
This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

ROAM: supporting safety critical applications in MANETs with cross-layer middleware

PLEASE CITE THE PUBLISHED VERSION

<http://dx.doi.org/10.1109/WoWMoM.2013.6583432>

PUBLISHER

© IEEE

VERSION

AM (Accepted Manuscript)

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Pease, Sarogini G.. 2019. "ROAM: Supporting Safety Critical Applications in Manets with Cross-layer Middleware". figshare. <https://hdl.handle.net/2134/18451>.

ROAM: Supporting Safety Critical Applications in MANETs with Cross-layer Middleware

Sarogini Grace Pease

Abstract—Provision of instantaneous, mobile and dependable media communications in military and disaster scenarios must overcome certain wireless network issues: lack of reliable existing infrastructure, immutability of safety-certified protocols and layer-2 dynamics with contributing factors including hidden transmitters and fading channels. This extended abstract investigates a cross-layer methodology to provide timely MANET communications through optimal channel selection and dynamic contention reduction, without protocol modification. This is done using ROAM: a new Real-time Optimised Ad hoc Middleware based architecture that has been implemented and validated in the ns2-MIRACLE simulator.

Keywords—Cross-layer Design, Inelastic Soft Real-time Applications, Mobile Ad Hoc Network

I. INTRODUCTION

MANETs are self-organising infrastructureless networks and MANET protocols work on a self-configuring basis to adaptively create network connections, without centralised management. This makes them ideal to media streaming and communications in military or disaster scenarios. However, shared channels, time-varying and complex topologies create multiple factors at layers 1 and 2 that contribute to high loss and variable end-to-end delay [1]. These must be addressed for MANETs to be viable solutions to real-time applications in these scenarios, where timeliness of packet delivery influences both usefulness and safety.

Over provisioning and resource management cannot provide absolute guarantees in wireless networks and widely used applications for VoIP and multimedia streaming have been designed to tolerate loss and delay. Regular frame loss and delays are not acceptable in safety-critical scenarios, such as the video transmission of friendly and non-friendly vehicles in military operations. These applications can operate within the remit of inelastic soft real-time [2] (ISRT) that tolerates loss, delay and jitter within acceptable and guaranteed bounds. Cross-layer responsiveness to layer 1 and 2 conditions enables higher layers to distinguish between causes of packet losses and errors, improving application performance [3]. For example, tuning transmission rate to channel quality prevents oversubscription of resources and related packet loss.

Withholding internal layer parameters from other layers facilitates fast development of interoperable systems and this functionality can be preserved by parameter monitoring and tuning, managed by external middleware. Current proposals for non-real-time applications have conceptualised but not implemented middleware that uses API access to protocol data structures [4]. These designs can enable generic support to contemporary and legacy network protocols, but require addition of optimising functionality to support delay and jitter sensitive ISRT.

Timing sensitive applications use UDP or RTP (Real-Time Protocol) at the transport layer, sacrificing reliability provided by TCP for reduced delay. When a hidden transmitter is present, the solution in IEEE 802.11 networks is to use handshaking of RTS/CTS packets, which has been shown to increase packet delay and congestion [5]. Lower layer signalling has been proposed to adjust transmission rates to individual link capacity and combined with resource reservation increased supported data rates [6].

Contention control can take place once a route has been selected, as end-to-end bandwidth and delay are dependent on channel quality and node capacity at each hop. Ad hoc 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. MANET routing with the lowest impact on delay is reactively set up as required (such as DSR, AODV and OSPF). These routinely select paths with minimum hop count, a metric that can be used instantaneously but does not always select the most robust link or with the least delay. Link metrics such as delay, ETX, MTM, WCETT can be implemented, but node mobility, link breakages [7] and the underlying process of the MAC layer such as repeated backoff introduce complexity in path selection and performance. Using dummy packets to probe instantaneous signal strength has been proposed as a reason to induce handoff between links of the same technology and resulted in reduced packet losses [8].

II. ROAM

Figure 1 gives an overview the proposed Real-time Optimised Ad hoc Middleware based architecture, consisting of the middleware (ROAM), multiple layer-specific APIs and associated cross-layer messages for the abstraction of protocol parameters. These have been designed to support safety-certified real-time protocols through minimal imposition on and concurrent execution with the protocol stack. Instead of modifying protocols, ROAM uses layer-specific API to access generic MANET protocol parameters. API are exported to the MAC, network and application layers.

The motivation for the optimal handoff functionality of ROAM is to ensure that a fading link is not automatically selected by a routing protocol that tends to select the shortest end-to-end path. The optimiser will register with the API to enable signalling of routing table values, retransmission counts and control packet RSSI to the middleware by the API. Fading is detected based on received signal strength indication (RSSI), rapid routing table changes and retransmissions. Rapid change in RSSI is used to identify sub-optimal channels to high velocity nodes. Comparative RSSI is then used to identify an optimal next hop.

