Fall 2010

Adaptive resource allocation for cognitive wireless ad hoc networks

Behdis Eslamnour

Department: Electrical and Computer Engineering

ADAPTIVE RESOURCE ALLOCATION FOR COGNITIVE WIRELESS AD HOC NETWORKS

by

BEHDIS ESLAMNOUR

A DISSERTATION

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

2010

Approved by

Jagannathan Sarangapani, Advisor
Maciej J. Zawodniok, Co-advisor
Ann Miller
R. Joe Stanley
Sanjay Madria

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following four articles that have been submitted for publication as follows:

Paper 1, pages 18-50, B. Eslamnour, S. Jagannathan, and M. J. Zawodniok, "Dynamic Channel Allocation in Wireless Networks Using Adaptive Learning Automata," in review, International Journal of Wireless Information Networks.

Paper 2, pages 51-94, B. Eslamnour and S. Jagannathan, "Adaptive Dynamic Routing for Hybrid Wireless Networks Using Nonlinear Optimal Framework," submitted to Wireless Communications and Mobile Computing, Wiley.

Paper 3, pages 95-147, B. Eslamnour and S. Jagannathan, "Dynamic Routing for Multi-Channel Multi-Interface Hybrid Wireless Networks Using Online Optimal Framework," submitted to IEEE Transactions on Vehicular Technology.

Paper 4, pages 148-181, B. Eslamnour, S. Jagannathan and M. J. Zawodniok, "Distributed Cooperative Resource Allocation for Primary and Secondary Users with Adjustable Priorities in Cognitive Radio Networks," submitted to Wireless Networks: The Journal of Mobile Communication, Computation and Information.

ABSTRACT

Widespread use of resource constrained wireless ad hoc networks requires careful management of the network resources in order to maximize the utilization. In cognitive wireless networks, resources such as spectrum, energy, communication links/paths, time, space, modulation scheme, have to be managed to maintain quality of service (QoS). Therefore in the first paper, a distributed dynamic channel allocation scheme is proposed for multi-channel wireless ad hoc networks with single-radio nodes. The proposed learning scheme adapts the probabilities of selecting each channel as a function of the error in the performance index at each step.

Due to frequent changes in topology and flow traffic over time, wireless ad hoc networks require a dynamic routing protocol that adapts to the changes of the network while allocating network resources. In the second paper, approximate dynamic programming (ADP) techniques are utilized to find dynamic routes, while solving discrete-time Hamilton-Jacobi-Bellman (HJB) equation forward-in-time for route cost. The third paper extends the dynamic routing to multi-channel multi-interface networks which are affected by channel uncertainties and fading channels. By the addition of optimization techniques through load balancing over multiple paths and multiple wireless channels, utilization of wireless channels throughout the network is enhanced.

Next in the fourth paper, a decentralized game theoretic approach for resource allocation of the primary and secondary users in a cognitive radio networks is proposed. The priorities of the networks are incorporated in the utility and potential functions which are in turn used for resource allocation. The proposed game can be extended to a game among multiple co-existing networks, each with different priority levels.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Jagannathan Sarangapani, for his guidance and help throughout this research. I have deep appreciation for the advice and direction he gave me, and for his patience and encouragement. I owe my academic progress to his generous spirit.

I would also like to thank my co-advisor, Dr. Maciej Zawodniok, for his valuable help and patience during the course of the research.

I also thank Dr. Ann Miller, Dr. Joe Stanley, and Dr. Sanjay Madria for serving on my committee and for their insightful comments.

I am grateful to Dr. Kelvin Erickson and Dr. Kurt Kosbar for their exceptional support over the course of my education in Missouri S&T.

I would also like to express my gratitude to Dr. Shoukat Ali, my advisor during my M.S. education, and my co-advisor during the first year of my Ph.D. study in Missouri S&T.

This research was supported in part by Intelligent Systems Center, Missouri S&T, Air Force Research laboratory contract through the Center for Aerospace Manufacturing Technologies, and National Science Foundation grant ECCS#0624644.

My special thanks go to Dr. Sahra Sedigh, who has always been there for me, in graduate study and in life.

Last, but not least, I am forever indebted to my parents, Farideh and Abbas, for their unconditional love and support. They are the light of my life.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

I. INTRODUCTION

The explosive growth of wireless networks and the limited spectrum resources requires intelligent and adaptive ways for resource management in wireless networks. Furthermore, measurements have shown that the spectrum is not efficiently utilized. Spectrum utilization can be improved using spatial techniques, frequency, modulation techniques, *etc*. As a consequence, newer concepts such as dynamic spectrum access, and policy-based intelligent radios known as cognitive radios and software-defined radios were introduced [1] to solve these current spectrum inefficiency problems.

NeXt Generation (xG) [2],[3] communication networks, also known as cognitive radio networks, will provide high bandwidth to mobile users through dynamic spectrum access techniques. Opportunistic access to licensed bands improves the spectrum utilization while avoiding interference with the existing primary users. While the conventional definition [4] of the spectrum opportunity is limited to only three dimensions frequency, time and space, the recent advances in technology provide newer dimensions to the spectrum space. The new spectrum hyperspace is defined to have several dimensions as frequency, time, location, angle of arrival, code, and possible others [6]-[8].

Two main characteristics of the cognitive radio can be defined [9],[10]:

• Cognitive capability: Cognitive capability refers to the ability of the radio technology to capture or sense the information from its radio environment. Through this capability, the portions of the spectrum that are unused at a specific time or location can be identified. Consequently, the best spectrum and appropriate operating parameters can be selected.

• Reconfigurability: The cognitive capability provides spectrum awareness while reconfigurability enables the radio to be dynamically programmed according to the radio environment.

Given the characteristics of the cognitive radios, the main functions for cognitive radios can be summarized as follows [7]:

• Spectrum sensing: Detecting unused spectrum and sharing the spectrum without harmful interference with other users. It can be performed via two different architectures: single-radio [11] and dual-radio [12].

• Spectrum management: Capturing the best available spectrum to meet user communication requirements.

• Spectrum mobility: Maintaining seamless communication requirements during the transition to better spectrum.

• Spectrum sharing: Providing the fair spectrum scheduling method among coexisting users.

Spectrum sensing and spectrum sharing can be done in centralized or distributed manner. Another classification for the spectrum sharing is cooperative (or collaborative), or non-cooperative (or non-collaborative, selfish) approaches. Cooperative solutions consider the effect of the node's communication on other nodes [13]-[15]. In other words, the interference measurements of each node are shared among other nodes. Furthermore, the spectrum allocation algorithms also consider this information. Centralized approach is not scalable because in the case of large number of users, the bandwidth required for communication between the cognitive nodes and the central node becomes huge. While

all the centralized solutions can be regarded as cooperative, there also exist distributed cooperative solutions.

Contrary to the cooperative solutions, non-cooperative solutions consider only the node at hand [16]-[18]. While non-cooperative solutions may result in reduced spectrum utilization, the minimal communication requirements among other nodes could be considered as a tradeoff.

The main two functions of cognitive radios, spectrum management and spectrum sharing are resource allocation problems which collect the required information through spectrum sensing. While resource allocation and management in cognitive radios cover a wide dimension of variables (such as frequency, time, location, modulation, angle of arrival, *etc*), multi-channel wireless ad hoc networks also face resource allocation challenge – though in less dimensions. In this dissertation the study begins with resource allocation problem for multi-channel wireless ad hoc networks and gradually move to the broader form of the problem, which is resource allocation for cognitive networks.

In the first paper focuses on channel allocation for wireless ad hoc networks. Although it does not explicitly consider the cognitive networks, the assumptions made for the first paper make it applicable in cognitive networks. In other words, the assumptions in Paper 1 imply that the nodes have cognitive capability (they are able to capture the information from their radio environment) and configurability. Cognitive capability lies within the capability of the nodes in measuring the performance index in different channels. On the other hand, ability of the nodes to switch between the different channels provides the configurability. Finally, the dynamic channel allocation scheme provides the spectrum sharing among the network nodes.

In the second paper, the communication links are considered as network resources, as well as time, location and node resources such as available energy. The dynamic routing problem is treated as a resource allocation problem in which the resource are communication links and node resources in various times and locations. During the course of routing, these resources are optimally allocated to data flows such that the data flows be directed to the destination while the routing (allocation) cost is minimized.

In the third paper, the problem is extended to the multi-channel multi-interface ad hoc networks, while the effect of channel uncertainties and fading channels are also taken into account. By extending the work to multi-channel multi-interface networks, the resource dimensions are increased. Furthermore, by considering the channel uncertainties, the cognitive capability of the nodes becomes more significant. Now the nodes not only have to detect the vacancies in the spectrum, they also have to estimate the channels conditions and quality in the presence of uncertainties. The channel quality information is then used in finding the routes and balancing the load among multiple links/paths.

The fourth paper focuses resource allocation in co-existing cognitive networks where each network has a different priority level for using the resources. In this paper a distributed cooperative game-theoretic approach is proposed in which the priorities are taken into account in the definition of utility functions and potential function. While the primary and secondary network users utilize their ability to sense the spectrum and each try to enhance their QoS, they should also limit the interference that they cause on the other nodes and networks.

1.1. ORGANIZATION OF THE DISSERTATION

The focus of this dissertation is on the resource allocation for cognitive networks (illustrated in Figure 1.1).

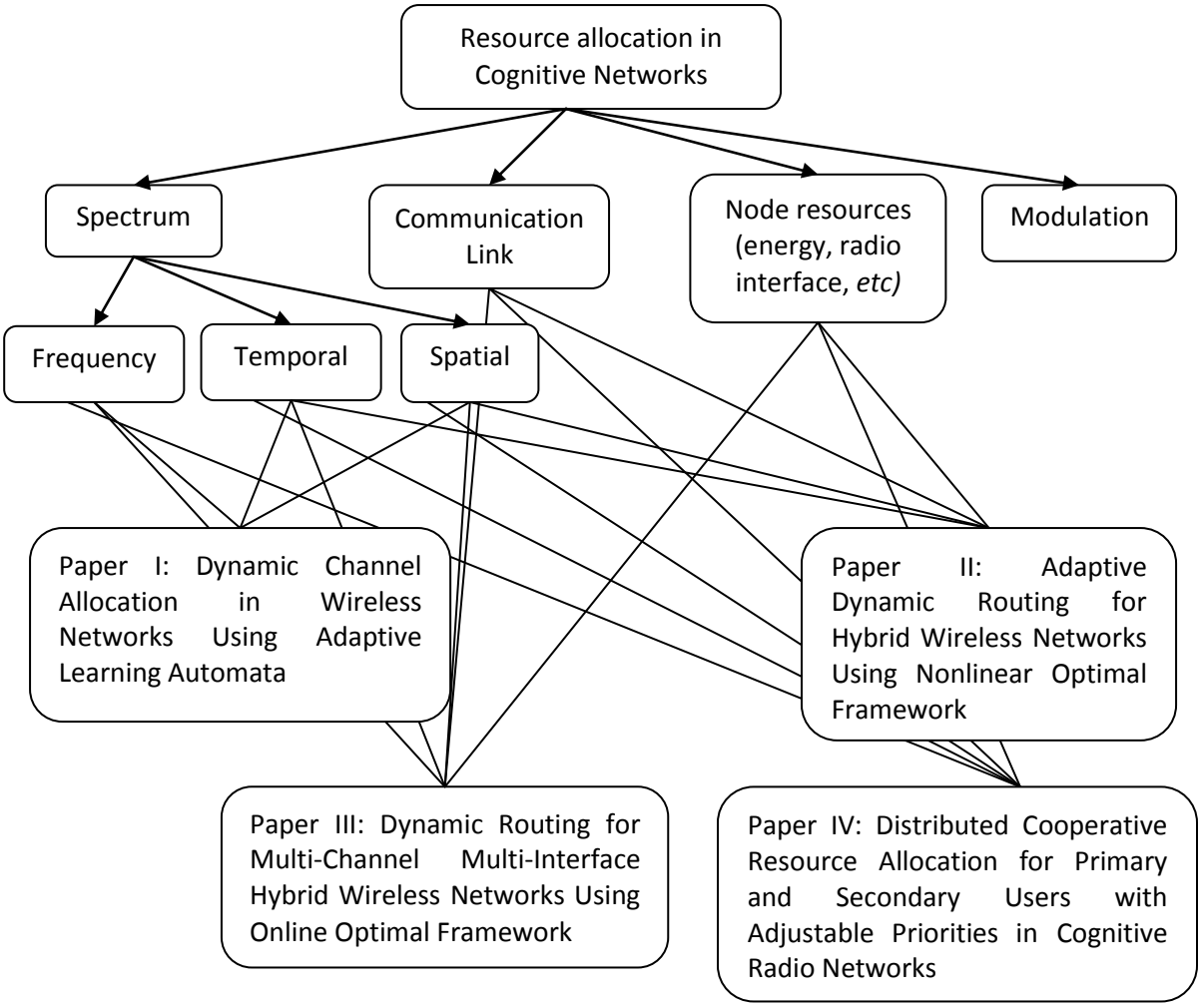

Figure 1.1. Dissertation outline.

First a method for dynamic channel allocation was proposed for wireless networks using adaptive learning automata. While the cognitive radios are not limited to spatial and temporal spectrum utilization, the spatial channel reuse approach in wireless

networks has been vastly investigated [19]-[24]. The bulk of the research on multiple channel allocation is notably done for mesh networks [20],[24], WLANs with infrastructure [21], cellular networks [3] and cognitive radio networks [22]. The multi-channel allocation problem has been investigated for the networks in which the nodes are equipped with either multiple-radio interface [24]-[27] or single-radio interface [19],[21],[28]-[30].

In the networks with infrastructure and access points [21], the channel co-ordination signals are exchanged through the wired distribution system connecting the access points. This practically eliminates the need for periodically switching to a common channel. In the case of multiple-radio interface approach, usually one interface is dedicated to the control signals by tuning on the control channel - eliminating the need for synchronization throughout the network. The remaining channels are allocated to the unused radio interfaces on the nodes for simultaneous transmission of data thus increasing temporal and spatial spectrum utilization.

In order to allocate channels for data transmission, the channel allocation problem has been addressed by many ways. SSCH [19] presents a channel hopping scheme for IEEE 802.11 ad-hoc wireless networks in which the nodes switch among a sequence of channels based on the random seed that has been assigned to the nodes. Unfortunately, the nodes continuously switch between the channels, even if there is no data to transmit. On the other hand, M-MAC [31] is a multi-channel MAC that uses beacons during ATIM windows to find the less occupied channels. Other schemes include graph coloring methods [21] and game theory approaches – either cooperative [22], or non-cooperative [3] which are good alternate approaches whereas they consume more time.

By contrast, Paper 1 proposes a distributed dynamic channel allocation scheme for wireless networks whose nodes are equipped with single-radio interfaces for simplicity. The adaptive pursuit learning algorithm runs periodically on the nodes, and adaptively finds the optimum channel allocation that provides the desired performance (or closest to the desired performance). The periodic nature of this algorithm makes it dynamic and enables the channel allocation to adapt to the topographic changes, possible loss of some channels, mobility of the nodes, and the traffic flow changes. The convergence of the algorithm is guaranteed by selecting a realistic yet desired performance metric. Extensive simulation results in static and mobile environments provide that using the proposed scheme for channel allocation in the multiple channel wireless networks significantly improves the throughput, dropped packet rate, energy consumption per packet and fairness index. The main contribution of the work includes a novel way of selecting channels online by using learning automata via a realistic performance index where the convergence proof is also included.

Paper 2 focuses on adaptive dynamic routing for wireless ad hoc networks. The traditional algorithms [32],[33] construct routing tables by exchanging topological data and discovering shortest paths [34]. However, mobility in an ad hoc network makes the nodes move in and out of range causing frequent updates on the routing tables as well as route selection. Also congestion and sudden node failure can cause route failure and reduced performance. As a consequence, the performance of the network degrades, or the frequently broken links can cause extra routing overhead to re-establish the routes. This excessive overhead due to unreliable links resulting from mobility can cause congestion and increased energy consumption.

Distance Vector Routing (DVR) [32], is a decentralized algorithm which associates a distance vector to each route. The vector magnitude represents the route cost and the direction identifies the next hop to be taken in order to reach the destination. Link State Routing (LSR) [33] creates a complete map of the network at each node. Unlike DSR, LSR requires many more updates in order to converge to a stable routing table. Destination Sequenced Distance Vector Routing or DSDV [34] routing is a derivative of DVR, which maintains a route table containing the cost and next-hop to each destination, and also a sequence number – to prevent routing loops. However, all the mentioned protocols are proactive. Hence, broken links and link failures caused by dynamic topology and channel uncertainties will demand route repair or new route discovery process.

Among on-demand routing protocols, Dynamic Source Routing (DSR) [35], Temporally Ordered Routing Algorithm (TORA) [36], Ad hoc On-demand Distance Vector or AODV [37] (which is a combination of the DSDV and DSR) represent a few of the schemes. When a route is required, the source node broadcasts a request message over the network. Once the destination receives the request, it sends back a route reply, and the path is cached in the reply message. If due to dynamic topology or channel uncertainties, a broken link occurs in a chosen path, the source must retransmit the message through an alternative route. Discovery of the new alternative routes demands extra routing overhead.

In the second paper, an on-demand adaptive dynamic routing protocol is proposed, which adapts the route based on link and channel conditions without flooding the network with new route request messages. Adaptive dynamic programming (ADP)

techniques are utilized to find dynamic routes, while solving discrete-time Hamilton-Jacobi-Bellman equation forward-in-time for route cost in an online manner. It uses a neural network (NN) or any online approximator (OLA) to approximate the route cost to the base station.

While our proposed method is based on dynamic programming, utilizing an OLA differentiates it from traditional dynamic programming approach such as Bellman-Ford. Moreover, it provides reactive and dynamic forward-in-time solutions – as opposed to proactive and static backward-in-time solutions in traditional dynamic programming. Furthermore, metrics such as available bandwidth, link delay, available energy and queue occupancy at nodes are taken into account in determining the route cost. Finally, mathematical analysis is included to show that the proposed approach renders a truly optimal route.

Paper 3 extends the area of our study to multi-channel multi-interface networks in the presence of fading channels and channel uncertainties. In the presence of fading channels, the networks also undergo a variety of changes caused by signal attenuation, interference, Doppler effect, shadowing effect, etc. These channel uncertainties can affect QoS of the network if they are not addressed by the data transmission protocols. Extensive studies on fading channels, interferences and fading models have been made [38]-[43]. Statistical models for Rayleigh fading [38],[39], Rician fading [43], lognormal shadowing [44] have been developed to capture and describe the environment. However, in order to make the analysis possible, a common assumption in the studies is that certain specifications such as mean signal power of both the desired signal and the interferer signals are known – if not variable. Another assumption is that the number of interferer

signals is known, or the speed at which the mobile nodes and their neighbors are moving is known [45] – and constant. As much as these assumptions make the analysis of the wireless channels possible, they cannot be used to develop dynamic and adaptive protocols for the network with varying environment.

In Paper 3, the channel characteristics are considered non-stationary and unknown. By utilizing MLE and Markov model for channel, and finding the time-varying and non-stationary probability outage, the effective available bandwidth is defined. The effective available bandwidth renders a more realistic measure of the channel condition. The outage probability and effective available bandwidth are also used in load balancing over multiple channels and also multiple routes – if one route fails to provide sufficient bandwidth. The method in Paper 2 is extended for the routing protocol by taking into account the link conditions as well as channel uncertainties. Furthermore, by the addition of optimization techniques through load balancing over multiple paths and multiple wireless channels, utilization of wireless channels throughout the network is enhanced.

Paper 4 focuses on resource allocation in co-existing networks exploiting cognitive radios while their access to resources is defined by different priorities. The two distinct characteristics of cognitive radios, cognitivity and configurability, enable the cognitive radios to sense the dynamic environment and adapt their parameters to optimize their performance. Game theoretic approach is one of the ways to perform resource allocation, organize the networks and their users' behavior within the network and also with respect to the co-existing networks [46]-[52],[53],[54]. Existing game theoretic approaches such as [52] and [53] have taken a non-cooperative approach to power, joint power and rate control with Quality of Service (QoS) constraints, respectively.

By contrast, a cooperative approach was taken in [47] to maximize the channel capacity. In [51], a framework for cooperation among the primary and secondary users is introduced. An interference avoidance protocol for peer-to-peer networks was presented in [48] that assumed either the network is centralized, or the receivers are co-located. Authors in [49],[51],[53] have provided the presence of a unique NE. However, the convergence of the combined game with primary and secondary users is not shown since finding a utility function that provides a unique NE for the combined game is rather difficult. Others assumed a single game with homogeneous players [48], or only consider a single PU node and a set of SU players [47],[49].

In the fourth paper, a decentralized game theoretic approach for resource allocation of the primary and secondary users in a cognitive radio network is proposed by relaxing these assumptions and defining priority levels. The priorities of the networks are incorporated in the proposed utility and potential functions. It is demonstrated analytically and through simulation studies that a unique NE exists for the combined game with primary users and secondary users (SU). Since the interaction among the networks and their priority levels are incorporated in the functions, the proposed game can be extended to a game among multiple co-existing networks, each with different priority levels.

1.2. CONTRIBUTIONS OF THE DISSERTATION

This dissertation provides contributions to the resource allocation problem in wireless ad hoc networks. The main contribution of the first paper includes a novel way of selecting channels online by using learning automata via a realistic performance index where the convergence proof is also included.

The contribution of the second paper is proposing an on-demand adaptive dynamic routing protocol which adapts the route based on link and channel conditions without flooding the network with new route request messages. While our proposed method is based on Bellman-Ford dynamic programming, utilizing online approximator differentiates it from dynamic programming. It provides reactive and dynamic forward-in-time solutions – as opposed to proactive and static backward-in-time solutions in dynamic programming.

The third paper extends the second paper to multiple-channel multiple-interface wireless networks with channel uncertainties. Time-varying outage probability for the channel links is introduced and determined by Markov models. Furthermore, channel estimation was utilized in load balancing over multiple links/channels in addition to determining the adaptive routes.

In the fourth paper a decentralized game theoretic approach for resource allocation of the primary and secondary users in a cognitive radio network is proposed. The contributions of this work include incorporation of priority levels of the networks in the utility and potential functions which are in turn used for resource allocation. It also provides analytical proofs of existence and uniqueness of NE for the PU and SUs by using these functions with priority levels, and existence of the NE for the combined game. The net result is a game theoretic approach applicable to multiple co-existing networks.

1.3. REFERENCES

[1] J. Mitola III., "Cognitive radio: an integrated agent architecture for software defined radio." PhD thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2000.

[2] DARPA XG WG, The XG Architectural Framework V1.0, 2003.

[3] DARPA XG WG, The XG Vision RFC V1.0, 2003.

[4] P. Kolodzy et al., "Next generation communications: Kickoff meeting," In *Proc. DARPA*, Oct. 2001.

[5] R. Matheson, "The electrospace model as a frequency management tool," in *Int. Symposium On Advanced Radio Technologies*, Boulder, Colorado, USA, Mar. 2003, pp. 126–132.

[6] A. L. Drozd, I. P. Kasperovich, C. E. Carroll, and A. C. Blackburn, "Computational electromagnetics applied to analyzing the efficient utilization of the RF transmission hyperspace," in *Proc. IEEE/ACES Int. Conf. on Wireless Communications and Applied Computational Electromagnetics*, Honolulu, Hawaii, USA, Apr. 2005, pp. 1077–1085.

[7] I. F. Akyildiz, W. Lee, M. C. Vuran and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey," Comput. Netw. 50, 13 (Sep. 2006), 2127-2159.

[8] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *Communications Surveys & Tutorials, IEEE* , vol.11, no.1, pp.116-130, First Quarter 2009.

[9] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications* 23 (2) (2005) 201–220.

[10] R.W. Thomas, L.A. DaSilva, A.B. MacKenzie, "Cognitive networks," in *Proc. IEEE DySPAN* 2005, November 2005, pp. 352–360.

[11] G. Vardoulias, J. Faroughi-Esfahani, G. Clemo, and R. Haines, "Blind radio access technology discovery and monitoring for software defined radio communication systems: problems and techniques," in *Proc. Int. Conf. 3G Mobile Communication Technologies*, London, UK, Mar. 2001, pp. 306–310.

[12] S. Shankar, C. Cordeiro, and K. Challapali, "Spectrum agile radios: utilization and sensing architectures," in *Proc. IEEE Int. Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Baltimore, Maryland, USA, Nov. 2005, pp. 160–169.

[13] V. Brik, E. Rozner, S. Banarjee and P. Bahl, "DSAP: a protocol for coordinated spectrum access," in *Proc. IEEE DySPAN* 2005, November 2005, pp. 611–614.

[14] L. Cao and H. Zheng, "Distributed spectrum allocation via local bargaining," in *Proc. IEEE Sensor and Ad Hoc Communications and Networks* (SECON) 2005, September 2005, pp. 475–486.

[15] J. Huang, R.A. Berry and M.L. Honig, "Spectrum sharing with distributed interference compensation," in *Proc. IEEE DySPAN* 2005, November 2005, pp. 88–93.

[16] S. Sankaranarayanan, P. Papadimitratos, A. Mishra and S. Hershey, "A bandwidth sharing approach to improve licensed spectrum utilization," in *Proc. IEEE DySPAN 2005, November* 2005, pp. 279–288.

[17] Q. Zhao, L. Tong and A. Swami, "Decentralized cognitive MAC for dynamic spectrum access," in *Proc. IEEE DySPAN* 2005, November 2005, pp. 224–232.

[18] H. Zheng and L. Cao, "Device-centric spectrum management," in *Proc. IEEE DySPAN* 2005, November 2005, pp. 56–65.

[19] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," In *Proceedings of the 10th Annual international Conference on Mobile Computing and Networking* (Philadelphia, PA, USA, September 26 - October 01, 2004). MobiCom '04. ACM, New York, NY, pp:216-230.

[20] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," In *Proceedings of the 11th Annual international Conference on Mobile Computing and Networking* (Cologne, Germany, August 28 - September 02, 2005). MobiCom '05. ACM, New York, NY, pp.58-72.

[21] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for WLANs," *SIGMOBILE Mob. Comput. Commun. Rev.* 9, 3 (Jul. 2005), pp: 19-31.

[22] N. Nie and C. Comaniciu, "Adaptive channel allocation spectrum etiquette for cognitive radio networks," *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, vol., no., pp.269-278, 8-11 Nov. 2005.

[23] J. Li, D. Chen, W. Li, and J. Ma, "Multiuser power and channel allocation algorithm in cognitive radio," *Parallel Processing, 2007. ICPP 2007. International Conference on* , vol., no., pp.72-72, 10-14 Sept. 2007.

[24] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh network," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 2, pp: 50-65, Apr. 2004.

[25] M. Felegyhazi, M. Cagalj, S. Bidokhti, and J.-P Hubaux, "Non-cooperative multi-radio channel allocation in wireless networks," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE,* vol., no., pp.1442-1450, May 2007.

[26] P. Kyasanur and N. H. Vaidya, "Routing in multi-channel multi-interface ad hoc wireless networks," technical report, Dec. 2004.

[27] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," Proc. of *Wireless Communications and Networking Conference,* vol.4, no., pp. 2051-2056 Vol. 4, 13-17 March 2005.

[28] J. So and N. Vaidya, "Routing and channel assignment in multi-channel multi-hop wireless networks with single-NIC devices," *Technical Report, University of Illinois at Urbana Champaign,* Dec. 2004.

[29] Z. Han, Z. Ji, and K.J.R. Liu, "Fair multiuser channel allocation for OFDMA networks using Nash bargaining solutions and coalitions," *Communications, IEEE Transactions on,* vol.53, no.8, pp. 1366-1376, Aug. 2005.

[30] J.A. Patel, H. Luo and I. Gupta, "A cross-layer architecture to exploit multi-channel diversity with a single transceiver," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, vol., no., pp.2261-2265, May 2007.

[31] J. So and N.H. Vaidya, "Multi-channel MAC for ad-hoc networks: handling multi-channel hidden terminals using a single transceiver," In *Proceedings of the 5th ACM international Symposium on Mobile Ad Hoc Networking and Computing*. MobiHoc '04. ACM, New York, NY.

[32] C. Hedrick, "Routing Information Protocol," Internet-Request for Comments 1058, IETF, June 1988.

[33] J. Moy, OSPF Version 2. RFC 1583, Network Working Group, March 1994.

[34] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," In *ACM SIGCOMM* '94, pages 234-244, 1994.

[35] D.B. Johnson, D.A. Maltz, Y. Hu, and J.G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)" Internet-Draft, IETF, November 2001. draft-ietfmanet-dsr-06.txt.

[36] V. Park and M. Corson, Temporally-Ordered Routing Algorithm (TORA): Version 1 Functional Specification. Internet-Draft, IETF, July 2001. draft-ietf-manet-tora-spec-04.txt.

[37] C. Perkins, E. Royer, and S. R. Das., "Ad hoc On-Demand Distance Vector (AODV) Routing," Internet-Draft, IETF, January 2002. draft-ietf-manet-aodv-10.txt.

[38] Y.-D. Yao and A.U.H. Sheikh, "Investigations into cochannel interference in microcellular mobile radio systems," *IEEE Trans. Veh. Technol*., vol.41, no.2, pp.114-123, May 1992.

[39] R. C. French, "The effect of fading and shadowing on channel reuse in mobile radio," *IEEE Trans. Veh. Technol*., vol. VT-28, pp. 171-181, 1979.

[40] H. Bischl and E. Lutz, "Packet error rate in the non-interleaved Rayleigh channel," *Communications, IEEE Transactions on* , vol.43, no.234, pp.1375-1382, Feb/Mar/Apr 1995.

[41] E. N. Gilbert, "Capacity of burst noise channels," Bell Syst. Tech. J., VO~39. , pp. 1253-1256, 1960.

[42] N. Weinberger and M. Feder, "Universal decoding for linear Gaussian fading channels in the competitive minimax sense," *Information Theory*, 2008. ISIT 2008. *IEEE International Symposium on* , vol., no., pp.782-786, 6-11 July 2008.

[43] L. Wang and Y. Cheng, "A statistical mobile-to-mobile Rician fading channel model," *Vehicular Technology Conference*, 2005. VTC 2005-Spring. 2005 IEEE 61st , vol.1, no., pp. 63- 67 Vol. 1, 30 May-1 June 2005.

[44] A.J. Coulson, A.G. Williamson, and R.G. Vaughan, "A statistical basis for lognormal shadowing effects in multipath fading channels," *IEEE Transactions on Communications*, vol.46, no.4, pp.494-502, Apr 1998.

[45] W. H. Shen and N. Moayeri, "Finite-state Markov channel-a useful model for radio communication channels," *IEEE Trans. Veh. Technol*., vol.44, no.1, pp.163-171, Feb 1995.

[46] B. Wang, Y. Wu, and K.J. R. Liu, "Game theory for cognitive radio networks: An overview," *Computer Networks,* (2010), doi:10.1016/j.comnet.2010.04.004.

[47] L. Giupponi and C. Ibars, "Distributed cooperation among cognitive radios with complete and incomplete information," *EURASIP Journal on Advances in Signal Processing*, Article ID 905185, 13 pages, 2009.

[48] R. Menon, A.B. MacKenzie, R.M. Buehrer, and J.H. Reed, "WSN15-4: A game-theoretic framework for interference avoidance in ad hoc networks," *Global Telecommunications Conference, GLOBECOM '06. IEEE* , vol., no., pp.1-6, 2006.

[49] Z. Zhang; J. Shi, H.-H. Chen, M. Guizani, and P. Qiu; , "A cooperation strategy based on Nash bargaining solution in cooperative relay networks," *Vehicular Technology, IEEE Transactions on* , vol.57, no.4, pp.2570-2577, July 2008.

[50] L. Tianming and S.K. Jayaweera, "A novel primary-secondary user power control game for cognitive radios with linear receivers," *Military Communications Conference, 2008. MILCOM 2008. IEEE* , vol., no., pp.1-7, 16-19 Nov. 2008.

[51] C. Ghosh, D.P. Agrawal, and M.B. Rao, "Channel capacity maximization in cooperative cognitive radio networks using game theory," *SIGMOBILE Mob. Comput. Commun. Rev.* 13, 2 (Sep. 2009), 2-13.

[52] C.U. Saraydar, N.B. Mandayam, and D.J. Goodman, "Efficient power control via pricing in wireless data networks," *Communications, IEEE Transactions on* , vol.50, no.2, pp.291-303, Feb 2002.

[53] F. Meshkati, H.V. Poor, S.C. Schwartz, and R.V. Balan, "Energy-efficient resource allocation in wireless networks with quality-of-service constraints," *Communications, IEEE Transactions on* , vol.57, no.11, pp.3406-3414, Nov. 2009.

[54] R.D. Yates, "A framework for uplink power control in cellular radio systems," *Selected Areas in Communications, IEEE Journal on* , vol.13, no.7, pp.1341-1347, Sep 1995.

PAPER

1. Dynamic Channel Allocation in Wireless Networks Using Adaptive Learning Automata[1,2]

Behdis Eslamnour[3], S. Jagannathan[4] and Maciej J. Zawodniok[5]

***Abstract*—** Single-channel based wireless networks have limited bandwidth and throughput and the bandwidth utilization decreases with increased number of users. To mitigate this problem, simultaneous transmission on multiple channels is considered as an option. This paper proposes a distributed dynamic channel allocation scheme using adaptive learning automata for wireless networks whose nodes are equipped with single-radio interfaces. The proposed scheme, Adaptive Pursuit learning automata runs periodically on the nodes, and adaptively finds the suitable channel allocation in order to attain a desired performance. A novel performance index, which takes into account the throughput and the energy consumption, is considered. The proposed learning scheme adapts the probabilities of selecting each channel as a function of the error in the performance index at each step. The extensive simulation results in static and mobile

---

[1] A conference version of this paper: B. Eslamnour, M. Zawodniok, S. Jagannathan, "Dynamic Channel Allocation in Wireless Networks Using Adaptive Learning Automata," was presented at *Wireless Communications and Networking Conference, WCNC'09.*

[2] This journal version has been submitted to the International Journal of Wireless Information Networks.

[3] Behdis Eslamnour(✉), S. Jagannathan, and Maciej J. Zawodniok
Department of Electrical and Computer Engineering
Missouri University of Science and Technology (formerly University of Missouri-Rolla)
Rolla, MO 65409, USA, e-mail: ben88@mail.mst.edu

[4] S. Jagannathan, e-mail: sarangap@mst.edu

[5] Maciej J. Zawodniok, e-mail: mjzx9c@mst.edu

environments provide that the proposed channel allocation schemes in the multiple channel wireless networks significantly improves the throughput, drop rate, energy consumption per packet and fairness index – compared to the 802.11 single-channel, and 802.11 with randomly allocated multiple channels. Also, it was demonstrated that the Adaptive Pursuit Reward-Only scheme guarantees updating the probability of the channel selection for all the links – even the links whose current channel allocations do not provide a satisfactory performance - thereby reducing the frequent channel switching of the links that cannot achieve the desired performance.

Keywords **—** Channel Allocation, Energy Efficiency, Fairness Index, Learning Automata, Multiple Channel MAC, Wireless Sensor Networks.

I. NOMENCLATURE

Symbol	Definition
N	number of channels
C	set of available channels, $C=\{c_1,c_2,\ldots c_N\}$
$p_i^j(k)$	probability of node i selecting channel j at time k, $\sum_{j=1}^{N} p_i^j(k) = 1$
$\mathbf{P}_i(k)$	probability vector of node i selecting any of the N channels at time k, $\mathbf{P}_i(k)=[p_i^1(k), p_i^2(k), \ldots, p_i^N(k)]$
$\beta_i^j(k)$	environment response at time k for selecting channel j by node i $\begin{cases} \text{if } \beta_i^j(k)=0, \text{the autumaton will be rewarded} \\ \text{if } \beta_i^j(k)=1, \text{the automaton will not be rewarded} \end{cases}$
$\hat{H}_i^j(k)$	average estimated throughput (percentage of successful transmissions) at time k over a window of M for channel j at node i $\hat{H}_i^j(k) = \frac{1}{M} \sum_{n=L_i^j(k)-M+1}^{L_i^j(k)} J_i^j(n)$

Symbol	Definition
$J_i^j(k)$	percentage of successful transmissions at time k over at node i if channel j is selected
$L_i^j(k)$	number of times that channel j was selected for node i from time 0 till k
$\hat{E}_i^j(k)$	average estimated consumed energy per packet at time k over a window of M for channel j at node i $\hat{E}_i^j(k) = \frac{1}{M} \sum_{n=L_i^j(k)-M+1}^{L_i^j(k)} e_i^j(k)$
ϕ^*	desired performance (joules/packet)$^{-1}$ $\phi^* = \left(\frac{H}{E}\right)_{desired}$
$\hat{\phi}_i^j(k)$	estimated performance of channel j at time k for node i $\hat{\phi}_i^j(k) = \frac{\hat{H}_i^j(k)}{\hat{E}_i^j(k)}$
$\hat{m}_i$	index of the channel that provides the maximum estimated performance for node i at time k $\hat{m}_i = \arg \max_j \hat{\phi}_i^j(k)$

II. INTRODUCTION

It is widely believed that the wireless networks are being limited by the lack of the available spectrum, and at the same time the spectrum is not efficiently utilized. Spectrum utilization can be improved using spatial techniques, frequency, modulation techniques, *etc*. As a consequence, newer concepts such as software-defined radios and cognitive radios were made possible [1]. While the cognitive radios are not limited to spatial and temporal spectrum utilization, the spatial channel reuse approach in wireless networks has been vastly investigated [2]-[7].

The bulk of the research on multiple channel allocation is notably done for mesh networks [3],[7], WLANs with infrastructure [4], cellular networks [3] and cognitive radio networks [5]. The multi-channel allocation problem has been investigated for the networks in which the nodes are equipped with either multiple-radio interface [7]-[10] or single-radio interface [2],[4],[11]-[13]. In the single-radio approach, the radios switch between the channels frequently in order to minimize interference and collision between the simultaneous transmissions in the same communication range. Usually in this approach, all the nodes periodically switch to a common channel for channel co-ordination, and then switch to different data channels to conduct the simultaneous transmissions. Therefore the switching delay (80-100 μs [2]) becomes one of the overheads increasing the network end-to-end delay. Additionally, synchronization is required in these schemes.

In the networks with infrastructure and access points [4], the channel co-ordination signals are exchanged through the wired distribution system connecting the access points. This practically eliminates the need for periodically switching to a

common channel. In the case of multiple-radio interface approach, usually one interface is dedicated to the control signals by tuning on the control channel. The remaining channels are allocated to the unused radio interfaces on the nodes for simultaneous transmission of data thus increasing temporal and spatial spectrum utilization. Dedicating one radio interface on each node for the control signals eliminates the need for synchronization throughout the network. Further, utilizing multiple radios reduces the need for frequent channel switching, and hence the switching overhead is significantly less than that in the single-radio approach. However, the cost of additional radios and their energy consumption must be taken into account.

In order to allocate channels for data transmission, the channel allocation problem has been addressed by many ways. SSCH [2] presents a channel hopping scheme for IEEE 802.11 ad-hoc wireless networks in which the nodes switch among a sequence of channels based on the random seed that has been assigned to the nodes. Unfortunately, the nodes continuously switch between the channels, even if there is no data to transmit. On the other hand, M-MAC [14] is a multi-channel MAC that uses beacons during ATIM windows to find the less occupied channels. Other schemes include graph coloring methods [4] and game theory approaches – either cooperative [5], or non-cooperative [3] which are good alternate approaches whereas they consume more time.

By contrast, in this paper, our previously proposed distributed dynamic channel allocation scheme [15] is extended for wireless networks whose nodes are equipped with single-radio interfaces for simplicity. This scheme can be particularly suitable for wireless sensor networks since due to the low cost and low energy consumption requirements, the nodes in wireless sensor networks rarely have the luxury of having

multiple radios. The adaptive pursuit learning algorithm runs periodically on the nodes, and adaptively finds the optimum channel allocation that provides the desired performance (or closest to the desired performance). Since this paper considers using single-radio interfaces on the nodes, and the algorithm is periodic, synchronization is required in this scheme. The periodic nature of this algorithm makes it dynamic and enables the channel allocation to adapt to the topographic changes, possible loss of some channels, mobility of the nodes, and the traffic flow changes. Unlike the linear and nonlinear schemes in which the reward and penalty values were functions of the probabilities of taking an action (in this case selecting a channel), this study examines an adaptive updating scheme in which the reward and penalty values are functions of the error between the desired and the estimated performance of the current channel allocation. The convergence of the algorithm is guaranteed by selecting a realistic yet desired performance metric. Extensive simulation results in static and mobile environments provide that using the proposed scheme for channel allocation in the multiple channel wireless networks significantly improves the throughput, dropped packet rate, energy consumption per packet and fairness index – compared to the 802.11 single-channel, and 802.11 randomly allocated multiple channels. The main contribution of the work includes a novel way of selecting channels online by using learning automata via a realistic performance index where the convergence proof is also included.

In Section III, the methodology and algorithms are presented. Simulation results and discussions are provided in Section IV. Section V concludes the paper. Proof of convergence of the algorithm is presented in Appendix A.

