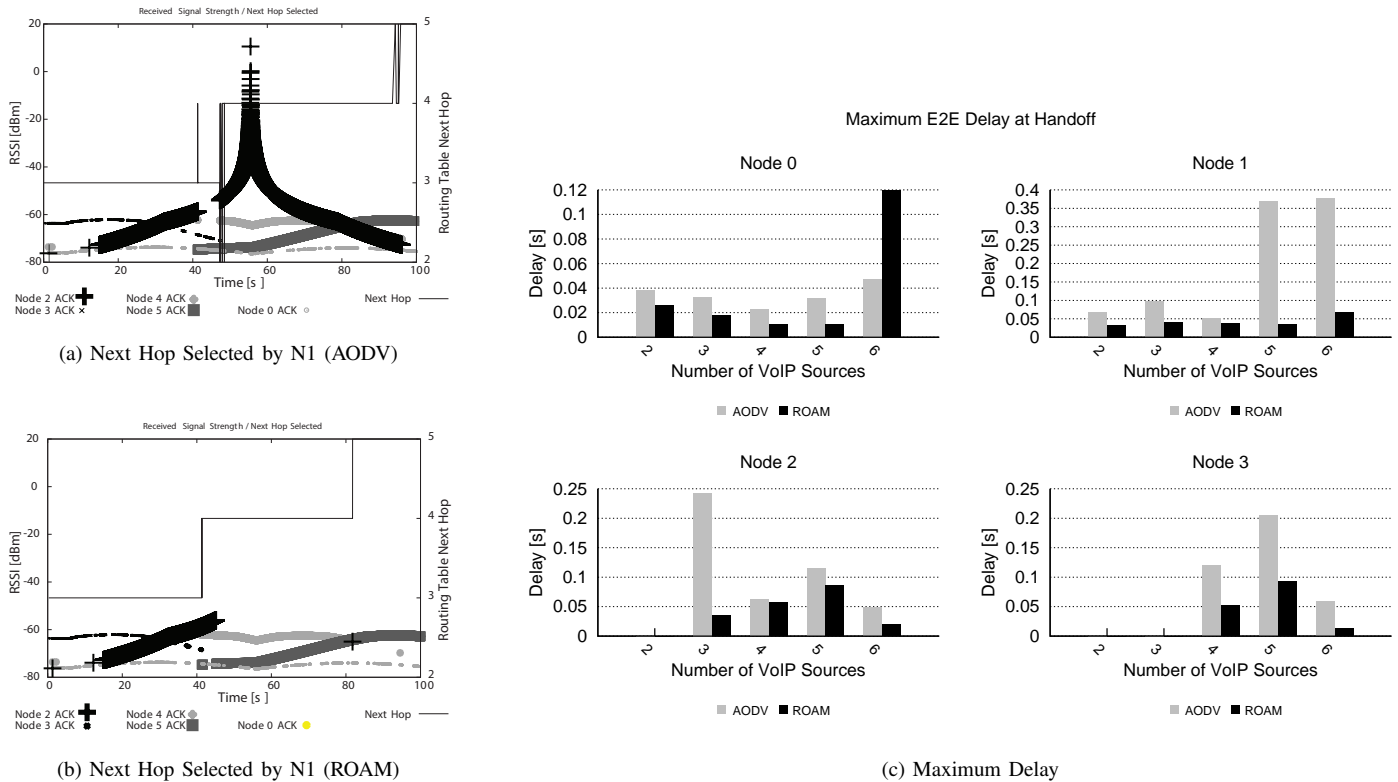(b) Next Hop Selected by N1 (ROAM)



(c) Maximum Delay

Fig. 9: Performance with Multiple VoIP Sources

CBR flows require the most stringent QoS from a network by transmitting and requiring receipt of a consistent stream of packets. The bidirectional, one-to-one VoIP sources used a variable traffic pattern model that differed from CBR through the inclusion of intervals of uplink and/or downlink silence amid bursts of VBR transmissions. Competition for medium access is more complex when traffic is bursty, arriving at inconsistent rates at forwarding nodes. Backoff and retransmission then have a greater impact on E2E delay and a sudden increase in traffic rate is more likely to overload IFQs in forwarding INs. Thus maximum delay was significantly lower for VoIP than CBR flows without ROAM.

With between 1-3 CBR sources or 2-5 VoIP sources present, performance improved for the ROAM node and other transmitters as rapid handoff curbed the rise in collisions and routing errors (figs. 8(c) and 9(c)). With multiple flows and increased link layer circulation, AODV in N1 repeatedly switched between fading and optimal paths (figs. 8(a)-8(b) and 9(a)-9(b)). Frames, ACKs or RREPs sucessfully received on a sub-optimal link induce AODV to use this as the next hop, thus handoff was faster with ROAM than AODV and switching in next hop selection was prevented.

However, with more than three sources, AODV provided comparable or better performance. With more nodes present and ROAM solely implemented in N1, punctual handoff for N1 was more likely to bring it in competition with a receding CBR source. With five CBR sources and six VoIP sources, receiver bottlenecks then resulted in increased enqueueing of

packets at N1. This corresponded to a rise in collisions and IFQ overflow outside of the local 1-hop neighbourhood in which ROAM is capable of improving performance. ROAMs ability to bound maximum delay for multiple real-time nodes was demonstrated with less than four transmitters, but not under control and data packet saturation. As competition for fading channels increased congestion.