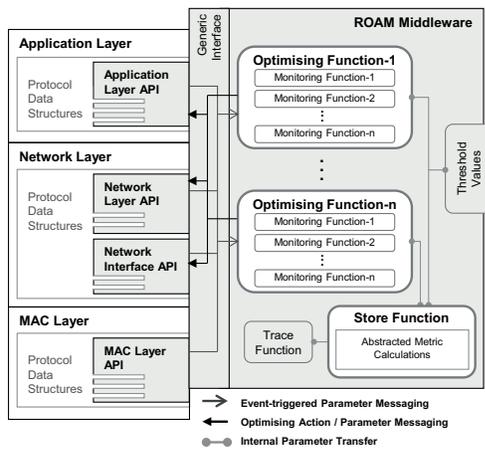


Fig. 1. ROAM Architecture Overview

To tune link selection, ROAM manages the replacement of the next hop in the routing table (with temporary pause of application transmission during removal) and temporary blacklisting of the fading path. ROAM signals the API with the old and new next hops that are replaced by the API within the protocol data structure. Therefore, while the approach is scalable to multiple protocols, the implementation is specific to MANET protocols as the API are used to manipulate and monitor particular protocol data structures. For example the transmission and receipt of routing control packets is assumed to be for setup and maintenance of paths.

The second, distributed contention function allows each instance of ROAM to monitor for repeated excess of retransmission count, increase in queue length, increasing channel busy time and delay between routing control packets. Rate of foreign ACKs intended for the next hop and ACKs intended for the node are compared to identify whether the same forwarding node is being used by a hidden node. Each ROAM node will reduce transmitted load by 15% following identification, to reduce contention. Previously discussed proposals for channel assignment and routing have relied on global signalling or reservation that competes with data for scarce network resources. In contrast, ROAM utilises available control packet information from unmodified protocols to gauge network conditions.

III. RESEARCH FINDINGS AND FUTURE WORK

Realistic simulation of cross-layer middleware and environmental interference conditions are supported by ns2-MIRACLE, which has been used for ROAM architecture simulation. ROAM is validated in MANET simulation scenarios that demonstrate independence from the key causes of network dynamics: application transmission setting (using CBR or VoIP over 802.11g) and MANET configuration, including mobility, speed and topology. Results with the horizontal handoff functionality have been compared to network performance using AODV. The hidden node contention reduction function has been tested against both CSMA and RTS/CTS.

Table I provides an example of performance for each node on the network, firstly with AODV managing handoff and then with the optimal handoff functionality of ROAM when the

TABLE I. PERFORMANCE COMPARISON OF ROAM AND AODV

Maximum End-to-end Delay [s] with AODV				
N. CBR Sources	N1 → N0	N2 → N0	N3 → N0	N4 → N0
1	0.022	-	-	-
2	0.023	0.019	-	-
3	0.024	0.021	0.024	-
4	0.015	0.016	0.014	0.014
Maximum End-to-end Delay [s] with ROAM				
N. CBR Sources	N1 → N0	N2 → N0	N3 → N0	N4 → N0
1	0.016	-	-	-
2	0.017	0.015	-	-
3	0.019	0.014	0.019	-
4	0.010	0.012	0.011	0.010
End-to-end PLR [%] with AODV				
N. CBR Sources	N1 → N0	N2 → N0	N3 → N0	N4 → N0
1	3.2	-	-	-
2	4.5	30.5	-	-
3	4.6	14.2	3.7	-
4	7.6	17.2	5.9	9.2
End-to-end PLR [%] with ROAM				
N. CBR Sources	N1 → N0	N2 → N0	N3 → N0	N4 → N0
1	6.4	-	-	-
2	7.1	30.5	-	-
3	8.4	14.2	3.1	-
4	10.4	17.2	4.8	6.6

number of CBR sources is varied. In these simulations, nodes moved 120m apart at speeds of 1.1m/s. Experimental results have shown that ROAM provides better performance, with bounded maximum delay and jitter, with marginally higher PLR (up to 5% increase) for some nodes.

Future work to extend this project will be to explore the feasibility of the middleware approach in a testbed environment. Kernel implementation and testing of cross-layer middleware in a mobile ad hoc network is still an open research area.

ACKNOWLEDGMENT

This work is sponsored by the Engineering and Physical Science Research Council (EPSRC) and BAE Systems, UK. I would like to thank Dr. Iain Phillips and Dr. Lin Guan for their invaluable supervision of this project.

REFERENCES

- [1] T. B. Reddy, I. Karthigeyan, B. S. Manoj, and C. S. R. Murthy, Quality of service 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] E. D. Jensen, Eliminating the ‘hard’/‘soft’ real-time dichotomy, *Computing and Control Engineering Journal*, vol. 8, no. 1, pp. 15–19, 1997.
- [3] D. Kliazovich and F. Granelli, Introduction: Why Cross-Layer? Its Advantages and Disadvantages, *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.
- [4] V. T. Raisinghani and S. Iyer, ECLAIR: An efficient cross layer architecture for wireless protocol stacks, *World Wireless Congress, SF*. Delson Group Inc, 2004.
- [5] J. L. Sobrinho, R. D. Haan, and J. M. Brazio, Why RTS-CTS is not your ideal wireless LAN multiple access protocol, *IEEE Wireless Communications and Networking Conference 2005*, volume 1, pp. 81–87, 2005.
- [6] 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*, 1, vol. 12, no. 4, pp. 59–65, 2005.
- [7] L. Qin and T. Kunz, Survey on Mobile Ad Hoc Network Routing Protocols and Cross-Layer Design, Technical report, Carleton University, Aug. 2004.
- [8] 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.