III. METHODOLOGY AND ALGORITHM

A. Methodology

In the proposed algorithm, the nodes periodically switch between the control hop, T_c, and data transmission hop, T_d (Figure 1). Each data transmission period, T_d, is comprised of the individual time slots, T_s. As an initial assumption, the network is assumed to be a peer-to-peer network in which all nodes are equipped with a single-radio interface. It is also assumed that routes have been established by a proactive routing protocol such as optimal link state routing (OLSR) [16] or optimal energy delay routing (OEDR) [17]. During T_c, all nodes are on one common channel to communicate the control signals. It is possible that one or more of the channels get highly affected by external interference and the network would lose these channels temporarily or permanently.

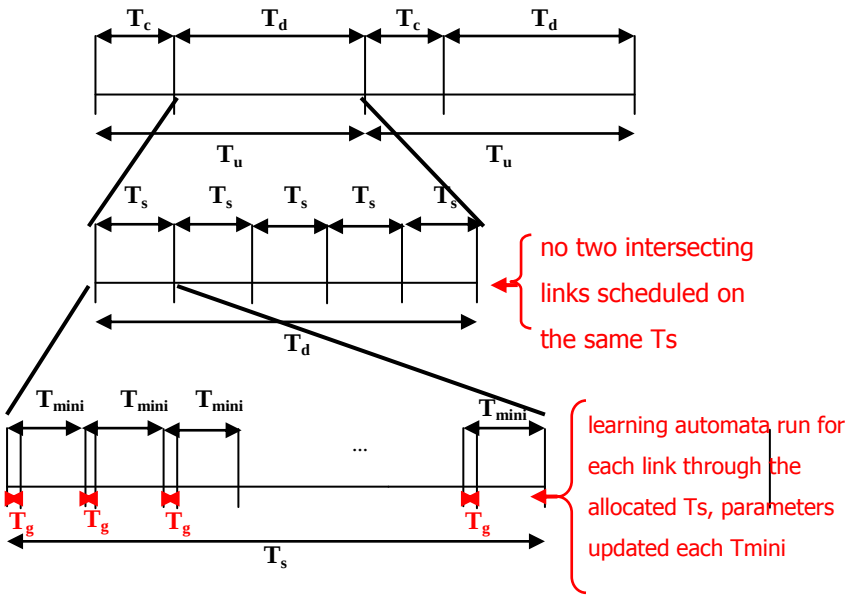

Figure 1. The two periods of control and data, and time slots within the data transmission period.

Any eligible external node that tries to join the network would send out join-request signals periodically on all the channels and listen in the intervals. It would be able to join the network during one of the T_c periods, and obtain the sequence and other necessary information about the network. The control signal also carries the schedule of the time slots for the links in the subsequent data transmission period. During the time scheduling, groups of non-intersecting links are scheduled for each T_s time slot. Also broadcast communications and route discovery are performed during T_c period.

After the T_c hop, the data transmission hop, T_d, begins. During each T_s time slot of T_d, channels are allocated to the links previously assigned to the T_s. The channel allocation algorithm is an iterative algorithm during which the channel allocation is refined. Due to the iterative nature of the algorithm, each T_s is divided into smaller time slots, T_{mini}, separated by T_g – guard bands. The probabilities of selecting the channels and parameters of the channel allocation algorithm are updated for each link at each T_{mini}. By periodically repeating the T_c and T_d hops, the channel allocation becomes dynamic. In addition, the network can adapt to the topographic changes, mobility of the nodes, and the changes in the traffic flow.

We also decided to use the control channel as one of the available channels for data transmission during the T_d period [18]. By utilizing this additional channel during T_d instead of dedicating it to the control signals and using it only during T_c, the spectrum utilization can be increased.

B. Algorithm

During each T_s, the learning algorithm is run on each transmitter node, *i*, separately. We first use the Adaptive Pursuit Reward-Inaction (PRI) which is an extended

version of Discretized PRI [19],[20]. Unlike the DPRI, in the Adaptive PRI scheme the update value, $\theta(k)$, of the channel selection probabilities is not a constant anymore. The update value of the probability is now a function of the error, $\Delta(k)$, of the performance metric. DPRI algorithm was chosen because of the faster convergence provided by it [19]. The Adaptive PRI algorithm is presented in Section 1). However, it appears that depending on the conditions that determine whether the environment response is satisfactory or unsatisfactory, the channel allocation on some links might always result in unsatisfactory response. This would result in 'left-out' links, whose channel selection probabilities are not updated due to the 'reward' property of the algorithm.

In order to eliminate this issue, we proposed and examined the Adaptive Pursuit Reward-Penalty (PRP) learning scheme. The 'reward' behavior of this scheme is the same as the Adaptive PRI. On the other hand, in the case of unsatisfactory environment response for a channel selection, the probability of selecting that channel (if that channel is not the channel with the highest performance among the channels) is decreased, and the probabilities of selecting the other channels are increased. The algorithm is presented in Section *B.2)*. Although this scheme eliminates the 'left-out' links problem, it has a rather slower convergence because of increasing the probabilities of some of the non-optimal channels in the 'penalty' scheme.

In the third effort, we proposed and examined using an Adaptive Pursuit Reward-Only (PRO) learning algorithm. This algorithm still uses a desired value of the performance for determining the magnitude of the update step in the probabilities, but it no longer uses the concept of 'satisfactory' or 'unsatisfactory' environment response. In

other words, the Adaptive PRO is the same as the Adaptive PRI, but the probabilities are guaranteed to be updated in a 'pursuit reward' manner at each iteration.

The performance metric of the network used in this paper was defined as

$$\phi^* = \left(\frac{H}{E}\right)_{desired} \tag{1}$$

where H is the desired percentage of the successful transmissions and E refers to the desired consumed energy per packet transmission. By this definition, the unit of the performance metric ϕ^* becomes packets/joule. Therefore, by selecting a realistic desired performance metric, the objective is to find the optimum channel allocation that provides a higher performance in terms of throughput defined in terms of a target value. A large value of ϕ^* indicates a large number of successful transmission of packets per joule. Hence, this performance metric covers both the throughput and the energy efficiency of the network.

1) The Adaptive Pursuit Reward-Inaction Algorithm

The steps of the Adaptive PRI, which runs on each individual link, are summarized as:

1) Initially, the probability of selecting any of the channels, $p_i^j(0)$, is set to $1/N$.

2) Select a channel using the probability $p_i^j(k)$. Transmit packets during the transmission interval.

3) Based on the measured feedback, update $J_i^j(n)$, $L_i^j(k)$ and $e_i^j(k)$.

4) If $L_i^j(k) \geq M$ (*i.e.* if node *i* has selected channel *j* more than *M* times so far), update $\hat{H}_i^j(k)$, $\hat{E}_i^j(k)$ and $\hat{\phi}_i^j(k)$ and continue on step 5. Otherwise, go to step 7.

$$5)\ \beta_i^j(k) = \begin{cases} 0, & \text{if } \dfrac{\phi^* - \hat{\phi}_i^j(k)}{\phi^*} < \delta \ \ (\textit{satisfactory response}) \\ 1 & \text{otherwise } \ (\textit{unsatisfactory response}) \end{cases} \tag{2}$$

6) Detect the channel index, $\hat{m}_i$, that provides the best estimated performance, $\hat{\phi}_i^j(k)$.

Update the probabilities if the environmental response was satisfactory, i.e. if $\beta_i^j(k) = 0$,

$$\begin{cases} p_i^{\hat{m}_i}(k+1) = 1 - \sum_{q=1, q\neq\hat{m}_i}^{N} p_i^q(k+1) \\ p_i^l(k+1) = \max(\, p_i^l(k) - \theta(k), \eta) \quad \forall l \neq \hat{m}_i \end{cases} \tag{3}$$

where $\theta(k) = \begin{cases} \gamma \cdot \left|\Delta(k)\right| / \phi^*, & \text{if } -\delta < \Delta(k)/\phi^* \\ \lambda \cdot \left|\Delta(k)\right| / \phi^* & \text{otherwise} \end{cases}$, such that $0 \le \theta(k) < 1$ and

$\Delta(k) = \phi^* - \hat{\phi}_i^j(k)$,

where η[6], the minimum possible probability of selecting a channel is chosen such that it guarantees all the channels be selected for a minimum certain number of times, K_I, during a certain number of iterations, M_I. This would keep the estimated values of channel performance up-to-date.

7) Continue to the next iteration, step 2.

2) The Adaptive Pursuit Reward-Penalty Algorithm

The steps in the Pursuit Reward-Penalty learning algorithm are the same as the steps in the Pursuit Reward-Inaction, except for Step 6 – the update law. In this step, when the environmental response is not satisfactory, the probability of selecting the

---

[6] The minimum probability of selecting a channel is determined such that it satisfies the non-equality below.
Pr{channel i being selected at least K_I times over M_I iterations} $\ge \rho$.

This implies that $\sum_{j=K_I}^{M_I} C(M_I, j).\eta^j.(1-\eta)^{M_I - j} \ge \rho$, where $M_I \ge K_I.N$ (N is the number of available channels).

current channel is reduced, and the probability of the other channels are increased as follows.

Detect the channel index, $\hat{m}_i$, that provides the best estimated performance, $\hat{\phi}_i^j(k)$. If the environmental response was satisfactory, i.e. $\beta_i^j(k) = 0$,

$$\begin{cases} p_i^{\hat{m}_i}(k+1) = 1 - \sum_{q=1, q\neq\hat{m}_i}^{N} p_i^q(k+1) \\ p_i^l(k+1) = \max(p_i^l(k) - \theta(k), \eta) \quad \forall l \neq \hat{m}_i \end{cases} \tag{4}$$

If the environmental response was unsatisfactory, i.e. $\beta_i^j(k) = 1$, and $j \neq \hat{m}_i$, which is

$$\begin{cases} p_i^j(k+1) = \max(1 - \sum_{q=1, q\neq j}^{N} p_i^q(k+1), \eta) \\ p_i^l(k+1) = \min(p_i^l(k) + \theta(k)/(N-1), 1-(N-1).\eta) \quad \forall l \neq j \end{cases} \tag{5}$$

3) The Adaptive Pursuit Reward-Only Algorithm

The steps in the Pursuit Reward-Penalty learning algorithm are the same as the steps in the Pursuit Reward-Inaction, except for Step 6 – the update law. In this scheme, the probabilities are updated such that selecting the channel with the highest performance is "pursued." This update is performed regardless of the "satisfactory" or "unsatisfactory" condition of the performance. In other words, we want to increase the probability of selecting the channel which provides the highest performance – even if this performance is less that the desired performance. However, the magnitude of the update step is determined by the relative error of the performance, $|\Delta(k)|/\phi^*$. The update law, Step 6, of the algorithm is as follows.

6) Detect the channel index, $\hat{m}_i$, that provides the best estimated performance, $\hat{\phi}_i^j(k)$. Update the probabilities regardless of the environmental response. The probability of

selecting the channel that provides the highest performance is increased and the probabilities of the other channels are reduced as following.

$$\begin{cases} p_i^{\hat{m}_i}(k+1) = 1 - \sum_{q=1, q\neq\hat{m}_i}^{N} p_i^q(k+1) \\ p_i^l(k+1) = \max(p_i^l(k) - \theta(k), \eta) \quad \forall l \neq \hat{m}_i \end{cases} \tag{6}$$

IV. Simulation Results and Discussions

This section presents the numerical results of running the learning algorithms in two different cases – static and mobile. It also compares the performance of these algorithms to the performance of the standard single-channel 802.11 and randomly allocated multi-channel 802.11. The simulations were performed using network simulator NS-2 where the simulator was modified to implement the three learning algorithms: Adaptive PRI, Adaptive PRP, and Adaptive PRO. The networks considered include peer-to-peer wireless networks with varying traffic, mobility, and node density. Traffic is generated by constant bit rate (CBR) sources with data rates equal to 2 Mbps and packet size equal to 1024 bytes. For each case, the simulation was repeated using 10 random scenarios, and the average values of the 10 trials were used in the analysis.

The simulations considered networks with up to eleven orthogonal channels whose bandwidths are set to 11 Mbps each. The objective of the multi-channel protocol is to allocate the available channels to the links such that the performance converges to a desired value as defined in Equation (1). The target value ϕ^* and the update parameters were set for various scenarios such that the desired performance is achievable. The nodes start without preferred channel and switch between channels until they find the one that provides the desired performance. The estimated performance of selecting each channel,

j, by each node, *i*, is averaged over 5 iterations (5 T_{mini}s) of selecting channel *j* by node *i*. This means that the width of the moving average window, *M*, was selected to be 5 iterations. The updating phase of the algorithm does not start at each node until each channel has been selected *M* times by the node. Therefore, before starting the update phase, at least $M \times N$ T_{mini}s (*N*: number of channels) are required. A larger *M* would give a better estimate, but at the same time it would increase the required time prior to the updating phase of the algorithm.

A. Static Scenario – starting flows at different times

This simulation scenario considers single time-slot duration, T_s, where all nodes are contending for the channels. The network consisted of 50 peer-to-peer single-radio wireless nodes located in an area of 100m×100m, while the communication range of the nodes is limited to 250m. As a result, a dense network topology is created where a single channel may not provide a sufficient quality of service (QoS). The three adaptive learning algorithms were run on the networks with up to eleven orthogonal channels. Three flows start at time t = 2 sec, then seven more flows start at time t = 3 sec, and finally an additional fifteen flows start at time t = 4 sec. The standard 802.11 protocol was also executed on the networks to contrast its performance with that of the proposed scheme. This was accomplished by *a)* using a single channel, and *b)* using 10 channels that were randomly allocated to the links. For each case, the simulation was repeated using 10 random scenarios, and the average of the 10 repeated trials were used in the analysis of the results. The achieved throughput is presented in Table I after using the different methods mentioned above.

Table I. Throughput, drop rate, energy consumption, and fairness index of the network with different channel allocation schemes.

	Throughput (Mbps)			Drop rate(Mbps)			Energy consumption (joules/packet)			Fairness index		
	3 flows	10 flows	25 flows	3 flows	10 flows	25 flows	3 flows	10 flows	25 flows	3 flows	10 flows	25 flows
802.11 – single data channel	4.20	3.89	3.00	0.77	15.98	47.00	0.00215	0.00807	0.01969	0.8028	0.4443	0.2157
Adaptive PRI – 2 data channels	6.15	8.25	7.83	0	11.75	42.94	0.00140	0.00331	0.00774	0.9620	0.7344	0.4082
Adaptive PRI – 3 data channels	6.12	12.44	12.19	0	5.82	38.80	0.00125	0.00235	0.00521	0.9716	0.8337	0.5129
Adaptive PRI – 6 data channels	6.10	19.35	24.80	0	0.26	26.13	0.00114	0.00153	0.00284	0.9789	0.9090	0.7244
Adaptive PRI – 8 data channels	6.10	20.34	32.70	0	0	17.76	0.00111	0.00135	0.00226	0.9811	0.9431	0.7689
Adaptive PRI – 10 data channels	6.15	20.57	39.58	0	0	10.16	0.00109	0.00130	0.00192	0.9824	0.9531	0.8022
802.11 – 10 data channels, random channel allocation	6.20	18.80	32.53	0	0.65	18.40	0.00105	0.00142	0.00219	0.9811	0.9475	0.7921

It is noticed that as the number of channels used in the adaptive PRI learning schemes increases, the throughput significantly improves when compared to the single-channel 802.11 scenario. The increased throughput is provided by the capacity of the additional channels. Naturally when there are only three flows in the network, one does not expect the throughput to improve by increasing the number of channels to higher than three. For the case of 25 flows, the adaptive PRI with 10 data channels provides an improvement of 13 times in throughput compared to the single-channel 802.11 scenario. When there are 25 flows in the network and only one channel is provided, the network is so congested that it provides a throughput of only three times for the 25 flows. However, when the adaptive PRI is used on 10 channels, it provides a higher capacity though not the capacity required to eliminate the congestion. The capacity provided by the 10 channels is almost 10 times the capacity of each channel. The capacity of each channel

for data packets in 802.11 is almost half of the channel bandwidth. We chose a standard channel bandwidth of 11Mbps in the simulations. Therefore the total throughput of 39.58 Mbps is reasonable compared to the total capacity of almost 50 Mbps, since there is a noticeable congestion in the network. Also for the same case of 25 flows, Adaptive PRI with 10 data channels provides an improvement of 1.22 times in throughput over random allocation of 10 channels. Using the Adaptive PRI algorithm for the networks of 6 nodes and 20 nodes, the maximum possible throughput (6 Mbps and 20 Mbps, respectively) can be achieved by utilizing 3 and 10 channels respectively, which will allocate a different channel to each link. However, for the network of 50 nodes saturation and high drop rate are inevitable, although the throughput is improved significantly by increasing the number of channels. As the number of nodes in the network increase, the number of contending nodes during the time slot, T_s, and mini slot, T_{mini}, increases. This can result in a case that some nodes do not get any chance to transmit during T_{mini}. Hence with a performance much smaller than the desired performance (i.e., unsatisfactory environment response), due to the "reward" characteristic of the learning algorithm, probabilities of channel selection would not be updated for them. This issue will be discussed in Section *C*.

Table I also presents the drop rate in the network using the different methods of channel allocations, and different number of channels. The results show that for the networks of 3 and 10 flows, the drop rate is significantly reduced by utilizing the Adaptive PRI learning scheme and more number of channels. The drop rate for the network of 25 flows is also reduced, but not as much as it was for the networks with smaller densities. This is due to the fact that the network is so dense and the number of

contending nodes is so high that the saturation is inevitable. It can be noticed by using the Adaptive PRI channel allocation and 10 data channels, in the worst case scenario (greatest number of flows), the drop rate is reduced by 78.38% compared to when using a single-channel 802.11. For the same case of 25 flows, PRI with 10 data channels provides a 44.78% reduction on drop rate over random allocation of 10 channels.

Table I also presents the energy consumption per packet in the network using the different methods of channel allocations, and different number of channels. The results show that using the PRI learning scheme and increasing the number of data channels significantly improves the energy consumption per packet. It can be noticed that by using PRI channel allocation and 10 data channels, in the worst case scenario (greatest number of flows), the energy consumption is reduced by 90.25% compared to when using a single-channel 802.11. Also using PRI with data channels reduces the energy consumption by 12.33%. For the same case of 25 flows, PRI with 10 data channels provides a 12.33% reduction in energy consumption per packet over random allocation of 10 channels.

Another performance metric that was used for evaluating the channel allocation schemes was fairness index [21]. Table I also presents the fairness index provided by using the different methods of channel allocations, and different number of channels. The results show that using the Adaptive PRI learning scheme and increasing the number of data channels improves the fairness index – especially when there are greater number of flows. It can be noticed that by using the Adaptive PRI channel allocation and 10 data channels, in the worst case scenario (greatest number of flows), the fairness index is increased by 3.7 times compared to when using a single-channel 802.11. Also using the

Adaptive PRI with 10 data channels increases the fairness index by 1.28%. For the same case of 25 flows, the Adaptive PRI with 10 data channels provides a 1.28% improvement in fairness over random allocation of 10 channels.

The other two channel allocation learning schemes, i.e. Adaptive PRP and Adaptive PRO, were also applied to the same networks and scenarios, with 10 data channels. Table II shows the throughput over the network when using the Adaptive PRI, PRP and PRO schemes and 10 data channels. It is noticed that for the greater number of flows, the Adaptive PRP schemes provides a slightly higher throughput compared to the other two learning schemes. Table II also shows the drop rate over the network when using the Adaptive PRI, PRP and PRO schemes and 10 data channels. It is noticed that for the greater number of flows, the PRI scheme provides a slightly higher (worse) drop rate compared to the other two learning schemes.

Table II. Throughput, drop rate, energy consumption, and fairness index when using the three learning schemes of channel allocation and 10 data channels.

	Throughput (Mbps)			Drop rate (Mbps)			Energy consumption (joules/packet)			Fairness index		
	3 flows	10 flows	25 flows	3 flows	10 flows	25 flows	3 flows	10 flows	25 flows	3 flows	10 f lows	25 flows
Adaptive PRI – 10 data channels	6.15	20.57	39.58	0	0	10.16	0.001085	0.001301	0.001924	0.9824	0.9531	0.8022
Adaptive PRP– 10 data channels	6.15	20.61	39.95	0	0	9.93	0.001085	0.001312	0.001917	0.9824	0.9507	0.8080
Adaptive PRO – 10 data channels	6.15	20.59	39.82	0	0	9.91	0.001085	0.001301	0.001915	0.9824	0.9527	0.8027

Table II shows the energy consumption per packet in the network when using the Adaptive PRI, PRP and PRO schemes and 10 data channels. The three methods do not show any significant difference in the sense of energy consumption. The fairness index of the network, when using the Adaptive PRI, PRP and PRO schemes and 10 data channels, is shown in Table II. It is noticed that for the greater number of flows, the Adaptive PRP scheme provides a slightly higher fairness compared to the other two learning schemes.

We also examined a case in which all the 25 flows started at t = 2 sec, then they were reduced to 10 flows at t = 3 sec, and finally reduced to 3 flows at t = 4 sec. Similarly the simulations were performed for 10 random scenarios for a network of 10 data channels, using the Adaptive PRI learning automata scheme. By comparing Table III to Table I, it can be concluded that by starting a greater number of flows at the same time, a smaller throughput can be achieved. That is, when 25 flows start at the same time, the achieved throughput is limited to 36.76 Mbps (Table III), while by adding 15 flows to the previously existing 19 flows (Table I) a throughput of 39.58 Mbps can be achieved. The reason for the smaller achieved throughput is the high collision in the case of the simultaneously starting greater number of flows.

Table III. Performance metrics of a network with flows. The channel allocation performed using the Adaptive PRI and 10 data channels.

	PRI, 10 data channels		
	25 flows	10 flows	3 flows
Throughput (Mbps)	36.76	25.03	6.14
Drop rate (Mbps)	4.90	0.09	0
Energy consumption (joules/packet)	0.002042	0.001249	0.000991
Fairness index	0.7501	0.9021	0.9832

B. Mobile Scenario

In Section A (static scenario) the network topology was assumed to be static during T_s. In this section a case is examined in which the network topology undergoes changes during the T_s period. A larger network (1000m×1000m) and greater number of flows (50 flows, i.e. 100 peer-to-peer nodes) are considered. Then the behavior of the single-channel 802.11, randomly allocated 10 channels using 802.11, and the Adaptive PRI learning scheme in the case of mobility of the nodes were examined. For four different values of maximum speed (5, 10, 15, and 20 m/s) and also static case (0 m/s), 10 random scenarios were generated and the average of these repeated simulations were used for comparison. Table IV presents the results for using the Adaptive PRI and 10 channels. The speed change does not show a significant effect on the performance. However, in general, these larger network scenarios with a higher traffic flow show a lower performance compared to the static case (Section A).

By using the Adaptive PRI learning scheme, the throughput, drop rate, energy consumption and fairness index show a significant improvement compared to the case that 802.11 is used with randomly allocated 10 data channels (Table IV). The throughput is improved by 19.6%, the drop rate is reduced by 47.6%, the energy consumption per packet is reduced by 10.6% and the fairness index is improved by 11.4%. Also compared to the single-channel 802.11, both Adaptive PRI and 802.11 over randomly allocated 10-data channel are performing significantly better.

Table IV. Performance of the Adaptive PRI with 10 data channels, single-channel 802.11, and randomly allocated 10 data channels using 802.11 on a network of 50 flows, while nodes moving in different speeds.

	Adaptive PRI, 10 data channels					802.11 - single channel	802.11 – 10 data channels, randomly allocated
	Static (0 m/s)	5 m/s	10 m/s	15 m/s	20 m/s	10 m/s	10 m/s
Throughput (Mbps)	84.31	83.68	82.96	81.84	79.44	15.51	69.97
Drop rate (Mbps)	13.35	14.10	14.62	15.71	17.78	80.43	26.92
Energy consumption (joules/ packet)	0.001734	0.001735	0.001741	0.001760	0.001811	0.008398	0.001940
Fairness index	0.7066	0.6975	0.6900	0.6868	0.6636	0.2169	0.6263

C. Comparison of the Three Schemes of the Learning Automata Regarding the Probability Update

Earlier we mentioned the problem of 'left-out' links in the PRI algorithm. This problem occurs when none of the channels provide a satisfactory performance, and hence the probabilities of channel selections are not updated at all. This case is examined below, where the Adaptive PRI is used for channel allocation in a peer-to-peer network of 50 nodes (25 links) using 10 channels.

It was observed that the channel allocations of 21 links out of 25 links converged. The channel allocations for the links 7, 9, 22, and 23 always provided a performance much smaller than the desired performance (i.e., unsatisfactory environment response). Due to the "reward-inaction" characteristic of the learning algorithm, probabilities of

channel selection for these links would not be updated. These links are ‘left-out’ of the update process. The probabilities of channel selections for one of the converged links (link 15), and one of the non-converged links (link 7) are shown in Figure 2 and Figure 3, respectively. Figure 2 shows how the probabilities of selecting the channels converge for link 15 while Figure 3 shows that these probabilities are not updated at all. All the channels keep their initial equal probability, 0.1, all the time. At each iteration, one of the channels is selected randomly.

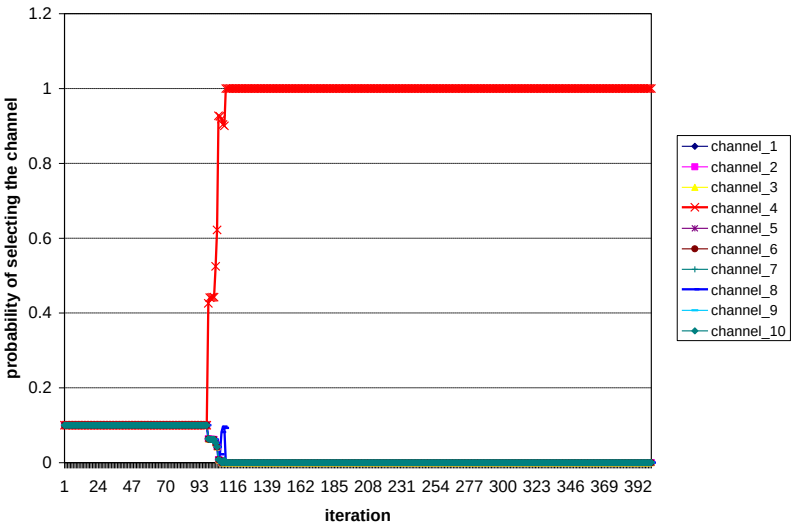

Figure 2. The probability of selecting the channels for link_15, using the Adaptive Pursuit Reward-Inaction algorithm.

By using the Pursuit Reward-Penalty algorithm, the ‘left-out’ links problem is eliminated and the probability of selecting the channels is updated even if the channel allocation is not providing a satisfactory performance. Although the probabilities of channel selections are updated, the channel allocations for 6 links (links 5, 7, 12, 21, 22,

and 23) do not converge yet by the end of the simulation. The channel allocations for the mentioned links provide a performance much smaller than the desired performance (i.e., unsatisfactory environment response). Hence the probabilities of channel selection for these links are updated through the "penalty" process of the algorithm.

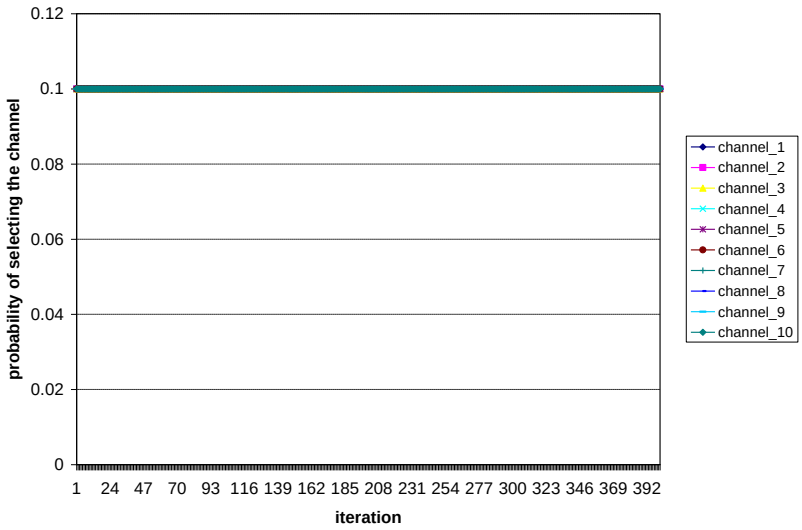

Figure 3. The probability of selecting the channels for link_7, using the Pursuit Reward-Inaction algorithm.

The probabilities of channel selections for one of the converged links (link 15), and one of the yet non-converged links (link 7) are shown in Figure 4 and Figure 5, respectively. Figure 4 shows how the probabilities of selecting the channels converges for link 15, and Figure 5 shows that these probabilities for link 7 are converging, though slowly (parameter adjustment might be needed or increasing the speed here).

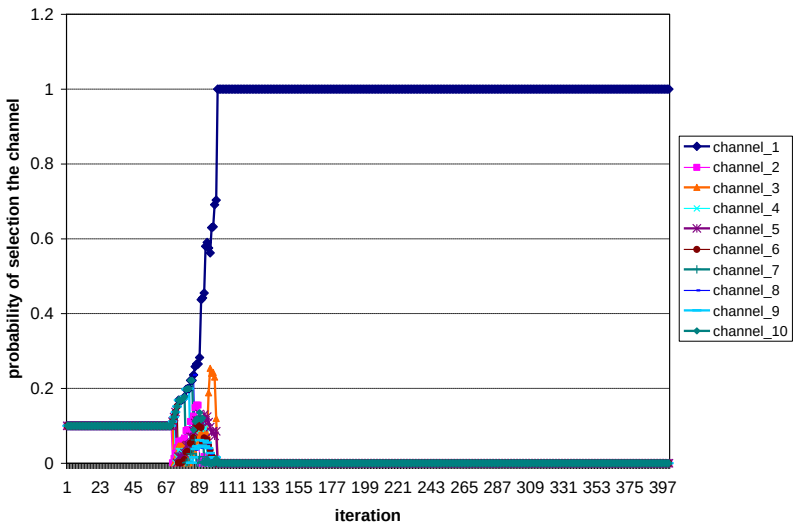

Figure 4. The probability of selecting the channels for link_15, using the Pursuit Reward-Penalty algorithm.

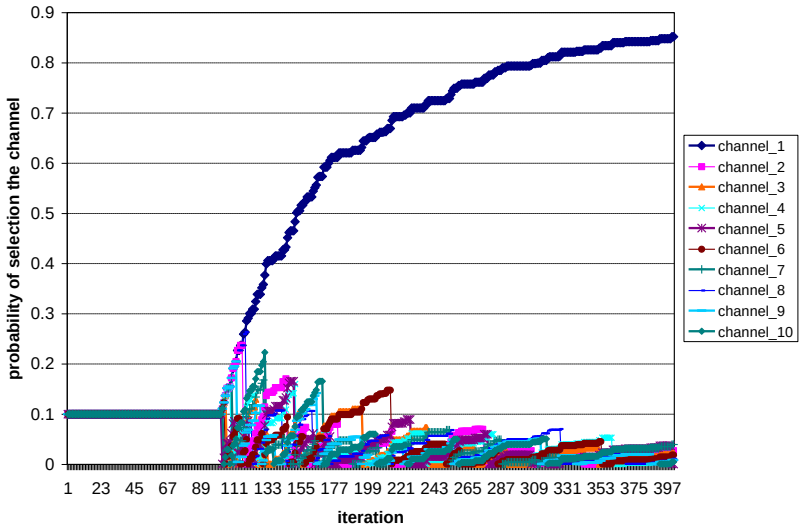

Figure 5. The probability of selecting the channels for link_7, using the Pursuit Reward-Penalty algorithm.

Figure 6 shows the changes in the channel allocations as the Pursuit Reward-Only algorithm runs on the network. It shows that the channel allocations of all the links converge. The probabilities of channel selection for all the links are updated with the

"pursuit" characteristic regardless of the environment response (channel performance). The updates are performed such that the probability of selecting the channel with the best performance is increased, and the probabilities of selecting the other channels are decreased. The magnitude of the relative error determines the magnitude of the update step.

Comparison of the results of the three algorithms shows that the Pursuit Reward-Penalty provides update and convergence for the cases that the channel performance is significantly smaller than the desired performance. The Pursuit Reward-Inaction did not guarantee the update for the less than desirable performance. This would result in "left-out" links; the links with no converged channel allocation. On the other hand, the Pursuit Reward-Only algorithm always increases the probability of the channel with the highest performance, whether the performance of the selected channel is satisfactory or not. This algorithm provides the fastest convergence among the three algorithms.

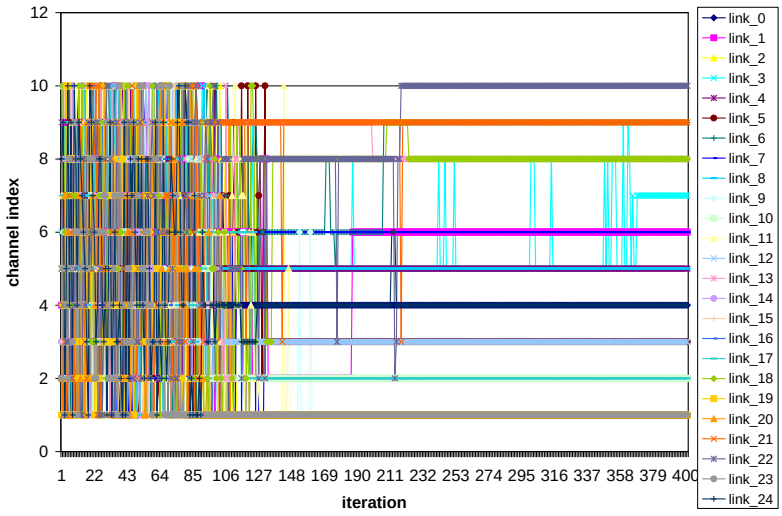

Figure 6. Channel allocation for 25 links in a network of 50 peer-to-peer nodes, using Pursuit Reward-Only learning automata. Channel allocations for all the links have converged.

V. CONCLUSIONS

In this paper we propose a distributed dynamic channel allocation algorithm for wireless networks whose nodes are equipped with single radio interface. The single-radio assumption was made for the sake of simplicity of the network, making it suitable to apply to wireless ad-hoc sensor networks. The periodic nature of the algorithm makes it dynamic and enables the channel allocation to adapt to the topographic changes, possible loss of some channels, mobility of the nodes, and the traffic flow changes. The Adaptive Pursuit learning algorithm runs periodically on nodes, and adaptively finds the optimum channel allocation that provides the desired performance. By selecting realistic desired performance metric, the convergence of the algorithm can be guaranteed. The analytical proof of convergence is presented in this appendix, and also the simulation results for networks of different densities and data channels were provided and showed a significant improvement in throughput, drop rate, energy consumption per packet, fairness index when compared to the single-channel 802.11 and 802.11 with randomly allocated multiple channels.

Also in order to avoid the ‘left-out’ links in the learning process in the first algorithm (Adaptive PRI), we proposed using the other two algorithms, Pursuit Reward-Penalty and Pursuit Reward-Only algorithms. The results of these two algorithms were compared to the results of the Pursuit Reward-Inaction, and showed that the Pursuit Reward-Penalty eliminates the ‘left-out’ links problem, and provides convergence using the same parameters as used in the Pursuit Reward-Inaction. The Pursuit-Only algorithm also eliminates the ‘left-out’ links problem. Also with the same parameters, it provides a faster convergence compared to the Pursuit Reward-Penalty algorithm.

ACKNOWLEDGEMENTS

The authors acknowledge the support of the Intelligent Systems Center and Air Force Research Lab.

APPENDIX A. PROOF OF CONVERGENCE

In Section B, the channel allocation algorithms were presented. In this section, the proofs of convergence of the algorithms are presented. The proofs follow the general method used in [19].

Proof of Convergence of the Adaptive Pursuit Reward-Inaction Algorithm

Theorem 1 establishes that for each node that is running the algorithm, if after a certain time, the channel allocation results in a greater performance for one channel compared to the other channels, the probability of selecting that channel tends to 1. Theorem 2 establishes that for each node and each channel, there exists a time that the channel has been selected by the node for at least *M* times. This guarantees having the average throughput, delay and consumed energy values, which are required for the performance evaluation.

Theorem 1. Suppose there exists an index m_i and a time instant $k_0 < \infty$ such that $\hat{\phi}_i^{m_i}(k) > \hat{\phi}_i^{j}(k)$ for all j such that $j \neq m_i$ and all $k \geq k_0$. Then there exists γ_0 and λ_0 such that for all resolution parameters ($\gamma < \gamma_0, \lambda < \lambda_0$), $p_i^{m_i}(k) \to 1$ with probability 1 as $k \to \infty$.

Proof: From the definition for Discrete Pursuit Reward-Inaction, it can be concluded that if m_i satisfies $m_i = \arg \max_j \hat{\phi}_i^{j}(k)$, where $\hat{\phi}_i^{m_i}(k) = \max_j \hat{\phi}_i^{j}(k)$, then $\hat{\phi}_i^{m_i}(k) > \hat{\phi}_i^{j}(k)$ for all $j \neq m_i$ and all $k \geq k_0$.

Therefore, for all $k > k_0$, $p_i^{m_i}(k+1) = \begin{cases} 1 - \sum_{j=1, j\neq m_i}^{N} (p_i^j(k) - \theta(k)), \\ \qquad \text{if } \beta_i^l(k) = 0 \quad \text{(w.p. } \zeta_i^{m_i}(k)) \\ p_i^{m_i}(k) \\ \qquad \text{if } \beta_i^l(k) = 1 \quad \text{(w.p. } 1 - \zeta_i^{m_i}(k)) \end{cases}$ (A.7)

If $p_i^{m_i}(k) = 1$, then the "pursuit" property of the algorithm trivially proves the result.

Assuming that the algorithm has not yet converged to the m_ith channel, there exists at least one nonzero component of $\mathbf{P}_i(k)$, $p_i^q(k)$, with $q \neq m_i$. Therefore one can write $p_i^q(k+1) = p_i^q(k) - \theta(k) < p_i^q(k)$. (A.8)

Since $\mathbf{P}_i(k)$ is a probability vector, $\sum_{j=1}^{N} p_i^j(k) = 1$, and $p_i^{m_i}(k) = 1 - \sum_{j=1, j\neq m_i}^{N} p_i^j(k)$. Therefore,

$$1 - \sum_{j=1, j\neq m_i}^{N} (p_i^j(k) - \theta(k)) > p_i^{m_i}(k). \tag{A.9}$$

As long as there is at least one nonzero component, $p_i^q(k)$ (where $q \neq m_i$), it is clear that one can decrement $p_i^q(k)$ and increment $p_i^{m_i}(k)$ by at least $\theta(k)$. Hence, $p_i^{m_i}(k+1) = p_i^{m_i}(k) + c(k) \cdot \theta(k)$, where $c(k) \cdot \theta(k)$ is an integral multiple of $\theta(k)$, and $0 < c(k) < N$, and

$$\theta(k) = \begin{cases} \gamma \cdot |\Delta(k)| / \phi^*, & \text{if } -\delta < \Delta(k) / \phi^* \\ \lambda \cdot |\Delta(k)| / \phi^* & \text{otherwise} \end{cases} \tag{A.10}$$

Therefore one can express the expected value of $p_i^{m_i}(k+1)$ conditioned on the current state of the channel, $\mathbf{Q}(k)$, $(\mathbf{Q}(k) = \{\mathbf{P}_i(k), \varphi_i(k)\})$ as follows

$$E[p_i^{m_i}(k+1)\,|\,\mathbf{Q}(k), p_i^{m_i}(k)\neq 1] = \zeta_i^{m_i}(k)\cdot[p_i^{m_i}(k)+c(k)\cdot\theta(k)] + (1-\zeta_i^{m_i}(k))\cdot p_i^{m_i}(k)$$
$$= p_i^{m_i}(k) + \zeta_i^{m_i}(k)\cdot c(k)\cdot\theta(k) \qquad \text{(A.11)}$$

Since all the previous terms have an upperbound of unity, $E[p_i^{m_i}(k+1)\,|\,\mathbf{Q}(k), p_i^{m_i}(k)\neq 1]$ is also bounded,

$$\sup_{k\geq 0} E[p_i^{m_i}(k+1)\,|\,\mathbf{Q}(k), p_i^{m_i}(k)\neq 1] < \infty. \qquad \text{(A.12)}$$

Thus one can write $E[p_i^{m_i}(k+1) - p_i^{m_i}(k)\,|\,\mathbf{Q}(k)] = \zeta_i^{m_i}(k)\cdot c(k)\cdot\theta(k) \geq 0, \quad$ for all $k \geq k_0$ implying that $p_i^{m_i}(k)$ is submartingale. By submartingale convergence theorem, the sequence $\{p_i^{m_i}(k)\}_{k\geq k_0}$ converges.

Therefore $E[p_i^{m_i}(k+1) - p_i^{m_i}(k)\,|\,\mathbf{Q}(k)] \to 0$ w.p.1, as $k\to\infty$.

This implies that $\zeta_i^{m_i}(k)\cdot c(k)\cdot\theta(k)\to 0$ w.p.1. This in turn implies that $c(k)\to 0$ w.p.1 $(\theta(k)\to 0$ w.p.1), which means there is no nonzero element in $\mathbf{P}_i(k)$ except for $p_i^{m_i}(k)$ (or $\Delta(k)\to 0$). Consequently, $\sum_{j=1, j\neq m_i}^{N} p_i^j(k)\to 0$ w.p.1 and

$p_i^{m_i}(k) = 1 - \sum_{j=1, j\neq m_i}^{N} p_i^j(k) \to 1$ w.p.1 ■

Theorem 2. For each node *i* and channel *j*, assume $p_i^j(0)\neq 0$. Then for any given constant $\delta_0 > 0$ and $M<\infty$, there exists $\gamma_0<\infty$, $\lambda_0<\infty$ and $k_0<\infty$ such that under the Discrete Pursuit Reward-Inaction algorithm, for all learning parameters $\gamma<\gamma_0$ and $\lambda<\lambda_0$ and all time $k>k_0$:

Pr{each channel chosen by node *i* more than *M* times at time *k*} $\geq 1-\delta_0$.