We can also observe the higher power packets received by N1 from source N2 in (in fig. 9(a)(b)). Oblivious AODV in N1 repeatedly selected N2 as a forwarding next hop, creating increased packet delay as N2 continued to use N3 as a forwarding node that it was receding from. ROAM ensures that a routing protocol does not select a link that is likely to have a very low coherence time, therefore delay was reduced for both the ROAM node and N2.

### C. Mobility Response in Association Topologies

MANET topology variation changes service requirements and contention and interference levels on different links. To test scalability, ROAM mobility response has been evaluated in topologies with varied mean shortest hop counts (HC): a tree (HC = 2.2), ring (HC = 2.3) and star (HC = 2.1).

Within a star or ring topology diverse available paths exist and each node has many 1-hop neighbours. This raises the network congestion threshold and, depending on link length, the collision threshold. Therefore, the poorest AODV performance was in the tree topology, where limitation of paths increases collisions and fading link selection (fig. 10).
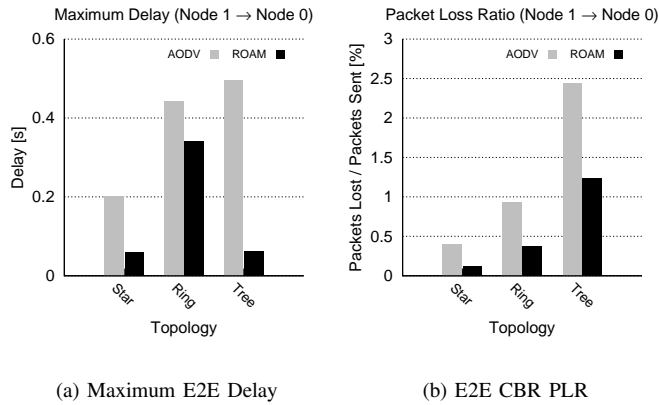
(a) Maximum E2E Delay      (b) E2E CBR PLR

Fig. 10: Performance in Varied Topologies



(a) Maximum E2E Delay      (b) E2E CBR PLR

Fig. 11: Performance with Mobile Node Speed

The ring topology has a drawback for a MANET transmitter: with a converging E2E path of increasing length, the performance of ad hoc routing degrades with each extra hop and the last hop becomes a bottleneck. Correspondingly, IFQ overflow at forwarding nodes was more prevalent. Fig. 10 shows that total PLR and maximum delay were reduced by ROAM, but the greatest improvement was in the tree topology.

### D. Mobility Response with Orbital Speed

Increasing node speed creates rapid topology changes, so ROAM performance becomes reliant on low levels of processing delay. Channel quality and next hop availability also change rapidly. Mobile transmitter speed was varied between 10-50m/s at increments of 10m/s.

Under a given TR, when nodes move in and out of range of each other at higher speeds, contention decreases but queueing increases as fewer packets are transmitted on each link before it begins to fade. With AODV alone, PLR increased incrementally with speed, whereas maximum delay was similar (fig. 11). Queueing and contention delay are key factors of E2E delay. ROAM reduces queueing delay and also the primary cause of packet dropping, IFQ overflow.

Multi-hop routing inefficiency impacts strongly on E2E delay when rapid, repeated handoff is required. Thus, at node speeds of 40-50m/s both AODV and ROAM provided unsatisfactory levels of performance. Repeated reconfiguration of topology leads to greater rate of change in channel quality and expedited path change reduced the timespan for which local information gathered by the ROAM was relevant. Fig. 11 shows that PLR improvement was promising but that the largest delay reduction was at lower speeds with ROAM.

Overall, capability to bound maximum delay and PLR under mobility has been demonstrated, supporting guarantees to timing-sensitive applications that maximum delay occurs only at initial path setup. However, contention induced delay is a key component of E2E delay, which must also be bounded to provide guaranteed performance to real-time traffic in MANETs. This is considered in the following section.
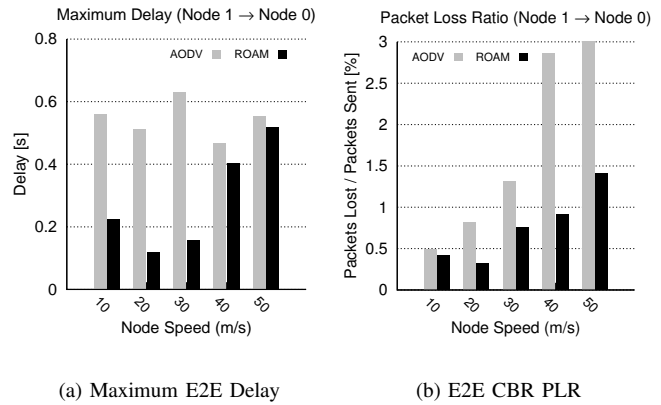
## VI. DISTRIBUTED CONTENTION RESPONSE OPTIMISER

The distributed contention response optimiser uses MAC overhearing to monitor queue length, channel busy time, path delay (time between RREQ-RREP and Data-ACK) and retransmission limit excess. Relative comparison is made between rate of packets intended for the CNH and for the ROAM node. This logic is used to learn whether a forwarding node is also used by a hidden terminal. Each ROAM node will reduce transmitted load following identification.

Previously discussed proposals for channel assignment and routing have relied on global signalling or reservation that competes with data for scarce resources. This optimiser uses available control packet information from unmodified protocols to gauge network conditions.

The optimiser uses three monitoring functions and the APIs use multiple callback functions to abstract parameters.