Proof: Define the random variable $Y_i^j(k)$ as the number of times that channel *j* was chosen by node *i* up to time *k*. Then it must be proved that $\Pr\{Y_i^j(k) > M\} \geq 1-\delta_0$.

This is equivalent to proving

$$\Pr\{Y_i^j(k) \le M\} \le \delta_0 . \tag{A.13}$$

The events $Y_i^j(k) = q$ and $Y_i^j(k) = s$ are mutually exclusive for $q \neq s$, so (A.13) can be rewritten as

$$\sum_{q=1}^{M} \Pr\{Y_i^j(k) = q\} \le \delta_0 \tag{A.14}$$

For any iteration of the algorithm, Pr**{**choosing channel *j***}** ≤ 1. Also the magnitude by which any channel selection probability can decrease in any iteration is bounded by $\gamma \cdot |\Delta(k)| / \phi^*$ (or $\lambda \cdot |\Delta(k)| / \phi^*$), where $\Delta(k) < \Delta$ for all *k*. During any of the first *k* iterations of the algorithm:

$$\Pr\{\text{channel } j \text{ is not chosen by node } i\} \le \left(1 - p_i^j(0) + k \cdot \gamma \cdot |\Delta| / \phi^*\right). \tag{A.15}$$

Using these upper bounds, the probability that channel *j* is chosen at most *M* times among *k* choices, has the following upper bound

$$\Pr\{Y_i^j(k) \le M\} \le \sum_{l=1}^{M} C(k,l)(1)^l (1 - p_i^j(0) + k \cdot \gamma \cdot |\Delta| / \phi^*)^{k-l} \tag{A.16}$$

In order to make a sum of *M* terms less than δ_0, it is sufficient to make each term less than δ_0 / M. Consider an arbitrary term, $l = m$. It must be shown that

$$C(k,m)(1)^m (1 - p_i^j(0) + k \cdot \gamma \cdot |\Delta| / \phi^*)^{k-m} < \delta_0 / M ,$$

or $$M \cdot C(k,m)(1)^m (1 - p_i^j(0) + k \cdot \gamma \cdot |\Delta| / \phi^*)^{k-m} < \delta_0 . \tag{A.17}$$

Knowing that $C(k,m) \le k^m$, it must be proven that $M \cdot k^m \left(1 - p_i^j(0) + k \cdot \gamma \cdot |\Delta| / \phi^*\right)^{k-m} \le \delta_0$.

Now in order to get the L.H.S of this term to be less than δ_0 as k increases, $\left(1-p_i^j(0)+k\cdot\gamma\cdot\left|\Delta\right|/\phi^*\right)$ must be strictly less than unity. In order to guarantee this, we bound the value of γ with respect to k must be bounded such that $\left(1-p_i^j(0)+k\cdot\gamma\cdot\left|\Delta\right|/\phi^*\right)<1$. It can be achieved by requiring that $\gamma<\frac{p_i^j(0)}{k\cdot|\Delta|}\cdot\phi^*$. Let

$$\gamma=\frac{p_i^j(0)}{2k\cdot|\Delta|}\cdot\phi^*. \tag{A.18}$$

With this value of γ, Equation (A.16) is simplified to $\Pr\{Y_i^j(k)\le M\}<M\cdot k^m\cdot\psi^{k-m}$, where $\psi=1-\frac{1}{2}p_i^j(0)$, and $0<\psi<1$. Now $\lim_{k\to\infty} M\cdot k^m\cdot\psi^{k-m}$ must be evaluated.

$$\lim_{k\to\infty} M\cdot k^m\cdot\psi^{k-m}=M\cdot\lim_{k\to\infty}\frac{k^m}{\left(1/\psi\right)^{k-m}}\text{, with }\gamma=\frac{p_i^j(0)}{2k\cdot|\Delta|}\cdot\phi^*.$$

By applying l'Hopital's rule:

$$\begin{aligned} M\cdot\lim_{k\to\infty}\frac{k^m}{\left(1/\psi\right)^{k-m}} &= M\cdot\lim_{k\to\infty}\frac{m!}{\left(\ln(1/\psi)\right)^m\left(1/\psi\right)^{k-m}} \\ &= 0 \qquad \text{with } \gamma=\frac{p_i^j(0)}{2k\cdot|\Delta|}\cdot\phi^* \end{aligned} \tag{A.19}$$

Therefore Equation (A.16) has a limit of zero as $k\to\infty$ and $\gamma\to 0$, whenever Equation (A.18) is satisfied.

Since the limit exists, for every channel j there is a $k(j)$ such that for all $k>k(j)$, Equation (A.16) holds.

Now set $\gamma(j) = \frac{p_i^j(0)}{2k(j)\cdot|\Delta|}\cdot\phi^*$. It remains to be shown that Equation (A.16) is satisfied for all $\gamma < \gamma(j)$, and for all $k > k(j)$. This is trivial because as γ decreases, the L.H.S of Equation (A.16) is monotonically decreasing, and so the inequality (A.16) is preserved.

Also for any $k > k(j)$, since $Y_i^j(k(j)) \geq M \Rightarrow Y_i^j(k) \geq M$, by the laws of probability:

$$\Pr\{Y_i^j(k) \geq M\} \geq \Pr\{Y_i^j(k(j)) \geq M\}. \qquad \text{(A.20)}$$

Thus in this case also, the inequality (A.16) still holds. Hence for any channel *j*, $\Pr\{Y_i^j(k) \leq M\} \leq \delta_0$ whenever $k > k(j)$ and $\gamma < \gamma(j)$. Since this argument can be repeated for all the channels, k_0 and γ_0 can be defined as $k_0 = \max_{1\leq j\leq N}\{k(j)\}$, and $\gamma_0 = \max_{1\leq j\leq N}\{\gamma(j)\}$. Thus for all *j*, it is true that for all $k > k_0$ and $\gamma < \gamma_0$ ($\lambda < \lambda_0$), the quantity $\Pr\{Y_i^j(k) \leq M\} \leq \delta_0$ and theorem is proved. ■

REFERENCES

[1] J. Mitola III., Cognitive Radio An Integrated Agent Architecture for Software Defined Radio. PhD thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2000.

[2] P. Bahl, R. Chandra, and J. Dunagan, SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks, In *Proceedings of the 10th Annual international Conference on Mobile Computing and Networking* (Philadelphia, PA, USA, September 26 - October 01, 2004). MobiCom '04. ACM, New York, NY, pp:216-230.

[3] M. Alicherry, R. Bhatia, and L. Li, Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks, In *Proceedings of the 11th Annual international Conference on Mobile Computing and Networking* (Cologne, Germany, August 28 - September 02, 2005). MobiCom '05. ACM, New York, NY, pp.58-72.

[4] A. Mishra, S. Banerjee, and W. Arbaugh, Weighted coloring based channel assignment for WLANs, *SIGMOBILE Mob. Comput. Commun. Rev.* 9, 3 (Jul. 2005), pp: 19-31.

[5] N. Nie and C. Comaniciu, Adaptive channel allocation spectrum etiquette for cognitive radio networks, *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, vol., no., pp.269-278, 8-11 Nov. 2005.

[6] J. Li, D. Chen, W. Li, and J. Ma, Multiuser power and channel allocation algorithm in cognitive radio, *Parallel Processing, 2007. ICPP 2007. International Conference on* , vol., no., pp.72-72, 10-14 Sept. 2007.

[7] A. Raniwala, K. Gopalan, and T. Chiueh, Centralized channel assignment and routing algorithms for multi-channel wireless mesh network, *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 2, pp: 50-65, Apr. 2004.

[8] M. Felegyhazi, M. Cagalj, S. Bidokhti, and J.-P Hubaux, Non-Cooperative Multi-Radio Channel Allocation in Wireless Networks, *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, vol., no., pp.1442-1450, May 2007.

[9] P. Kyasanur and N. H. Vaidya, Routing in Multi-Channel Multi-Interface Ad Hoc Wireless Networks, technical report, Dec. 2004.

[10] P. Kyasanur and N.H. Vaidya, Routing and interface assignment in multi-channel multi-interface wireless networks, Proc. of *Wireless Communications and Networking Conference*, vol.4, no., pp. 2051-2056 Vol. 4, 13-17 March 2005.

[11] J. So and N. Vaidya, Routing and channel assignment in multi-channel multi-hop wireless networks with single-NIC devices, *Technical Report, University of Illinois at Urbana Champaign*, Dec. 2004.

[12] Z. Han, Z. Ji, and K.J.R. Liu, Fair Multiuser Channel Allocation for OFDMA Networks Using Nash Bargaining Solutions and Coalitions, *Communications, IEEE Transactions on*, vol.53, no.8, pp. 1366-1376, Aug. 2005.

[13] J.A. Patel, H. Luo and I. Gupta, A Cross-Layer Architecture to Exploit Multi-Channel Diversity with a Single Transceiver, *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, vol., no., pp.2261-2265, May 2007.

[14] J. So and N.H. Vaidya, Multi-channel MAC for Ad-Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver, In *Proceedings of the 5th ACM international Symposium on Mobile Ad Hoc Networking and Computing*. MobiHoc '04. ACM, New York, NY.

[15] B. Eslamnour, M. Zawodniok, and S. Jagannathan, Dynamic Channel Allocation in Wireless Networks using Adaptive Learning Automata, *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE* , vol., no., pp.1-6, 5-8 April 2009

[16] T. Clausen and P. Jacquet, Optimized Link State protocol (OLSR), 2003.

[17] N. Regatte and S. Jagannathan, Optimized Energy-Delay Routing in Ad Hoc Wireless Networks, *Proc. of the World Wireless Congress,* May 2005.

[18] R. Maheshwari, H. Gupta, and S.R. Das, Multichannel MAC Protocols for Wireless Networks, *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on* , vol.2, no., pp.393-401, 28-28 Sept. 2006.

[19] B.J. Oommen and J.K. Lanctot, Discretized pursuit learning automata, *Systems, Man and Cybernetics, IEEE Transactions on*, vol.20, no.4, pp.931-938, Jul/Aug 1990.

[20] M.A. Haleem and R. Chandramouli, Adaptive downlink scheduling and rate selection: a cross-layer design, *Selected Areas in Communications, IEEE Journal on*, vol.23, no.6, pp. 1287-1297, June 2005.

[21] R. Jain, D. Chiu, and W. Hawe, A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems, DEC Research Report TR-301, September 1984.

2. Adaptive Dynamic Routing for Hybrid Wireless Networks Using Nonlinear Optimal Framework

B. Eslamnour, *Student Member IEEE* and S. Jagannathan, *Senior Member IEEE*

***Abstract*—**Wireless ad hoc networks are subject to frequent changes in topology and channel uncertainties over time. Therefore, a dynamic routing protocol that adapts to the changes of the network would provide an improved performance compared to static routes. With the use of static routes, broken links and failed routes require route repair or route discovery. In this paper, approximate dynamic programming (ADP) techniques are utilized to find dynamic routes, while solving discrete-time Hamilton-Jacobi-Bellman (HJB) equation forward-in-time for route cost. First we investigate the effect of the queue occupancy in the one-hop neighbors in the dynamic route selection. Then the route selection by a forward-in-time approach is presented. It is shown that when the number of hops increases, the proposed route selection results in truly optimal route for the performance index selected. The performance of the proposed optimal route selection approach is evaluated by extensive simulations and by comparing it to AODV.

***Index Terms*—Dynamic Routing Protocol, Approximate Dynamic Programming, Energy Efficient Routing, Multihop Routing**

I. NOMENCLATURE

Symbol	Definition
k	hop on the route (k = 0 at the source node)
$C(k)$	initial route cost from the node at hop k to BS (Base Station)
$LC_{k,k+1}$	link cost between the node at hop k and the next node (at hop k+1)
$d_{k,k+1}$	link delay between the node at hop k and the next node (at hop k+1)
PTT	packet transmission time $PTT = \frac{PacketSize}{DataRate}$
BW_k^d	desired bandwidth at the node at hop k
BW_{k+1}^a	available bandwidth at the node at hop k+1
E_k^{init}	initial energy at the node at hop k
E_k^a	available energy at the node at hop k
q_k	queue length at the node at hop k
$q_k^{\max}$	queue limit at the node at hop k
$f(.)$	a convex function of queue occupancy. It progressively increases as the queue occupancy ratio approaches 100%
$J(k)$	route cost to BS for the kth hop on the route (at source, k=0) cost function represented by OLA: $J(k) = W_J^T \sigma(z(k)) + \varepsilon_J(k)$
$z(k)$	state vector at the kth hop $z(k) = [z_1(k) \quad z_2(k) \quad z_3(k)]^T$
W_J	constant target OLA parameter vector
$\sigma(\bullet)$	vector activation function for the cost approximation OLA scheme
$\varepsilon_J(k)$	bounded cost approximation error
$\hat{J}(k)$	approximation of route cost from the kth hop on the route to BS
$\hat{W}_J(k)$	actual parameter vector for the target OLA parameter vector W_J
$e_J(k)$	error in the OLA route cost at the kth hop
$\tilde{W}_J(k)$	OLA parameter estimation error, $\tilde{W}_J(k) = W_J - \hat{W}_J(k)$

II. INTRODUCTION

Traditional routing algorithms [1], [2] construct routing tables by exchanging topological data and discovering shortest paths [3]. Usually the routing table entries for each destination contain the next hop and link cost. Once data packets become available for transmission to a destination, the next hop node is chosen by table look-up.

However, mobility in an ad hoc network make the nodes move in and out of range causing frequent updates on the routing tables as well as route selection. As a consequence, the performance of the network degrades, or the frequently broken links can cause extra routing overhead to re-establish the routes. This excessive overhead due to unreliable links resulting from mobility can cause congestion, increased energy consumption and reduced performance. In an attempt to solve this problem, the objective has been to minimize the routing overhead to conserve energy [4] and improve network scalability.

Distance Vector Routing (DVR)[1], also known as Distributed Bellman-Ford [9], [10], is a decentralized algorithm which associates a distance vector to each route. The vector magnitude represents the route cost and the direction identifies the next hop to be taken in order to reach the destination. However, DVR is a proactive routing and the distance vectors are static. Hence, broken links and link failures caused by topology changes and channel uncertainties will demand route repair or new route discovery process. In addition, DVR does not prevent routing loops.

On the other hand, Link State Routing [2] creates a complete map of the network at each node. This is done in two phases, recording the state of the links connected to each node, and distributing this information to the other nodes in the network. Routing

tables are constructed by calculating the cheapest path to each destination using link state information received from other nodes. Entries in the routing table contain destination address, cost and next hop. Unlike DSR, LSR requires many more updates in order to converge to a stable routing table. However, frequent updates and the fact that every change in the routing table of a node should be informed to all the other nodes in the network, increase the message overhead in LSR. In the meanwhile, un-updated routing tables in the network can cause routing loops.

Destination Sequenced Distance Vector Routing or DSDV [10] routing is a derivative of DVR. DSDV maintains a route table containing the cost and next-hop to each destination, and also a sequence number. The sequence number is generated by the destination and indicates the time of its creation and is propagated along with the link updates. This allows the nodes to identify the most recent updates, and also prevents routing loops. However, in case of dynamic topologies, routing overhead will be high due to frequent transmissions of route update packets.

Dynamic Source Routing (DSR) [12],[14] is an on-demand routing protocol. When a route is required, the source node broadcasts a request message over the network. Once the destination receives the request, it sends back a route reply, and the path is cached in the reply message. The path taken by a source routed message is determined by the source in the beginning of the transmission, and cannot be changed during transit. If a broken link occurs in a chosen path, the source must retransmit the message through an alternative route.

DSR can be optimized by limiting the initial route request to a single hop, allowing only immediate neighbors to respond with a previously cached [15]-[17] route.

If no route is discovered within a timeout period the normal route requests flood the network. However, this optimization is only effective when the route caches are up-to-date [17]. Besides, it can cause joint paths between multiple flows, causing a degraded throughput.

Temporally Ordered Routing Algorithm (TORA) [18], [19] is a link reversal algorithm which routes messages by assigning a height value to each node. Heights are negotiated between immediate neighbors on-demand by sending a query message, triggering further queries which elevate the height of the originator. Intermediate nodes are given heights in descending order toward the destination. Messages will be directed only through nodes in descending height order. TORA finds multiple on-demand paths to destination and is suitable for dense networks. However, it does not scale well, and among on-demand routing protocols, AODV outperforms TORA. In addition, TORA assumes that all the nodes have synchronized clocks.

The Ad hoc On-demand Distance Vector or AODV [20] is a combination of the DSDV and DSR algorithms. Routes are created by exchanging distance vectors on-demand. The route request continues searching until a node owning a fresh route or the final destination is reached. In either case a reply is sent back to the originator of the request. Reply messages create a path for messages to flow between origin and destination. The freshness of the routes is determined by sequence numbers.

Periodically, nodes send HELLO messages to indicate their presence to neighbors. In the absence of receiving a HELLO message from a neighbor within a predefined interval, the neighbor is eliminated from the neighbor's list, and an error

message is sent to the origin of the route. However, unstable network topologies can cause a high communication overhead due to re-discovering route requests.

Some alternative routing approaches for ad hoc networks are based on Cluster Based Routing Protocols (CBRP) [21] -[23] which partition the network into disjoint sets. Cluster-heads are selected [24] based on their position and coverage. Route discovery is limited to the cluster-heads, and hence the amount of route discovery packets flooded into the network is reduced. However, when the network topology changes due to mobility, the cluster selection process may be repeated. Optimal selection of the cluster-heads would require frequent message exchange among the neighbors.

The Optimized Link State Routing protocol (OLSR) [25],[26], Topology Dissemination Based Reverse Path (TDBRP) [27] and Zone Routing Protocol (ZRP) [28] are derivatives of Link State Routing. In fact these methods share similarities with clustering because they select a subset of the nodes to perform routing. The term Multi-Point Relay (MPR) is used instead of cluster-head and clusters are replaced with zones.

On the other hand, Associativity Based Routing (ABR) [29],[30] attempts to limit network traffic by discovering and using the most temporally stable routes between nodes. In ABR, broken links are locally repaired – eliminating the need for starting the new route discovery by the source node. However during the local route repair, local query broadcasts can result in high delays.

Other link quality factors have also been used in defining route cost metric. Expected Transmission Count (ETX) in [31] is the number of transmission attempts in order to achieve one successful transmission, and evaluates the link quality by the expected transmission counts. Weighted Cumulative Expected Transmission Time

(WCETT) [32] assigns ETT to each link of the network. The route metric is a weighted sum of the cumulative ETT of the links and the ETT of the bottleneck link along the route. However, both ETX and WCETT are designed for stationary mesh networks. Furthermore, since the routes discovery is based on the route cost from source to destination, the routes are established either by a centralized approach, or upon receiving the requests at the destination. While the former seems unrealistic, the latter requires a waiting time for receiving the route reply message from the destination – similar to all the other methods that route selection decision is made in the destination. However, we are interested a routing protocol that is able to find the route on-the fly.

Optimized Energy-Delay Routing (OEDR) [33] and Optimized Energy-Delay Sub-network Routing (OEDSR) [34] are proposed for wireless ad hoc networks and wireless sensor networks, respectively. OEDR uses an energy-delay metric in selecting the next relay nodes in order to maximize the number of two hop neighbors. In OEDSR, the link cost factor is defined by the available energy at the relay node, link-delay between the two nodes, and the distance between the relay node and the base station.

MMCR [35] is a multi-interface multi-channel proactive routing protocol that utilized MPRs and a new cost metric to discover and establish routes in wireless ad hoc networks. MMCR is built concepts of OEDSR except it utilizes multiple channels. The link cost between two nodes is defined proportional to the link delay, and inversely proportional to bandwidth factor and energy utilization between the nodes and guarantees certain levels of QoS. However, it is a proactive protocol, and the routes need to be re-discovered in case of changes in topology or link failure.

In this paper, an on-demand adaptive dynamic routing protocol is proposed in which the routes are adapted based on link and channel conditions without flooding the network with new route request messages. Adaptive dynamic programming techniques are utilized to find dynamic routes, while solving discrete-time Hamilton-Jacobi-Bellman equation forward-in-time for route cost in an online manner. It uses a neural network (NN) or any online function approximator to approximate the route cost to the base station. While our proposed method is based on Bellman-Ford dynamic programming, utilizing online approximator differentiates it from dynamic programming. It provides reactive and dynamic forward-in-time solutions – as opposed to proactive and static backward-in-time solutions in dynamic programming.

In the proposed protocol, after the initialization phase, the route discovery and adaptation are based on the local information on neighbors and channel quality, hence avoiding flooding the network with route discovery or route repair messages. Furthermore, metrics such as available bandwidth, link delay, available energy and queue occupancy at nodes are taken into account in determining the route cost. In comparison to hop count metric, this provides a more effective definition of route cost in the presence of congestion and channel uncertainties.

In Section III problem statement is described. Section IV presents the methodology. First it examines the effect of using queue occupancy, available bandwidth, available energy and link delay in route cost, and it is shown that it improves the performance of the network compared to AODV. Then it presents the adaptive dynamic routing protocol that uses an OLA (On-Line Approximator) to estimate the remaining route cost forward-in-time. The OLA estimates the remaining route cost of the next node

to BS using queue occupancy, link delay, and the initial cost to BS. In Section V simulation results and analysis are presented, and Section VI concludes the paper.

III. PROBLEM STATEMENT

We consider a wireless ad hoc network in which any node can start a flow towards the base station. Our objective is to achieve a dynamic routing scheme capable of adapting to the channel uncertainties and topology changes. In the initial phase, the base station sends a beacon throughout the network. As the beacon propagates in the network, its sequence number is incremented by each forward (to avoid loops), and nodes are initialized by their route cost to the base station. Also each node identifies its one-hop neighbors towards the base station. Each node establishes a neighbors table in which it lists all the neighbors, whether they are towards the bases station, their initial route cost to base station, $C(k)$, and several other entries that will be explained later.

In order to sustain the neighbors list, nodes periodically send HELLO messages to indicate their presence to neighbors. In the absence of receiving a HELLO message from a neighbor within a predefined interval, the neighbor is removed from the neighbor's list.

Once a node has packets to transmit to the base station, the next node is selected based on the estimation of its route cost to the base station and its link cost to the current node. The routing process is observed as a hopd process. Starting at the source node, the packet is at hop $k = 0$. Depending upon the topology and channel conditions, there can be several candidate neighbor nodes to be chosen as the next node of the route to the base station. Fig. 1 illustrates this view of the network. It must be note that the nodes vertically aligned at each hop represent the possible list of the 'next nodes' from the point of view

of the node in the previous hop – not the precise geographical location of the nodes in the network.

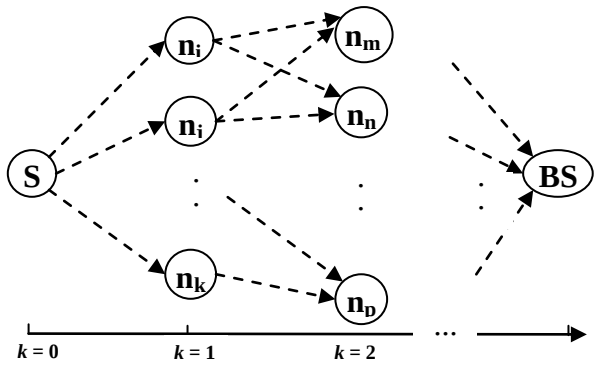

Fig. 1. Wireless network from the routing algorithm point of view.

At each hop, the route cost to the base station is the cumulative sum of the costs of the links on the route.

$$J(k) = \sum_{i=k}^{\infty} LC_{i,i+1} . \tag{1}$$

It can be rewritten as [36]

$$J(k) = LC_{k,k+1} + \sum_{i=k+1}^{\infty} LC_{i,i+1} = LC_{k,k+1} + J(k+1) . \tag{2}$$

In order to find the optimum route, the objective is to minimize (2) at each hop. In other words at each hop the minimum cost is

$$J^*(k) = \min_{n_{k+1} \in N_k^1} \left\{ LC_{k,k+1} + J^*(k+1) \right\} . \tag{3}$$

where N_k^1 is the list of one-hop neighbors of the nodes at hop *k.*

Since the most up-to-date optimum route cost to BS is not always available, in the paper instead of $J^*(k+1)$ in (3), an estimation of this route cost is used. However, the up-to-date value of the link cost is available through frequent updates of the neighbors list.

In the next section, this problem is solved via two approaches. In the first approach, the initial route cost to BS, $C(k+1)$, is used as the estimation of $J^*(k+1)$, and the effect of a new queue-occupancy-aware link cost metric is examined. After concluding that the new link cost metric provides a reasonable measurement of the neighborhood, this link cost metric is used to define a model for the network to solve the discrete-time solution of HJB equation forward-in-time while using an OLA for approximating the second term in (3).

IV. METHODOLOGY

A. Queue-Occupancy-Aware Dynamic Routing

In this approach, the performance index (3) is rewritten as

$$J^*(k) = \min_{n_{k+1} \in N_k^1} \left\{ LC_{k,k+1} + C(k+1) \right\}. \tag{4}$$

In other words, at each hop the algorithm tries to find the next node among the one-hop neighbors that minimizes (4). It must be noted that $C(k+1)$ is the initialized route cost to BS, and is not updated frequently. On the other hand, the link cost between the neighbors is updated frequently through exchanging HELLO messages. We introduce the new link cost metric that takes into account the link delay, available bandwidth, available energy, and queue-occupancy of the receiver node of the link

$$LC_{k,k+1} = \frac{d_{k,k+1}}{PTT} + \frac{BW_k^d}{BW_{k+1}^a} + \frac{E_{k+1}^{init}}{E_{k+1}^a} + f\left(\frac{q_{k+1}}{q_{k+1}^{\max}}\right), \tag{5}$$

where $LC_{k,k+1}$ is the link cost between the node at hop k and the node at hop k+1, $d_{k,k+1}$ is the link delay between the two nodes, BW_k^d is the desired bandwidth at the node at hop k, BW_{k+1}^a is the available bandwidth at the node at hop k+1. E_{k+1}^{init} and E_{k+1}^a are the initial energy and available energy at the node at hop k+1, respectively. q_{k+1} and $q_{k+1}^{\max}$ are queue length and queue limit at the node at hop k+1, respectively. $f(.)$ is a convex function of queue occupancy. It progressively increases as the queue occupancy ratio approaches 100%. In order to reduce the computation load, we suggest using a piecewise linear approximation of $f(.)$ as follows (and illustrated in Fig. 2)

$$f'(x) = \begin{cases} 0 & x = 0 \\ 1 & 0 < x < 0.3 \\ 5 & 0.3 \le x < 0.6 \\ 30 & 0.6 \le x < 0.7 \\ 50 & 0.7 \le x < 0.8 \\ 100 & 0.8 \le x < 0.9 \\ 200 & 0.9 \le x < 1 \\ 500 & x = 1 \end{cases} \tag{6}$$

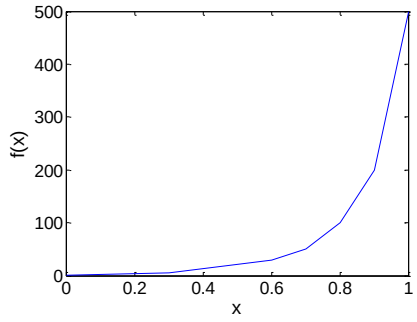

Fig. 2. Linear approximation of function *f*(.).

In order to evaluate the effectiveness of the new route cost definition, a set of simulations were performed and comparisons were made between the new protocol (q-aware) and AODV protocols. The simulation settings and results are presented in the next section.

1) Queue-Occupancy-Aware Dynamic Routing: Simulation Results

This section presents the simulation results of running the dynamic q-aware routing protocol and AODV on various scenarios and topologies. Network Simulator NS-2 [37] was modified, and the proposed routing protocol was implemented in NS-2. The simulation parameters are listed in TABLE I.

TABLE I.

SIMULATION PARAMETERS

Routing Protocol	AODV and Q-AWARE
Number of Nodes	50
Simulation Area	1000 m×1000 m
Mobility Model	Random way-point
Propagation Model	Two-ray Ground Reflection
Transmission Range	250 m
Traffic Type	CBR (UDP)
Number of Connections (Flows)	10, 12, 14, …, 20
Packet Size	210 bytes
Data Rate	20 kbps, 100 kbps, 200 kbps, 500 kbps, and 1000 kbps
Interface Queue	50 pkts

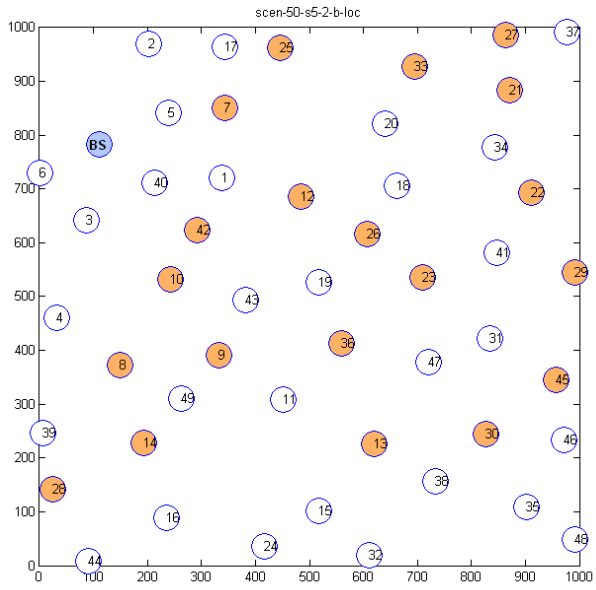

Fig. 3. A sample topology with 20 sources sending data flows towards BS.

Random topologies were generated such that one Base Station and 49 nodes are randomly placed in an area of 1000 m×1000 m. Also various random scenarios were generated to provide a variable range of flows (10, 12, …, 20) towards Base Station. Fig. 3 illustrates one of the topologies with 20 flows.

Fig. 4 shows the average throughput for various number of flows (10, 12, …, 20) and various values of data rate (20k bps, 100 kbps, 200 kbps, 500 kbps, and 1000 kbps) , using dynamic q-aware routing protocol and AODV. Results for Q-aware and AODV protocols are shown in solid lines and dashed lines, respectively. It can be noted that Q-aware protocol in general provides a larger average throughput compared to AODV. For small data rates, the two protocols provide similar throughputs. However, as the data rate increases (*i.e.* it is more likely to have congested areas and failed links), Q-aware protocol provides a considerably higher average throughput compared to AODV.

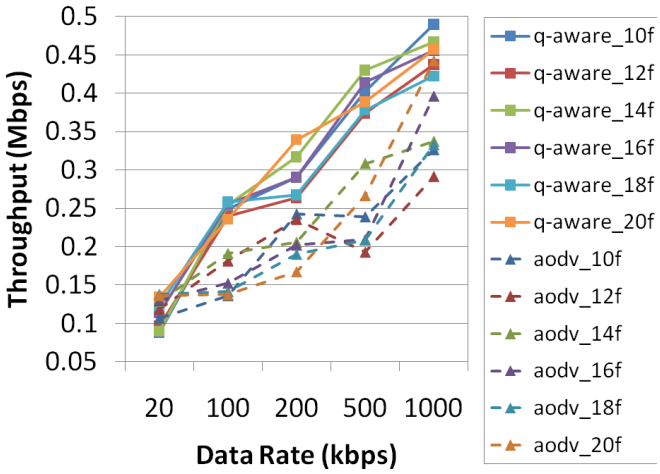

Fig. 4. Average throughput for various number of flows and various data rates.

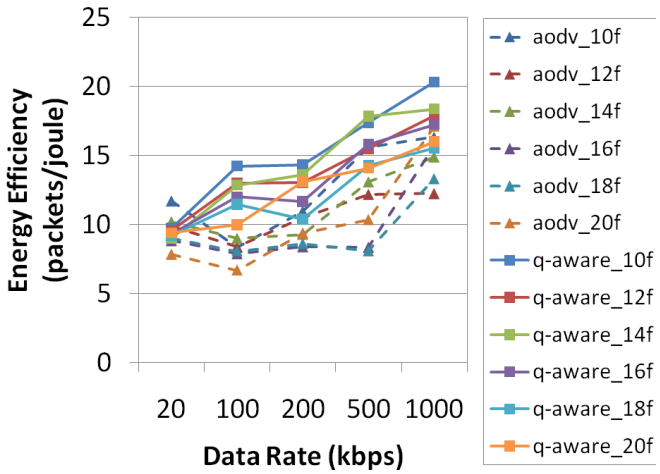

Fig. 5. Energy efficiency for various number of flows and various data rates.

Fig. 5 shows the energy efficiency for various number of flows (10, 12, …, 20) and various values of data rate (20k bps, 100 kbps, 200 kbps, 500 kbps, and 1000 kbps) , using dynamic Q-aware routing protocol and AODV. Results for Q-aware and AODV

protocols are shown in solid lines and dashed lines, respectively. It is noticed that in general, Q-aware protocol provides higher energy efficiency compared to AODV. In particular, for each value of number of flows in the network, Q-aware protocol provides higher energy efficiency compared to AODV.

These preliminary results illustrate that a dynamic routing protocol which is aware of the queue occupancy of the neighbor nodes provides an improved performance compared to AODV. In the next section, this idea will be used to develop an adaptive dynamic queue-occupancy-aware routing that utilizes online approximators (OLA) to estimate the cost of the remaining route to Base Station.

B. Adaptive Dynamic Queue-Occupancy-Aware Routing

In this approach, instead of being limited to the initialized cost to BS, $C(k)$ s, the estimation of cost to BS is approximated by a NN as a function of queue occupancy, link delay, and the initial cost to BS, $C(k)$. State vector is defined as $z(k) = [z_1(k) \quad z_2(k) \quad z_3(k)]^T$, where $z_1(k)$, $z_2(k)$ and $z_3(k)$ are the terms associated with queue occupancy, link delay, and the initial cost to BS, $C(k)$, respectively.

At each hop, the next nodes are sorted according to their initial cost to Base Station, $C(k)$.Then in general, if at hop *k*, there are *M* possible next nodes, state equation will be

$$x(k+1) = x(k) + B(k)u(k), \tag{7}$$

where

$B = [0 - x(k) \quad 1 - x(k) \quad \cdots \quad (M-1) - x(k)]$,
$u(k) = [1 \quad 0 \quad \cdots \quad 0]^T$, or $[0 \quad 1 \quad \cdots \quad 0]^T$, ... or $[0 \quad \cdots \quad 0 \quad 1]^T$ ($u(k)$ is a vector of length *M*).
$x(k+1) = 0, 1, \ldots, M-1$

The relationship between the state vector, $z(k+1)$, and state $u(k)$ is

$$z(k+1) = \underline{\mathbf{Z}}(k+1).u(k) \tag{8}$$

where the columns of $\underline{\mathbf{Z}}(k+1)$ are the state vectors of the *M* possible next hops, *i.e.*

$\underline{\mathbf{Z}}(k+1) = [z_0(k+1), z_1(k+1), \ldots, z_{M-1}(k+1)]$

Where $u(k) = [1 \quad 0 \quad \cdots \quad 0]^T$, or $[0 \quad 1 \quad \cdots \quad 0]^T$, ... or $[0 \quad \cdots \quad 0 \quad 1]^T$ and $z_i(k+1)$ is the state vector of the *i*th node at (*k*+1)th hop.

It must be noted that (8) also bonds the state vector, $z(k+1)$, to the scalar value of the state, $x(k+1)$.

In other words, where *u(k)* in (7) gives the scalar state, *x*(*k*+1), for the next hop, it also selects the *x*(*k*+1)th column of $\underline{\mathbf{Z}}(k+1)$ as the next state vector.

We are trying to minimize the infinite horizon cost function, (2) given by

$$J(k) = \sum_{i=k}^{\infty} LC_{i,i+1} = LC_{k,k+1} + J(k+1) \tag{9}$$

where $LC_{k,k+1} = u^T(k)R(k)u(k)$.

Assume $R(k)$ is an $M \times M$ diagonal matrix whose diagonal elements are the updated link costs from the current node to each of the next nodes.

The optimal decision for (7) that also minimizes (9) is found by [36]

$$u^*(k) = -\frac{1}{2} R^{-1}(k) B^T \frac{\partial J^*(k+1)}{\partial x(k+1)} \tag{10}$$

where $u^*(k)$ and $J^*(k)$ are the optimal decision and smooth optimal cost function, respectively.

Using the approximation property of OLAs [38], the cost function can be represented by an OLA as

$$J(k) = W_J^T \sigma(z(k)) + \varepsilon_J(k) \tag{11}$$

where W_J is the constant target OLA parameter, $\varepsilon_J(k)$ is the bounded approximation error, and $\sigma(\bullet)$ is the vector activation function for the cost approximation OLA scheme. The approximation holds for all *z* in a compact set ***S***. The approximation error is assumed to be bounded [38] as

$$\left\| \varepsilon_J(k) \right\| \le \varepsilon_{JM} \tag{12}$$

where ε_{JM} is a known bound dependent on ***S***.

Approximating the cost function by an OLA renders

$$\hat{J}(k) = \hat{W}_J^T(k)\sigma(z(k)) = \hat{W}_J^T(k)\sigma(k) \ , \tag{13}$$

where $\hat{J}(k)$ is the approximation of the cost function, and $\hat{W}_J(k)$ is the actual parameter vector for the target OLA parameter vector, W_J. Vector $\sigma(k)$ is a set of bounded activation functions whose elements are chosen to be linearly independent basis functions. Defining the error in the cost function as in [39]

$$e_J(k) = LC_{k-1,k} + \hat{W}_J^T(k)\sigma(k) - \hat{W}_J^T(k)\sigma(k-1) . \tag{14}$$

The error can be rewritten as

$$e_J(k+1) = LC_{k,k+1} + \hat{W}_J^T(k+1)\left(\sigma(k+1) - \sigma(k)\right) . \tag{15}$$

Then the auxiliary cost error is defined as

$$E_J^T(k+1) = Y^T(k) + X^T(k)\hat{W}_J(k+1) \ E_J^T(k+1) = Y^T(k) + X^T(k)\hat{W}_J(k+1) , \tag{16}$$

where $Y(k) = [LC_{k,k+1} \quad LC_{k-1,k} \cdots LC_{k-j,k-j+1}]$ and

$X(k) = [\Delta\sigma(k+1) \quad \Delta\sigma(k) \cdots \Delta\sigma(k-j+1)]$ with $\Delta\sigma(k+1) = \sigma(k+1) - \sigma(k)$, $0 < j < k-1 \in \mathbf{N}$. Following the steps in [39], the cost function parameter update will be

$$\hat{W}_J(k+1) = X(k)\left(X^T(k)X(k)\right)^{-1}\left(\alpha_J E_J^T(k) - Y^T(k)\right), \tag{17}$$

where $0 < \alpha_J < 1$. Replacing (17) in (16) results in

$$E_J(k+1) = \alpha_J E_J(k). \tag{18}$$

Definition 1: A set of functions $\sigma(z) = \{\sigma_i(z)\}_1^L$ is linearly independent when $\sum_{i=1}^{L} \lambda_i \sigma_i(z) = 0$ holds only if $\lambda_1 = \lambda_2 = \cdots = \lambda_L = 0$.

Remark 1: In two consecutive hops of the route, *k* and *k*+1, it is impossible to have $z(k) = z(k+1)$. The two state vectors are different in at least one element, $z_i(k)$, and $z_i(k+1)$ – the one associated with the initial route cost to BS. By definition, the route cost to BS is initialized by propagating a beacon from BS. This cost, $C(k)$, cannot be the same in two consecutive hops.

Lemma 1: Let $z(k)$ and $z(k+1)$ be the state vectors in hops *k* and *k*+1. Also let $\sigma(z(k)) = \{\sigma_i(z(k))\}_1^L$ be a set of linearly independent functions. Then the set $\Delta\sigma(z(k+1)) = \{\sigma_i(z(k+1)) - \sigma_i(z(k))\}_1^L$ is also linearly independent.

Proof: Suppose that *Lemma 1* is not true. Then there exists some set $\{\lambda_i\}_1^L$ with some nonzero λ_is such that

$$\sum_{i=1}^{L} \lambda_i \Delta\sigma_i(z(k+1)) = 0. \tag{19}$$

It can be rewritten as

$$\begin{aligned}&\sum_{i=1}^{L}\lambda_i\left(\sigma_i(z(k+1))-\sigma_i(z(k))\right)\\&=\sum_{i=1}^{L}\lambda_i\left(\sigma_i(k+1)-\sigma_i(k)\right)\\&=\sum_{i=1}^{L}\lambda_i\sigma_i(k+1)-\sum_{i=1}^{L}\lambda_i\sigma_i(k)\\&=0\end{aligned} \tag{20}$$

Equality (20) may hold in two cases.

Case I. $\sum_{i=1}^{L}\lambda_i\sigma_i(k+1)=\sum_{i=1}^{L}\lambda_i\sigma_i(k)\neq 0$. This case is rejected by *Remark 1***.**

Case II. $\sum_{i=1}^{L}\lambda_i\sigma_i(k+1)=\sum_{i=1}^{L}\lambda_i\sigma_i(k)=0$. This case contradicts the hypothesis of linear independency of $\sigma(k)=\{\sigma_i(k)\}_1^L$ (and also that of $\sigma(k+1)=\{\sigma_i(k+1)\}_1^L$).