**Monitor_RTS-CTS()** permits execution of the algorithm if CSMA/CA is not used.

**Find_Hidden_Node()** calls a sequence of tuning functions:

- *OnEvent1()*: responds to increasing MAC queue length or path delay to monitor retransmission limit exceeded rate; ACK rate for the ROAM node and rate of ACK packets overheard from the CNH but intended for another node ($ACK_{nbr}$).
- *OnEvent2()*: responds to retransmission limit exceeded and $ACK_{nbr}$ rates that are higher than ACK rate, to assume the presence of a hidden terminal. ROAM tunes the application layer for a one-off constraint of TR and PS, reducing load without throttling.
- *OnEvent3()*: restores original application settings if previously monitored parameters fall below a threshold.

**Find_Exposed_Node()** runs continually with two tuning functions:

- *OnEvent1()*: responds to neighbour frame RSSI within range of CNH RSSI and $ACK_{nbr}$ interception to identify the presence of an exposed terminal. The function to find a hidden node is then interrupted and obstructed.
- *OnEvent2()*: permits the function to find a hidden node to be called if the exposed node disappears.
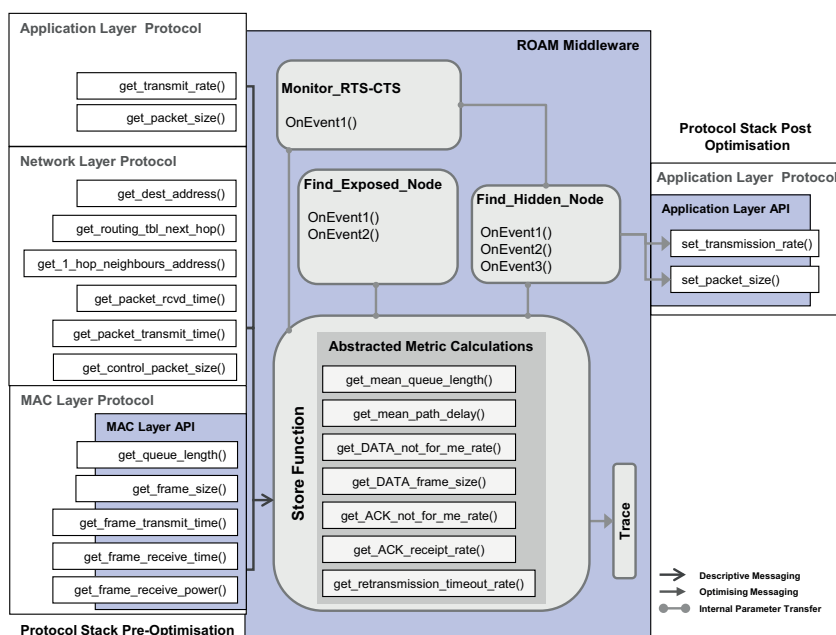
Fig. 12: ROAM with Contention Response Optimiser

In a shared medium, transmissions that are spatially exposed to each other contend for bandwidth. With a single antenna, transmission and collision detection (CSMA/CD) cannot be simmultaneous and sectionIII-B demonstrated some resulting media access control problems. However, the repercussions of flow admission oblivious to contention conditions are error recovery, backoff and overflow of resources; inducing violation of packet timing guarantees. In this section, results with ROAM are compared to CSMA/CA that depends on RTS/CTS handshaking.

### A. Contention Response with Mixed Traffic Load

ROAM was implemented over CSMA as this is the default for IEEE 802.11. As the contention response optimiser functions on a distributed basis it was implemented in all sources. ROAM was validated with heterogeneous CBR settings and increasing transmitted load: $TR_1$ of N1 varied between 1-5Mbps and $TR_2$ of N2 between 2-6Mbps, at intervals of 1Mbps. PS varied from 500-1300B at increments of 200B. ROAM has been tested with homogeneous settings and mixed PS but results are not included in this paper, as the lowest level
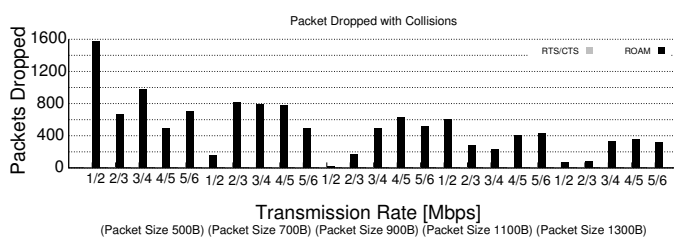


(a) Collision Count
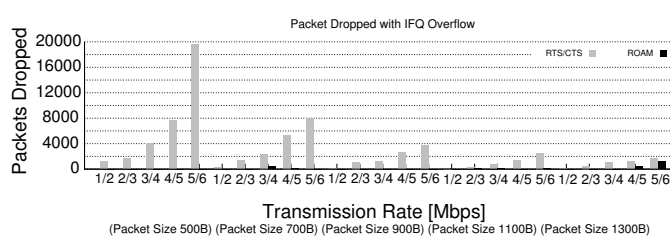


(a) IFQ Full
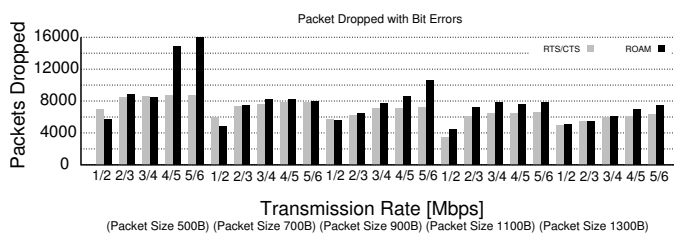


(b) Packet Bit Errors



(b) No Route

Fig. 13: Packet Loss with Mixed Traffic

Fig. 14: Packet Drops with Mixed Traffic

(a) Delay (RTS/CTS), Mixed Traffic
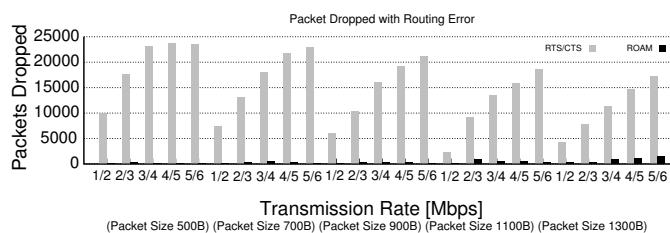
(b) Delay (RTS/CTS), 5 Sources