Therefore set $\Delta\sigma(z(k+1))=\{\sigma_i(z(k+1))-\sigma_i(z(k))\}_1^L$ is linearly independent. ■

Remark 2. Lemma 1 implies that matrix $X^T(k)X(k)$ in (17) is invertible if $z(k)\neq 0$. Recalling the definition of the cost function (2) and OLA approximation (13), it can be noted that when $z(k)=0$, both definitions of the cost become zero. State vector $z(k)$ is zero only at BS, because only at BS all the three elements of $z(k)$ (queue length, link delay, and initial cost to BS) are zero. In other words, once the system states converge to zero (upon arrival at BS) the cost function approximation can no longer be updated. This can be viewed as a persistency of excitation (PE) requirement for the inputs to the cost function OLA. That is, the system states must be persistently exiting long enough for the OLA to learn the optimal cost function.

Remark 3. It is important to use bounded activation functions in the OLA, *e.g.*

radial basis functions or saturated polynomial basis function such as $\sigma(z) = S(\theta(z)) = \left(1 - e^{-\theta(z)}\right) \Big/ \left(1 + e^{-\theta(z)}\right)$. This guarantees $\|\sigma(k+1)\| \le \sigma_M$, which will be used in the future proofs.

Let the OLA parameter estimation error be defined as

$$\tilde{W}_J(k) = W_J - \hat{W}_J(k). \tag{21}$$

Recalling the ideal OLA cost function (11), the general definition of the cost function (2) can be rewritten as

$$W_J^T \sigma(k) + \varepsilon_J(k) = LC_{k,k+1} + W_J^T \sigma(k+1) + \varepsilon_J(k+1). \tag{22}$$

Rearranging the terms, the link cost can be derived as

$$LC_{k,k+1} = W_J^T \sigma(k) - W_J^T \sigma(k+1) + \varepsilon_J(k) - \varepsilon_J(k+1). \tag{23}$$

Recalling that $\Delta\sigma(k+1) = \sigma(k+1) - \sigma(k)$ and defining

$$\Delta\varepsilon_J(k) = \varepsilon_J(k+1) - \varepsilon_J(k), \tag{24}$$

equation (23) can be rewritten as

$$LC_{k,k+1} = -W_J^T \Delta\sigma(k) - \Delta\varepsilon_J(k). \tag{25}$$

Substituting $LC_{k-1,k}$ and $LC_{k,k+1}$ in (14) and(15), respectively, one gets

$$\begin{aligned} e_J(k) &= -W_J^T \Delta\sigma(k) - \Delta\varepsilon_J(k-1) \\ &\quad + \hat{W}_J^T(k)\left(\sigma(k) - \sigma(k-1)\right) \\ &= -\tilde{W}_J^T(k)\Delta\sigma(k) - \Delta\varepsilon_J(k-1) \end{aligned}, \tag{26}$$

and

$$e_J(k+1) = -\tilde{W}_J^T(k+1)\Delta\sigma(k+1) - \Delta\varepsilon_J(k). \tag{27}$$

Recalling $e_J(k+1)=\alpha_J e_J(k)$ from (18), one will get

$$\begin{aligned}\tilde{W}_J^T(k+1)&=\alpha_J\tilde{W}_J^T(k)\Delta\sigma(k)\Delta\sigma^T(k+1)\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\\&+\left(\alpha_J\Delta\varepsilon_J(k-1)-\Delta\varepsilon_J(k)\right)\Delta\sigma^T(k+1)\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\\&=\left(\alpha_J\tilde{W}_J^T(k)\Delta\sigma(k)+\alpha_J\Delta\varepsilon_J(k-1)-\Delta\varepsilon_J(k)\right)\\&\cdot\Delta\sigma^T(k+1)\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\end{aligned}\tag{28}$$

And consequently,

$$\begin{aligned}\tilde{W}_J\ (k+1)=&\left(\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\Delta\sigma(k+1)\right)\\&\cdot\left(\alpha_J\tilde{W}_J^T(k)\Delta\sigma(k)+\alpha_J\Delta\varepsilon_J(k-1)-\Delta\varepsilon_J(k)\right)^T\end{aligned}\tag{29}$$

In the following, the boundedness of the cost function error (14), and the OLA parameter estimation error will be investigated.

Definition 3 [38]*:* An equilibrium point z_e is said to be *uniformly ultimately bounded (UUB)* if there exists a compact set $S\subset\Re^{n^*}$ so that for all initial states $z_e\in S$ there exists a bound $B\geq 0$ and a time $T(B,z_e)$ such that $\left\|z(k)-z_e\right\|\leq B$ for all $k\geq k_0+T$.

Theorem 1: *(Boundedness of the Cost OLA Errors).* Let $z(k)$ be any state vector in the network along the possible routes from the source node to BS, and let the cost OLA parameter be updated as in (17). Then, the cost errors (14) and (29) are *UUB*.

Proof: Consider the Lyapunov function candidate

$$V_J(k)=e_J^2(k)+\frac{1}{\beta}tr\left\{\tilde{W}_J^T(k)\tilde{W}_J\ (k)\right\}\tag{30}$$

where $\beta>0$.

The first difference of (30) will be

$$\begin{aligned}\Delta V_J(k) &= V_J(k+1) - V_J(k) \\ &= \left(e_J^2(k+1) - e_J^2(k)\right) \\ &+ \frac{1}{\beta} tr\left\{\tilde{W}_J^T(k+1)\tilde{W}_J(k+1) - \tilde{W}_J^T(k)\tilde{W}_J(k)\right\}\end{aligned} \quad . \tag{31}$$

Using (28), and recalling $e_J(k+1) = \alpha_J e_J(k)$ yields

$$\begin{aligned}\Delta V_J(k) &= \left(\alpha_J^2 - 1\right) e_J^2(k) \\ &- \frac{1}{\beta} tr\left\{\tilde{W}_J^T(k)\tilde{W}_J(k)\right\} \\ &+ \frac{1}{\beta} tr\left\{\tilde{W}_J^T(k+1)\tilde{W}_J(k+1)\right\}\end{aligned} \quad . \tag{32}$$

Recalling $0 < \alpha_J < 1$, and $tr\left\{\tilde{W}_J^T(k)\tilde{W}_J(k)\right\} = \left\|\tilde{W}_J(k)\right\|_F^2$, it can be noted that the first two terms of (32) are less than zero. To simplify the third term,

$$\begin{aligned}&\tilde{W}_J^T(k+1)\tilde{W}_J(k+1) = \left(\alpha_J \tilde{W}_J^T(k)\Delta\sigma(k) + \alpha_J \Delta\varepsilon_J(k-1) + \Delta\varepsilon_J(k)\right) \\ &.\Delta\sigma^T(k+1)\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1} \\ &.\left(\Delta\sigma\ (k+1)\Delta\sigma^T(k+1)\right)^{-1}\Delta\sigma(k+1) \\ &.\left(\alpha_J \tilde{W}_J^T(k)\Delta\sigma(k) + \alpha_J \Delta\varepsilon_J(k-1) + \Delta\varepsilon_J(k)\right)^T\end{aligned} \tag{33}$$

It can be written as

$$\tilde{W}_J^T(k+1)\tilde{W}_J(k+1) = \mathbf{\Psi} + \mathbf{\Pi_1} + \mathbf{\Pi_2} + \mathbf{E}, \tag{34}$$

where

$$\begin{aligned}&\mathbf{\Psi} = \alpha_J^2 \tilde{W}_J^T(k)\Delta\sigma(k)\Delta\sigma^T(k+1)\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1} \\ &.\left(\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\right)^T \Delta\sigma(k+1)\Delta\sigma^T(k)\tilde{W}_J(k)\end{aligned} \quad , \tag{35}$$

and

$$\mathbf{\Pi_1} = \alpha_J \tilde{W}_J^T(k)\Delta\sigma(k)\Delta\sigma^T(k+1)\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}$$
$$.\left(\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\right)^T \Delta\sigma(k+1)\left(\alpha_J\Delta\varepsilon_J(k-1)+\Delta\varepsilon_J(k)\right) \quad (36)$$

$$\mathbf{\Pi_2} = \left(\alpha_J\Delta\varepsilon_J(k-1)+\Delta\varepsilon_J(k)\right)\Delta\sigma^T(k+1)$$
$$.\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\left(\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\right)^T \quad (37)$$
$$.\Delta\sigma(k+1).\alpha_J\Delta\sigma^T(k)\tilde{W}_J(k)$$

$$\mathbf{E} = \left(\alpha_J\Delta\varepsilon_J(k-1)+\Delta\varepsilon_J(k)\right)^2\Delta\sigma^T(k+1)$$
$$.\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\left(\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\right)^T \quad (38)$$
$$.\Delta\sigma(k+1)$$

Let $\mathbf{\Sigma} = \left(\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\right)^T$, and $y = \Delta\sigma(k+1)$, where $\mathbf{\Sigma} \in \Re^{L\times L}$, and $y \in \Re^{L\times 1}$. It can be shown that

$y^T\mathbf{\Sigma}^T\mathbf{\Sigma}y = \|\mathbf{\Sigma}y\|_2^2$. This equality will be used to simplify (35), (36), (37) and (38) as follows.

$$\mathbf{\Psi} = \alpha_J^2.\|\mathbf{\Sigma}y\|_2^2.\tilde{W}_J^T(k)\Delta\sigma(k)\Delta\sigma^T(k)\tilde{W}_J(k), \quad (39)$$

$$\mathbf{\Pi_1} = \mathbf{\Pi_2}$$
$$= \alpha_J.\|\mathbf{\Sigma}y\|_2^2.\left(\alpha_J\Delta\varepsilon_J(k-1)+\Delta\varepsilon_J(k)\right).\tilde{W}_J^T(k)\Delta\sigma(k), \quad (40)$$

$$\mathbf{E} = \left(\alpha_J\Delta\varepsilon_J(k-1)+\Delta\varepsilon_J(k)\right)^2.\|\mathbf{\Sigma}y\|_2^2, \quad (41)$$

Now the third term of (32) can be rewritten as

$$tr\{\mathbf{\Psi}+\mathbf{\Pi_1}+\mathbf{\Pi_2}+\mathbf{E}\} = tr\{\mathbf{\Psi}\}+2tr\{\mathbf{\Pi_1}\}+tr\{\mathbf{E}\}$$
$$= \alpha_J^2.\|\mathbf{\Sigma}y\|_2^2.tr\left\{\tilde{W}_J^T(k)\Delta\sigma(k)\Delta\sigma^T(k)\tilde{W}_J(k)\right\}$$
$$+2\alpha_J.\|\mathbf{\Sigma}y\|_2^2.\left(\alpha_J\Delta\varepsilon_J(k-1)+\Delta\varepsilon_J(k)\right).tr\left\{\tilde{W}_J^T(k)\Delta\sigma(k)\right\} \quad (42)$$
$$+\left(\alpha_J\Delta\varepsilon_J(k-1)+\Delta\varepsilon_J(k)\right)^2.\|\mathbf{\Sigma}y\|_2^2$$

It can be simplified as

$$\begin{aligned} & tr\left\{\boldsymbol{\Psi}+\boldsymbol{\Pi}_{\mathbf{1}}+\boldsymbol{\Pi}_{\mathbf{2}}+\mathbf{E}\right\} \\ & =\alpha_J^2.\left\|\boldsymbol{\Sigma} y\right\|_2^2.\left\|\tilde{W}_J^T(k)\Delta\sigma(k)\right\|_F^2 \\ & +2\alpha_J.\left\|\boldsymbol{\Sigma} y\right\|_2^2.\left(\alpha_J\Delta\varepsilon_J(k-1)+\Delta\varepsilon_J(k)\right).tr\left\{\tilde{W}_J^T(k)\Delta\sigma(k)\right\} \\ & +\left(\alpha_J\Delta\varepsilon_J(k-1)+\Delta\varepsilon_J(k)\right)^2.\left\|\boldsymbol{\Sigma} y\right\|_2^2 \end{aligned} \tag{43}$$

Note that $\left\|\boldsymbol{\Sigma} y\right\|_2 \le \left\|\boldsymbol{\Sigma}\right\|_F \cdot \left\|y\right\|_2$. Also from ***Remark 2*** it can be concluded that $\Delta\sigma(k+1)\Delta\sigma^T(k+1)$ is invertible, and $\left\|\boldsymbol{\Sigma}\right\|_F^2 \le \frac{1}{\Delta\sigma_{\min}^2}$.In consecutive hops of the route, there exists a computable positive constant such that $\left\|\Delta\sigma(k)\right\| \le \Delta\sigma_M$. Also recall (12) expressing that $\left\|\varepsilon_J(k)\right\| \le \varepsilon_{JM}$. With the above assumptions, and substituting (43) in (32), the first difference of the Lyapunov candidate is simplified as

$$\begin{aligned} \Delta V_J(k) \le & -\left(\alpha_J^2-1\right)e_J^2(k)-\left(\frac{1}{\beta}-\alpha_J^2.\frac{\Delta\sigma_M^4}{\beta\Delta\sigma_{\min}^2}\right)\left\|\tilde{W}_J(k)\right\|_F^2 \\ & -2\alpha_J\cdot\frac{\Delta\sigma_M^2}{\beta\Delta\sigma_{\min}^2}\left(\alpha_J+1\right)\varepsilon_{JM}.\left(\left|e_J(k)\right|+\varepsilon_{JM}\right) \\ & +\left(\alpha_J+1\right)^2\varepsilon_{JM}^2\cdot\frac{\Delta\sigma_M^2}{\beta\Delta\sigma_{\min}^2} \end{aligned} \quad . \tag{44}$$

The first three terms of (44) are less than zero if $\alpha_J^2 < \min\left\{1, \Delta\sigma_M^4 / \Delta\sigma_{\min}^2\right\}$. Further, $\Delta V_J(k)$ is less than zero if

$$\left|e_J(k)\right| > \sqrt{\frac{\left(\alpha_J+1\right)^2\varepsilon_{JM}^2\Delta\sigma_M^2}{\beta\left(1-\alpha_J^2\right)\Delta\sigma_{\min}^2}} \tag{45}$$

or

$$\left\|\tilde{W}_J(k)\right\|_F > \sqrt{\frac{\left(\alpha_J+1\right)^2\varepsilon_{JM}^2\Delta\sigma_M^2}{\left(\Delta\sigma_{\min}^2-\alpha_J^2\Delta\sigma_M^4\right)}}\,. \tag{46}$$

According to standard Lyapunov theory [38], it can be concluded that $\Delta V_J(k)$ is less than zero outside the compact set given above. That is, the cost function error (14), and the OLA parameter estimation error (21) are *UUB.* ■

Next in the following corollary, it will be shown that the decision vector approaches the optimal value.

Corollary 1: *(Boundedness of the decision vector).* Let $\hat{u}(k)$ be any decision vector in the network along the possible routes from the source node to BS, which is updated as

$$\hat{u}(k) = -\frac{1}{2} R^{-1}(k) B^T \frac{\partial \hat{J}(k+1)}{\partial x(k+1)} \quad . \tag{47}$$

Then there exists a bound $\varepsilon_r \geq 0$ such that $\left\| \hat{u}(k) - u^*(k) \right\| \leq \varepsilon_r$ as $k \to \infty$.

Proof: Recall the optimal decision policy

$$u^*(k) = -\frac{1}{2} R^{-1}(k) B^T \frac{\partial J^*(k+1)}{\partial x(k+1)} .$$

Also recall that cost is estimated by an OLA as $\hat{J}(k) = \hat{W}_J^T(k)\sigma(k)$, where

$\hat{W}_J(k+1) = X(k)\left(X^T(k)X(k)\right)^{-1}\left(\alpha_J E_J^T(k) - Y^T(k)\right)$.

Then

$$\begin{aligned} &\left\| \hat{u}(k) - u^*(k) \right\| = \\ &\left\| -\frac{1}{2} R^{-1}(k) B^T \frac{\partial \hat{J}(k+1)}{\partial x(k+1)} + \frac{1}{2} R^{-1}(k) B^T \frac{\partial J^*(k+1)}{\partial x(k+1)} \right\| \\ &\leq \left\| -\frac{1}{2} R^{-1}(k) B^T \right\| \cdot \left\| \frac{\partial J^*(k+1)}{\partial x(k+1)} - \frac{\partial \hat{J}(k+1)}{\partial x(k+1)} \right\| \end{aligned} \tag{48}$$

The second multiplicand of (48) can be simplified as

$$\left\|\frac{\partial J^*(k+1)}{\partial x(k+1)}-\frac{\partial \hat{J}(k+1)}{\partial x(k+1)}\right\|=$$

$$\left\|\frac{\partial}{\partial x(k+1)}\left(W_J^{*T}\sigma(k+1)+\varepsilon_J(k+1)-\hat{W}_J^T(k+1)\sigma(k+1)\right)\right\|$$

$$=\left\|\frac{\partial}{\partial x(k+1)}\left(\tilde{W}_J^T(k+1)\sigma(k+1)+\varepsilon_J(k+1)\right)\right\|$$

$$\le\left\|\frac{\partial}{\partial x(k+1)}\left(\tilde{W}_J^T(k+1)\sigma(k+1)\right)\right\|+\left\|\frac{\partial \varepsilon_J(k+1)}{\partial x(k+1)}\right\|$$

$$\le\left\|\frac{\partial \tilde{W}_J^T(k+1)}{\partial x(k+1)}\sigma(k+1)+\tilde{W}_J^T(k+1)\frac{\partial \sigma(k+1)}{\partial x(k+1)}\right\|+\left\|\frac{\partial \varepsilon_J(k+1)}{\partial x(k+1)}\right\|$$

$$\le\left\|\frac{\partial \tilde{W}_J^T(k+1)}{\partial x(k+1)}\right\|\cdot\left\|\sigma(k+1)\right\|+\left\|\tilde{W}_J^T(k+1)\right\|_F\cdot\left\|\frac{\partial \sigma(k+1)}{\partial x(k+1)}\right\|$$

$$+\left\|\frac{\partial \varepsilon_J(k+1)}{\partial x(k+1)}\right\| \tag{49}$$

Recalling that $\left\|\sigma(k+1)\right\|\le\sigma_M$ (***Remark 3***), $\left\|\tilde{W}_J^T(k+1)\right\|_F\le\beta_{WJ}$ (***Theorem 1***),

$\left\|\frac{\partial \varepsilon_J(k+1)}{\partial x(k+1)}\right\|\le\varepsilon'_{JM}$, and $\left\|\frac{\partial \sigma(k+1)}{\partial x(k+1)}\right\|\le\sigma'_M$, (49) can be rewritten as

$$\left\|\frac{\partial J^*(k+1)}{\partial x(k+1)}-\frac{\partial \hat{J}(k+1)}{\partial x(k+1)}\right\|\le\left\|\frac{\partial}{\partial x(k+1)}\tilde{W}_J^T(k+1)\right\|.\sigma_M \cdot \\ +\beta_{WJ}\cdot\sigma'_M+\varepsilon'_{JM} \tag{50}$$

$$\frac{\partial \tilde{W}_J^T(k+1)}{\partial x(k+1)}=\frac{\partial}{\partial x(k+1)}\left(\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\Delta\sigma(k+1)\right)$$

$$=\frac{\partial\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}}{\partial x(k+1)}\cdot\Delta\sigma(k+1) \\ +\frac{\partial\Delta\sigma(k+1)}{\partial x(k+1)}\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1} \tag{51}$$

Given that $\frac{\partial\left(\mathbf{X}^{-1}\right)}{\partial z}=-\mathbf{X}^{-1}\frac{\partial \mathbf{X}}{\partial z}\mathbf{X}^{-1}$, (51) can be rewritten as

$$\begin{aligned}&\frac{\partial \tilde{W}_J^T(k+1)}{\partial x(k+1)}=\\&-\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\cdot\frac{\partial\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)}{\partial x(k+1)}\\&\cdot\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\\&+\frac{\partial\Delta\sigma(k+1)}{\partial x(k+1)}\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)^{-1}\end{aligned}\tag{52}$$

Recalling that $\left\|\left(\Delta\sigma(k+1)\Delta\sigma^T(k+1)\right)^{-1}\right\|_F\le\frac{1}{\Delta\sigma_{\min}}$,

$\left\|\frac{\partial\left(\Delta\sigma^T(k+1)\Delta\sigma\ (k+1)\right)}{\partial x(k+1)}\right\|\le\Delta\sigma'_{\min}$, $\left\|\Delta\sigma(k+1)\right\|\le\Delta\sigma_M$, and $\left\|\frac{\partial\Delta\sigma(k+1)}{\partial x(k+1)}\right\|\le\Delta\sigma'_M$, one

can write

$$\left\|\frac{\partial\tilde{W}_J^T(k+1)}{\partial x(k+1)}\right\|\le\frac{\Delta\sigma'_{\min}}{\Delta\sigma^2_{\min}}\cdot\Delta\sigma_M+\frac{\sigma'_M}{\Delta\sigma_{\min}}.\tag{53}$$

Then (50) can be rewritten as

$$\left\|\frac{\partial J^*(k+1)}{\partial x(k+1)}-\frac{\partial\hat{J}(k+1)}{\partial x(k+1)}\right\|\le\left(\frac{\Delta\sigma'_{\min}}{\Delta\sigma^2_{\min}}\cdot\Delta\sigma_M+\frac{\sigma'_M}{\Delta\sigma_{\min}}\right).\sigma_M+\beta_{WJ}\cdot\sigma'_M+\varepsilon'_{JM}.\tag{54}$$

Substituting (50) in (48),

$$\begin{aligned}&\left\|\hat{u}(k)-u^*(k)\right\|\le\left\|-\frac{1}{2}R^{-1}(k)B^T\right\|\cdot\left\|\frac{\partial J^*(k+1)}{\partial x(k+1)}-\frac{\partial\hat{J}(k+1)}{\partial x(k+1)}\right\|\\&\le\left\|-\frac{1}{2}R^{-1}(k)B^T\right\|\cdot\\&\left(\left(\frac{\Delta\sigma'_{\min}}{\Delta\sigma^2_{\min}}\cdot\Delta\sigma_M+\frac{\sigma'_M}{\Delta\sigma_{\min}}\right).\sigma_M+\beta_{WJ}\cdot\sigma'_M+\varepsilon'_{JM}\right)\end{aligned}\tag{55}$$

Assuming $\left\| -\frac{1}{2} R^{-1}(k) B^T \right\| \le \gamma_J$, it can be written

$$\left\| \hat{u}(k) - u^*(k) \right\| \le \gamma_J \cdot \left(\left(\frac{\Delta\sigma'_{\min}}{\Delta\sigma^2_{\min}} \cdot \Delta\sigma_M + \frac{\sigma'_M}{\Delta\sigma_{\min}} \right) . \sigma_M + \beta_{WJ} \cdot \sigma'_M + \varepsilon'_{JM} \right) \tag{56}$$

Therefore,

$$\left\| \hat{u}(k) - u^*(k) \right\| \le \varepsilon_r \ . \blacksquare \tag{57}$$

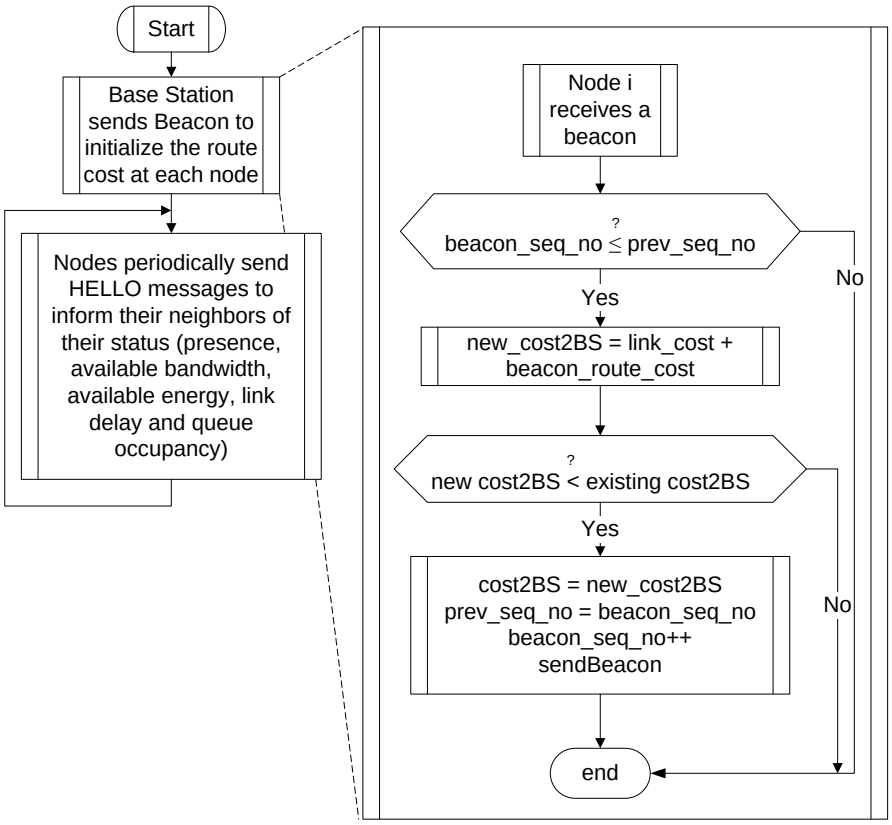

Fig. 6. Flowchart of initialization and topology update.

Fig. 6 represents the route cost initialization phase and also neighbor status update. Fig. 7 illustrates the on-line route discovery. Upon receiving a data packet towards BS, the node updates the NN cost approximator weights (using (17)), then using the updated weights, estimates the cost to Base Station for each of the one-hop neighbors. The next hop for the route is chosen such that the summation of the estimated cost to Base Station and the up-to-date link cost (see (3)) be minimized.

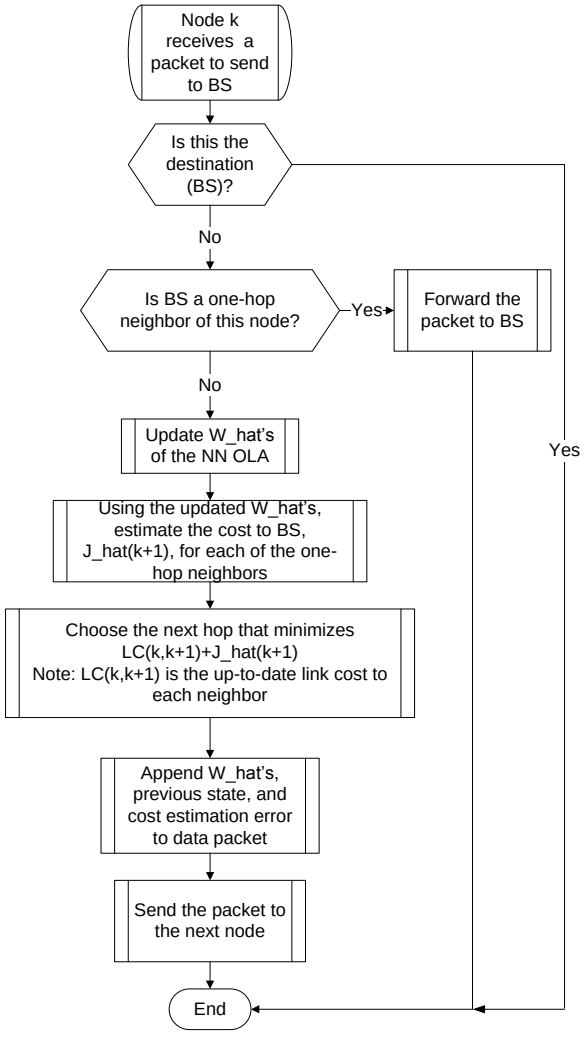

Fig. 7. Flowchart of next hop discovery upon receiving a packet towards BS.

C. An Example of Adaptive Dynamic Queue-Occupancy-Aware Routing

This section presents an example case in which the route for a source to Base Station is found by the proposed algorithm. Note that *k* represents the hop with source as the origin. Also *C*(*k*) is the initial cost to Base Station – found by the initial beacon. Fig. 8 shows three steps of an example network. The updated link costs between the nodes are shown in red. At each hop, the next hop nodes are sorted according to their initial cost to Base Station , *C*(*k*). The estimated route cost of the nodes at hop k , $\hat{J}(k)$ is calculated by the OLA, (13) , $\hat{J}(k) = \hat{W}_J^T(k)\sigma(k)$

At hop $k = n$, the state equation is

$$x(n+1) = x(n) + B(n)u(n)$$

where $x(n) = 2,$

$$B(n) = [-2 \quad -1 \quad 0],$$
$$u(n) = [1 \quad 0 \quad 0]^T \text{ or } [0 \quad 1 \quad 0]^T \text{ or } [0 \quad 0 \quad 1]^T$$

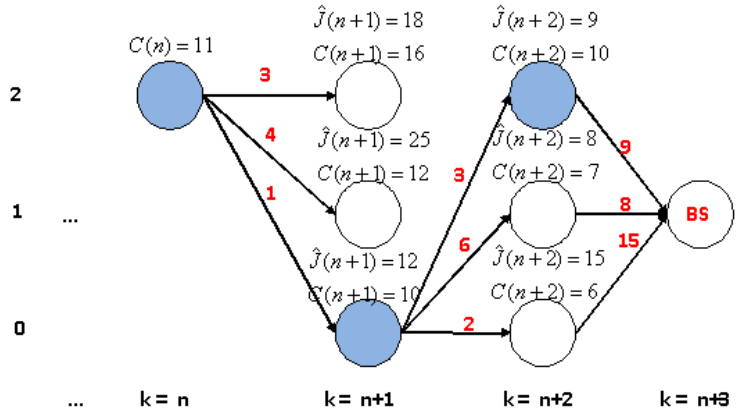

Fig. 8. Example network – finding the route to BS using the adaptive routing method.

Then from (47), one can calculate the

$$\hat{u}^*(n) = -\frac{1}{2} R^{-1}(n) B^T(n) \frac{\partial \hat{J}^*(n+1)}{\partial x(n+1)}$$

where $R(n) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 3 \end{bmatrix}$, $\hat{J}(n+1)$ is calculated by the OLA, (13) ,

$\hat{J}(k) = \hat{W}_J^T(k)\sigma(z(k)) = \hat{W}_J^T(k)\sigma(k)$,

and $\frac{\partial \hat{J}^*(n+1)}{\partial x(n+1)} = 13$.

Then the closest admissible $u(n)$ to $\hat{u}^*(n)$ is found (using Euclidean distance). It will be $u(n) = [1 \quad 0 \quad 0]^T$.

Then,

$$\begin{aligned} x(n+1) &= 2 + [-2 \quad -1 \quad 0][1 \quad 0 \quad 0]^T \\ &= 0 \end{aligned}$$

Now at $k = n+1$, the state equation will be

$$x(n+2) = x(n+1) + B(n+1)u(n+1)$$

where $x(n+1) = 0$,

$$\begin{aligned} &B(n+1) = [0 \quad 1 \quad 2], \\ &u(n+1) = [1 \quad 0 \quad 0]^T \text{ or } [0 \quad 1 \quad 0]^T \text{ or } [0 \quad 0 \quad 1]^T \end{aligned}$$

Then from (47), it can be written

$$\hat{u}^*(n+1) = -\frac{1}{2} R^{-1}(n+1) B^T(n+1) \frac{\partial \hat{J}^*(n+2)}{\partial x(n+2)}$$

where $R(n+1)=\begin{bmatrix}2 & 0 & 0\\ 0 & 6 & 0\\ 0 & 0 & 3\end{bmatrix}$ and $\frac{\partial \hat{J}^*(n+2)}{\partial x(n+2)}=-1$.

Then the closest admissible $u(n+1)$ to $\hat{u}^*(n+1)$ is found (using Euclidean distance). It will be $u(n+1)=[0 \quad 0 \quad 1]^T$.

Then,

$$x(n+2)=0+[0 \quad 1 \quad 2][0 \quad 0 \quad 1]^T=2.$$

At this hop, Base Station is the one-hop neighbor (with a link cost of 9) of the current node. It is selected as the next node, and therefore the route to Base Station is completed.

It must be noted that at each hop, the next node is not the node with the smallest estimated cost to Base Station, neither is the one with the smallest link cost to the current node. The next node is selected such that the summation of these two parameters be minimized. That is, at each hop the route is found such that the remaining route cost be minimized. This is in fact Bellman's principle of optimality [36].

V. SIMULATION RESULTS

This section presents the simulation results of running the adaptive dynamic routing protocol (using the OLA) and AODV on various scenarios and topologies. Network Simulator NS-2 [37] was modified, and the proposed routing protocol was implemented in NS-2. The simulation parameters are listed in TABLE II.

TABLE II

SIMULATION PARAMETERS

Routing Protocol	AODV and OLA
Average Hop Distance form BS	20, 30, 40, 50 hops
Simulation Area	Variable – depending on the size of the network
Number of Nodes	Variable - depending on the size of the network (190, 280, 370 and 460)
Mobility Model	Random way-point
Propagation Model	Two-way Ground Reflection
Transmission Range	250 m
Traffic Type	CBR (UDP)
Number of Connections (Flows)	10, 12, 14, …, 20
Packet Size	256, 512, 1024 and 2048 bytes
Data Rate	20 kbps, 100 kbps, 200 kbps, 500 kbps, 1000 kbps and 2000 kbps
Interface Queue	50 pkts

Various random scenarios were generated to provide a variable range of flows (10, 12, …, 20) with variable range of average distance from Base Station (20, 30, 40 and 50 hops).

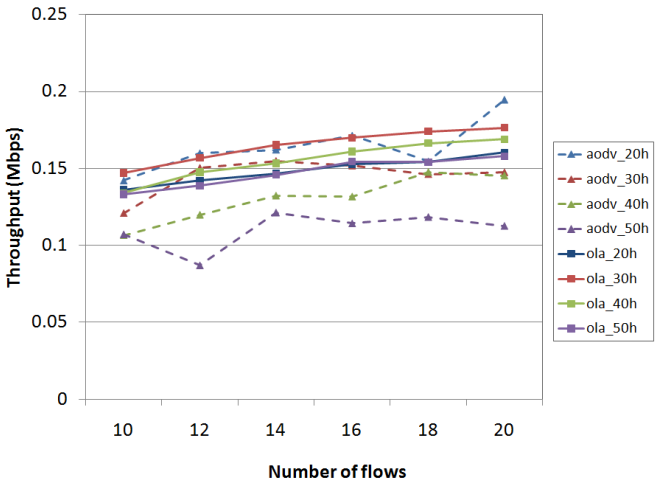

Fig. 9. Average throughput for variable number of flows. Average distance from BS: 20, 30, 40 and 50 hops. Packet size: 256 bytes. Data rate: 20 kbps.

Fig. 9 shows the average throughput of the network for various number of flows and various average distance from Base Station (route length), where packet size is 256 bytes and data rate is 20 kbps. It can be noted that as the average distance of the source to Base Station increases, our proposed method provides a higher throughput compared to AODV. Only for the average distance of 20 hops, AODV performs better. The reason is that the proposed method requires a minimum number of hops to converge. On the other hand, when the average distance from source to Base Station is 50 hops, the proposed scheme provides up to 40.18% improvement in throughput.

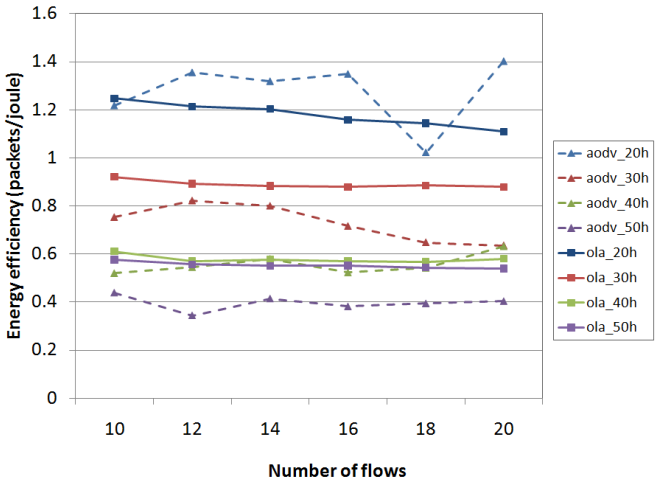

Fig. 10. Energy efficiency vs. number of flows. Average distance from BS: 20, 30, 40 and 50 hops. Packet size: 256 bytes. Data rate: 20 kbps.

Fig. 10 shows the energy efficiency of the network vs. number of flows for various average distance from Base Station (route length), where packet size is 256 bytes and data rate is 20 kbps. It is noticed that the energy efficiency is almost constant for the

various number of flows. However, as the average distance of the source to Base Station increases, our proposed method provides a higher energy efficiency compared to AODV. Fig. 11 presents a better view of this fact. Only for the average distance of 20 hops, AODV performs better. As mentioned earlier, the he proposed method requires a minimum number of hops to converge. On the other hand, when the average distance from source to Base Station is 50 hops, the proposed provides up to 33.46% improvement in energy efficiency.

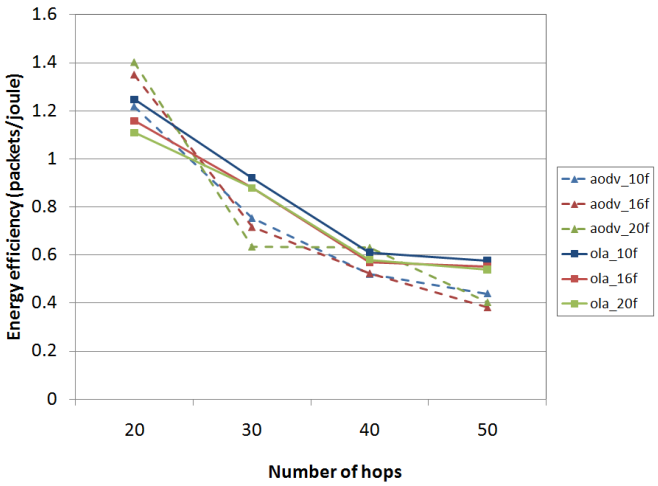

Fig. 11. Energy Efficiency vs. average distance from BS. Number of flows:10, 12, 14, 16, 18 and 20. Packet size: 256 bytes. Data rate: 20 kbps.

In order to evaluate the proposed cost metric, we calculated the route cost per flow using AODV and the proposed method. The link costs in both method were calculated using (5). Fig. 12. Shows the route cost per flow for six flows in the network, where packet size is 256 bytes and data rate is 20 kbps, and number of flows is 20. It can be noticed that in overall, the proposed method selects the routes with smaller costs (compared to AODV).

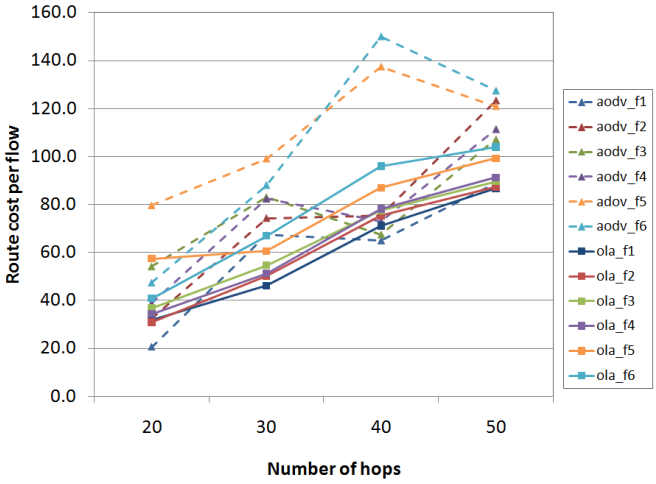

Fig. 12. Route cost per flow vs. average distance from BS. Number of flows: 20. Packet size: 256 bytes. Data rate: 20 kbps.

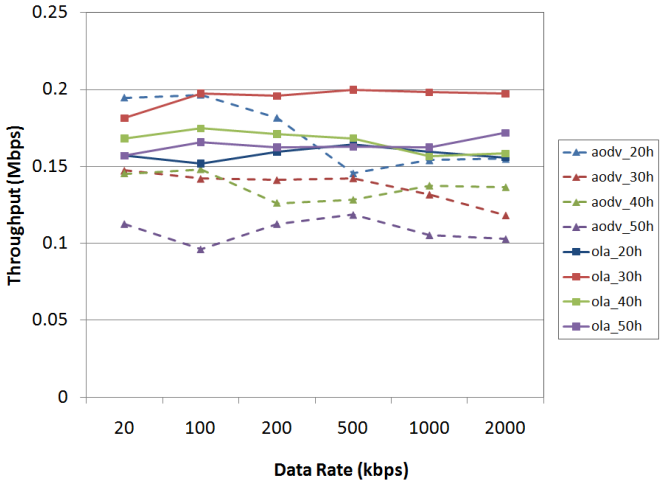

Fig. 13. Average throughput vs. data rate. Average distance from BS: 20, 30, 40 and 50 hops. Packet size: 256 bytes. Number of flows: 20.

Fig. 13 shows the average throughput of the network for variable data rate and various average distance from Base Station (route length), where packet size is 256 bytes

and number of flows is 20. It can be noticed that except for three cases, the proposed algorithm provides a higher throughput compared to AODV. Those three cases occur when the average distance to Base Station is 20 hops and data rate is 20k, 100k, or 200k. Recall the weakness of the proposed algorithm for short routes. However, when the average distance of the source increases, the proposed method provides improvements as high as 67.13% in throughput.

Fig. 14 shows the energy efficiency of the network vs. number of hops and for data rates (20kbps, 500kbps and 2Mbps), where packet size is 256 bytes and number of flows is 20. It can be noticed that as the average distance of the source to Base Station increases, our proposed method provides a higher energy efficiency compared to AODV. Only for the average distance of 20 hops and data rate of 20kbps, AODV performs better. The proposed algorithm can improve the energy efficiency by as high as 63.90%.

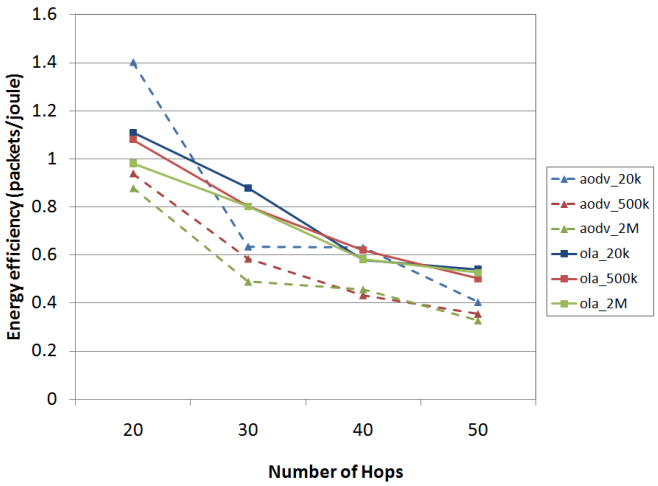

Fig. 14. Energy Efficiency vs. average distance from BS. Data Rates:20kbps, 500kbps and 1Mbps. Packet size: 256 bytes. Number of flows: 20.

Finally, the effect of varying packet size and average distance to Base Station was examined, where data rate is 2 Mbps and number of flows is 20.

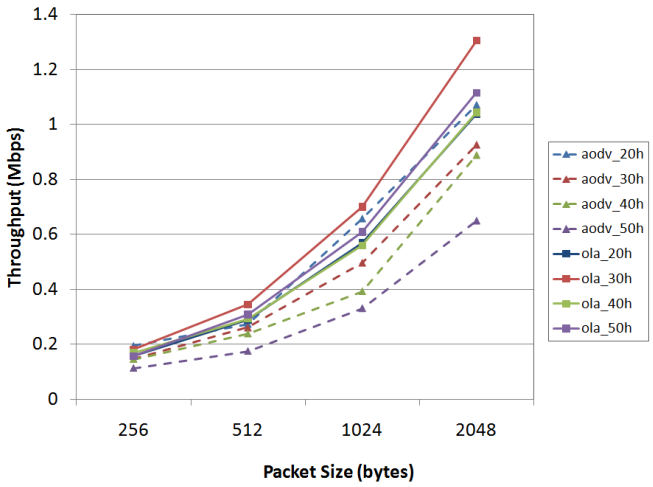

Fig. 15. Average throughput vs. packet size. Average distance from BS: 20, 30, 40 and 50 hops. Number of flows: 20. Data rate: 2 Mbps.

Fig. 15 illustrates the average throughput of the network vs. packet size. It is noticed that in overall, our proposed method provides a higher throughput. Also as the packet size increases, the throughput of both methods (AODV and our proposed method) increases. Furthermore, as the packet size increases, the improvement achieved by our proposed method increases (as high as 71.97%).

Fig. 16 presents the energy efficiency vs. packet size. It is noticed that in overall (except for the small packets and short routes), our proposed method provides a higher energy efficiency. Also as the average distance of the source nodes from Base Station (*i.e.* route length) increases, the improvement in the energy efficiency increases (as high as 65.72% - for packet size of 2048 bytes and average route length of 50 hops). This fact is illustrated in Fig. 17.

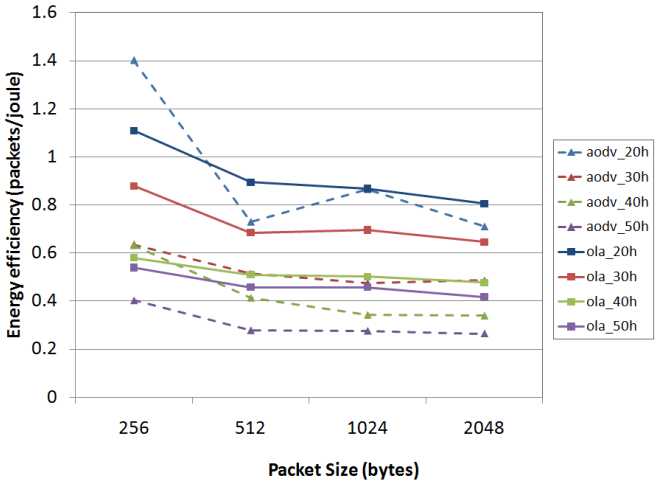

Fig. 16. Energy Efficiency vs. packet size. Average distance from BS: 20, 30, 40 and 50 hops. Number of flows: 20. Data rate: 2 Mbps.

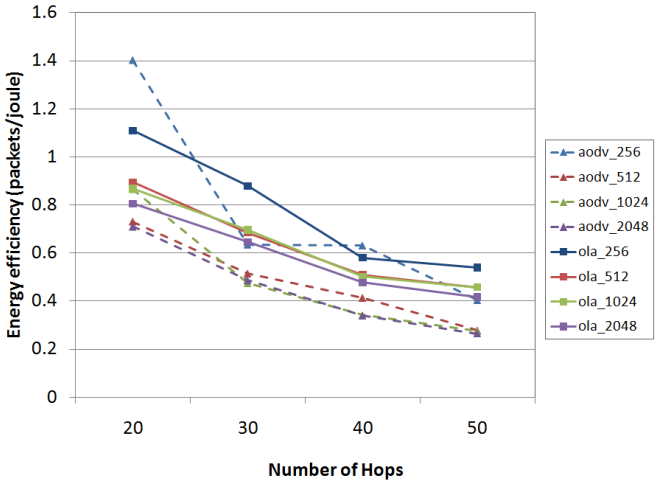

Fig. 17. Energy Efficiency vs. average distance from BS. Packet size: 256, 512, 1024, 2048. Number of flows: 20. Data rate: 2 Mbps.

VI. CONCLUSIONS

In this work, Bellman's principle of optimality and dynamic programming techniques, as well as OLA were utilized to find the routes reactively. A model for the network, as well as the state equation and cost function was presented. After finding the

initial cost to Base Station at each node (by propagating a beacon from Base Station), the link cost to the one-hop neighbors are updated frequently, and OLA is used to estimate the cost of the next node to Base Station. The boundedness of the OLA error and cost, and also the convergence of the decision signal to the optimal solution were proved.

Simulation results for various packet size, data rate, number of flows and average distance of the source node to Base Station verify that the proposed algorithm can improve the throughput and energy efficiency as high as 71.97% and 65.72%, respectively (compared to AODV). In the proposed algorithm, in case of link failure or congestion, there is no need to re-discover a new route. Instead, the next node is selected in real time. The proposed algorithm appears to outperform AODV in higher congestions and longer routes, also for larger packets. In the absence of congestion, the simple route cost metric of AODV, hop counts, is sufficient to find the optimum route. Using the proposed method in low congestions only increases the overhead, because it sends the OLA parameters along with the data packet. The same reasoning applies to the case of small packet size. Since the OLA parameters are piggy-backed to the data packets, a larger data packet would transmit a larger amount of data with the same amount of overhead. Finally, the proposed method performs better as the route length increases. This is due to the fact that the cost OLA needs to converge so that the optimal decision would be achieved.

ACKNOWLEDGMENT

Research supported in part by Intelligent Systems Center, Missouri S&T.

REFERENCES

[1] C. Hedrick, Routing Information Protocol, Internet-Request for Comments 1058, IETF, June 1988.

[2] J. Moy. OSPF Version 2. RFC 1583, Network Working Group, March 1994.

[3] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, 1:269-271, 1959.

[4] Suresh Singh, Mike Woo, and C. S. Raghavendra. "Power-aware routing in mobile ad hoc networks." *In Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 181-190, 1998.

[5] Javier Gomez, Andrew T. Campbell, Mahmoud Nagshineh, and Chatschik Bisdikian. "Power-aware routing in wireless packet networks," *In Proc. of Sixth IEEE International Workshop on Mobile Multi-media Communications*, November 1999.

[6] Qun Li, Javed Aslam, and Daniela Rus, "Online power-aware routing in wireless ad-hoc networks," *In MobiCom*, 2001.

[7] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks." *In Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, pages 8596, Rome, Italy, July 2001.

[8] Ivan Stojmenovic and Xu Lin. "Power-aware localized routing in wireless networks." *IEEE Trans. Parallel Distrib. Syst.*, 12(11):1122-1133, 2001.

[9] R. Bellman. Dynamic Programming. Princeton University Press, Princeton, New Jersey, 1957.

[10] L.R. Ford Jr. and D. R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, New Jersey, 1962.

[11] Charles Perkins and Pravin Bhagwat. "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers." *In ACM SIGCOMM '94*, pages 234-244, 1994.

[12] D. Johnson, "Routing in ad hoc networks of mobile host," *In Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, U.S., 1994.

[13] David B Johnson and David A Maltz, "Dynamic source routing in ad hoc wireless networks," In Imielinski and Korth, editors, Mobile Computing, volume 353. Kluwer Academic Publishers, 1996.

[14] David B Johnson, David A Maltz, Yih-Chun Hu, and Jorjeta G. Jetcheva, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). Internet-Draft, IETF, November 2001. draft-ietfmanet-dsr-06.txt.

[15] Yih-Chun Hu and David B. Johnson, "Caching strategies in on-demand routing protocols for wireless ad hoc networks," *In Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, August 2000.

[16] M. K. Marina and S. R. Das, "Performance of route caching strategies in dynamic source routing," In Proceedings of the 2nd Wireless Networking and Mobile Computing (WNMC), April 2001.

[17] Malcolm C. Easton and Ronald Fagin, "Cold-start vs. warm-start miss ratios," *Communications ACM*, 21(10):866{872, 1978.

[18] Vincent D. Park and M. Scott Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," *In INFOCOM '97*, pages 1405-1413, 1997.

[19] V. Park and M. Corson, Temporally-Ordered Routing Algorithm (TORA): Version 1 Functional Specification. Internet-Draft, IETF, July 2001. draft-ietf-manet-tora-spec-04.txt.

[20] Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das, Ad hoc On-Demand Distance Vector (AODV) Routing. Internet-Draft, IETF, January 2002. draft-ietf-manet-aodv-10.txt.

[21] M. Jiang, J. Li, and Y. Tay. Cluster Based Routing Protocol (CBRP) Functional Specification. Internet-Draft, IETF, August 1998. draft-ietfmanet-cbrp-spec-00.

[22] Ching-Chuan Chiang and Mario Gerla, "Routing and multicast in multihop, mobile wireless networks," *In Proceedings of ICUPC '97*, 1997.

[23] P. Krishna, N.H. Vaidya, M. Chatterjee, and D.K. P Radhan, "A clusterbased approach for routing in dynamic networks," *In Proceedings of the Second USENIX Symposium on Mobile and Location-Independent Computing*, 1995.

[24] Ching-Chuan Chiang, Hsiao-Kuang Wu, Winston Liu, and Mario Gerla, "Routing in clustered multihop mobile wireless networks with fading channel," *In Proceedings of IEEE Singapore International Conference on Networks (SICON'97)*, 1997.

[25] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, and A. Qayyum et L. Viennot, "Optimized link state routing protocol," *In IEEE INMIC*, 2001.

[26] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," Internet-Request for Comments 3626, IETF, October 2003.

[27] R. Ogier, F. Templin, and M. Lewis, "Topology dissemination based on reverse-path forwarding (TBRPF)," Internet-Request for Comments 3684, IETF, Febuary 2004.

[28] Z. Haas and M. Pearlman, "The zone routing protocol for highly reconfigurable ad-hoc networks," *In Proceedings of ACM SIGCOMM 98,*August 1998.

[29] C-K Toh, "Associativity-based routing for ad-hoc mobile networks," *Wireless Personal Communications Journal*, 4(2):103-139, 1997.

[30] C.-K. Toh, Long-Lived Ad Hoc Routing based on the Concept of Associativity. Internet-Draft, IETF, March 1999. draft-ietf-manet-longlivedadhoc-routing-00.txt.

[31] D. Decouto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless networks," *in Proceedings of MobiCom*, 2003.

[32] J. Padhye, R. Draves, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," *in Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM '04)*, pp. 114–128, Philadelphia, Pa, USA, 2004.

[33] N. Regatte and S. Jagannathan, "Optimized energy-delay routing in ad hoc wireless networks," *in Proc. Of the World Wireless Congress*, May 2005.

[34] S. Ratnaraj, S. Jagannathan, V. Rao, "OEDSR: optimized energy-delay sub-network routing in wireless sensor network," *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*, vol., no., pp.330-335.

[35] R. Anguswamy, M. Zawodniok, S. Jagannathan, "A multi-interface multi-channel routing (MMCR) protocol for wireless ad hoc networks," *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, vol., no., pp.1-6, 5-8 April 2009.

[36] F. L. Lewis and V. L. Syrmos, Optimal Control, 2nd ed. Hoboken, NJ: Wiley, 1995.

[37] http://www.isi.edu/nsnam/ns/

[38] S. Jagannathan, Neural network control of nonlinear discrete-time systems, CRC Press, April 2006.

[39] T. Dierks and S. Jagannathan, "Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics," *Decision and Control, 2009. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, vol., no., pp.6750-6755, 15-18 Dec. 2009.

3. Dynamic Routing for Multi-Channel Multi-Interface Hybrid Wireless Networks Using Online Optimal Framework

Behdis Eslamnour and S. Jagannathan
Department of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, MO, USA
e-mail: {ben88, sarangap}@mst.edu

Abstract— A dynamic routing protocol is necessary to provide better performance in the presence of multiple wireless channels by adapting to the topology changes of wireless ad hoc networks such as hybrid networks and vehicular ad hoc networks under fading channels or channel uncertainties. In this paper, on-line estimators and Markov models are utilized to estimate fading channel conditions. Using the channel estimates, queue occupancy, available energy and link delay, approximate dynamic programming (ADP) techniques are utilized to find dynamic routes, while solving discrete-time Hamilton-Jacobi-Bellman (HJB) equation forward-in-time for route cost. Routing is accomplished for multiple paths and multiple channels through load balancing. The multi-path routes were shown to be near optimal. The performance of the proposed load balancing and the optimal route selection schemes for multi-channel multi-interface wireless network in the presence of fading channels is evaluated by simulations and comparing it to AODV.

Keywords- Adaptive Dynamic Routing; Approximate Dynamic Programming; Wireless Ad hoc Networks; Multi-path; Load Balancing; Multi-channel; Multi-Interface

I. INTRODUCTION

Wireless ad hoc networks are subject to frequent changes due to changes in their topology and traffic load. In the presence of fading channels, the networks also undergo a variety of changes caused by signal attenuation, interference, Doppler effect, shadowing effect, etc. These channel uncertainties can affect quality of Service (QoS) of the network if they are not addressed by the data transmission protocols. Extensive studies on fading channels, interferences and fading models have been made [1]-[4]. Statistical models for Rayleigh fading [1], Rician fading [4], and lognormal shadowing [5], have been developed to capture and describe the wireless environment. However, in order to make the analysis possible, assumptions are made [1][2][6] such as mean signal power of both the desired signal and the interferer signals are known – if not variable and the number of interferer signals is known. In addition, the speed at which the mobile nodes and their neighbors are moving is also considered known [45] – and constant. As much as these assumptions make the analysis of the wireless channels possible, they cannot be used to develop dynamic and adaptive protocols for the network with varying environment.

The dynamic environment makes the proactive routing protocols such as [7],[9] less effective, since it requires frequent updates throughout the network. On the other hand, most of the on-demand routing protocols [9],[10] establish the routes by sending route request messages and receiving a reply message either from the destination, or a node that has a route to the destination. Routing protocols for multi-radio multi-channel networks have been proposed [11] to improve the channel utilization and throughput. However, their reliance on route request and reply messages makes them susceptible to

increased routing overhead in dynamic environments due to frequent changes in routes. Furthermore, even in the interference-aware method [11], where the available bandwidth is estimated, the estimation is based on the idle time of the channel. However, the idle time of the channel does not give a realistic estimation of the available bandwidth because in fading channels, the channel may be left unused due to the RTS messages that never receive any CTS. This causes a large idle time of the channel, and consequently giving a wrong estimate of the available bandwidth.

In this paper, multiple channels and multiple interfaces are utilized throughout the network to improve the network capacity via learning automata-based dynamic channel allocation scheme [12]. The proposed routing protocol takes into account the link conditions as well as channel uncertainties. The channel characteristics are considered non-stationary and unknown in this routing scheme. By utilizing mean least square error (MLE) and Markov models for channel, and finding the time-varying and non-stationary probability outage, the effective available bandwidth is defined in order to realize a more realistic measure of the channel condition. The time-varying outage probability and effective available bandwidth are also used to balance load over multiple channels and multiple routes – if one route fails to provide sufficient bandwidth.

The proposed routing protocol is for ad hoc networks with base station. While it is not common to associate ad hoc networks with any base station or access point, base stations are utilized for ad hoc networks such as “multi-hop hybrid networks” (a combination of ad hoc and cellular networks) [15], “hybrid networks” [16],[17] and hierarchical ad hoc networks” [18]. Furthermore, VANETS (Vehicular Ad Hoc Networks) are another known field of ad hoc networks with roadside base stations. The

proposed routing protocol adapts the route based on link and channel conditions without flooding the network with new route request messages. Adaptive dynamic programming (ADP) techniques are utilized to find dynamic routes, while solving discrete-time Hamilton-Jacobi-Bellman equation forward-in-time for route cost in an online manner. It uses a neural network (NN) or any online approximator (OLA) to approximate the route cost to the destination. It provides reactive and dynamic forward-in-time solutions – as opposed to proactive and static backward-in-time solutions in traditional dynamic programming-based schemes [7]-[10]. Analytical proofs are offered wherever possible which is not common in standard dynamic-programming schemes. Furthermore, metrics such as effective available bandwidth, link delay, available energy and queue occupancy at nodes are taken into account in determining the route cost.

In summary, this paper introduces a time-varying definition of outage probability to estimate the channel condition which in turn is used in load balancing over multiple channel links. Moreover, on-demand adaptive dynamic routes are found forward-in-time by using an OLA. In order to determine the route cost, the OLA uses link and channel conditions and node status, i.e. link delay, available bandwidth, outage probability, queue occupancy and available energy. The paper is organized as follows. Section II presents the methodology that covers channel allocation, channel modeling, routing protocol and load balancing algorithm. Section III presents the simulation results, and Section IV concludes the paper.

II. METHODOLOGY

A. Channel Allocation

In [12], we proposed a distributed dynamic channel allocation scheme for wireless networks using adaptive learning automata whose nodes are equipped with single radio interfaces so that a more suitable channel can be selected. The proposed scheme, Adaptive Pursuit Reward-Inaction, runs periodically on the nodes, and adaptively finds the suitable channel allocation in order to attain a desired performance. This scheme is utilized in this work.

B. Available Bandwidth and Channel Uncertainties

In the presence of path loss, shadowing effect and path loss, the wireless links exhibit uncertainties which will reduce the available bandwidth. For such environments, the outage probability is defined as a way to quantify these uncertainties. The classic definition of outage probability of a communication link is [24]

$$P_{out} = \Pr\left(r < \gamma_{th}\right) = \int_{0}^{\gamma_{th}} p_r(r)dr \tag{1}$$

where r is signal-to-interference ratio (SIR), $p_r(r)$ is the probability distribution function (pdf) of SIR, and γ_{th} is the SIR threshold.

In this paper, we use the Markov model for communication link/channel. Consider a wireless communication link in the presence of fading channel. It can be modeled [45], [23] as a finite-state Markov model with N states (Fig. 1). A special case of this model is the widely known Gilbert-Elliot model [41], where a two-state Markov chain is used to model the channel as “good” and “bad” states.

In the following a case will be investigated with the assumption of unknown stationary probabilities, and then the unknown non-stationary case will be considered.

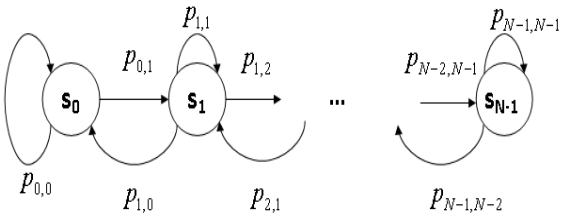

Fig. 1. A wireless communication link in the presence of Rayleigh fading channel and interference, modeled as an N-state Markov chain.

Case I. Estimation of Stationary Model

The assumption of known environment does seem a strong assumption for environments with mobile nodes moving in speeds and directions which are most likely unknown to their peers. By utilizing Maximum Likelihood Estimation (MLE) [22], the probability values can be learned on-line, and the *n*-step transition probabilities are proven to converge to certain time-invariant values as in stationary known environment (see Appendix A). Let the stationary system with *N* states be observed until *n* transitions have occurred. Then the MLE of the transition probabilities would be

$$\hat{p}_{i,j} = \frac{q_{i,j}}{q_i} \tag{2}$$

where $q_{i,j}$ is the total number of state transitions from s_i to s_j ($q_{i,j} = \sum_{k=1}^{n-1} \mathrm{I}_{X_k}(s_i).\mathrm{I}_{X_{k+1}}(s_j)$),

q_i is the number of occurrences of state s_i ($q_i = \sum_{j=0}^{N-1} q_{i,j}$), and $\sum_{j=0}^{N-1} q_i = n$. Recall that $\mathrm{I}_A(.)$ is the indicator function such that $\mathrm{I}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$.

As shown in Appendix A, as *n* increases, the estimated *n*-step transition probabilities converge to their true long-run probabilities. The convergence time *n* for the estimation can be determined as the smallest time within which a tolerable estimation error, ε, can be achieved such that

$$\left| \hat{p}_{i,j}(n) - p_{i,j}^{\infty} \right| \le \varepsilon \tag{3}$$

where $\hat{p}_{i,j}(n)$ is the estimated *n*-step transition probability, and $p_{i,j}^{\infty}$ is the long-run transition probability. As an example, consider a two state Markov chain with estimated 1-step transition probability matrix $\hat{P} = \begin{bmatrix} \hat{p}_{0,0} & \hat{p}_{0,1} \\ \hat{p}_{1,0} & \hat{p}_{1,1} \end{bmatrix}$. The *n*-step transition probability matrix can be obtained as

$$\hat{P}(n) = \hat{P}^n = \begin{bmatrix} \dfrac{\hat{p}_{1,0} + \hat{p}_{0,1}(1 - \hat{p}_{0,1} - \hat{p}_{1,0})^n}{\hat{p}_{0,1} + \hat{p}_{1,0}} & \dfrac{\hat{p}_{0,1} - \hat{p}_{0,1}(1 - \hat{p}_{0,1} - \hat{p}_{1,0})^n}{\hat{p}_{0,1} + \hat{p}_{1,0}} \\ \dfrac{\hat{p}_{1,0} - \hat{p}_{1,0}(1 - \hat{p}_{0,1} - \hat{p}_{1,0})^n}{\hat{p}_{0,1} + \hat{p}_{1,0}} & \dfrac{\hat{p}_{1,0} + \hat{p}_{1,0}(1 - \hat{p}_{0,1} - \hat{p}_{1,0})^n}{\hat{p}_{0,1} + \hat{p}_{1,0}} \end{bmatrix}, \tag{4}$$

where $0 \le \hat{p}_{0,0}, \hat{p}_{0,1}, \hat{p}_{1,0}, \hat{p}_{1,1} \le 1$, and $\left| 1 - \hat{p}_{0,1} - \hat{p}_{1,0} \right| < 1$. Then the long-run transition probability matrix can be obtained as

$$P^{\infty} = \lim_{n\to\infty} \hat{P}(n) = \begin{bmatrix} \dfrac{\hat{p}_{1,0}}{\hat{p}_{0,1}+\hat{p}_{1,0}} & \dfrac{\hat{p}_{0,1}}{\hat{p}_{0,1}+\hat{p}_{1,0}} \\ \dfrac{\hat{p}_{1,0}}{\hat{p}_{0,1}+\hat{p}_{1,0}} & \dfrac{\hat{p}_{1,0}}{\hat{p}_{0,1}+\hat{p}_{1,0}} \end{bmatrix} \tag{5}$$

In other words,

$$\hat{P}(n) = \begin{bmatrix} p_{0,0}^{\infty} & p_{0,1}^{\infty} \\ p_{1,0}^{\infty} & p_{1,1}^{\infty} \end{bmatrix} + \begin{bmatrix} \dfrac{\hat{p}_{0,1}(1-\hat{p}_{0,1}-\hat{p}_{1,0})^n}{\hat{p}_{0,1}+\hat{p}_{1,0}} & -\dfrac{\hat{p}_{0,1}(1-\hat{p}_{0,1}-\hat{p}_{1,0})^n}{\hat{p}_{0,1}+\hat{p}_{1,0}} \\ -\dfrac{\hat{p}_{1,0}(1-\hat{p}_{0,1}-\hat{p}_{1,0})^n}{\hat{p}_{0,1}+\hat{p}_{1,0}} & \dfrac{\hat{p}_{1,0}(1-\hat{p}_{0,1}-\hat{p}_{1,0})^n}{\hat{p}_{0,1}+\hat{p}_{1,0}} \end{bmatrix} \tag{6}$$

Recall that $\left|1-\hat{p}_{0,1}-\hat{p}_{1,0}\right|<1$, and notice the presence of $(1-\hat{p}_{0,1}-\hat{p}_{1,0})^n$ in the elements of the second term of $\hat{P}(n)$ in (6). Thus as *n* increases, the elements of the second term of $\hat{P}(n)$ decrease. Therefore *n* can be selected such that the second terms in $\hat{P}(n)$ of (6) be made smaller than the estimation error ε in (3)**.**

Case II. Non-Stationary Channel Model

Network topology and obstacles, on-going traffic, channel assignment (if multiple channels have been utilized) and mobility of the nodes cause channel variations which make the stationary model assumption unrealistic. In such cases, non-stationary Markov models for the communication links are necessary. The *N*-state Markov chain in Fig. 1 is valid with a change: the transition probabilities, $p_{i,j}$s, are time-varying, i.e. the probability of transition from state s_i at time step *k* to state s_j at time step *k*+1 is defined as

$$p_{i,j}(k) = P(X_{k+1} = s_j \mid X_k = s_i) \tag{7}$$

where $p_{i,j}(k)$ is not necessarily equal to $p_{i,j}(l)$, for $l \neq k$.

In such cases, we still use the MLE for transition probabilities, but instead of using the entire history of the state transitions, a sliding window of the recent history is used [21]. Here a sliding window W_i of length L to each state s_i (Fig. 2) is assigned. The window W_i stores the previous L transitions from state s_i.

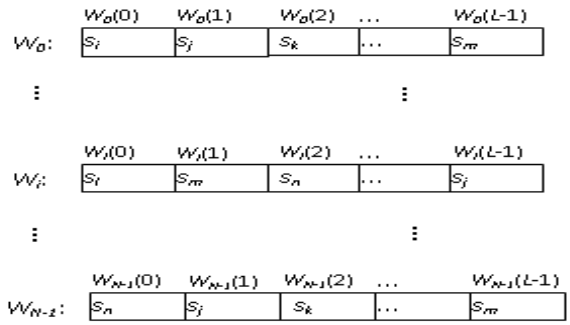

Fig. 2. Sliding windows of length L for the N state non-stationary communication link Markov model.

At each time step, depending on the previous state, the sliding window associated to that state is updated. Let the previous state at time k-1 be s_i , i.e. $s(k) = s_i$. Let the sliding window at time k-1 be W_i, and the state at time k becomes s_j. The sliding window W_i will be updated such that the elements of W_i be shifted towards the end of the window, and the new state s_j will be inserted at the beginning of the window as

$$\begin{cases} W_i(n+1) = W_i(n) & \text{for } n = 0,1,...,L-2 \\ W_i(0) = s_j \end{cases} \tag{8}$$

where L is the size of the window, and s_j is the new state. Note that at each time step, only one of the sliding windows is updated. Then the MLE of the transition probabilities at each time point k is calculated as

$$\hat{p}_{i,j}(k) = \frac{\sum_{l=0}^{L-1} \mathrm{I}_{W_i(l)}(s_j)}{L}, \text{ for all } i \text{ and } j, \tag{9}$$

where $\mathrm{I}_A(.)$ is the indicator function. From(9), the estimation of transient state probability, $\hat{p}_j(k)$, can be calculated.

Note that each transmitter node builds up sliding windows for its neighbors and updates them based on the ACK messages that it receives after each transmission. Then it can estimate the outage probability of the links to its neighbors as the probability of those links going to "bad" state, which is defined when SIR being less than a redefined threshold. It must be noted that although we assume a non-stationary representation for the channel, the channel variations are assumed to be slow enough (compared to the probability updates), such that the MLE probabilities converge to the transient state probabilities, $p_j(k)$.

1) Time-Varying Outage Probability

In the rest of this paper, let the communication link be modeled as a 2-state Markov model, where the state at time step *k* is defined as $X_k = \begin{cases} s_0 & \text{if } r(k) > \gamma_{th} \\ s_1 & \text{if } r(k) \le \gamma_{th} \end{cases}$, where *r*(*k*) is the SIR of the channel/link at time point *k*. Let the time-varying MLE transition probabilities be calculated using (9).

Definition 4. For the 2-state Markov model, the outage probability at time step *k* is the probability that the SIR of the channel/link be less that a threshold value, i.e. the channel/link at state s_1. The time-varying outage probability at time step *k* is defined as

$$P_{out}(k) = \Pr\left(r(k) < \gamma_{th}\right) = P\left(X_k = s_1\right) = \hat{p}_1(k) \tag{10}$$

2) Effective Available Bandwidth

Let the available bandwidth of a link, measured by listening to the ongoing traffic be BW^a. In the presence of fading channels, interferers, and other environmental uncertainties, this definition of available bandwidth would not be realistic due to packet losses that cannot be accurately measured. Therefore, the available bandwidth that takes into account the uncertainties is defined.

Definition 5. Let the outage probability of the link at time point k be $P_{out}(k)$. Then the effective available bandwidth at time step k, $EBW^a(k)$ is defined as

$$EBW^a(k) = \left(1 - P_{out}(k)\right).BW^a \tag{11}$$

C. Link Cost

The link cost between the neighbors is updated frequently through exchanging HELLO messages. By modifying the metric in [14], we introduced a new link cost metric that takes into account the link delay, available bandwidth, available energy, and queue-occupancy of the receiver node of the link

$$LC_{k,k+1} = \frac{d_{k,k+1}^{C_i}}{PTT} + \frac{BW_k^d}{BW_{k+1}^{a,C_i} - BW_k^d} + \frac{E_{k+1}^{init}}{E_{k+1}^{a}} + f\left(\frac{q_{k+1}}{q_{k+1}^{\max}}\right), \tag{12}$$

where $LC_{k,k+1}^{C_i}$ is the link cost between the node at hop k and the node at hop k+1, $d_{k,k+1}^{C_i}$ is the link delay between the two nodes when channel C_i is used, PTT is packet transmission time ($PTT = PacketSize/DataRate$), BW_k^d is the desired bandwidth at the node at hop k, BW_{k+1}^{a,C_i} is the available bandwidth on channel C_i at the node at hop k+1.

E_{k+1}^{init} and E_{k+1}^{a} are the initial energy and available energy at the node at hop k+1, respectively. q_{k+1} and $q_{k+1}^{\max}$ are queue length and queue limit at the node at hop k+1, respectively. $f(.)$ is a convex function of queue occupancy. It progressively increases as the queue occupancy ratio approaches 100%.

We modify this link cost such that the channel uncertainties caused by fading channels would be taken into account. In order to do so, the available bandwidth, BW_{k+1}^{a,C_i}, is replaced with the effective bandwidth, (11), which was earlier defined in the previous section.

D. Dynamic Routing

In the following, system specifications and the approximate dynamic programming method for multi-channel multi-interface routing are presented. During the initial phase, the base station sends a beacon throughout the network. As the beacon propagates in the network, nodes are initialized by their route cost to the base station. Also each node identifies its one-hop neighbors towards the base station.

Each node establishes a neighbors table in which it lists all the neighbors, whether they are towards the base station, their initial route cost to base station, $C(k)$, and several other entries such as the channels being used, link delay, available bandwidth, queue occupancy and available energy. In order to sustain the neighbors list, nodes periodically send HELLO messages to indicate their presence to neighbors (similar to AODV). Once a node has packets to transmit to the base station, the next node is selected based on the estimation of its route cost to the base station and its link cost to the current node.

The routing process is observed as a staged process. Starting at the source node, the packet is at hop k = 0. Depending on the topology and channel conditions, there can

be several candidate neighbor nodes to be chosen as the next node in the route to the base station. Fig. 3 illustrates this view of the network. It must be noted that the nodes vertically aligned at each hop represent the possible list of the 'next hop nodes and channels' from the point of view of the node in the previous hop – not the precise geographical location of the nodes in the network. In fact, if there are *N* channels available for the communication between node *i* and *m* (located at hops *k* and *k*+1), there will be *N* virtual copies of node *m* representing a possible next node for the packets at node *i* – each one representing the link between the nodes using one of the *N* channels.

At each hop, the route cost to the base station, $J(k)$, is the cumulative sum of the costs of the links on the route, which can be written as

$$J(k) = LC_{k,k+1} + \sum_{i=k+1}^{\infty} LC_{i,i+1} = LC_{k,k+1} + J(k+1) \tag{13}$$

where $LC_{k,k+1}$ is the link cost between the node at hop k and the node at hop k+1. In order to find the optimum route, the objective is to minimize (13) at each hop. In other words at each hop the minimum cost is

$$J^*(k) = \min_{\substack{n_{k+1} \in N_k^1, \\ C_{k,k+1}}} \left\{ LC_{k,k+1} + J^*(k+1) \right\}. \tag{14}$$

where N_k^1 is the list of one-hop neighbors at hop *k*., and $c_{k,k+1}$ is the channel of the link between hop *k* and *k*+1.

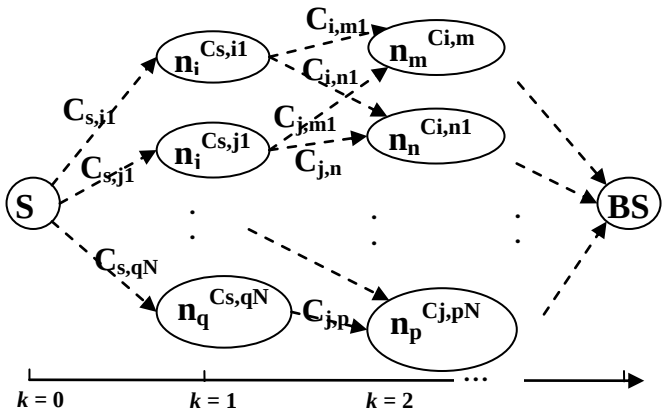

Fig. 3. Wireless network from routing point of view.

Since the most up-to-date optimum route cost to BS (Base Station) is not always available, instead of $J^*(k+1)$ in (14), an estimate of this route cost is utilized [14]. The up-to-date value of the link cost is available through frequent updates of the neighbors list. Approximate dynamic programming (ADP) techniques are utilized to find dynamic routes, while solving discrete-time Hamilton-Jacobi-Bellman (HJB) equation forward-in-time for route cost. The estimation of cost to BS is approximated by a NN as a function of state vector of the nodes.

The dynamic routes in [14] were found based on the assumption that there was no fading channel or shadowing effect in the environment. In that case, measuring the available bandwidth by just listening to the ongoing traffic could be an acceptable method. However, in the presence of fading channel and other uncertainties, the available bandwidth measured by the method above cannot be realistic. Due to uncertainties of the channel, each communication link has an outage probability [24]. We define an "effective available bandwidth" that takes into account the time-varying outage probability of the

link. Using this new definition of the available bandwidth, the link cost calculation and the selection of the next node along the route is performed by using the outage probability of the link.

In case of insufficient bandwidth on a certain link, other channels and radio interfaces between the nodes are utilized, and load balancing among those links and paths is achieved by using optimization techniques and constraints that are aware of the channel uncertainties. The estimation of cost to BS is approximated by a neural network (NN) as a function of state vector of the node at each hop. State vector is defined as $z(k) = [z_1(k) \quad z_2(k) \quad z_3(k)]^T$, where $z_1(k)$, $z_2(k)$ and $z_3(k)$ are the terms associated with queue occupancy, effective available bandwidth, and the initial cost to BS, $C(k)$, respectively.

At each hop, the next nodes are sorted according to their initial cost to Base Station, $C(k)$ .Then in general, if at hop *k*, there are *M* possible next nodes, state equation will be

$$x(k+1) = x(k) + Bu(k), \tag{15}$$

where $B = [0 - x(k) \quad 1 - x(k) \quad \cdots \quad (M-1) - x(k)]$, and

$u(k) = [1 \quad 0 \quad \cdots \quad 0]^T$, or $[0 \quad 1 \quad \cdots \quad 0]^T$, ... or $[0 \quad \cdots \quad 0 \quad 1]^T$, $x(k+1) = 0,1,\ldots,M-1$ and *u(k)* is a vector of length *M*. The relationship between the state vector, $z(k+1)$, and state $u(k)$ is

$$z(k+1) = \underline{\mathbf{Z}}(k+1).u(k) \tag{16}$$

where the columns of $\underline{\mathbf{Z}}(k+1)$ are the state vectors of the *M* possible next hops, i.e. $\underline{\mathbf{Z}}(k+1) = [z_0(k+1), z_1(k+1), \ldots, z_{M-1}(k+1)]$, where $u(k) = [1 \quad 0 \quad \cdots \quad 0]^T$, or

$[0 \quad 1 \quad \cdots \quad 0]^T$, ... or $[0 \quad \cdots \quad 0 \quad 1]^T$ and $z_i(k+1)$ is the state vector of the *i*th node at (*k*+1)th hop. It must be noted that (8) also bonds the state vector, $z(k+1)$, to the scalar value of the state, $x(k+1)$; state vector $z(k+1)$ depends on the node that is selected to be (*k*+1)th hop, in other words it depends on $x(k+1)$. Note that $x(k+1)$ is the node index from hop *k*'s point of view. Here the hop index '*k*' can be viewed as the time index provided the sample interval is one.

We are trying to minimize the infinite horizon cost function, (13), where

$$LC_{k,k+1} = u^T(k)R(k)u(k). \tag{17}$$

Assume $R(k)$ is an $M \times M$ diagonal matrix whose diagonal elements are the updated link costs from the current node to each of the next nodes.

The optimal decision for (7) that also minimizes (17) is found by [36]

$$u^*(k) = -\frac{1}{2}R^{-1}(k)B^T \frac{\partial J^*(k+1)}{\partial x(k+1)}, \tag{18}$$

where $u^*(k)$ and $J^*(k)$ are the optimal decision and optimal cost function, respectively.

The cost function is a smooth function. Using the approximation property of OLAs [38], the cost function can be represented by an OLA:

$$J(k) = W_J^T \sigma(z(k)) + \varepsilon_J(k) \tag{19}$$

where W_J is the constant target OLA parameter, $\varepsilon_J(k)$ is the bounded approximation error, and $\sigma(\cdot)$ is the vector activation function for the cost approximation OLA scheme.

The approximation holds for all z in a compact set $\boldsymbol{S}$. The approximation error is assumed to be bounded [38] as $\|\varepsilon_J(k)\| \le \varepsilon_{JM}$,where ε_{JM} is a known bound dependent on $\boldsymbol{S}$.

Approximating the cost function by an OLA renders

$$\hat{J}(k) = \hat{W}_J^T(k)\sigma(z(k)) = \hat{W}_J^T(k)\sigma(k) \ , \tag{20}$$

where $\hat{J}(k)$ is the approximation of the cost function, and $\hat{W}_J(k)$ is the actual parameter vector for the target OLA parameter vector, W_J . Vector $\sigma(k)$ is a set of bounded activation functions whose elements are chosen to be linearly independent basis functions. Defining the error in the route cost function as in [20]

$$e_J(k) = LC_{k-1,k} + \hat{W}_J^T(k)\sigma(k) - \hat{W}_J^T(k)\sigma(k-1) \ . \tag{21}$$

Then the auxiliary route cost error is defined as

$$E_J^T(k+1) = Y^T(k) + X^T(k)\hat{W}_J(k+1), \tag{22}$$

where $Y(k) = [LC_{k,k+1} \quad LC_{k-1,k} \cdots LC_{k-j,k-j+1}]$ and $X(k) = [\Delta\sigma(k+1) \quad \Delta\sigma(k) \cdots \Delta\sigma(k-j+1)]$ with $\Delta\sigma(k+1) = \sigma(k+1) - \sigma(k)$, $0 < j < k-1 \in \mathbb{N}$. Following the steps in [14], the cost function parameter update is defined as

$$\hat{W}_J(k+1) = X(k)\left(X^T(k)X(k)\right)^{-1}\left(\alpha_J E_J^T(k) - Y^T(k)\right), \tag{23}$$

where $0 < \alpha_J < 1$. Replacing (17) in (16) results in

$$E_J(k+1) = \alpha_J E_J(k) \tag{24}$$

Definition 6. A set of functions $\sigma(z) = \{\sigma_i(z)\}_1^L$ is linearly independent when $\sum_{i=1}^{L} \lambda_i \sigma_i(z) = 0$ holds only if $\lambda_1 = \lambda_2 = \cdots = \lambda_L = 0$.