(c) Goodput (RTS/CTS), 5 Sources

(d) Delay (ROAM), Mixed Traffic

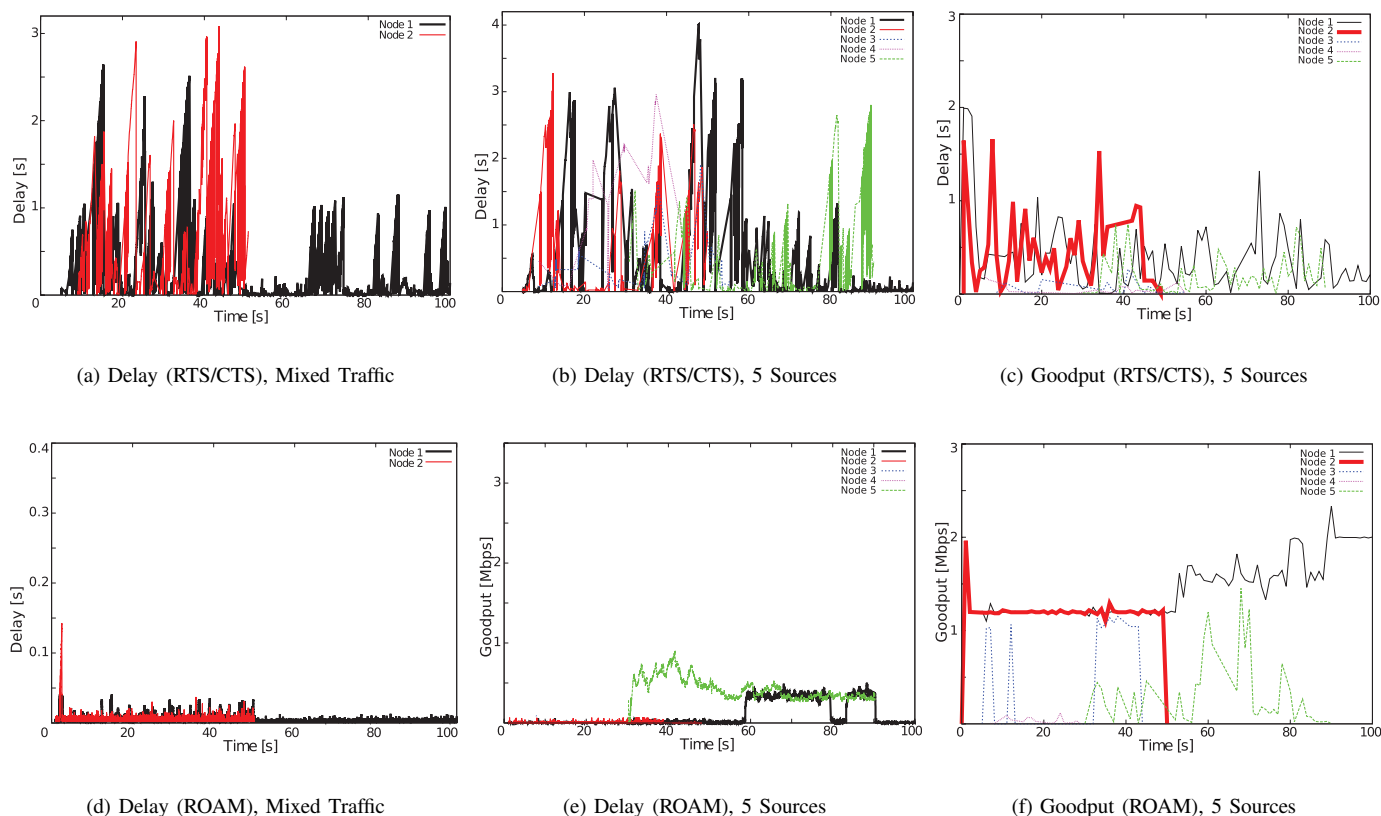(e) Delay (ROAM), 5 Sources

(f) Goodput (ROAM), 5 Sources

Fig. 15: Performance with Mixed Traffic and with Five CBR Sources

of ROAM performance was demonstrated with mixed TR.

Fairness of bandwidth distribution decreases as higher TR flows selfishly overload a shared channel. When bandwidth requirements differ between nodes, network packet delays in enqueueing, dequeueing and transmission become more varied. Good performance of ROAM under heterogeneous conditions is essential as the optimiser tunes both TR and PS to improve performance.

RTS/CTS relies on consistent TR and node synchronisation for collision avoidance, whereas ROAM does not directly prevent transmissions on a busy channel. Figs. 13-14 shows routing and bit errors were high with RTS/CTS as a result of control packet traffic. Bit error rates were similar to ROAM and higher at high TR, with more packets transmitted per second.

With flows sharing a receiver, IFQs that filled faster under higher TR were not serviced as fast at bottleneck links. Packets are increasingly buffered during handshaking and if RTS, CTS or routing packets are dropped, E2E queueing and path coherence times increase and the IFQ eventually overflows. Corresponding, PLR was lower with ROAM than RTS/CTS.