Remark 1. Without loss of generality, we assume that the cost function parameter vectors are updated at the nodes in discrete-time intervals $\{0,1,2,...,k,...\}$. Similarly, packets are updated at each node which is referred to as *hop* along the routing path as the packet is moving toward the destination. Therefore, index *k* represents both discrete-time interval and hop along the route toward the destination.

Remark 2. In two consecutive hops of the route, *k* and *k*+1, it is impossible to have $z(k) = z(k+1)$. The two state vectors, $z(k)$ and $z(k+1)$, are different in at least one element, $z_i(k)$, and $z_i(k+1)$ – the one associated with the initial route cost to BS. By definition, the route cost to BS is initialized by propagating a beacon from BS. This cost, $C(k)$, cannot be the same in two consecutive hops.

Lemma 1. Let $z(k)$ and $z(k+1)$ be the state vectors in hops *k* and *k*+1. Also let $\sigma(z(k)) = \{\sigma_i(z(k))\}_1^L$ be a set of linearly independent functions. Then the set $\Delta\sigma(z(k+1)) = \{\sigma_i(z(k+1)) - \sigma_i(z(k))\}_1^L$ is also linearly independent.

Proof. Suppose that *Lemma 1* is not true. Then there exists some set $\{\lambda_i\}_1^L$ with some nonzero λ_is such that $\sum_{i=1}^{L} \lambda_i \Delta\sigma_i(z(k+1)) = 0$.

It can be rewritten as

$$\sum_{i=1}^{L} \lambda_i \left(\sigma_i(z(k+1)) - \sigma_i(z(k))\right) = \sum_{i=1}^{L} \lambda_i \sigma_i(k+1) - \sum_{i=1}^{L} \lambda_i \sigma_i(k) = 0 \qquad (25)$$

Equality (20) may hold in two cases.

Case I. $\sum_{i=1}^{L}\lambda_i\sigma_i(k+1)=\sum_{i=1}^{L}\lambda_i\sigma_i(k)\neq 0$. This case is rejected by *Remark 2*.

Case II. $\sum_{i=1}^{L}\lambda_i\sigma_i(k+1)=\sum_{i=1}^{L}\lambda_i\sigma_i(k)=0$. This case contradicts the hypothesis of linear independency of $\sigma(k)=\{\sigma_i(k)\}_1^L$ (and also that of $\sigma(k+1)=\{\sigma_i(k+1)\}_1^L$).

Therefore set $\Delta\sigma(z(k+1))=\{\sigma_i(z(k+1))-\sigma_i(z(k))\}_1^L$ is linearly independent. ■

Remark 3. *Lemma 1* implies that matrix $X^T(k)X(k)$ in (17) is invertible if $z(k)\neq 0$. Recalling the definition of the cost function (11) and OLA approximation (13), it can be noted that when $z(k)=0$, both definitions of the cost become zero. State vector $z(k)$ is zero only at BS, because only at BS all the three elements of $z(k)$ (queue length, link delay, and initial cost to BS) are zero. In other words, once the system states converge to zero (upon arrival at BS) the cost function approximation can no longer be updated. This can be viewed as a persistency of excitation (PE) requirement for the inputs to the cost function OLA. That is, the system states must be persistently exiting long enough for the OLA to learn the optimal cost function.

Remark 4. It is important to use bounded activation functions in the OLA, *e.g.* radial basis functions or saturated polynomial basis function such as $\sigma(z)=S(\theta(z))=\left(1-e^{-\theta(z)}\right)/\left(1+e^{-\theta(z)}\right)$. This guarantees $\|\sigma(k+1)\|\leq\sigma_M$, which will be used in the future proofs.

Let the OLA parameter estimation error be defined as

$$\tilde{W}_J(k)=W_J-\hat{W}_J(k). \tag{26}$$

Recalling the ideal OLA cost function (11), the general definition of the cost function can be rewritten as

$$W_J^T \sigma(k) + \varepsilon_J(k) = LC_{k,k+1} + W_J^T \sigma(k+1) + \varepsilon_J(k+1). \tag{27}$$

Rearranging the terms, the link cost can be derived as

$$LC_{k,k+1} = W_J^T \sigma(k) - W_J^T \sigma(k+1) + \varepsilon_J(k) - \varepsilon_J(k+1). \tag{28}$$

Recalling that $\Delta\sigma(k+1) = \sigma(k+1) - \sigma(k)$ and defining

$$\Delta\varepsilon_J(k) = \varepsilon_J(k+1) - \varepsilon_J(k), \tag{29}$$

equation (23) can be rewritten as

$$LC_{k,k+1} = -W_J^T \Delta\sigma(k) - \Delta\varepsilon_J(k). \tag{30}$$

Substituting $LC_{k-1,k}$ and $LC_{k,k+1}$ in (14) and (16), respectively, it yields

$$\begin{aligned} e_J(k) &= -W_J^T \Delta\sigma(k) - \Delta\varepsilon_J(k-1) + \hat{W}_J^T(k)\left(\sigma(k) - \sigma(k-1)\right) \\ &= -\tilde{W}_J^T(k)\Delta\sigma(k) - \Delta\varepsilon_J(k-1) \end{aligned}, \tag{31}$$

and $$e_J(k+1) = -\tilde{W}_J^T(k+1)\Delta\sigma(k+1) - \Delta\varepsilon_J(k). \tag{32}$$

Recalling $e_J(k+1) = \alpha_J e_J(k)$ from (16), (26) is rewritten as

$$\tilde{W}_J(k+1) = \left(\Delta\sigma(k+1)\left(\Delta\sigma^T(k+1)\Delta\sigma(k+1)\right)^{-1}\right) \cdot \left(\alpha_J \Delta\sigma^T(k)\tilde{W}_J(k) + \alpha_J\Delta\varepsilon_J(k-1) - \Delta\varepsilon_J(k)\right). \tag{33}$$

In the following, the boundedness of the cost function error (14), and the OLA parameter estimation error will be investigated.

Definition 7. [38] An equilibrium point z_e is said to be *uniformly ultimately bounded (UUB)* if there exists a compact set $S \subset \Re^{n^*}$ so that for all initial states $z_e \in S$ there exists a bound $B \geq 0$ and a time $T(B, z_e)$ such that $\left\| z(k) - z_e \right\| \leq B$ for all $k \geq k_0 + T$.

Theorem 1. *(Boundedness of the Cost OLA Errors).* Let $z(k)$ be any state vector in the network along the possible routes from the source node to BS, and let the cost OLA parameter be updated as in (17). Then, the cost errors (14) and (29) are *UUB*.

Proof. Consider the Lyapunov function candidate

$$V_J(k) = e_J^2(k) + \frac{1}{\beta} tr\left\{ \tilde{W}_J^T(k) \tilde{W}_J(k) \right\} \tag{34}$$

where $\beta > 0$.

Similar to *Theorem* 1 in Paper 2, it can be proved that $\Delta V_J(k)$ is less than zero if

$$\left| e_J(k) \right| > \sqrt{\frac{\left(\alpha_J + 1\right)^2 \varepsilon_{JM}^2 \Delta\sigma_M^2}{\beta\left(1 - \alpha_J^2\right)\Delta\sigma_{\min}^2}} \tag{35}$$

or

$$\left\| \tilde{W}_J(k) \right\|_F > \frac{\left(1 - \alpha_J\right) \varepsilon_{JM} \Delta\sigma_M}{\sqrt{\Delta\sigma_{\min}^2 - \alpha_J^2 \Delta\sigma_M^2}} . \tag{36}$$

where $\alpha_J^2 < \min\left\{1, \Delta\sigma_{\min}^2 / \Delta\sigma_M^2\right\}$ and $\beta > 0$.

According to standard Lyapunov theory [38], it can be concluded that $\Delta V_J(k)$ is less than zero outside the compact set given above. That is, the cost function error (14), and the OLA parameter estimation error (26) are *UUB*. ■

Next in the following corollary, it will be shown that the decision vector approaches the optimal value in an approximate manner.

Corollary 1. *(Boundedness of the decision vector).* Let $\hat{u}(k)$ be any decision vector for the selection of the next hop node in the network along the possible routes from the source node to BS, which is updated as

$$\hat{u}(k) = -\frac{1}{2} R^{-1}(k) B^T \frac{\partial \hat{J}(k+1)}{\partial x(k+1)} \quad . \tag{37}$$

Then there exists a bound $\varepsilon_r \geq 0$ such that $\left\| \hat{u}(k) - u^*(k) \right\| \leq \varepsilon_r$ as $k \to \infty$. In other words, a near optimal path results as the number of hops increases.

Proof. See *Corollary* 1 in Paper 2.

Remark 5. If the size of the approximator is increased (the number of neurons are increased), the approximation error becomes zero. This reduces the error bound on the optimal decision.

E. Load Balancing

Once the next node in the route is determined by the routing algorithm in section D, the effective available bandwidth is compared to the desired bandwidth/rate at the link which is defined by the received data packets that need to be relayed, or the packets that are generated by the current node. If the link is not able to provide the desired bandwidth, then the other channel links between the two nodes are examined. If they can cumulatively provide the desired bandwidth, the flow is directed to the next node through load balancing over the multiple channels between the two nodes. This is done by utilizing optimization techniques [25].

1) Load Balancing over Multiple Channels between a Pair of Nodes

Recall that in the link cost,(5), the term associated with bandwidth is inversely proportional to the effective bandwidth. Therefore, this term of the link cost for choosing the link between the nodes *i* and *j*, using channel *m* can be defined as

$$C_{i,j}^{m}(r_{i,j}^{m}) = \frac{r_{i,j}^{m}}{EBW_{i,j}^{m} - r_{i,j}^{m}} \tag{38}$$

where $r_{i,j}^{m}$ is the rate assigned to the link, and $EBW_{i,j}^{m}$ is the effective available bandwidth on channel *m* between nodes *i* and *j*. Also assume that there are *M* channels available between nodes *i* and *j*. Then the load balancing problem can be written as

$$\text{minimize} \quad C_{i,j}(r_{i,j}) = \sum_{m=1}^{M} C_{i,j}^{m}(r_{i,j}^{m}) \tag{39}$$

$$\text{subject to: } 0 \le r_{i,j}^{m} \le EBW_{i,j}^{m}, \text{ for } m = 1,2,...,M \tag{40}$$

$$\sum_{m=1}^{M} r_{i,j}^{m} = r_{i,j}^{d} \tag{41}$$

where $r_{i,j}^{d}$ is the desired rate between nodes *i* and *j*. Minimization of (39) implies that

$$\frac{\partial}{\partial r_{i,j}^{m}}\left(\sum_{m=1}^{M} C_{i,j}^{m}(r_{i,j}^{m})\right) = 0, \text{ for } m = 1,2,...,M \tag{42}$$

Hence the optimization problem can be written as

$$\sum_{m=1}^{M} \frac{\partial}{\partial r_{i,j}^{m}} C_{i,j}(r_{i,j}^{*})\left(r_{i,j}^{m} - r_{i,j}^{*m}\right) \ge 0 \ , \tag{43}$$

and $0 \le r_{i,j}^{m} \le EBW_{i,j}^{m}$, for $m = 1, 2, ..., M$, and $\sum_{m=1}^{M} r_{i,j}^{m} = r_{i,j}^{d}$, where $r_{i,j}^{*} = \left\{ r_{i,j}^{*m} \right\}$ is the optimal solution. It turns into [25]

$$r_{i,j}^{*} > 0 \text{ only if } \frac{\partial}{\partial r_{i,j}^{\sim m}} C_{i,j}(r_{i,j}^{*}) \ge \frac{\partial}{\partial r_{i,j}} C_{i,j}(r_{i,j}^{*}), \text{ for all } \sim m \in \{1, 2, ..., M\} - \{*m\}. \quad (44)$$

In other words, (44) implies that [25] in optimal solution, the first derivative of the cost should be equal on all the paths that are assigned a part of the flow $r_{i,j}^{d}$, the first derivatives of the cost of optimum paths with respect to their flow should be equal or smaller than the first derivative of the cost of the other paths with respect to their own flow.

From (38) it can be concluded that

$$\frac{\partial}{\partial r_{i,j}^{m}} C_{i,j}^{m}(r_{i,j}^{m}) = \frac{EBW_{i,j}^{m}}{\left(EBW_{i,j}^{m} - r_{i,j}^{m} \right)^{2}}. \quad (45)$$

Let flow $r_{i,j}^{d}$ be split among *m* out of *M* channels. Conditions (a) and (b) require that the first derivatives of the *m* channels must be equal, and they should also be smaller than or equal to the first derivatives of the remaining channels which is given by

$$\frac{EBW_{i,j}^{1}}{\left(EBW_{i,j}^{1} - r_{i,j}^{*1} \right)^{2}} = \frac{EBW_{i,j}^{2}}{\left(EBW_{i,j}^{2} - r_{i,j}^{*2} \right)^{2}} = \cdots = \frac{EBW_{i,j}^{m}}{\left(EBW_{i,j}^{m} - r_{i,j}^{*m} \right)^{2}} \quad (46)$$

and

$$\frac{EBW_{i,j}^{n}}{\left(EBW_{i,j}^{n} - r_{i,j}^{*n} \right)^{2}} \le \frac{1}{EBW_{i,j}^{l}}, \quad (47)$$

for $n=1,2,...,m$ and $l=m+1,m+2,...,M$, where $r_{i,j}^{*1}>0$, $r_{i,j}^{*2}>0$, …, $r_{i,j}^{*m}>0$ and $r_{i,j}^{*m+1}=r_{i,j}^{*m+2}=\cdots=r_{i,j}^{*M}=0$. $r_{i,j}^{*m}$ is the optimum rate assigned to the link, and $EBW_{i,j}^{m}$ is the effective available bandwidth on channel *m* between nodes *i* and *j*. The channels are sorted in descending order with respect to their effective available bandwidths, i.e. $EBW_{i,j}^{1}\geq EBW_{i,j}^{2}\geq\cdots\geq EBW_{i,j}^{M}$.

Therefore, the proof in [25] for multiple channels over a link provides that (46) and (47) are the requirements for the optimal load balancing. Now recall (11), where at each time point *k*, the effective available bandwidth, between nodes *i* and *j* using channel *m* is

$$EBW_{i,j}^{m}(k)=\left(1-P_{out_{i,j}}^{m}(k)\right).BW_{i,j}^{am}. \tag{48}$$

By substituting (48) in (46) and (47), it yields

$$\frac{\left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a1}}{\left(\left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a1}-r_{i,j}^{*1}\right)^{2}}=\frac{\left(1-P_{out_{i,j}}^{2}(k)\right).BW_{i,j}^{a2}}{\left(\left(1-P_{out_{i,j}}^{2}(k)\right).BW_{i,j}^{a2}-r_{i,j}^{*2}\right)^{2}}=\cdots=\frac{\left(1-P_{out_{i,j}}^{m}(k)\right).BW_{i,j}^{am}}{\left(\left(1-P_{out_{i,j}}^{m}(k)\right).BW_{i,j}^{am}-r_{i,j}^{*m}\right)^{2}} \tag{49}$$

and

$$\frac{\left(1-P_{out_{i,j}}^{n}(k)\right).BW_{i,j}^{an}}{\left(\left(1-P_{out_{i,j}}^{n}(k)\right).BW_{i,j}^{an}-r_{i,j}^{*n}\right)^{2}}\leq\frac{1}{\left(1-P_{out_{i,j}}^{l}(k)\right).BW_{i,j}^{al}} \tag{50}$$

for $n=1,2,...,m$ and $l=m+1,m+2,...,M$, where $r_{i,j}^{*1}>0$, $r_{i,j}^{*2}>0$, …, $r_{i,j}^{*m}>0$ and $r_{i,j}^{*m+1}=r_{i,j}^{*m+2}=\cdots=r_{i,j}^{*M}=0$.

Theorem 2. For multi-channel multi-hop networks, available bandwidth-based load balancing results in inefficient routing of flows among the links.

Proof. Proof is provided by a counterexample. Consider the load balancing problem for a two-channel/link case in which $BW_{i,j}^{a1} \le BW_{i,j}^{a2}$. Using Available bandwidth [26] in load balancing means that the outage probabilities are assumed to be zero, and the case is reduced to

Case I. $r_{i,j}^{*2} = r_{i,j}^{d}$ and $r_{i,j}^{*1} = 0$. From [25] and [26],

$$\frac{BW_{i,j}^{a2}}{\left(BW_{i,j}^{a2} - r_{i,j}^{*2}\right)^2} \le \frac{1}{BW_{i,j}^{a1}} \tag{51}$$

It is simplified as

$$r_{i,j}^{*2} \le BW_{i,j}^{a2} - \sqrt{BW_{i,j}^{a1}.BW_{i,j}^{a2}}\,, \tag{52}$$

Case II: $r_{i,j}^{*1} > 0$, $r_{i,j}^{*2} > 0$ and $r_{i,j}^{*1} + r_{i,j}^{*2} = r_{i,j}^{d}$. From [25] and [26],

$$\frac{BW_{i,j}^{a1}}{\left(BW_{i,j}^{a1} - r_{i,j}^{*1}\right)^2} = \frac{BW_{i,j}^{a2}}{\left(BW_{i,j}^{a2} - r_{i,j}^{*2}\right)^2}\,.$$

It can be simplified as

$$\frac{r_{i,j}^{*2}}{r_{i,j}^{*1}} = \sqrt{\frac{BW_{i,j}^{a2}}{BW_{i,j}^{a1}}}.\left(\frac{r_{i,j}^{d} - BW_{i,j}^{a1} + \sqrt{BW_{i,j}^{a1} \cdot BW_{i,j}^{a2}}}{r_{i,j}^{d} - BW_{i,j}^{a2} + \sqrt{BW_{i,j}^{a1} \cdot BW_{i,j}^{a2}}}\right) \ge 1 \tag{53}$$

Note that if only the available bandwidths are taken into account in load balancing, the link using Channel 2 is assigned a greater rate, i.e. $r_{i,j}^{*1} \le r_{i,j}^{*2}$. However, in the presence of fading channels, mobility, etc, it is quite possible that the link with the

greater bandwidth have a greater outage possibility. In this case, assigning a greater flow rate to the link using Channel 2 only increases the dropped packet rate on that link. ■

Theorem 3. For multi-channel multi-hop networks, effective bandwidth-based load balancing yields an efficient distribution of flow among the links.

Proof. Consider the same two channel/link case in *Theorem 1*. Assume that $BW_{i,j}^{a1} \le BW_{i,j}^{a2}$, while the fading channels and interference causes Channel 2 to have a greater outage probability and consequently a smaller Effective Bandwidth, $EBW_{i,j}^{a1} \ge EBW_{i,j}^{a2}$. In other words, $\left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a1} \ge \left(1-P_{out_{i,j}}^{2}(k)\right).BW_{i,j}^{a2}$. Now the load-balancing problem according to Effective bandwidth can be solved as following.

Case I. $r_{i,j}^{*1} = r_{i,j}^{d}$ and $r_{i,j}^{*2} = 0$. Recall (50) where

$$\frac{\left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a1}}{\left(\left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a1} - r_{i,j}^{*1}\right)^2} \le \frac{1}{\left(1-P_{out_{i,j}}^{2}(k)\right).BW_{i,j}^{a2}} \tag{54}$$

It is simplified as

$$\left(1-P_{out_{i,j}}^{2}(k)\right).BW_{i,j}^{a1}.\left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a2} \le \left(\left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a1} - r_{i,j}^{*1}\right)^2, \tag{55}$$

and then

$$r_{i,j}^{*1} \le \beta_1 - \sqrt{\beta_1 \cdot \beta_2}, \tag{56}$$

where $\beta_1 = \left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a1}$, $\beta_2 = \left(1-P_{out_{i,j}}^{2}(k)\right).BW_{i,j}^{a2}$ and $\beta_1 \ge \beta_2$.

Case II: $r_{i,j}^{*1} > 0$, $r_{i,j}^{*2} > 0$ and $r_{i,j}^{*1} + r_{i,j}^{*2} = r_{i,j}^{d}$. Recall (49), where

$$\frac{\left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a1}}{\left(\left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a1}-r_{i,j}^{*1}\right)^{2}} = \frac{\left(1-P_{out_{i,j}}^{2}(k)\right).BW_{i,j}^{a2}}{\left(\left(1-P_{out_{i,j}}^{2}(k)\right).BW_{i,j}^{a2}-r_{i,j}^{*2}\right)^{2}}.$$

It can be simplified as

$$\sqrt{\beta_2}\left(\beta_1 - r_{i,j}^{*1}\right) = \sqrt{\beta_1}\left(\beta_2 - r_{i,j}^{*2}\right) \tag{57}$$

where $\beta_1 = \left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a1}$, $\beta_2 = \left(1-P_{out_{i,j}}^{2}(k)\right).BW_{i,j}^{a2}$ and $\beta_1 \geq \beta_2$. Then (57) can be rewritten as

$$r_{i,j}^{*1}\sqrt{\beta_2} - \beta_1\sqrt{\beta_2} = r_{i,j}^{*2}\sqrt{\beta_1} - \beta_2\sqrt{\beta_1}. \tag{58}$$

Recalling $r_{i,j}^{*1} + r_{i,j}^{*2} = r_{i,j}^{d}$, (58) is simplified as

$$\frac{r_{i,j}^{*1}}{r_{i,j}^{*2}} = \sqrt{\frac{\beta_1}{\beta_2}}.\left(\frac{r_{i,j}^{d} - \beta_2 + \sqrt{\beta_1\beta_2}}{r_{i,j}^{d} - \beta_1 + \sqrt{\beta_1\beta_2}}\right) \geq 1, \tag{59}$$

where $\beta_1 = \left(1-P_{out_{i,j}}^{1}(k)\right).BW_{i,j}^{a1}$, $\beta_2 = \left(1-P_{out_{i,j}}^{2}(k)\right).BW_{i,j}^{a2}$ and $\beta_1 \geq \beta_2$.

It can be seen that our proposed load-balancing algorithm assigns a higher data rate to Channel 1, whose effective bandwidth is greater. In other words, unlike the Available bandwidth method, it avoids allocating a higher data rate to a link which is more likely to have an outage. ■

Remark 6. For the cases with more than two channels/links, the proof can be done by induction.

Remark 7. It is also noteworthy if the outage probabilities are zero, the solution in (49) and (50) would be the same as the solution in [25].

2) Load Balancing over Multiple Paths with Common Source Node

Recall that the link cost for choosing the link between nodes *i* and *j*, can be defined as (38) , $C_{i,j}^{m}(r_{i,j}^{m}) = \frac{r_{i,j}^{m}}{EBW_{i,j}^{m} - r_{i,j}^{m}}$, where $r_{i,j}^{m}$ is the rate assigned to the link, and $EBW_{i,j}^{m}$ is the effective available bandwidth on channel *m* between nodes *i* and *j*. Also assume that channels $p \in P$ ($P \subseteq \{1,2,...,M\}$) are used for communication between nodes *i* and *j*. In addition, assume a similar multiple channel communication link between nodes *i* and *l*, with the link cost for channel *n* as $C_{i,l}^{n}(r_{i,l}^{n}) = \frac{r_{i,l}^{n}}{EBW_{i,l}^{n} - r_{i,l}^{n}}$, where channels $q \in Q$ ($Q \subseteq \{1,2,...,M\}$) are used for communication between nodes *i* and *l*.

Then the load balancing problem over the two paths (*i,j*) and (*i,l*) the can be written as

$$\text{minimize} \quad C_{i,j}(r_{i,j}) + C_{i,l}(r_{i,l}) = \sum_{p \in P} C_{i,j}^{p}(r_{i,j}^{p}) + \sum_{q \in Q} C_{i,l}^{q}(r_{i,l}^{q}) \tag{60}$$

$$\text{subject to: } 0 \le r_{i,j}^{p} \le EBW_{i,j}^{p}, \text{ for } p \in P, \quad P \subseteq \{1,2,...,M\}$$

$$0 \le r_{i,l}^{q} \le EBW_{i,l}^{q}, \text{ for } q \in Q, \quad Q \subseteq \{1,2,...,M\} \tag{61}$$

$$\sum_{p \in P} r_{i,j}^{p} + \sum_{q \in Q} r_{i,l}^{q} = r_{i}^{d} \quad , \tag{62}$$

where r_{i}^{d} is the desired flow rate at node *i*.

Sort all the communications links according to their effective available bandwidth, e.g.

$$\underbrace{EBW_{i,j}^{p_1} \geq EBW_{i,l}^{q_1} \geq \cdots \geq EBW_{i,j}^{p_{\mu(P)}}}_{\mu(P)+\mu(Q)} \tag{63}$$

where $\mu(P)$ and $\mu(Q)$ are the size of set P and set Q, respectively. Notice that the intersection of P and Q can be non-empty. In order to avoid the interference between two neighbor links that are using the same channel, the total set of available channels over links (i,j) and (i,l) should be reduced as presented in Table I.

Now the optimization problem (60) is reduced to

$$\text{minimize:} \quad C_{i,j}(r_{i,j}) + C_{i,l}(r_{i,l}) = \sum_{p \in P_R} C_{i,j}^{p}(r_{i,j}^{p}) + \sum_{q \in Q_R} C_{i,l}^{q}(r_{i,l}^{q}) \tag{64}$$

$$\text{subject to: } 0 \leq r_{i,j}^{p} \leq EBW_{i,j}^{p}, \text{ for } p \in P_R$$

$$0 \leq r_{i,l}^{q} \leq EBW_{i,l}^{q}, \text{ for } q \in Q_R \tag{65}$$

$$\sum_{p \in P_R} r_{i,j}^{p} + \sum_{q \in Q_R} r_{i,l}^{q} = r_i^{d} \quad , \tag{66}$$

where r_i^d is the desired flow rate at node i. Solutions in (49) and (50) still apply, however, instead of M channels between nodes i and j, it is over $\mu(P_R) + \mu(Q_R)$ channels and over two paths (i,j) and (i,l). Note that the solutions above can be extended to multiple links. The reduction algorithm must be run for the channel sets of all the paths.

TABLE I. PSEUDOCODE FOR REDUCING THE CHANNEL SETS TO AVOID CHANNEL INTERFERENCE.

REDUCTION(P,Q)

1 sort elements of P such that
$EBW_{i,j}^{p_1} \geq EBW_{i,j}^{p_2} \geq \cdots \geq EBW_{i,j}^{p_{\mu(P)}}$
2 sort elements of Q such that
$EBW_{i,l}^{q_1} \geq EBW_{i,l}^{q_2} \geq \cdots \geq EBW_{i,l}^{q_{\mu(Q)}}$
3 $P_R \leftarrow \varnothing$ and $Q_R \leftarrow \varnothing$
4 **for** each $p_n \in P$
5 **for** each $q_m \in Q$
6 **if** $EBW_{i,j}^{p_n} \geq EBW_{i,l}^{q_l}$
7 **then** $P_R \leftarrow P_R \cup p_n$
8 **if** $p_n \in Q$
9 **then** $Q \leftarrow \{q \in Q \mid q \neq p_n\}$
10 **else**
11 $Q_R \leftarrow Q_R \cup q_m$
12 **if** $q_m \in P$
13 **then** $P \leftarrow \{p \in P \mid p \neq q_m\}$
14 **return** P_R and Q_R

F. Analysis of the Cost of Multi-path Route

When the flow is divided among multiple paths, the cost of routing the flow is no longer the cost of a single path. In order to find the *equivalent single-path cost for flow* f_m for multiple paths, the route cost should be taken into account along all the paths the flow is traversing. The following equation for the routing cost of flow f_m at hop k, $J_{fm}(k)$, should be satisfied

$$J_{f_m}(k) = \sum_{p \in P} J_p(k) = \sum_{p \in P} \left(LC_{k,k+1;p} + J_p(k+1) \right) \tag{67}$$

$J_p(k)$ is the route cost on path p from hop k to BS. Set P is the set of the paths that flow f_m

is distributed among at hop k. $LC_{k,k+1;p}$ is the link cost between hop k and hop k+1 on path p. $J_p(k+1)$ is the route cost from hop k+1 on path p to BS. Subsequently, the routing optimization problem would be

$$J^*_{f_m}(k) = \min_{\substack{n_{k+1}\in N_k^1,\\ c_{k,k+1}}} \left\{ \sum_{p\in P} \left(LC_{k,k+1;p} + J^*_p(k+1) \right) \right\}, \tag{68}$$

where N_k^1 is the list of one-hop neighbors at hop k, and $c_{k,k+1}$ is the channel of the link between hop k and k+1. It must be noted that minimization of the route cost of the individual paths is taken care of in (14). In other words,

$$J^*_{f_m}(k) = \min_{\substack{p\in P,\\ n_{k+1}\in N_k^1,\\ c_{k,k+1}}} \left\{ \sum_{p\in P} \left(J^*_p(k) \right) \right\} \tag{69}$$

where $J^*_p(k)$ is the route cost from hop k to BS on path p. Note that if the available bandwidth on one path/link is sufficient, the route will be single-path, and consequently $J^*_{f_m}(k) = J^*_p(k)$. However, if the available bandwidth on one path/link is not sufficient, additional links are selected and the flow is distributed among them based on the load balancing algorithm in Section *E*. The steps are presented in Fig.4.

In the following, the multi-path routing which exploits the forward-in-time adaptive dynamic routing (Section *D*) and load balancing (Section *E*), is proven to provide the near optimal routes.

Theorem 4. Let $\hat{J}_{f_m}(k)$ be the estimated cost of the multi-path route for flow f_m from hop k to BS. Then a bound ξ_J exists such that

$$\left|\hat{J}_{f_m}(k) - J^*_{f_m}(k)\right| \le \xi_J \text{, for} \quad k \ge k_0 \tag{70}$$

Proof. Recall that at each hop k, the multi-path route cost is calculated by

$$\hat{J}_{f_m}(k) = \sum_{p \in P}\left(\hat{J}_p(k)\right)$$

where $\hat{J}_p(k)$ is the estimated route cost along the individual path p. It is estimated by the forward-in-time dynamic routing in Section *D*, (20)

$$\hat{J}_p(k) = \hat{W}^T_{J;p}(k)\sigma(z_p(k)) = \hat{W}^T_{J;p}(k)\sigma_p(k) \tag{71}$$

where $\hat{W}_{J;p}(k)$ is the cost OLA parameter along path p (which is updated using (23)). In *Theorem* 1 it was proved that the OLA parameter estimation error is bounded ($\left\|W_J - \hat{W}_J(k)\right\|_F \le \frac{\left(1-\alpha_J\right)\ \varepsilon_{JM}\Delta\sigma_M}{\sqrt{\Delta\sigma^2_{\min} - \alpha_J^2\Delta\sigma^2_M}}$). Also in *Corollary* 1 it was shown that the decision vector results in near optimal routes as the number of hops increase ($\left\|\hat{u}(k) - u^*(k)\right\| \le \varepsilon_r$). Subsequently, one can conclude that

$$\left|\hat{J}_p(k) - J^*_p(k)\right| \le \varsigma_J \text{, for} \quad k \ge k_0 \tag{72}$$

On the other hand, recall that the multi-path route cost can be written as

$$\hat{J}_{f_m}(k) = \sum_{p \in P}\hat{J}_p(k) = \sum_{p \in P}\left(LC_{k,k+1;p} + \hat{J}_p(k+1)\right)$$

where $LC_{k,k+1;p}$ contains the term $\frac{r_{k,k+1}^m}{EBW_{k,k+1}^m - r_{k,k+1}^m}$. Recall that $r_{k,k+1}^m$ is the data rate assigned to the link between hop k and k+1 using channel m, and $EBW_{k,k+1}^m$ is the Effective available bandwidth on that link. Recall that $C_{k,k+1}^m(r_{k,k+1}^m) = \frac{r_{k,k+1}^m}{EBW_{k,k+1}^m - r_{k,k+1}^m}$ is the cost that was used in load balancing algorithm to assign the rates among several links such that

$$\text{minimize} \quad \sum_{\substack{p \in P, \\ k+1 \in p, \\ n_{k+1} \in N_k^1}} C_{k,k+1}^m(r_{k,k+1}^m) \tag{73}$$

subject to: $0 \le r_{k,k+1}^m \le EBW_{k,k+1}^m$, for $p \in P$

$$\sum_{\substack{p \in P, \\ k+1 \in p, \\ n_{k+1} \in N_k^1}} r_{k,k+1}^m = r_k^d \quad , \tag{74}$$

Combining (72) and (73), one can write

$$\left| \sum_{p \in P} \hat{J}_p(k) - \sum_{p \in P} J_p^*(k) \right| \le \xi_J, \text{ for } \quad k \ge k_0 \tag{75}$$

and subsequently

$$\left| \hat{J}_{f_m}(k) - J_{fm}^*(k) \right| \le \xi_J, \text{ for } \quad k \ge k_0 . \tag{76}$$ ■

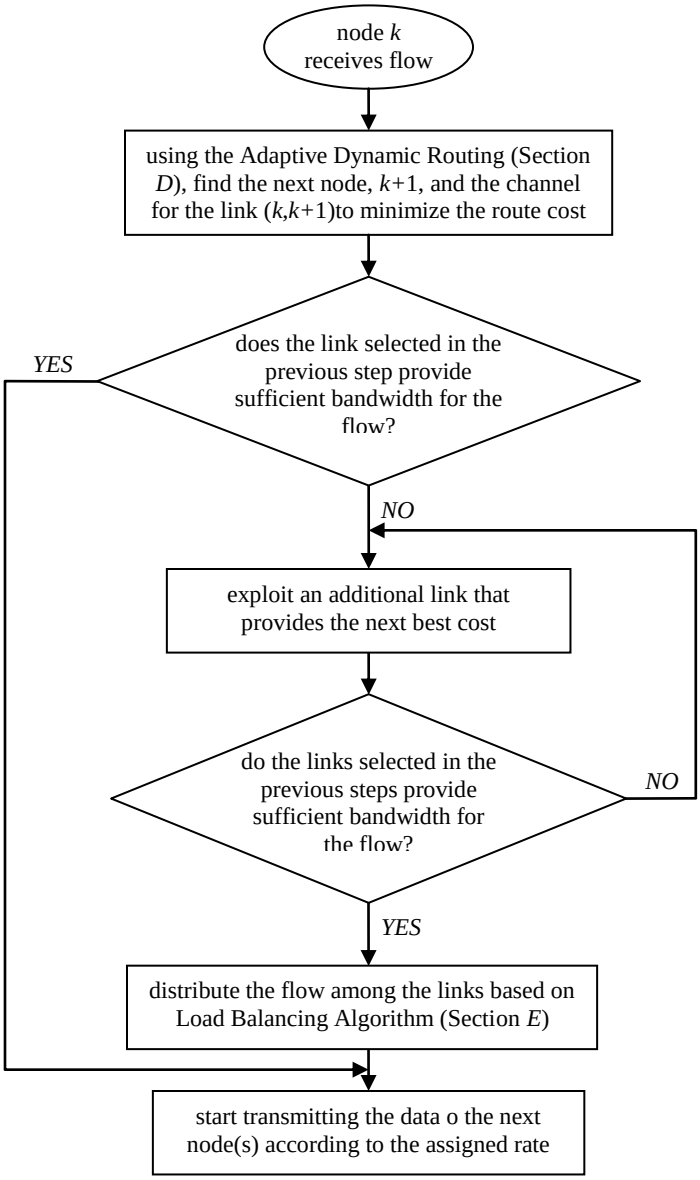

Fig. 4. Flowchart of multi-path routing.

G. Analysis of End-to-End Delay

As a packet enters the networks and traverses through it, until it arrives its destination node, it faces an end-to-end delay. The end-to-end delay is the sum of the delays on each communication link taken by the packet. The link delay is consisted of the

following components: processing delay (d_{proc}), queuing delay (d_q), transmission delay (d_{tx}), and propagation delay (d_{prop}). Subsequently, the link delay between nodes *i* and *j*, can be written as following

$$d_l(i, j) = d_{proc}(i) + d_q(i) + d_{tr}(i) + d_{prop}(i, j) \tag{77}$$

Processing delay at node *i*, $d_{proc}(i)$, is the delay between the time the packet is received at node *i*, and the time the packet is assigned to an outgoing link queue. It includes the time that is required by node *i*, to decide to which node the packet should be sent; i.e. the route discovery/recovery delay.

Queuing delay is the delay between the time that a packet is assigned to an outgoing queue for transmission, and the time it starts being transmitted. It can include the time that it takes for node *i* to detect a clear carrier/receiver. If the protocol utilizes RTS/CTS messages, the queuing delay includes the delay between the time node *i* sends the RTS message and the time it receives the CTS message from node *j*.

Transmission delay is the delay between the times that the first and last bits of the packet are transmitted. It depends on the packet size and data rate of the transmitter node.

Propagation delay is the delay between the time that the last bit of the packet is transmitted at the transmitter node (node *i*), and the time that it is received at the receiver node (node *j*). It must be noted that the propagation delay depends on the physical characteristics of the link, and is independent of the traffic carried by the link.

Subsequently, the end-to-end delay of a pack that traverses a path *P*, can be written as the sum of the link delays along path *P*.

$$\begin{aligned} d_{EndToEnd} &= \sum_{\substack{l \in P, \\ i,j \in l}} d_l(i,j) \\ &= \sum_{\substack{l \in P, \\ i,j \in l}} \left(d_{proc}(i) + d_q(i) + d_{tx}(i) + d_{prop}(i,j) \right) \end{aligned} \tag{78}$$

Let find each of the delay components assuming that the adaptive dynamic routing (proposed in Section II.D) is utilized.

Processing Delay. Once a packet is received at a node, the online estimator parameters, $\hat{W}_J(k)$, should be updated so that the remaining route cost can be estimated and the node with the lowest cost be selected as the next node. Recalling (14) and (17), the computation cost of updating $\hat{W}_J(k)$ can be written as

$$C_{\hat{W}_J} = C_1 + C_2 + C_3 + C_4 + C_5 + C_6, \tag{79}$$

where C_1 is the computation cost of calculating $E_J(k)$ (in (14)). It is derived as

$$C_1 = L_{Mem}\left(1 + 3L_{\hat{W}}\right) \text{ instructions}, \tag{80}$$

where L_{Mem} is the length of the memory window that is used in parameter estimation (i.e. the size of vector $E_J(k)$), and $L_{\hat{W}}$ is the size of the parameter vector $\hat{W}_J(k)$.

C_2 is the computation cost of calculating $\alpha_J E_J^T(k) - Y^T(k)$ (in (17)) – given that the vectors $E_J(k)$ and $Y_J(k)$ are provided. It is derived as

$$C_2 = 2L_{Mem} \text{ instructions}. \tag{81}$$

C_3 is the computation cost of calculating $X^T(k)X(k)$ (in (17)). It is derived as

$$C_3 = \left(L_{Mem}\right)^2 \cdot \left(2L_{\hat{W}}\right) \text{ instructions.} \tag{82}$$

C_4 is the computation cost of calculating the inverse of $X^T(k)X(k)$ (in (17)). It depends on the algorithm that is used for matrix inversion. Assuming that DCM (Divide-and Conquer Matrix inversion) algorithm [27] be used for matrix inversion, the cost is derived as

$$C_4 = \frac{1}{2}\left(L_{Mem}\right)^3 + \frac{1}{2}\left(L_{Mem}\right)^2 \text{ instructions.} \tag{83}$$

C_5 is the computation cost of multiplying $\left(X^T(k)X(k)\right)^{-1}$ and $\left(\alpha_J E_J^T(k) - Y^T(k)\right)$ (in (17)). It is derived as

$$C_5 = L_{Mem}\left(2L_{Mem}\right) \text{ instructions.} \tag{84}$$

Finally, C_6 is the computation cost of multiplying $X(k)$ and $\left(X^T(k)X(k)\right)^{-1}\left(\alpha_J E_J^T(k) - Y^T(k)\right)$. It is calculated as

$$C_6 = L_W\left(2L_{Mem}\right) \text{ instructions.} \tag{85}$$

Therefore, cost of updating the estimation parameter vector, $\hat{W}_J(k)$, is calculated as

$$C_{\hat{W}_J} = \frac{1}{2}\left(L_{Mem}\right)^3 + \left(\frac{5}{2} + 2L_{\hat{W}}\right)\left(L_{Mem}\right)^2 + \left(3 + 2L_{\hat{W}}\right)L_{Mem} \text{ instructions.} \tag{86}$$

Subsequently, the estimation of the remaining route cost can be calculated as in (13), with computation cost of

$$C_{\hat{j}} = \frac{1}{2}\left(L_{Mem}\right)^3 + \left(\frac{5}{2} + 2L_{\hat{W}}\right)\left(L_{Mem}\right)^2 + \left(3 + 2L_{\hat{W}}\right)L_{Mem} + 2L_{\hat{W}} \quad instructions. \qquad (87)$$

Assuming that the processor the node has a speed of S_{proc} MIPS (Million Instructions Per Second), the processing delay of the packet at node *i* would be

$$\begin{aligned} d_{proc}(i) &= C_{\hat{j}} / S_{proc} \\ &= \left(\frac{1}{2}\left(L_{Mem}\right)^3 + \left(\frac{5}{2} + 2L_{\hat{W}}\right)\left(L_{Mem}\right)^2 + \left(3 + 2L_{\hat{W}}\right)L_{Mem} + 2L_{\hat{W}}\right) \Big/ S_{proc} \end{aligned} \qquad (88)$$

Queuing Delay. Let assume that 802.11 MAC is utilized and RTS/CTS messages are used to find the proper time to transmit the package. The queuing delay would be a function of the number of the packets that are already in the queue, and the RTS/CTS delay.

$$d_q(i) = d_{existinPackets}(i) + d_{RTS/CTS}(i), \qquad (89)$$

where $d_{existingPackets}(i)$ is the delay associated with the existing packets in the queue , and $d_{RTS/CTS}(i)$ is the RTS/CTS delay.