For heterogeneous traffic with $TR_1$ 3Mbps, $TR_2$ 4Mbps, PS 900B, Fig. 15(a) demonstrates that RTS/CTS injects large artificial delays and jitter into flows. In contrast to wired networks, multi-rate mechanism and distributed coordination function (DCF) employment mean high TR corresponds to high E2E delay. Following packet errors and with queue

variation and node mobility these introduce jitter in the stream.

When a hidden source was identified by ROAM, PS and TR were adjusted to reduce the pressure on the shared channel by 30%. When the nodes were out of range (after 50s) ROAM tuned load to return to its previous value. Competition for the shared receiver was reduced and the queue backlog was allowed to empty, ensuring efficient use of available bandwidth during contention. In reducing contention without handshaking and buffering, ROAM makes a greater contribution to overall ISRT performance, to bound maximum delay and jitter (figs. 15(a) and (d))

### B. Contention Response with Competing Sources

CBR QoS requirements are the most stringent amongst ISRT applications and provide a representative benchmark for other ISRT scenarios. Increasing numbers of CBR flows creates variation in contention from data and control packets and dynamic changes in available bandwidth.

Sources were hidden terminals in pairs: N1/N2, N3/N4 during 10-50s of the simulation. Load reduction under contention tuned by the middleware provided comparable performance to RTS/CTS. Reducing TR lowers competition for resources, but ROAM also increases PS to avoid application throttling.

The addition of further sources increased congestion along the E2E path, but each was farther from the receiver and subject to low available bandwidth. This is because increasing circulation of flows and control packets through the network
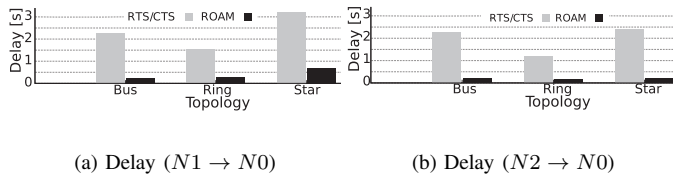
(a) Delay ($N1 \rightarrow N0$)

(b) Delay ($N2 \rightarrow N0$)

Fig. 16: E2E Delay during Contention with Topology

TABLE I: Packets Dropped by Topology

| Drop | RTS/CTS | | | ROAM | | |
|---|---|---|---|---|---|---|
| Reason | Ring | Bus | Star | Ring | Bus | Star |
| COL | 400 | 200 | 900 | 1229 | 121 | 1247 |
| ERR | 9825 | 5683 | 8225 | 14216 | 8647 | 12051 |
| IFQ Full | 186 | 836 | 2509 | 0 | 0 | 0 |
| No Route | 26717 | 12621 | 20029 | 1410 | 247 | 1114 |

contributed to reduced channel quality and increased and varying contention delay. Thus goodput for nodes 3-5 was similarly throttled under congestion with RTS/CTS and ROAM (figs. 15(c)-(d)).

Fig. 15(f) indicates that ROAM was able to bound maximum delay by more than 50%, in comparison to RTS/CTS. However, as N5 did not compete with a hidden terminal this node transmitted its flows at the full initial TR, increasing IFQ backlogs along the shared E2E path. This had an impact on E2E delay for N2, even when the remaining transmitters had left the network.

This scenario has shown that ROAM is capable of being implemented on a large scale in multiple hidden transmitter pairs. Performance is improved for ISRT nodes with short E2E paths but a distributed, local contention response cannot prevent congestion in a large multi-hop MANET. Load control would need to be proportional to the number of hidden terminal pairs to prevent the creation of a bottleneck at a shared receiver.

### C. Contention Response in Hidden Node Topologies

ROAM is able to bound maximum delay and jitter, under a range of configurations when the network is not congested. However, the previously implemented bus topology (HC = 2.5) used limited, convergent E2E paths that force nodes to compete for available bandwidth and channel access.

Therefore, ROAM has been evaluated with two further topologies of varied HC: a star (HC = 2.1) and a ring (HC = 2.3). ROAM uses conditional comparison between packets, giving independence from varying inter-nodal distance. In each topology, N1 and N2 were hidden from each other.

Table I shows that due to the size and structure of the ring topology, collisions, packet and routing errors were higher than in the star and bus topologies, even with consistent TR. Bit errors were a significant challenge in the ring and star topologies, that placed multiple nodes within interference range.

Spatial diversity of E2E paths was more limited in the ring and bus than the star, that could offer more robust paths to the routing protocol. Collisions and errors were more prevalent with ROAM than RTS/CTS in all topologies but ROAM reduced IFQ overflow and routing errors that were the key cause of packet drops with RTS/CTS.

While ROAM avoids contention on a single shared channel, this is not implemented along the E2E path. Whereas, RTS/CTS is implemented across the network, rather than solely in high collision areas. Handshaking takes place whenever a packet is to be transmitted, enabling the continual detection of contention but also leading to much higher E2E delay. Thus, unnecessarily elevated per hop delay compounded to high E2E delay and IFQ overflow in the ring and star topologies.

As multiple nodes are in interference range and will contend for channel access with each other in a star or ring topology, MAC layer delay and consequently E2E delay is higher than in a bus topology. During the period when N1 and N2 were transmitting, although ROAM had reduced transmitted load, packets were delayed due to a buffer backlog. As the transmitters orbited the star and ring topologies, handoff occurred, during which packets were buffered as new E2E paths were set up. Packets are continually enqueued as the IFQ begins to empty. Therefore, ROAM performance improvement marginally decreases with the number of handoffs required.