It must be note that the RTS/CTS delay is consisted of transmission delay for RTS packet, propagation delay for RTS packet, responding delay at the receiver node, transmission delay for CTS packet at the receiver node, and propagation delay for the CTS packet. Assuming that the receiver node has not failed, and the communication link between the transmitter and receiver nodes is not jammed, the RTS/CTS delay is bounded

as

$$d_{RTS/CTS}^{\min} \le d_{RTS/CTS} \le d_{RTS/CTS}^{\max} \tag{90}$$

where $d_{RTS/CTS}^{\min} = \frac{L_{RTS}}{DataRate_i} + \frac{L_{CTS}}{DataRate_j} + 2d_{prop}$, and L_{RTS} and L_{CTS} are the size of the RTS and CTS packets, respectively. $DataRate_i$ and $DataRate_j$ are the data rates at the transmitter node (node *i*) and the receiver node (node *j*), respectively. d_{Prop} is the propagation delay of the packet between nodes *i* and *j*. Note that the propagation delay between the two nodes is assumed to be the same in both directions.

Assuming that the receiver node is alive, and the communication link between the transmitter and receiver node is not jammed, it can be assumed that the receiver node can receive an RTS message and respond to it properly at least every $\delta_{RTS/CTS}$. In other words, $d_{RTS/CTS}^{\max} = d_{RTS/CTS}^{\min} + \delta_{RTS/CTS}$.

The portion of the queuing delay that is associated with the existing packets in the queue can be calculated as

$$d_{existingPackets}(i) = \left(qLen(i) - 1\right) \cdot \left(d_{RTS/CTS} + d_{tx}(i)\right), \tag{91}$$

where $qLen(i)$ is the queue length at the transmitter node (node *i*), and $d_{tx}(i)$ is the transmission delay of the packet at node *i*. Assuming that the maximum queue length at the nodes is $qLen^{max}$, the queuing delay at node *i* will be bounded as following

$$d_q(i) \le \left(qLen^{\max} - 1\right) \cdot \left(d_{RTS/CTS}^{\max} + d_{tx}(i)\right) + d_{RTS/CTS}^{\max}. \tag{92}$$

In the following derivation of the transmission delay is presented.

Transmission Delay. Assuming that data rate at node i is $DataRate_i$, and the size of the packet is L_p, transmission rate of the packet is calculated as

$$d_{tx}(i) = \frac{L_p}{DataRate_i}. \tag{93}$$

Propagation Delay. Propagation delay of a packet only depends on the physical distance between the transmitter and receiver nodes, and the speed of light. It is calculated as

$$d_{prop}(i, j) = \frac{dist(i, j)}{c}, \tag{94}$$

where $dist(i,j)$ is the distance between the transmitter node i, and the receiver node j, and $c \approx 3\times10^8$ m/s is the speed of light.

Subsequently, recalling (78), and using (88), (92), (93) and (94), the upperbound for the end-to-end delay of a packet traversing along path P can be derived as

$$\begin{aligned} d_{EndToEnd} &= \sum_{\substack{l\in P,\\ i,j\in l}} \left(d_{proc}(i) + d_q(i) + d_{tx}(i) + d_{prop}(i, j) \right) \\ &\le \sum_{\substack{l\in P,\\ i,j\in l}} \left[\left(\frac{1}{2}\left(L_{Mem}\right)^3 + \left(\frac{5}{2} + 2L_{\hat{W}} \right)\left(L_{Mem}\right)^2 + \left(3 + 2L_{\hat{W}}\right)L_{Mem} + 2L_{\hat{W}} \right) \Big/ S_{proc} \right. \\ &\quad + \left(qLen^{\max} - 1\right)\cdot \left(d^{\max}_{RTS/CTS} + \frac{L_p}{DataRate_i} \right) + d^{\max}_{RTS/CTS} \\ &\quad \left. + \frac{L_p}{DataRate_i} + \frac{dist(i, j)}{c} \right] \end{aligned} \tag{95}$$

It must be noted that the availability of fast processors makes the processing delay very small. The processor speed, S_{proc}, can vary from 7.37 MIPS for Crossbow MICA2 motes [28] to 100 MIPS for Missouri S&T G4-SSN motes [29] and 235 MIPS for iPaQ

motes [30]. Furthermore, the propagation delay is significantly small except for the satellite links – which is not the case in the networks studied in this research.

III. SIMULATION RESULTS

This section compares the performance of three methods of load balancing: a) load is equally distributed among the channels, b) load balancing with respect to Available bandwidth [26], and c) our proposed method, load balancing with respect to Effective bandwidth. In the rest of this paper, for brevity, they are referred to as EQ, ABW, and EBW respectively. The protocols were implemented and evaluated in NS-2 [37].

Each node is equipped with three transmitter radios and three receiver radios. Each pair of the transmitter/receiver is tuned on a different channel (Channel 1, Channel 2 and Channel 3 of IEEE802.11). Packet size was set to 256 bytes, and data rate of 15Mbps was started at t = 5 sec. MAC rate was set to 11Mps so that the MAC layer bandwidth do not limit the overall bandwidth of the network. Queue length was set to 50, and Rayleigh fading was implemented as a special case of Nakagami fading [32], where $m0\_ = m1\_ = m2\_ = 1.0$. In order to create a non-stationary environment, a distortion was injected to channel 1 between t = 7sec and t = 11sec. This was done by varying the *gamma_* values of the Nakagami fading [32], and generating interference on Channel 1 between t = 7sec and t = 11sec. The interference at the receiver, received on Channel 1, Channel 2 and Channel 3, is illustrated in Fig.5. It shows the interference on Channel 1, induced between t = 7sec and t= 11sec.

Fig. 6 and Fig. 7 illustrate the SIR level of Channel 1, and channels 2 and 3 when

the three load balancing methods (EQ, ABW and EBW) were applied respectively. The distortion period in Channel 1 can be seen in the three cases. It can also be noticed that the SIR level in Channel 2 and Channel 3 remains almost constant. The outage probability of the three channels for the three cases of load balancing is shown in Fig. 8. It can be noticed how the outage probability in Channel 1 is highly variable during the distortion time. It can also be notice how the outage probability in Channel 1 increases during this time because of the increased distortion on the channel.

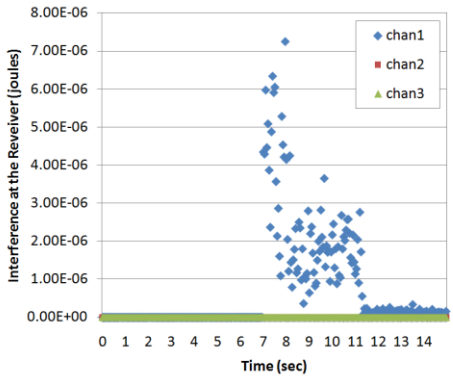

Fig. 5. Interference at the receiver on Channels 1,2, and 3, when interference is induced on Channel 1 between *t*=7 sec and *t*=11sec.

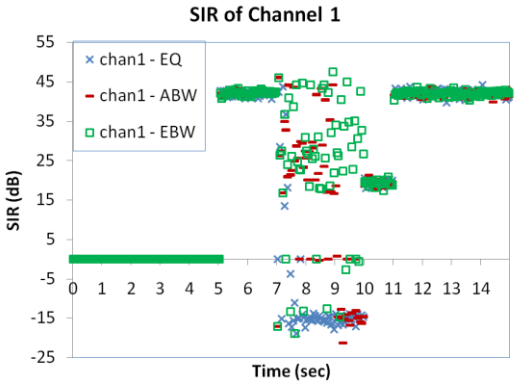

Fig. 6. SIR of channel 1 between two nodes.

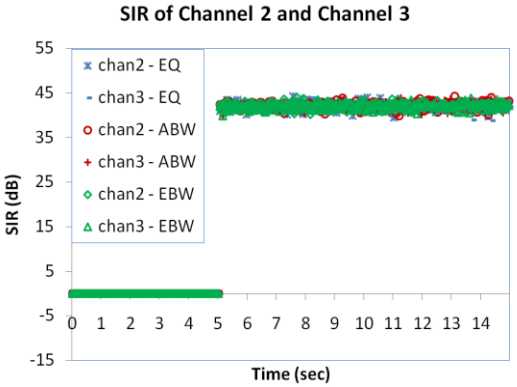

Fig. 7. SIR of channel 2 and channel 3 between two nodes.

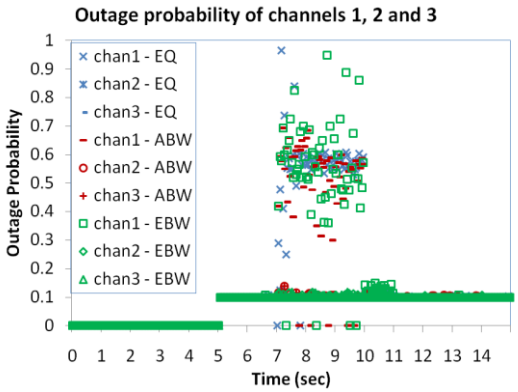

Fig. 8. Outage probability of the three channels between two nodes.

Fig. 9 represents the available bandwidth and effective bandwidth of Channel 1 when the load balancing methods EQ, ABW and EBW are applied respectively. It must be noticed how the Effective bandwidth is smaller than the Available bandwidth in all the cases, especially during the distortion, when the outage probability is high. It shows that the Effective bandwidth provides more realistic information about the channel conditions in the presence of uncertainties and distortions.

Fig. 10 shows the Available bandwidth and Effective bandwidth of Channel 2 when the load balancing methods EQ, ABW and EBW are applied respectively. It can be seen that the Available bandwidth and Effective bandwidth of Channel 2 are more stationary compared to Channel 1. Also the difference between the values of these two bandwidths is not as significant as it was in Channel 1during the distortion. However, it can also be noticed that the distortion in Channel 1 slightly affects the Available bandwidth and Effective bandwidth in Channel 2. The available bandwidth and effective bandwidth in Channel 3 are similar to those of Channel 2. Therefore, for the sake of saving space, they are not presented in this paper.

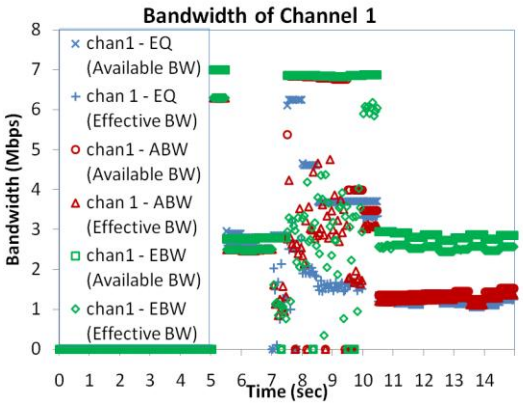

Fig. 9. Available bandwidth and effective bandwidth of channel 1 between two nodes.

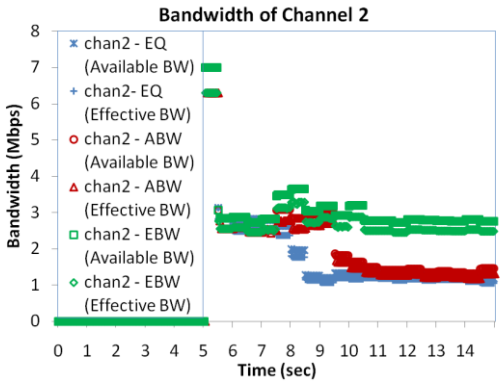

Fig. 10. Available bandwidth and effective bandwidth of channel 2 between two nodes.

Finally, Fig. 11 and Fig. 12 present the throughput in the three links on Channels 1, 2 and 3, using EQ, ABW and EBW, respectively. Note how equally distributing the flow among the links degrades the throughput in Channel 1, and also in the other two channels. Note how the EBW method is able to maintain the throughput in the other two channels and provides a slightly better throughput in Channel 1. Using the EBW method for load balancing, and also using the new proposed routing method, simulation was run for a network of 100 nodes in a 500m×4000m area, and throughput at the destination node was measured (see Fig. 13). It can be seen that the channel distortion affects the total throughput when the routing protocol is AODV. However, NEW protocol is able to compensate for the drop in throughput by finding alternative route and load balancing.

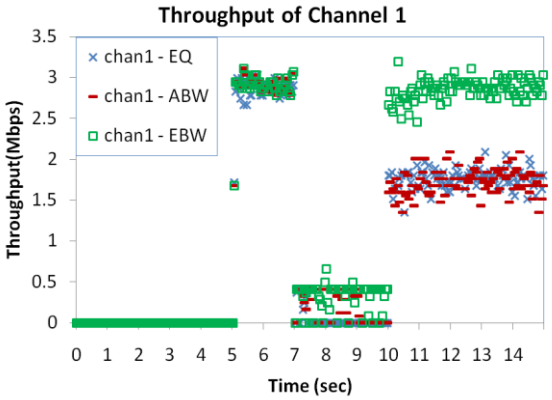

Fig. 11. Throughput of channel 1 between two nodes

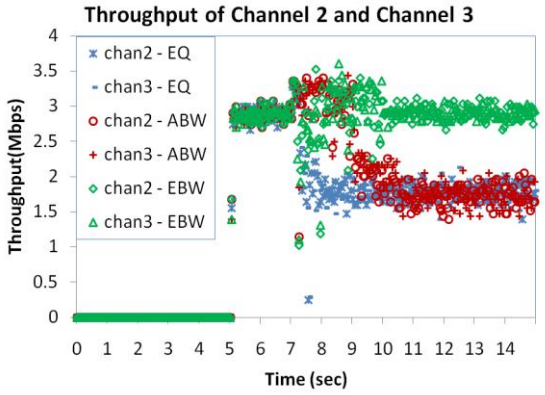

Fig. 12. Throughput of channel 2 and channel 3 between two nodes.

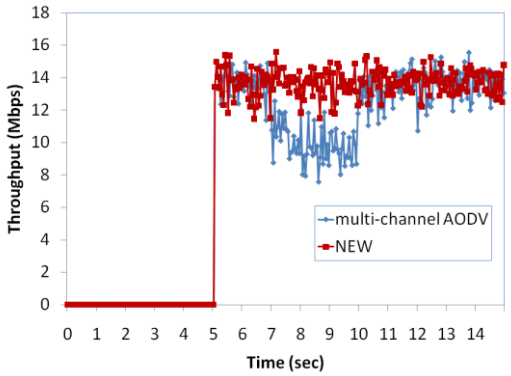

Fig. 13. Throughput of the network for one data flow in the presence of Rayleigh fading over channels, and distortion on one wireless link.

Figs. 14 and 15 illustrate the throughput packet drop rate in mobile case for the above scenario for various node speeds (5, 10, 15 and 20 m/s) and various hop distance from the destination (BS) (20 and 50 hops). It can be noticed how the NEW routing protocol, using Effective bandwidth in link cost metric and load balancing outperforms AODV when the average hop distance between source and destination is 50 hops. On the other hand, when the average hop is small, AODV provides a higher throughput. The reason is that our proposed routing protocol requires a minimum number of iterations (hops) to converge; therefore, for small hop distances AODV has a better performance.

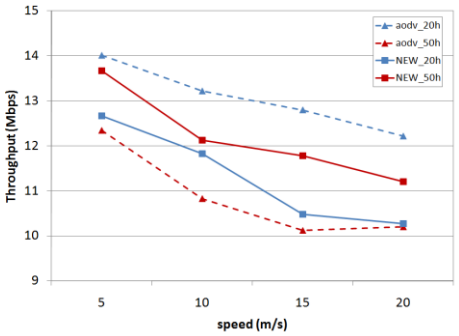

Fig. 14. Throughput of the network in the presence of Rayleigh fading and distortion.

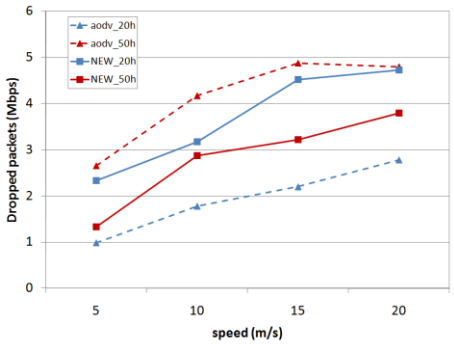

Fig. 15. Packet drop rate of the network in the presence of Rayleigh fading and distortion.

Fig.16 shows the energy efficiency (joules/packet) in mobile case for the same scenario for various node speeds and various hop distance from the destination (BS). The NEW routing protocol that utilizes available energy in link cost, and also uses Effective bandwidth in link cost metric and load balancing outperforms AODV. Furthermore, the dynamic and forward-in-time nature of the NEW routing protocol eliminates the frequent route discovery and route repair messages, and thus contributing to higher energy efficiency of the NEW routing protocol compared to AODV.

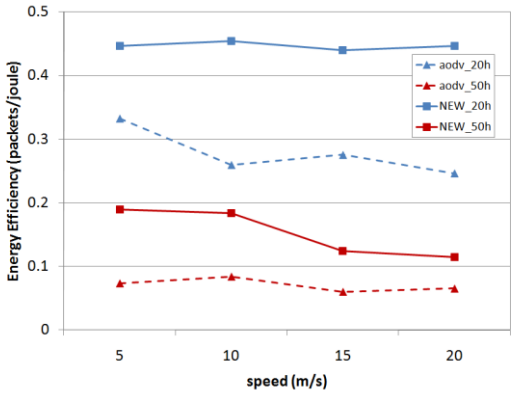

Fig. 16. Energy efficiency of the network in the presence of Rayleigh fading over channels, and distortion on one wireless link.

Figs. 17 and 18 illustrate the network throughput and the packet drop rate for varying data rates. The node movement speed was 5m/s and various hop distance from the destination (BS) (20 and 50 hops) was examined. It is notices that for the low data rates, where the channel capacity can handle the flow, the drop rate is zero. As the data

rate increases, and exceeds the capacity of one channel (here it happens at data rate of 10Mps), packet loss starts, and the additional channels (in this scenario Channel 2 and Channel 3) and load balancing are utilized. It can be noticed that when the average hop is larger, our proposed routing protocol provides a higher throughput. The reason is that our proposed routing protocol requires a minimum number of iterations (hops) to converge; therefore, for small hop distances AODV has a better performance.

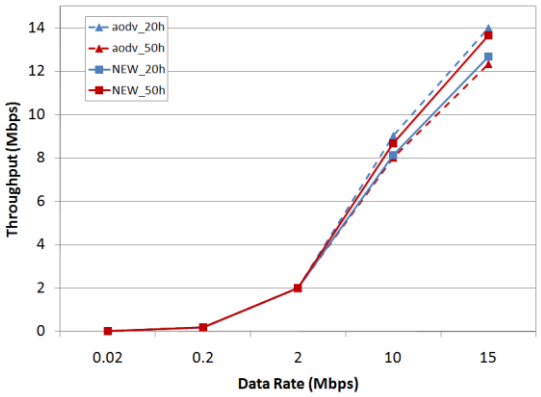

Fig. 17. Throughput of the network for varying data rate in the presence of Rayleigh fading and distortion.

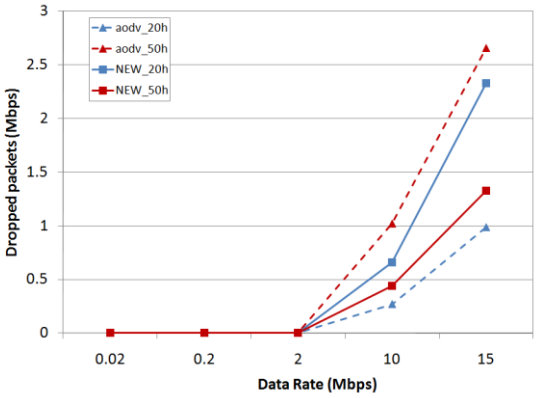

Fig. 18. Packet drop rate of the network for varying data rate in the presence of Rayleigh fading and distortion.

IV. CONCLUSIONS

This paper proposed an adaptive routing protocol for multi-channel multi-interface wireless ad hoc networks in the presence of channel uncertainties. Time-varying outage probability for the channel links were determined by modeling the channel as a Markov model, and subsequently used to estimate the channel condition by MLE.

The outage probability and channel estimation were utilized in load balancing over multiple links. Bellman's principle of optimality and nonlinear optimal framework, as well as OLA, was utilized to find the routes reactively. The OLA used the link and channel conditions such as the effective bandwidth (which is defined by outage probability and available bandwidth), link delay, queue occupancy and available energy in the next node to estimate the route cost from the next node to Base Station.

Simulation results in the presence of Rayleigh fading over the wireless channels, and distortion over one channel show that the effective bandwidth as a function of outage probability can be a more precise estimation of the available bandwidth of the channel. Once used in load balancing, it outperforms methods EQ and ABW. Simulation results also showed that the proposed routing method is able to compensate for the drop in throughput by load balancing and finding alternative route. Furthermore, by performing the forward-in-time route discovery, it improves the energy efficiency by avoiding flooding the network with route repair/route discovery messages.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Maciej Zawodniok for his precious comments and also providing the source code for measuring the interference signal in NS-2.

REFERENCES

[1] Y.-D. Yao and A.U.H. Sheikh, "Investigations into cochannel interference in microcellular mobile radio systems," *Vehicular Technology, IEEE Transactions on*, vol.41, no.2, pp.114-123, May 1992.

[2] E. N. Gilbert, "Capacity of burst noise channels," *Bell Syst. Tech. J.,* VO~39. , pp. 1253-1256, 1960.

[3] N. Weinberger and M. Feder, "Universal decoding for linear Gaussian fading channels in the competitive minimax sense," *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, vol., no., pp.782-786, 6-11 July 2008.

[4] L. Wang and Y. Cheng, "A statistical mobile-to-mobile Rician fading channel model," *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol.1, no., pp. 63- 67 Vol. 1, 30 May-1 June 2005.

[5] A.J. Coulson, A.G. Williamson, and R.G. Vaughan, "A statistical basis for lognormal shadowing effects in multipath fading channels," *Communications, IEEE Transactions on* , vol.46, no.4, pp.494-502, Apr 1998.

[6] W. H. Shen and N. Moayeri, "Finite-state Markov channel-a useful model for radio communication channels," *Vehicular Technology, IEEE Transactions on*, vol.44, no.1, pp.163-171, Feb 1995.

[7] C. Hedrick, "Routing information protocol," Internet-Request for Comments 1058, IETF, June 1988.

[8] R. Bellman. Dynamic Programming. Princeton University Press, Princeton, New Jersey, 1957.

[9] D.B. Johnson, D.A. Maltz, Y. Hu, and J.G. Jetcheva, "The dynamic source routing protocol for mobile ad hoc networks (DSR)" Internet-Draft, IETF, November 2001. draft-ietfmanet-dsr-06.txt.

[10] C. Perkins, E. Royer, and S. R. Das., "Ad hoc on-demand distance vector (AODV) routing," Internet-Draft, IETF, January 2002. draft-ietf-manet-aodv-10.txt.

[11] T. Liu and W. Liao, "Interference-aware QoS routing for multi-rate multi-radio multi-channel IEEE 802.11 wireless mesh networks," *Wireless Communications, IEEE Transactions on* , vol.8, no.1, pp.166-175, Jan. 2009.

[12] B. Eslamnour, M. Zawodniok and S. Jagannathan , "Dynamic channel allocation in wireless networks using adaptive learning automata," *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE* , vol., no., pp.1-6, 5-8 April 2009

[13] F. L. Lewis and V. L. Syrmos, Optimal Control, 2nd ed. Hoboken, NJ: Wiley, 1995.

[14] B. Eslamnour and S. Jagannathan, "Adaptive dynamic routing for wireless ad hoc networks using nonlinear optimal framework." unpublished, submitted.

[15] T. Lochmatter, "Base stations in mobile ad hoc networks," Thesis, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, July 2004.

[16] O. Dousse, P. Thiran, and M. Hasler, “Connectivity in ad-hoc and hybrid networks,” in Proc. IEEE INFOCOM’02, vol. 2, New York, USA, June 2002, pp. 1079–1088.

[17] R.-S. Chang, W.-Y. Chen, and Y.-F. Wen, “Hybrid wireless network protocols,” *Vehicular Technology, IEEE Transactions on* , vol.52, no.4, pp. 1099- 1109, July 2003.

[18] M. Popa, C. Moica, A.S. Popa, and D. Mnerie, “Hierarchical ad hoc networks,” *EUROCON, 2007. The International Conference on "Computer as a Tool"* , vol., no., pp.2509-2516, 9-12 Sept. 2007.

[19] S. Jagannathan, Neural Network Control of Nonlinear Discrete-Time Systems, CRC Press, April 2006.

[20] T. Dierks and S. Jagannathan, “Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics,” *Decision and Control, 2009. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on* , vol., no., pp.6750-6755, 15-18 Dec. 2009.

[21] C. Eui-Young, L. Benini, A. Bogliolo, L. Yung-Hsiang and G. De Micheli, “Dynamic power management for nonstationary service requests,” *Computers, IEEE Transactions on* , vol.51, no.11, pp. 1345- 1361, Nov 2002.

[22] T. W. Anderson and L. A. Goodman, “Statistical inference about Markov chains,” *The Annals of Mathematical Statistics*, Vol. 28, No. 1 (Mar., 1957), pp. 89-110.

[23] R. Fantacci and M. Scardi, “Performance evaluation of preemptive polling schemes and ARQ techniques for indoor wireless networks,” *Vehicular Technology, IEEE Transactions on* , vol.45, no.2, pp.248-257, May 1996.

[24] A. G. Williamson and J. D. Parsons, “Outage probability in a mobile radio system subject to fading and shadowing,” *Electron. Lett.*, vol. 21,pp. 622-623, 1985.

[25] D. Bertsekas and R. Gallager, Data Networks, Prentice Hall 1987.

[26] R. Anguswamy, M. Zawodniok, S. Jagannathan, “A multi-interface multi-channel routing (MMCR) protocol for wireless ad hoc networks,” *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, vol., no., pp.1-6, 5-8 April 2009.

[27] S. Eberli, D. Cescato and W. Fichtner, “Divide-and-conquer matrix inversion for linear MMSE detection in SDR MIMO receivers," *NORCHIP, 2008.* , vol., no., pp.162-167, 16-17 Nov. 2008.

[28] http://www.xbow.com/

[29] J.W. Fonda, M.J. Zawodniok, S. .Jagannathan, A. Salour and D. Miller, "Missouri S&T mote-based demonstration of energy monitoring solution for network enabled manufacturing using wireless sensor networks (WSN)," *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on* , vol., no., pp.559-560, 22-24 April 2008.

[30] Compaq iPAQ, http://www.compaq.com/products/iPAQ

[31] NS-2: http://www.isi.edu/nsnam/ns/

[32] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi and H. Hartenstein, "Overhaul of IEEE 802.11 modeling and simulation in NS-2," *In Proceedings of the 10th ACM/IEEE International Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems* (MSWiM), pp. 159-168, Chania, Greece, October 2007.

APPENDIX A

Theorem A.1: Consider a Markov chain with *N* states. Let the MLE of the transition probabilities be defined as $\hat{p}_{i,j} = \frac{q_{i,j}}{q_i}$, where $q_i = \sum_{j=0}^{N-1} q_{i,j}$, and $\sum_{j=0}^{N-1} q_i = n$. Prove that

$$\lim_{n\to\infty} \hat{p}_{i,j} = p_{i,j}^{\infty}, \tag{A.1}$$

where $p_{i,j}^{\infty}$ is the long-run transition probability from state s_i to state s_j $p_{i,j}^{\infty} = P(X_{k+1} = s_j \mid X_k = s_i)$.

Proof: Recall that $n_{i,j} = \sum_{k=1}^{n-1} \mathrm{I}_{X_k}(s_i).\mathrm{I}_{X_{k+1}}(s_j)$, where $\mathrm{I_A}(.)$ is the indicator function. It can be written

$$\frac{q_{i,j}}{n-1} = \frac{1}{n-1}\sum_{k=1}^{n-1} \mathrm{I}_{X_k}(s_i).\mathrm{I}_{X_{k+1}}(s_j) \tag{A.2}$$

The RHS of the above equation is the time average of indictor function, and

converges to expectation

$$\lim_{n\to\infty}\frac{q_{i,j}}{n-1}=E\left[\mathrm{I}_{X_k}(s_i).\mathrm{I}_{X_{k+1}}(s_j)\right] \tag{A.3}$$

Also recall that for indicator function, $E\left[\mathrm{I}_A(x)\right]=\int_A d\mathbf{P}=P(A)$, therefore

$$E\left[\mathrm{I}_{X_k}(s_i).\mathrm{I}_{X_{k+1}}(s_j)\right]=P\left(X_k=s_i,X_{k+1}=s_j\right)=P(X_{k+1}=s_j\mid X_k=s_i).P\left(X_k=s_i\right). \tag{A.4}$$

Recall that the long-run transition probability, and long-run state probability are respectively $p_{i,j}^{\infty}=P(X_{k+1}=s_j\mid X_k=s_i)$, and $p_i=P(X_k=s_i)$. Therefore

$$\lim_{n\to\infty}\frac{q_{i,j}}{n-1}=p_{i,j}.p_i\ . \tag{A.5}$$

Now consider $q_i=\sum_{j=0}^{N-1}q_{i,j}=\sum_{k=1}^{n-1}\mathrm{I}_{X_k}(s_i)$. Similar to what was done in (A.2), by dividing both sides by *n*-1, it yields

$$\lim_{n\to\infty}\frac{q_i}{n-1}=\lim_{n\to\infty}\left(\frac{1}{n-1}\sum_{k=1}^{n-1}\mathrm{I}_{X_k}(s_i)\right)=E\left[\mathrm{I}_{X_k}(s_i)\right]=P\left(X_k=s_i\right)=p_i\ . \tag{A.6}$$

Thus it can be written

$$\lim_{n\to\infty}\hat{p}_{i,j}=\lim_{n\to\infty}\frac{q_{i,j}}{q_i}=\lim_{n\to\infty}\left(\frac{q_{i,j}}{n-1}\Big/\frac{q_i}{n-1}\right)=\frac{p_{i,j}^{\infty}.p_i}{p_i}=p_{i,j}^{\infty} \tag{A.7}$$ ■

4. Distributed Cooperative Resource Allocation for Primary and Secondary Users with Adjustable Priorities in Cognitive Radio Networks

Behdis Eslamnour, S. Jagannathan and Maciej Zawodniok
Department of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, MO, USA
e-mail: {ben88, sarangap, mjzx9c}@mst.edu

***Abstract*—** The ability of cognitive radio to sense the spectrum and adapt its parameters in response to the dynamic environment makes it an ideal candidate for resource allocation of spectrum in wireless networks, especially co-existing and emergency networks. In the two latter networks, the secondary users should sense the spectrum and adapt their parameters such that they can use these resources without causing a degradation or interference to the performance of the primary/licensed users. Therefore, in this paper, a decentralized game theoretic approach for resource allocation of the primary and secondary users in a cognitive radio networks is proposed. In this work, the priorities of the networks are incorporated in the utility and potential functions which are in turn used for resource allocation. It is demonstrated analytically by using the potential and utility functions and through simulation studies that a unique NE exists for the combined game with primary users (PU) and secondary users (SU), and the combined game converges to the NE.

Keywords-***cognitive radio netoworks; resource allocation; game theory; Nash Equilibrium; potential games; Bayesian games; imperfect information***

I. INTRODUCTION

Cognitive networks [1] have emerged as a new communication mode promising improved spectrum utilization and faster and more reliable network service. The two distinct characteristics of cognitive radios are cognitivity and configurability which enable the cognitive radios to sense the dynamic environment and adapt their parameters to optimize their performance.

Game theoretic approach is one of the ways to perform resource allocation, organize the networks and their users' behavior within the network and also with respect to the co-existing networks [2]-[8],[10],[13]. Existing game theoretic approaches such as [8] and [10] have taken a non-cooperative approach to power, joint power and rate control with Quality of Service (QoS) constraints, respectively. By contrast, a cooperative approach was taken in [3] to maximize the channel capacity. In [7], a framework for cooperation among the primary and secondary users is introduced. An interference avoidance protocol for peer-to-peer networks was presented in [4] that assumed either the network is centralized, or the receivers are co-located.

The utility function used in the literature is a function of bit-error-rate [5],[8],[10], a combination of the signal power and interference power at the receiver [3] , or signal-to-interference ratio [5]. Depending on the approach, the convergence and optimality of the algorithms are proven by using the existence of either Pareto optimality or Nash equilibrium (NE), existence of potential functions as well as the uniqueness of NE. However, the existing approaches [4], [5], [8] and [10] are applicable to a single network, or networks with equal priorities.

In [5], where primary and secondary networks are considered, it is assumed that there is only one primary user in the network. Though this work can be generalized to more than one primary user, the utility function is not discussed which takes into account the interference caused by the additional primary users (PUs). The work of [3] considers primary and secondary networks with the primary network using time division multiple access (TDMA) protocol where PUs cannot transmit at the same time. Unless the PUs communicate to a base station, the interference caused by a hidden node during communication among PUs, is not addressed.

On the other hand, authors in [5],[7],[10] have provided the presence of a unique NE. However, the convergence of the combined game with primary and secondary users is not shown since finding a utility function that provides a unique NE for the combined game is rather difficult. Others assumed a single game with homogeneous players [4], or only consider a single PU node and a set of SU players [3],[5]. By contrast, in this paper a decentralized game theoretic approach for resource allocation of the primary and secondary users in a cognitive radio network is proposed by relaxing these assumptions and defining priority levels. The priorities of the networks are incorporated in the proposed utility and potential functions. It is demonstrated analytically and through simulation studies that a unique NE exists for the combined game with primary users and secondary users (SU). Since the interaction among the networks and their priority levels are incorporated in the functions, the proposed game can be extended to a game among multiple co-existing networks, each with different priority levels.

The contributions of the work include the incorporation of priority levels in utility and potential functions, analytical proofs of existence and uniqueness of NE for the PU

and SUs by using these functions with priority levels, and existence of the NE for the combined game. The net result is a game theoretic approach applicable to multiple co-existing networks.

Therefore in this paper, after presenting the system representation of a cognitive network in Section II, the individual games for primary and secondary networks are proposed, and the existence and uniqueness of NE for each of the networks is shown in Section III. Then by incorporating priority parameters, these two games are combined and the existence of NE for the combined game is proven through exact potential function. Section IV presents the simulation results, showing the convergence of the games and effect of priority parameter. Finally, Section V concludes the paper.

II. System Model

We consider a cognitive radio network consisted of a network of primary users (PUs), and a network of secondary users (SUs). The network can be heterogeneous, but it is assumed that that the wireless devices are cognitive and capable of sensing the environment and adapting their parameters. An example of the cognitive network is illustrated in Fig. 1.

In this paper, M PU nodes and N SU nodes with Q orthogonal channels are considered. The transmission power at the ith PU is denoted by p_i^P and the transmission power at the jth SU is given by p_j^S while it is assumed that the SU transmitter/receiver pairs are within the communication range of each other. The communication between the SU or PU pairs can experience interference from transmissions emanating from other

PUs or SUs that are using the same channel, and are within the sensing range of the receivers.

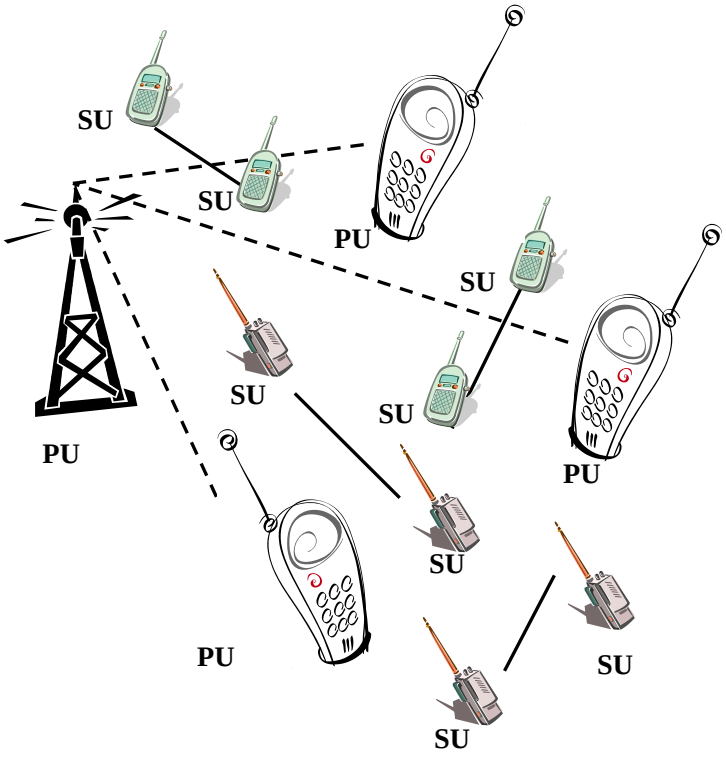

Fig. 1. Cognitive radio network architecture.

Given the channel and transmission power as the network resources, the objective is to allocate these resources such that a) the interference that SUs cause to the PUs will be minimized, and b) SUs be able to communicate to each other. It must be noted that PU nodes do have a very little incentive in reducing their interference on the SU nodes. The objective of the PU network is to satisfy its QoS requirements with the minimum energy consumption. On the other hand, the SU nodes should avoid adding interference to the PU nodes while trying to find a spectrum hole for their own communications.

III. POWER CONTROL AND CHANNEL ALLOCATION GAME

In this section the PU and SU games and their utility functions are introduced. The priorities of the PU and SU networks are incorporated in the utility functions such that the SU nodes receive a larger penalty for causing interference to the PU nodes. On the other hand, the penalty that the PU nodes receive for causing interference to the SU communications is not significant. Subsequently, the existence and uniqueness of NE for the PU and SU games are proven and then the existence of the NE for the combined game is shown by presenting an exact potential function.

In the following, the PU and SU games are introduced and existence of NE is demonstrated.

A. Primary User Game

Consider a normal[1] form game represented as

$$\Gamma^P = \left\{ M, \left\{ A_i^P \right\}_{i \in M}, \left\{ u_i^P \right\}_{i \in M} \right\} \tag{1}$$

where M is the finite set of players, i.e., the PUs. A_i^P is the set of actions available for PU player i, and u_i^P is the utility function associated with player i. If all the available actions for the PU players are collected in action space and denoted by $A^P = \underset{i \in M}{\times} A_i^P = A_1^P \times A_2^P \times \cdots \times A_M^P$, then $u_i^P : A^P \to \mathbb{R}$. The decision making by players is accomplished in a distributed manner, but at the same time the decision of each player affects the other players' decisions. The goal of the game is to achieve a set of actions for

[1] A normal (strategic) form game is a game with three elements: the set of players, $i \in M$, the strategy space A_i for each player i, and payoff functions that give player i's utility u_i for each profile of strategies [9].

the players in order to attain an equilibrium point such that no player has any interest in choosing another action unilaterally.

Each PU player action consists of: (a) transmission power, p_i^P, and (b) channel, c_i^P, where $c_i^P \in \{c_1, c_2, \ldots, c_Q\}$ represents the channel that PU_i is using (superscript P denoting primary users), and Q represents the number of channels. Let the utility function for the PU player i be defined as

$$\begin{aligned} u_i^P(a_i, a_{-i}) = {} & \alpha_P \log_2\left(1 + p_i^P h_{ii}^P\right) \\ & -\beta_P\left(\sum_{k\neq i}^{M} p_k^P h_{ki}^P e\left(c_k^P, c_i^P\right) + \sum_{k=1}^{N} p_k^S h_{ki}^S e\left(c_k^S, c_i^P\right)\right) \\ & -\left(\eta_P \sum_{k\neq i}^{M} p_i^P h_{ik}^P e\left(c_k^P, c_i^P\right) + \lambda_P \sum_{k=1}^{N} p_i^P h_{ik}^P e\left(c_k^P, c_i^S\right)\right) \\ & -\kappa_P p_i^P\left(\sum_{k\neq i}^{M} p_k^P + \sum_{k=1}^{N} p_k^S\right) \end{aligned} \tag{2}$$

where h_{ij}^P is the channel gain between the transmitter PU node i and receiver j, h_{ij}^S is the channel gain between the transmitter SU node i and receiver node j, and $e(\cdot)$ is a Boolean function defined as $e(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$.

In the utility function, the first term is associated with the payoff at the receiver node of PU transmitter i while the second term represents the interference that other nodes cause at PU_i . The third term accounts for the interference that PU_i causes at the other PU players and also at the SU players whereas the fourth term accounts for the price (penalty) that the user pays for consuming more energy.

Remark 1. It must be noted that the summation of the first and the second terms resembles the Shannon capacity [16] with a twist. The interference term has been moved out of the *log* term so that the required conditions for the utility functions and potential functions (sections *C*, *D* and *E*) will be satisfied.