### D. Contention Response with Source Speed

In future military networks, ISRT will require support under the rapidly changing conditions created by increasing node speed. E2E routes will be setup and torn down on a more regular basis, with nodes moving more frequently between busy and available channels. To evaluate the architecture under these conditions, speed was elevated between 10-50m/s at intervals of 10m/s. N1 and N2 were hidden terminals from 0-50s into the simulation.

E2E delay depends on efficient E2E path maintenance at high speed. As RTS and CTS packets interfere with routing control packets, path maintenance is slower. This results in increased buffer backlogs and eventual overflow. ROAM performs well at high speeds by implementing a short-term reduction in resource requirements. By offering lower incidence of routing errors it enables timely maintenance of rapidly changing E2E paths, lowering overall PLR.

Fig. 17 demonstrates the reduced maximum delay with ROAM. While ROAM reduces the rate at which packets are enqueued by the application, at 20m/s error recovery resulted in high delay. Additionally, each subsequent handoff compounded the time taken for the IFQ to drain, even after the competing node had left the network. However, at 50m/s rapid path change reduced the contention period.

With RTS/CTS, goodput periodically dropped to a negligible value but was regularly higher than with ROAM. RTS/CTS waits for a free channel before transmitting thus MAC rates can be stepped up, draining the queue but increasing burstiness and unfair bandwidth distribution. ROAM reduces the achievable goodput of both contending nodes but demonstrates more fairness between N1 and N2.
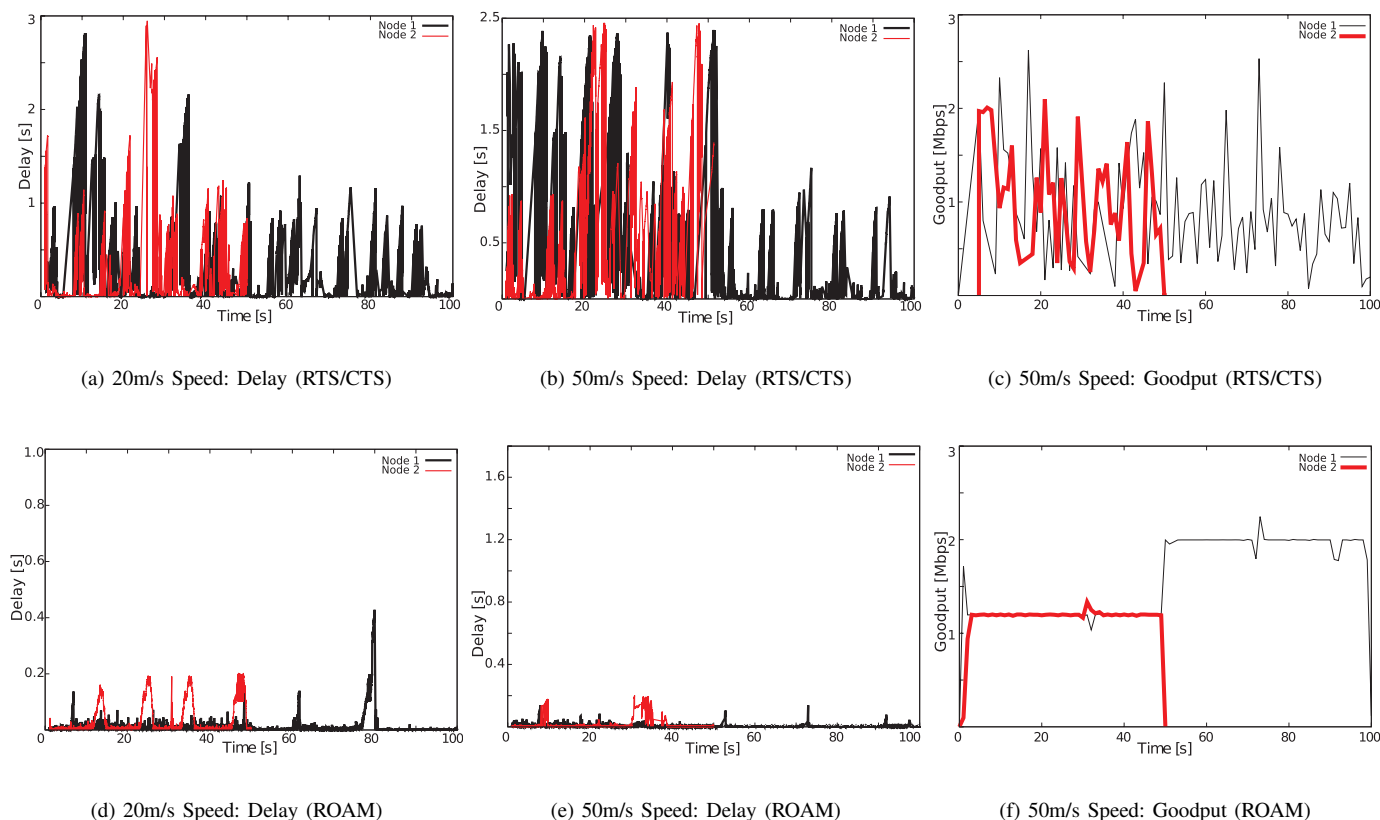
(a) 20m/s Speed: Delay (RTS/CTS)     (b) 50m/s Speed: Delay (RTS/CTS)     (c) 50m/s Speed: Goodput (RTS/CTS)

(d) 20m/s Speed: Delay (ROAM)     (e) 50m/s Speed: Delay (ROAM)     (f) 50m/s Speed: Goodput (ROAM)

Fig. 17: Mobile Node Speed

## VII. CONCLUSIONS AND FUTURE WORK

The novel architecture and associated optimisers proposed in this paper are capable of providing assurances of bounded maximum delay, loss and jitter and that the lowest system performance levels are solely associated with initial path setup. This enables the provision of guarantees to loss-sensitive real-time applications, without recourse to protocol modification, complex protocol interactions or the use of congestion-inducing signalling. Real-time QoS awareness can therefore be added to devices as desired, without the need to encourage widespread uptake of a novel protocol or architecture.

However, assumptions made in the design of the optimisers limits their operation to MANET protocols as the API is used to manipulate and monitor particular protocol data structures, for example the transmission and receipt of routing control packets is assumed to be only for the purpose of maintaining or setting up a new path. ROAM also does not provide direct delay control through traffic conditioning. Therefore, if a requirement to provide optimal horizontal handoff or contention control does not occur, ROAM is incapable of providing bounded delay, jitter or packet loss ratio.

Our simulation results have demonstrated that the ROAM architecture outperforms benchmark MANET protocols such as AODV and RTS/CTS. Future work will include large scale evaluation of the architecture in a real-world testbed and extension of the ROAM optimisers to address the problem of MANET-specific congestion. We intend to investigate perfor-mance alongside adaptive traffic admission with heterogeneous traffic and intelligent buffer management with homogeneous traffic. ROAM contention control may also be extended to include low level signalling of the presence of multiple hidden node clusters. This will enable tuning according to distance from the receiver, to extend the contention reduction to a congestion reduction benefit across the network.

## ACKNOWLEDGMENT

## REFERENCES

[1] T. B. Reddy, I. Karthigeyan, B. S. Manoj, and C. S. R. Murthy, "Quality of sService Provisioning in Ad hoc Wireless Networks: a Survey of Issues and Solutions," *Ad Hoc Netw.*, vol. 4, no. 1, pp. 83–124, Jan. 2006.

[2] M. Kumar and R. Mishra, "An Overview of MANET: History, Challenges and Applications," *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. 3, no. 1, Mar. 2012.

[3] W. Jia, H. Wang, M. Tang, and W. Zhao, "Effective Delay Control for High Rate Heterogeneous Real-time Flows," in *ICDCS '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society, 2003, p. 367.

[4] E. D. Jensen, "Eliminating the 'Hard'/'soft' Real-time Dichotomy," *Computing and Control Engineering Journal*, vol. 8, no. 1, pp. 15–19, 1997.

[5] J. L. Sobrinho, R. D. Haan, and J. M. Brazio, "Why RTS-CTS is not your Ideal Wireless LAN Multiple Access Protocol," in *IEEE Wireless Communications and Networking Conference 2005*, vol. 1, 2005, pp. 81–87.

[6] D. Kliazovich and F. Granelli, "Introduction: Why Cross-Layer? Its Advantages and Disadvantages," in *Cross Layer Designs in WLAN Systems*, ser. Emerging Communication and Service Technologies Series, N. Zorba, C. Skianis, and C. Verikoukis, Eds. Troubadour Publishing Limited, 2011, no. v. 1.

[7] S. Pease, L. Guan, I. Phillips, and A. Grigg, "Cross-layer Signalling and Middleware: A Survey for Inelastic Soft Real-time Applications in MANETs," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1928–1941, 2011.

[8] S. Sharma, R. Mishra, and K. Singh, "Current Trends and Future Aspects in Cross-Layer Design for the Wireless Networks," in *Advances in Computer Science and Information Technology. Computer Science and Engineering: Second International Conference*, ser. CCSIT '12, 2012, pp. 283–296.

[9] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, "Cross-layer Design of Ad hoc Networks for Real-time Video Streaming," *Wireless Communications, IEEE*, vol. 12, no. 4, pp. 59–65, 2005.

[10] K. Hong, S. Sengupta, and R. Chandramouli, "Cross-Layer MAC Enabling Virtual Link for Multi-Hop Routing in Wireless Ad Hoc Networks," *Communications Society*, 2010.

[11] Y. Yamao, Y. Kida, and Y. Kadowaki, "Cross-Layer Multi-Hopping Scheme for Efficient and Reliable Transmission in Fading Environment," in *VTC Fall*, 2010, pp. 1–5.

[12] H. Aiache, V. Conan, J. Leguay, and M. Levy, "Xian: Cross-layer Interface for Wireless Ad hoc Networks," in *Mediterranean Ad Hoc Networking Workshop, 2006. Med-Hoc-Net 2006. Proceedings of the*, Jun. 2006.

[13] L. Qin and T. Kunz, "Survey on Mobile Ad Hoc Network Routing Protocols and Cross-Layer Design," Carleton University, Tech. Rep., Aug. 2004.

[14] T. Taleb, K. Kashibuchi, A. Leonardi, S. Palazzo, K. Hashimoto, N. Kato, and Y. Nemoto, "A Cross-Layer Approach for an Efficient Delivery of TCP/RTP-Based Multimedia Applications in Heterogeneous Wireless Networks," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 6, pp. 3801–3814, 2008.

[15] E. da Silva and L. Albini, "Middleware Proposals for Mobile Ad Hoc Networks," *Journal of Network and Computer Applications*, vol. 43, pp. 103–120, Aug. 2014.

[16] V. T. Raisinghani and S. Iyer, "Cross-layer Feedback Architecture for Mobile Device Protocol Stacks," *Communications Magazine, IEEE*, vol. 44, no. 1, pp. 85–92, Jan. 2006.

[17] Z. Chang, G. Gaydadjiev, and S. Vassiliadis, "Infrastructure for Cross-Layer Designs Interaction," in *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, Aug. 2007, pp. 19–25.

[18] H. Aiache, V. Conan, L. Lebrun, J. Leguay, S. Rousseau, and D. Thoumin, "XIAN Automated Management and Nano-Protocol to Design Cross-Layer Metrics for Ad Hoc Networking," in *Lecture Notes in Computer Science: Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, ser. Lecture Notes in Computer Science, A. Das, H. K. Pung, F. B. S. Lee, and L. W. C. Wong, Eds. Springer Berlin Heidelberg, 2008, vol. 4982, pp. 1–13.

[19] M. Conti, G. Maselli, and G. Turi, "Design of a Flexible Cross-Layer Interface for Ad Hoc Networks," in *Challenges in Ad Hoc Networking*, ser. IFIP International Federation for Information Processing. Springer Boston, 2006, vol. 197, pp. 189–198.

[20] K. M. E. Defrawy, M. S. E. Zarki, and M. M. Khairy, "Proposal for a Cross-layer Coordination Framework for Next Generation Wireless Systems," in *IWCMC '06: Proceedings of the 2006 international conference on Wireless communications and mobile computing*. New York, NY, USA: ACM, 2006, pp. 141–146.

[21] X. Gu, X. Fu, H. Tschofenig, and L. Wolf, "Towards Self-optimizing Protocol Stack for Autonomic Communication: Initial Experience," in *Autonomic Communication*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006, vol. 3854, pp. 186–201.

[22] X. Ren and J. Zhang, "A Novel Cross-layer Architecture for Wireless Protocol Stacks," in *International Conference on Multimedia Technology (ICMT)*, Ningbo, China, 2010.

[23] M. A. Razzaque, S. Dobson, and P. Nixon, "Cross-layer Optimisations for Autonomic Networks," in *Advanced Autonomic Networking and Communication*, ser. Whitestein Series in Software Agent Technologies and Autonomic Computing. Birkh{ä}user Basel, 2008, pp. 127–148.

[24] V. T. Raisinghani and S. Iyer, "ECLAIR: An Efficient Cross layer Architecture for Wireless Protocol Stacks," in *In World Wireless Congress, SF*. Delson Group Inc, 2004.

[25] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *Mobile Ad Hoc Networking: The Cutting Edge Directions*, ser. IEEE Series on Digital & Mobile Communication. Hoboken, NJ, USA: Wiley, 2013.

[26] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, and M. Zorzi, "ns2-MIRACLE: a Modular Framework for Multi-technology and Cross-layer Support in Network Simulator 2," in *ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*. ICST, Brussels, Belgium, Belgium; Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering: ICST, 2007, pp. 1–8.

**Sarogini Grace Pease** received the Ph.D. degree in computer science from Loughborough University, Loughborough, U.K., in 2013, the M.Sc. degree in information technology from the Department of Computer Science, Loughborough University, in 2008, and the LL.B.(Hons.) degree in law from the University of Essex, Essex, U.K., in 2003.

She is currently a Research Associate at the Wolfson School of Mechanical and Manufacturing Engineering at Loughborough University, Loughborough, U.K. She previously held a Communications Manager position at H.M. Prison Service, and was Communications Coordinator at Dresdner Kleinwort Investment Bank and The London Museums Hub. Her research interests are in the design and performance improvement of real-time, embedded systems for mobile ad hoc and wireless sensor networks; industrial energy monitoring and indoor tracking.

Dr Pease's Ph.D. project was funded under CASE award from the E.P.S.R.C. and B.A.E. Systems.

**Iain W Phillips** received the Ph.D. degree in computer science from the University of Manchester, Manchester, U.K. in 1994 and the B.Sc. degree from the University of Manchester in 1989.

He has been involved in computer networks research for over 20 years. Since 1992 he has worked at Loughborough University, Loughborough, U.K., as a Research Associate, then Research Fellow, Lecturer and Senior Lecturer. From 2008-11 he served as Head of the Department of Computer Science. During 2011-12 he took a sabbatical year spending time at the University of Cambridge, Cambridge, UK and the University of Adelaide, Adelaide, Australia. He has over 100 publications in Computer Networking and Distributed Computing. From 2012-14 he was Chair of the Council of Professors and Heads of Computing (CPHC), UK.

Dr Phillips is a Chartered Engineer, Chartered IT Professional, Fellow of the British Computer Society and Member of the ACM. He has reviewed for IEEE journals.

**Lin Guan** received the Ph.D. degree from the University of Bradford, Bradford, UK, in 2005.

She is a Senior Lecturer in the Department of Computer Science at Loughborough University, Loughborough, U.K. After her Ph.D, she was also appointed as a Research Associate at the University of Bradford. She held a project manager/software engineer position in Simulation Systems Ltd. shortly, prior to moving to Loughborough University as a Lecturer in 2006. Her research interests focus on performance modelling/evaluation of heterogeneous computer networks, QoS analysis and enhancements, mobile computing and wireless sensor networks. She has published over 70 journal and conference papers and she has been serving as guest co-editor for several international journals, such as those published by Elsevier and Springer.

Dr Guan was awarded the British Federation of Women Graduates Foundation Main Grant in 2004, during her Ph.D. She holds two CASE awards and one Eng.D. on E.P.S.R.C./B.A.E. funded projects. She received a prestigious award as Royal Society Industry Fellow and E.P.S.R.C. KTA project.