Rewrite (2) as

$$\begin{aligned} u_i^P(a_i, a_{-i}) = {} & \alpha_P \log_2\left(1 + h_{ii}^P p_i^P\right) \\ & -\beta_P\left(\sum_{k\neq i}^{M} p_k^P h_{ki}^P e\left(c_k^P, c_i^P\right) + \sum_{k=1}^{N} p_k^S h_{ki}^S e\left(c_k^S, c_i^P\right)\right) \\ & -p_i^P\left(\eta_P \sum_{k\neq i}^{M} h_{ik}^P e\left(c_k^P, c_i^P\right) + \lambda_P \sum_{k=1}^{N} h_{ik}^P e\left(c_k^P, c_i^S\right)\right) \\ & -\kappa_P p_i^P\left(\sum_{j=1, j\neq i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S\right) \end{aligned} \quad . \tag{3}$$

B. Secondary User Game

Consider a normal form game represented as

$$\Gamma^S = \left\{N, \left\{A_i^S\right\}_{i\in N}, \left\{u_i^S\right\}_{i\in N}\right\} \tag{4}$$

where N is the finite set of players, i.e., the SUs, A_i^S is the set of actions available for SU player *i*, and u_i^S is the utility function associated with player *i*. If all available actions for all SU players are represented by $A^S = \underset{i\in N}{\times} A_i^S$, then $u_i^S : A^S \to \mathbb{R}$. The decision making by players is distributed and it affects the decisions for the other players. The goal of the SU game is to achieve a set of actions for the players (equilibrium point) such that no player has any interest in choosing another action unilaterally. The action for each SU player is consisted of two components: (a) transmission power, p_i^S, and (b) channel, c_i^S, where $c_i^S \in \left\{c_1, c_2, \ldots, c_Q\right\}$.

Let the utility function for SU player *i* be defined as

$$u_i^S(a_i,a_{-i})=\alpha_S\log_2\left(1+p_i^S h_{ii}^S\right)-\beta_S\left(\sum_{k=1}^{M}p_k^P h_{ki}^P e\left(c_k^P,c_i^S\right)+\sum_{k\neq i}^{N}p_k^S h_{ki}^S e\left(c_k^S,c_i^S\right)\right)$$
$$-\left(\eta_S\sum_{k=1}^{M}p_i^S h_{ik}^S e\left(c_k^S,c_i^P\right)+\lambda_S\sum_{k\neq i}^{N}p_i^S h_{ik}^S e\left(c_k^S,c_i^S\right)\right)-\kappa_S p_i^P\left(\sum_{k\neq i}^{M}p_k^P+\sum_{k=1}^{N}p_k^S\right) \quad (5)$$

where h_{ij}^P and h_{ij}^S are the channel gains, and $e(\cdot)$ is a Boolean function defined as

$$e(x,y)=\begin{cases}1 & \text{if } x=y\\ 0 & \text{if } x\neq y\end{cases}.$$

In this utility function, the first term is associated with the payoff at the receiver node of SU transmitter *i* while the second term represents the interference that SU_i causes at the PU players the other SU players. The third term accounts for the interference that SU_i causes at the PU players and also at the other SU players whereas the fourth term accounts for the price that it pays for consuming more energy.

Rewrite (5) as

$$u_i^S(a_i,a_{-i})=\alpha_S\log_2\left(1+p_i^S h_{ii}^S\right)-\beta_S\left(\sum_{k=1}^{M}p_k^P h_{ki}^P e\left(c_k^P,c_i^S\right)+\sum_{k\neq i}^{N}p_k^S h_{ki}^S e\left(c_k^S,c_i^S\right)\right)$$
$$-p_i^S\left(\eta_S\sum_{k=1}^{M}h_{ik}^S e\left(c_k^S,c_i^P\right)+\lambda_S\sum_{k\neq i}^{N}h_{ik}^S e\left(c_k^S,c_i^S\right)\right)-\kappa_S p_i^P\left(\sum_{k\neq i}^{M}p_k^P+\sum_{k=1}^{N}p_k^S\right) \quad (6)$$

Next it will be shown that NE exists and it is unique.

C. Existence of Nash Equilibrium

In this section the existence of NE for the PU and SU games is demonstrated. Before proceeding, the following is needed.

Definition 1. The normal game $\Gamma = \left\{N, \left\{A_i\right\}_{i\in N}, \left\{u_i\right\}_{i\in N}\right\}$ has a Nash Equilibrium if for all $i \in N$, $\left\{A_i\right\}_{i\in N}$ is a nonempty convex compact subset of a Euclidean space, and $u_i(a)$ is continuous and concave/quasiconcave on A_i [9].

Theorem 1. The PU game $\Gamma^P = \left\{M, \left\{A_i^P\right\}_{i\in M}, \left\{u_i^P\right\}_{i\in M}\right\}$ has a unique NE.

Proof. In the game $\Gamma^P = \left\{M, \left\{A_i^P\right\}_{i\in M}, \left\{u_i^P\right\}_{i\in M}\right\}$, $\left\{A_i^P\right\}_{i\in M}$ is a nonempty compact subset of Euclidean space $\mathbb{R}\times\mathbb{N}$. Also recall (3), where the utility function is continuous in A_i^P such that

$$u_i^P(a_i, a_{-i}) = \alpha_P \log_2\left(1 + h_{ii}^P p_i^P\right) - \beta_P\left(\sum_{k\neq i}^{M} p_k^P h_{ki}^P e\left(c_k^P, c_i^P\right) + \sum_{k=1}^{N} p_k^S h_{ki}^S e\left(c_k^S, c_i^P\right)\right)$$
$$- p_i^P\left(\eta_P \sum_{k\neq i}^{M} h_{ik}^P e\left(c_k^P, c_i^P\right) + \lambda_P \sum_{k=1}^{N} h_{ik}^P e\left(c_k^P, c_i^S\right)\right) - \kappa_P p_i^P\left(\sum_{j=1, j\neq i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S\right).$$

The first derivative of the utility function is given by

$$\frac{\partial u_i^P}{\partial p_i^P} = \frac{\alpha_P}{\ln 2}\cdot\frac{h_{ii}^P}{1 + p_i^P h_{ii}^P} - \left(\eta_P \sum_{k\neq i}^{M} h_{ik}^P e\left(c_k^P, c_i^P\right) + \lambda_P \sum_{k=1}^{N} h_{ik}^P e\left(c_k^P, c_i^S\right)\right) - \kappa_P\left(\sum_{k\neq i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S\right) \quad (7)$$

Consequently, the second derivative of the utility function is given by

$$\frac{\partial^2 u_i^P}{\partial p_i^P \partial p_i^P} = -\frac{\alpha_P}{\ln 2}\cdot\frac{\left(h_{ii}^P\right)^2}{\left(1 + p_i^P h_{ii}^P\right)^2} \quad (8)$$

Hence $\frac{\partial^2 u_i^P}{\partial p_i^P \partial p_i^P} < 0$, and it can be concluded that the utility function, u_i^P, is concave on P_i. Therefore, recalling *Definition* 1, it is shown that NE exists for the PU game Γ^P. ■

Theorem 2. The SU game $\Gamma^S = \left\{ N, \left\{ A_i^S \right\}_{i \in N}, \left\{ u_i^S \right\}_{i \in S} \right\}$ has a unique NE.

Proof. Follow similar steps as the PU game. ■

D. Uniqueness of Nash Equilibrium

The best-response correspondence[2] of the PU and SU games were found in the previous section. However, they must satisfy certain conditions so that a unique NE can be attained for the PU and SU games. First a definition of a unique NE is introduced followed by a theorem to show that indeed the NE for the PU and SU are unique.

Definition 2. A Nash equilibrium is unique if the best-response correspondence, $r(\mathbf{p})$, is a *standard function* [13] satisfying the following properties: (a) *Positivity*: $r(\mathbf{p}) > 0$, (b) *Monotonicity*: if $\mathbf{p} \geq \mathbf{p}'$, then $r(\mathbf{p}) \geq r(\mathbf{p}')$, and (c) *Scalability*: for all $\omega > 1$, $\omega r(\mathbf{p}) \geq r(\omega \mathbf{p})$.

Definition 3. A Nash equilibrium is unique if the best-response correspondence, $r(\mathbf{p})$ is a *type II standard function* [14] with following properties: (a) *Positivity*: $r(\mathbf{p}) > 0$,

[2] Best-response correspondence (function) for player i is a function that yields a set of actions for player i, $r_i(a_{-i})$, when the other players' actions are given by a_{-i}, *i.e.*

(b) *Type II Monotonicity*: if $\mathbf{p} \le \mathbf{p}'$, then $r(\mathbf{p}) \ge r(\mathbf{p}')$, and (c) *Type II Scalability*: for all $\omega > 1$, $r(\omega\mathbf{p}) \ge (1/\omega) r(\mathbf{p})$.

Theorem 3. NE for the PU game and SU game are unique.

Proof. The uniqueness of the NE for the PU and SU games is proved by showing that the best-response correspondences for them are type II standard functions.

Uniqueness of NE for the PU game. The best-response correspondence of the PU game is found by making $\frac{\partial u_i^P}{\partial p_i^P} = 0$. Recalling (7), the best-response correspondence will be derived as following

$$\frac{\alpha_P}{\ln 2} \cdot \frac{h_{ii}^P}{1 + p_i^P h_{ii}^P} - \left(\eta_P \sum_{k \ne i}^{M} h_{ik}^P e\left(c_k^P, c_i^P\right) + \lambda_P \sum_{k=1}^{N} h_{ik}^P e\left(c_k^P, c_i^S\right) \right) - \kappa_P \left(\sum_{k \ne i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S \right) = 0 \qquad (9)$$

$$r_i^P(\mathbf{p}) = p_i^P = \frac{\alpha_P}{\ln 2} \cdot \frac{1}{\eta_P \sum_{k \ne i}^{M} h_{ik}^P + \lambda_P \sum_{k=1}^{N} h_{ik}^P + \kappa_P \left(\sum_{k \ne i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S \right)} - \frac{1}{h_{ii}^P} \quad . \qquad (10)$$

However, the coefficients must satisfy certain conditions so that a positive real value can be achieved for p_i^P that maximizes the utility function. This p_i^P is the unique NE.

Now let's study the three properties of a standard function (or type II standard function), and examine the best-response correspondence function.

$r_i(a_{-i}) = \{a_i \text{ in } A_i : u_i(a_i, a_{-i}) \ge u_i(a_i', a_{-i}) \text{ for all } a_i' \text{ in } A_i\}$. Any action in that set is at least as good for player i as every other action of player i, when the other players' actions are given by a_{-i}. [9]

Positivity: $r(\mathbf{p}) > 0$. The best response correspondence, $r_i^P(\mathbf{p})$, has this property if

$$\frac{\alpha_P}{\ln 2} \cdot \frac{1}{\left(\eta_P \sum_{k\neq i}^{M} h_{ik}^P e\left(c_k^P, c_i^P\right) + \lambda_P \sum_{k=1}^{N} h_{ik}^P e\left(c_k^P, c_i^S\right)\right) + \kappa_P \left(\sum_{k\neq i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S\right)} - \frac{1}{h_{ii}^P} > 0.$$

In other words, $\frac{\alpha_P . h_{ii}}{\ln 2} > \left(\eta_P (M-1) + \lambda_P N\right) + \kappa_P (M+N-1) P_{\max}$, where P_{max} is a the maximum transmission power at the nodes. The condition can be satisfied by selecting the proper values for α_P, η_P, λ_P and κ_P.

Type II monotonicity: if $\mathbf{p} \le \mathbf{p}'$, then $r(\mathbf{p}) \ge r(\mathbf{p}')$. Recall that

$$r_i^P(\mathbf{p}) = \frac{\alpha_P}{\ln 2} \cdot \frac{1}{\left(\eta_P \sum_{k\neq i}^{M} h_{ik}^P e\left(c_k^P, c_i^P\right) + \lambda_P \sum_{k=1}^{N} h_{ik}^P e\left(c_k^P, c_i^S\right)\right) + \kappa_P \left(\sum_{k\neq i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S\right)} - \frac{1}{h_{ii}^P}.$$

Therefore, if $\mathbf{p} \le \mathbf{p}'$, then $\sum_{k\neq i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S \le \sum_{k\neq i}^{M} p_j'^P + \sum_{k=1}^{N} p_k'^S$, and eventually $r(\mathbf{p}) \ge r(\mathbf{p}')$.

Type II scalability: for all $\omega > 1$, $r(\omega\mathbf{p}) \ge (1/\omega) r(\mathbf{p})$.

$$r_i^P(\omega\mathbf{p}) = \frac{\alpha_P}{\ln 2} \cdot \frac{1}{\left(\eta_P \sum_{k\neq i}^{M} h_{ik}^P e\left(c_k^P, c_i^P\right) + \lambda_P \sum_{k=1}^{N} h_{ik}^P e\left(c_k^P, c_i^S\right)\right) + \kappa_P \left(\omega\sum_{k\neq i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S\right)} - \frac{1}{h_{ii}^P}$$

$$> \frac{\alpha_P}{\ln 2} \cdot \frac{1}{\omega\eta_P \sum_{k\neq i}^{M} h_{ik}^P e\left(c_k^P, c_i^P\right) + \omega\lambda_P \sum_{k=1}^{N} h_{ik}^P e\left(c_k^P, c_i^S\right) + \omega\kappa_P \left(\sum_{k\neq i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S\right)} - \frac{1}{\omega h_{ii}^P}$$

Therefore, the best-response correspondence is a type II standard function. Recalling *Definition 3*, it can be concluded that the NE obtained by the best-response correspondence is unique [14].

Uniqueness of NE for the SU game. The uniqueness of NE for the SU game $\Gamma^S = \left\{N, \left\{A_i^S\right\}_{i\in N}, \left\{u_i^S\right\}_{i\in N}\right\}$ can be shown by following the same line as it was done for the PU game. ■

In this section it was shown that the PU and SU games have unique NE. In the next section, the combined game is considered and existence of the NE is demonstrated.

E. Existence of Nash Equilibrium - Exact Potential Function

In this section, the combined PU and SU game is considered, and the existence of NE for the combined game is shown. This is done by proving that the combined game is a potential game. A potential game [11] is a normal game such that any change in the utility function of a player caused by a unilateral change in the player's strategy, is reflected in a potential function which represents the aggregated payoff of all the players. In other words, any single player's interest in increasing its own payoff is aligned with the group's interest. Furthermore, Hofbauer and Sandholm in [12] presented that "in potential games, the aggregate payoff function (i.e. potential function) serves as a Lyapunov function for a broad class of evolutionary dynamics."

Definition 4. A game $\Gamma = \left\{N, \left\{A_i\right\}_{i\in N}, \left\{u_i\right\}_{i\in N}\right\}$ is a *potential game* [11] if there is a potential function $V : A \to \mathbb{R}$ such that one of the following conditions holds. A game satisfying the first condition is an *exact potential game*, and a game satisfying the second condition is an *ordinal potential game*.

$$V(a_i, a_{-i}) - V(a_i', a_{-i}) = u(a_i, a_{-i}) - u(a_i', a_{-i}) \text{ for } \forall\ i \in N,\ a \in A, \text{ and } a_i' \in A_i .$$

$$u(a_i, a_{-i}) > u(a_i', a_{-i}) \Leftrightarrow V(a_i, a_{-i}) > V(a_i', a_{-i}) \text{ for } \forall\ i \in N,\ a \in A, \text{ and } a_i' \in A_i .$$

Theorem 4. The combined game of PU and SU converges to NE for all the players.

Proof. If the game can be shown to be an exact potential game (EPG) with a potential function, then it has been proved that that the game strategy towards the *best response* converges to NE [11].

Consider the following function as the potential function candidate

$$\begin{aligned}
V(a) &= \alpha_P \sum_{i=1}^{M} \log_2\left(1 + p_i^P h_{ii}^P\right) + \alpha_S \sum_{i=1}^{N} \log_2\left(1 + p_i^S h_{ii}^S\right) \\
&\quad - a\beta_P \sum_{j=1}^{M} \left(\sum_{k \neq j}^{M} p_k^P h_{kj}^P e\left(c_k^P, c_j^P\right) + \sum_{k=1}^{N} p_k^S h_{kj}^S e\left(c_k^S, c_j^P\right) \right) - b\beta_S \sum_{j=1}^{N} \left(\sum_{k=1}^{M} p_k^P h_{kj}^P e\left(c_k^P, c_j^S\right) + \sum_{k \neq j}^{N} p_k^S h_{kj}^S e\left(c_k^S, c_j^S\right) \right) \\
&- c \sum_{j=1}^{M} \left(\eta_P \sum_{k \neq j}^{M} p_j^P h_{jk}^P e\left(c_k^P, c_j^P\right) + \lambda_P \sum_{k=1}^{N} p_j^P h_{jk}^P e\left(c_k^P, c_j^S\right) \right) - d \sum_{j=1}^{N} \left(\eta_S \sum_{k=1}^{M} p_j^S h_{jk}^S e\left(c_k^S, c_j^P\right) + \lambda_S \sum_{k \neq j}^{N} p_j^S h_{jk}^S e\left(c_k^S, c_j^S\right) \right) \\
&- e\kappa_P \sum_{j=1}^{M} p_j^P \left(\sum_{k \neq j}^{M} p_k^P + \sum_{k=1}^{N} p_k^S \right) - f\kappa_P \sum_{j=1}^{N} p_j^S \left(\sum_{k=1}^{M} p_k^P + \sum_{k \neq j}^{N} p_k^S \right)
\end{aligned} \tag{11}$$

It can be rewritten as

$$\begin{aligned}
V(a_i, a_{-i}) &= \alpha_P \log_2\left(1 + p_i^P h_{ii}^P\right) \\
&- \left(a\beta_P + c\eta_P\right) \sum_{k \neq i}^{M} p_k^P h_{ki}^P e\left(c_k^P, c_i^P\right) - \left(a\beta_P + d\eta_S\right) \sum_{k=1}^{N} p_k^S h_{ki}^S e\left(c_k^S, c_i^P\right) \\
&- \left(c\eta_P + a\beta_P\right) \sum_{k \neq i}^{M} p_i^P h_{ik}^P e\left(c_i^P, c_k^P\right) - \left(c\lambda_P + b\beta_S\right) \sum_{k=1}^{N} p_i^P h_{ik}^P e\left(c_i^P, c_k^S\right) \\
&- \left(2e\kappa_P\right) p_i^P \sum_{k \neq i}^{M} p_k^P - \left(e\kappa_P + f\kappa_S\right) p_i^P \sum_{k=1}^{N} p_k^S + \alpha_P \sum_{k \neq i}^{M} \log_2\left(1 + p_k^P h_{kk}^P\right) + \alpha_S \sum_{k=1}^{N} \log_2\left(1 + p_k^S h_{kk}^S\right) \\
&- a\beta_P \sum_{j \neq i}^{M} \left(\sum_{k \neq j, k \neq i}^{M} p_k^P h_{kj}^P e\left(c_k^P, c_j^P\right) + \sum_{k=1}^{N} p_k^S h_{kj}^S e\left(c_k^S, c_j^P\right) \right) - b\beta_S \sum_{j=1}^{N} \left(\sum_{k \neq i}^{M} p_k^P h_{kj}^P e\left(c_k^P, c_j^S\right) + \sum_{k \neq j}^{N} p_k^S h_{kj}^S e\left(c_k^S, c_j^S\right) \right) \\
&- c \sum_{j \neq i}^{M} \left(\eta_P \sum_{k \neq j, k \neq i}^{M} p_j^P h_{jk}^P e\left(c_j^P, c_k^P\right) + \lambda_P \sum_{k=1}^{N} p_j^P h_{jk}^P e\left(c_j^P, c_k^S\right) \right) - d \sum_{j=1}^{N} \left(\eta_S \sum_{k \neq i}^{M} p_j^S h_{jk}^S e\left(c_j^S, c_k^P\right) + \lambda_S \sum_{k \neq j}^{N} p_j^S h_{jk}^S e\left(c_j^S, c_k^S\right) \right) \\
&- e\kappa_P \sum_{j \neq i}^{M} p_j^P \left(\sum_{k \neq j, k \neq i}^{M} p_k^P + \sum_{k=1}^{N} p_k^S \right) - f\kappa_S \sum_{j=1}^{N} p_j^S \left(\sum_{k \neq i}^{M} p_k^P + \sum_{k \neq j}^{N} p_k^S \right)
\end{aligned} \tag{12}$$

Therefore, the potential function, $V(a)$, can be written with respect to a PU as

$$V(a_i, a_{-i}) = u_i^P(a_i, a_{-i}) + F^P(a_{-i}) \tag{13}$$

where $F^P(a_{-i})$ is a function independent of the strategy of player i, a_i, as following

$$\begin{aligned}
F^P(a_{-i}) &= \alpha_P \sum_{k\neq i}^{M} \log_2\left(1+ p_k^P h_{kk}^P\right) + \alpha_S \sum_{k=1}^{N} \log_2\left(1+ p_k^S h_{kk}^S\right) - a\beta_P \sum_{j\neq i}^{M}\left(\sum_{k\neq j, k\neq i}^{M} p_k^P h_{kj}^P e\left(c_k^P, c_j^P\right) + \sum_{k=1}^{N} p_k^S h_{kj}^S e\left(c_k^S, c_j^P\right)\right) \\
&-b\beta_S \sum_{j=1}^{N}\left(\sum_{k\neq i}^{M} p_k^P h_{kj}^P e\left(c_k^P, c_j^S\right) + \sum_{k\neq j}^{N} p_k^S h_{kj}^S e\left(c_k^S, c_j^S\right)\right) - c\sum_{j\neq i}^{M}\left(\eta_P \sum_{k\neq j, k\neq i}^{M} p_j^P h_{jk}^P e\left(c_j^P, c_k^P\right) + \lambda_P \sum_{k=1}^{N} p_j^P h_{jk}^P e\left(c_j^P, c_k^S\right)\right) \\
&-d\sum_{j=1}^{N}\left(\eta_S \sum_{k\neq i}^{M} p_j^S h_{jk}^S e\left(c_j^S, c_k^P\right) + \lambda_S \sum_{k\neq j}^{N} p_j^S h_{jk}^S e\left(c_j^S, c_k^S\right)\right) \\
&-e\kappa_P \sum_{j\neq i}^{M} p_j^P \left(\sum_{k\neq j, k\neq i}^{M} p_k^P + \sum_{k=1}^{N} p_k^S\right) - f\kappa_S \sum_{j=1}^{N} p_j^S \left(\sum_{k\neq i}^{M} p_k^P + \sum_{k\neq j}^{N} p_k^S\right)
\end{aligned} \tag{14}$$

where

$$a\beta_P + c\eta_P = \beta_P, \tag{15}$$

$$a\beta_P + d\eta_S = \beta_P, \tag{16}$$

$$a\beta_P + c\eta_P = \eta_P, \tag{17}$$

$$c\lambda_P + b\beta_S = \lambda_P, \tag{18}$$

$$2e\kappa_P = \kappa_P, \text{ and} \tag{19}$$

$$e\kappa_P + f\kappa_S = \kappa_P \tag{20}$$

From (15), (16) and (17) it can be concluded that $\beta_P = \eta_P$, and $c = 1 - a$. From (19), one can conclude $e = \frac{1}{2}$.

Hence, if PU player *i* changes its strategy to a_i', the potential function can be written as

$$V(a_i', a_{-i}) = u_i^P(a_i', a_{-i}) + F^P(a_{-i}) \tag{21}$$

and consequently

$$V(a_i, a_{-i}) - V(a_i', a_{-i}) = u_i^P(a_i, a_{-i}) - u_i^P(a_i', a_{-i}). \tag{22}$$

Therefore, the potential function is an exact potential function for PUs.

Similarly, the potential function, $V(a_i, a_{-i})$, can be written with respect to a SU as

$$V(a_i, a_{-i}) = u_i^S(a_i, a_{-i}) + F^S(a_{-i}) \tag{23}$$

where $F^S(a_{-i})$ is a function independent of the strategy of player *i*, a_i, as following

$$\begin{aligned} F^S(a_{-i}) = {} & \alpha_P \sum_{k=1}^{M} \log_2\left(1 + p_k^P h_{kk}^P\right) + \alpha_S \sum_{k\neq i}^{N} \log_2\left(1 + p_k^S h_{kk}^S\right) - a\beta_P \sum_{j=1}^{M}\left(\sum_{k\neq j}^{M} p_k^P h_{kj}^P e\left(c_k^P, c_j^P\right) + \sum_{k\neq i}^{N} p_k^S h_{kj}^S e\left(c_k^S, c_j^P\right)\right) \\ & - b\beta_S \sum_{j\neq i}^{N}\left(\sum_{k=1}^{M} p_k^P h_{kj}^P e\left(c_k^P, c_j^S\right) + \sum_{k\neq j, k\neq i}^{N} p_k^S h_{kj}^S e\left(c_k^S, c_j^S\right)\right) - c\sum_{j=1}^{M}\left(\eta_P \sum_{k\neq j}^{M} p_j^P h_{jk}^P e\left(c_j^P, c_k^P\right) + \lambda_P \sum_{k\neq i}^{N} p_j^P h_{jk}^P e\left(c_j^P, c_k^S\right)\right) \\ & - d\sum_{j\neq i}^{N}\left(\eta_S \sum_{k=1}^{M} p_j^S h_{jk}^S e\left(c_j^S, c_k^P\right) + \lambda_S \sum_{k\neq j, k\neq i}^{N} p_j^S h_{jk}^S e\left(c_j^S, c_k^S\right)\right) \\ & - e\kappa_P \sum_{j=1}^{M} p_j^P \left(\sum_{k\neq j}^{M} p_k^P + \sum_{k=1, k\neq i}^{N} p_k^S\right) - f\kappa_S \sum_{j\neq i}^{N} p_j^S \left(\sum_{k=1}^{M} p_k^P + \sum_{k\neq j, k\neq i}^{N} p_k^S\right) \end{aligned} \tag{24}$$

where

$$b\beta_S + c\lambda_P = \beta_S, \tag{25}$$

$$b\beta_S + d\lambda_S = \beta_S, \tag{26}$$

$$d\eta_S + a\beta_P = \eta_S, \tag{27}$$

$$d\lambda_S + b\beta_S = \lambda_S, \tag{28}$$

$$2f\kappa_S = \kappa_S, \text{ and} \tag{29}$$

$$e\kappa_P + f\kappa_S = \kappa_S \tag{30}$$

From (25), (26) and (28) it can be concluded that $\beta_S = \lambda_S$, and $d = 1-b$. From (29), one can conclude $f = \frac{1}{2}$.

Hence, if player *i* changes its strategy to a_i', the potential function can be written as

$$V(a_i', a_{-i}) = u_i^S(a_i', a_{-i}) + F^S(a_{-i}) \tag{31}$$

and consequently

$$V(a_i, a_{-i}) - V(a_i', a_{-i}) = u_i^S(a_i, a_{-i}) - u_i^S(a_i', a_{-i}). \tag{32}$$

Therefore, the potential function is an exact potential function for SUs. ■

Note that from (15)-(20) and (25)-(30) it can be concluded that (for $a, b, c, d > 0$)

$$a = b, \tag{33}$$

$$c = d = 1-a, \tag{34}$$

$$e = f = \frac{1}{2}, \tag{35}$$

$$\beta_P = \eta_P = \eta_S, \tag{36}$$

$$\beta_S = \lambda_S = \lambda_P, \text{ and} \tag{37}$$

$$\kappa_P = \kappa_S . \tag{38}$$

Therefore, in order for the game to be exact potential, the utility functions and the potential functions should be as following

$$\begin{aligned} u_i^P(a_i, a_{-i}) = {} & \alpha_P \log_2\left(1 + h_{ii}^P p_i^P\right) \\ & -\beta_P\left(\sum_{k\neq i}^{M} p_k^P h_{ki}^P e\left(c_k^P, c_i^P\right) + \sum_{k=1}^{N} p_k^S h_{ki}^S e\left(c_k^S, c_i^P\right)\right) \\ & -\left(\beta_P \sum_{k\neq i}^{M} p_i^P h_{ik}^P e\left(c_k^P, c_i^P\right) + \beta_S \sum_{k=1}^{N} p_i^P h_{ik}^P e\left(c_k^P, c_i^S\right)\right) \\ & -\kappa p_i^P\left(\sum_{j=1, j\neq i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S\right) \end{aligned}$$

$$\begin{aligned} u_i^S(a_i, a_{-i}) = {} & \alpha_S \log_2\left(1 + p_i^S h_{ii}^S\right) \\ & -\beta_S\left(\sum_{k=1}^{M} p_k^P h_{ki}^P e\left(c_k^P, c_i^S\right) + \sum_{k\neq i}^{N} p_k^S h_{ki}^S e\left(c_k^S, c_i^S\right)\right) \\ & -\left(\beta_P \sum_{k=1}^{M} p_i^S h_{ik}^S e\left(c_k^S, c_i^P\right) + \beta_S \sum_{k\neq i}^{N} p_i^S h_{ik}^S e\left(c_k^S, c_i^S\right)\right) \\ & -\kappa p_i^P\left(\sum_{k\neq i}^{M} p_k^P + \sum_{k=1}^{N} p_k^S\right) \end{aligned}$$

$$\begin{aligned} V(a) = {} & \alpha_P \sum_{i=1}^{M} \log_2\left(1 + p_i^P h_{ii}^P\right) + \alpha_S \sum_{i=1}^{N} \log_2\left(1 + p_i^S h_{ii}^S\right) \\ & -a\beta_P \sum_{j=1}^{M}\left(\sum_{k\neq j}^{M} p_k^P h_{kj}^P e\left(c_k^P, c_j^P\right) + \sum_{k=1}^{N} p_k^S h_{kj}^S e\left(c_k^S, c_j^P\right)\right) \\ & -a\beta_S \sum_{j=1}^{N}\left(\sum_{k=1}^{M} p_k^P h_{kj}^P e\left(c_k^P, c_j^S\right) + \sum_{k\neq j}^{N} p_k^S h_{kj}^S e\left(c_k^S, c_j^S\right)\right) \\ & -(1-a)\sum_{j=1}^{M}\left(\beta_P \sum_{k\neq j}^{M} p_j^P h_{jk}^P e\left(c_k^P, c_j^P\right) + \beta_S \sum_{k=1}^{N} p_j^P h_{jk}^P e\left(c_k^P, c_j^S\right)\right) \\ & -(1-a)\sum_{j=1}^{N}\left(\beta_P \sum_{k=1}^{M} p_j^S h_{jk}^S e\left(c_k^S, c_j^P\right) + \beta_S \sum_{k\neq j}^{N} p_j^S h_{jk}^S e\left(c_k^S, c_j^S\right)\right) \\ & -\frac{\kappa}{2}\sum_{j=1}^{M} p_j^P\left(\sum_{k\neq j}^{M} p_k^P + \sum_{k=1}^{N} p_k^S\right) - \frac{\kappa}{2}\sum_{j=1}^{N} p_j^S\left(\sum_{k=1}^{M} p_k^P + \sum_{k\neq j}^{N} p_k^S\right) \end{aligned}$$

Remark 2. The priorities of the networks are incorporated in the utility functions as the β_P / β_S ratio.

Remark 3. Note that the coefficients β_P and β_S can be chosen such that the term associated with the interference caused by the PUs at SUs be less significant that the term associated with the interference caused by the SUs at PUs. It can be achieved by selecting $\beta_P >> \beta_S$.

Definition 5. *P* is potential for game $\Gamma = \left\{ N, \left\{ A_i \right\}_{i \in N}, \left\{ u_i \right\}_{i \in N} \right\}$ if and only if *P* is differentiable, and [11] $\dfrac{\partial u_i}{\partial a_i} = \dfrac{\partial P}{\partial a_i}$ for all $i \in N$.

Theorem 5. *V* is a potential for the PU game $\Gamma^P = \left\{ M, \left\{ A_i^P \right\}_{i \in M}, \left\{ u_i^P \right\}_{i \in M} \right\}$, and the SU game $\Gamma^S = \left\{ N, \left\{ A_i^S \right\}_{i \in N}, \left\{ u_i^S \right\}_{i \in N} \right\}$.

Proof. Recall (13), where the potential function *V* can be written as

$$V(a_i, a_{-i}) = u_i^P(a_i, a_{-i}) + F^P(a_{-i}) .$$

The first derivative of the potential function will be $\dfrac{\partial V}{\partial p_i^P} = \dfrac{\partial u_i^P}{\partial p_i^P} + \dfrac{\partial F^P}{\partial p_i^P} = \dfrac{\partial u_i^P}{\partial p_i^P} + 0 = \dfrac{\partial u_i^P}{\partial p_i^P}$.

Therefore, *V* is a potential for the PU game. Recalling (23), it can be similarly shown that *V* is a potential for the SU game. ■

Definition 6. A game $\Gamma = \left\{ N, \left\{ A_i \right\}_{i \in N}, \left\{ u_i \right\}_{i \in N} \right\}$ is an exact potential game if and only if [11] $\dfrac{\partial^2 u_i}{\partial a_i \partial a_j} = \dfrac{\partial^2 u_j}{\partial a_i \partial a_j}$ for all $i, j \in N$.

Theorem 6. The PU and SU games are exact potential games.

Proof. For nodes *i* being PU, recall the first derivative, (7), where

$$\frac{\partial u_i^P}{\partial p_i^P} = \frac{\alpha_P}{\ln 2} \cdot \frac{h_{ii}^P}{1 + p_i^P h_{ii}^P} - \beta_P \sum_{k \neq i}^{M} h_{ik}^P e\left(c_i^P, c_k^P\right) - \beta_S \sum_{k=1}^{N} h_{ik}^P e\left(c_i^P, c_k^S\right) - \kappa \left(\sum_{k \neq i}^{M} p_j^P + \sum_{k=1}^{N} p_k^S \right).$$

Then the second derivative with respect to a PU(or SU) node, *j*, will be $\frac{\partial^2 u_i^P}{\partial p_i^P \partial p_j^P} = \frac{\partial^2 u_i^P}{\partial p_i^P \partial p_j^S} = -\kappa$. On the other hand, $\frac{\partial u_j^P}{\partial p_i^P} = -\beta_P h_{ij}^P - \kappa p_j^P$, and $\frac{\partial u_j^P}{\partial p_i^S} = -\beta_P h_{ij}^S e\left(c_i^P, c_j^S\right) - \kappa p_j^P$. Therefore, $\frac{\partial^2 u_j^P}{\partial p_i^P \partial p_j^P} = \frac{\partial^2 u_j^P}{\partial p_i^S \partial p_j^P} = -\kappa$. Consequently, $\frac{\partial^2 u_j^P}{\partial p_i \partial p_j} = \frac{\partial^2 u_i^P}{\partial p_i \partial p_j} = -\kappa$ Similarly, for SU utility function it can be shown that $\frac{\partial^2 u_j^S}{\partial p_i \partial p_j} = \frac{\partial^2 u_i^S}{\partial p_i \partial p_j} = -\kappa$.

Therefore, both PU game and SU game are exact potential games[11] (*Definition 6*). ■

F. Resource Allocation Game with Incomplete Information

In case of incomplete information, the stochastic values of the parameters will appear in the utility and potential function equations. In this case, each parameter with incomplete information will be treated with respect to its probability distribution function, and the exact values of parameters (for the complete information case) in the equations will be replaced with their expected values (for the incomplete information case). Consequently, the game will be a Bayesian game [15], and the equilibrium points will be Bayesian Nash equilibriums. In this section, the existence of Bayesian Nash equilibriums, and also Bayesian potential games and their convergence will be shown.

Definition 7. A Bayesian equilibrium in a game of incomplete information with a finite number of types θ_i for each player *i*, prior distribution *f*, and pure-strategy spaces A_i is a Nash equilibrium of the expanded game in which each player *i*'s space of pure strategies is the set $A_i^{\Theta_i}$ of maps from Θ_i to A_i [15].

Fudenberg and Tirole in [15]showed that the strategy a_i is a Bayesian Nash equilibrium if a_i solves

$$a_i(\theta_i) \in \arg\max_{a_i' \in A_i} \sum_{\theta_{-i}} f(\theta_{-i} \mid \theta_i) u_i\left(a_i', a_{-i}(\theta_{-i}), (\theta_i \mid \theta_{-i})\right). \tag{39}$$

In the following two cases of incomplete information are considered: a) incomplete information about the type of the other nodes, i.e whether they are primary or secondary nodes, b) incomplete information about the channel gain. In the following these two cases are studied.

<u>*Incomplete information about the node types.*</u> It is assumed that nodes do not have the precise information about the type of the other nodes; in other words, they do not know whether the other nodes in their neighborhood are PU or SU. Instead, they know the probability of those nodes being PU or SU. Also the total number of the nodes, *K*, is known. Then the utility function for a PU node will be

$$\begin{aligned} u_i^{P,Bayesian}(a_i, a_{-i}) = {} & \alpha_P \log_2\left(1 + h_{ii}^P p_i^P\right) \\ & -\beta_P \sum_{k=1,k\neq i}^{K} \left(f(\theta_k = PU) \cdot p_k^P h_{ki}^P e\left(c_k^P, c_i^P\right) + f(\theta_k = SU) \cdot p_k^S h_{ki}^S e\left(c_k^S, c_i^P\right)\right) \\ & - \sum_{k=1,k\neq i}^{K} \left(\beta_P \cdot f(\theta_k = PU) \cdot p_i^P h_{ik}^P e\left(c_k^P, c_i^P\right) + \beta_S \cdot f(\theta_k = SU) \cdot p_i^P h_{ik}^P e\left(c_k^P, c_i^S\right)\right) \\ & -\kappa p_i^P \sum_{k=1,k\neq i}^{K} \left(f(\theta_k = PU) \cdot p_k^P + f(\theta_k = SU) \cdot p_k^S \right) \end{aligned} \tag{40}$$

Similarly, the utility function for a SU mode will be

$$
\begin{aligned}
u_i^{S,Bayesian}(a_i,a_{-i}) = {} & \alpha_S \log_2\left(1+h_{ii}^S p_i^S\right) \\
& -\beta_S \sum_{k=1,k\neq i}^{K}\left(f\left(\theta_k=PU\right)\cdot p_k^P h_{ki}^P e\left(c_k^P,c_i^S\right)+f\left(\theta_k=SU\right)\cdot p_k^S h_{ki}^S e\left(c_k^S,c_i^S\right)\right) \\
& -\sum_{k=1,k\neq i}^{K}\left(\beta_P\cdot f\left(\theta_k=PU\right)\cdot p_i^S h_{ik}^S e\left(c_i^S,c_k^P\right)+\beta_S\cdot f\left(\theta_k=SU\right)\cdot p_i^S h_{ik}^S e\left(c_i^S,c_k^S\right)\right) \\
& -\kappa p_i^S \sum_{k=1,k\neq i}^{K}\left(f\left(\theta_k=PU\right)\cdot p_k^P + f\left(\theta_k=SU\right)\cdot p_k^S\right). \qquad (41)
\end{aligned}
$$

The PU and SU utility functions can be combined, and a single utility function can be derived as

$$
\begin{aligned}
& u_i^{Bayesian}(a_i,a_{-i}) = f\left(\theta_i=PU\right)\cdot\alpha_P\cdot\log_2\left(1+h_{ii}^P p_i^P\right) \\
& -f\left(\theta_i=PU\right)\cdot\beta_P\sum_{k=1,k\neq i}^{K}\left(f\left(\theta_k=PU\right)\cdot p_k^P h_{ki}^P e\left(c_k^P,c_i^P\right)+f\left(\theta_k=SU\right)\cdot p_k^S h_{ki}^S e\left(c_k^S,c_i^P\right)\right) \\
& -f\left(\theta_i=PU\right)\cdot\sum_{k=1,k\neq i}^{K}\left(\beta_P\cdot f\left(\theta_k=PU\right)\cdot p_i^P h_{ik}^P e\left(c_k^P,c_i^P\right)+\beta_S\cdot f\left(\theta_k=SU\right)\cdot p_i^P h_{ik}^P e\left(c_k^P,c_i^S\right)\right) \\
& -\kappa\cdot f\left(\theta_i=PU\right)\cdot p_i^P\sum_{k=1,k\neq i}^{K}\left(f\left(\theta_k=PU\right)\cdot p_k^P + f\left(\theta_k=SU\right)\cdot p_k^S\right)+f\left(\theta_i=SU\right)\cdot\alpha_S\cdot\log_2\left(1+h_{ii}^S p_i^S\right) \\
& -f\left(\theta_i=SU\right)\cdot\beta_S\sum_{k=1,k\neq i}^{K}\left(f\left(\theta_k=PU\right)\cdot p_k^P h_{ki}^P e\left(c_k^P,c_i^S\right)+f\left(\theta_k=SU\right)\cdot p_k^S h_{ki}^S e\left(c_k^S,c_i^S\right)\right) \\
& -f\left(\theta_i=SU\right)\cdot\sum_{k=1,k\neq i}^{K}\left(\beta_P\cdot f\left(\theta_k=PU\right)\cdot p_i^S h_{ik}^S e\left(c_i^S,c_k^P\right)+\beta_S\cdot f\left(\theta_k=SU\right)\cdot p_i^S h_{ik}^S e\left(c_i^S,c_k^S\right)\right) \\
& -\kappa\cdot f\left(\theta_i=SU\right)\cdot p_i^S\sum_{k=1,k\neq i}^{K}\left(f\left(\theta_k=PU\right)\cdot p_k^P + f\left(\theta_k=SU\right)\cdot p_k^S\right) \qquad (42)
\end{aligned}
$$

Note that (42) can be rewritten as

$$
\begin{aligned}
u_i^{Bayesian}(a_i,a_{-i}) &= f\left(\theta_i=PU\right)\cdot u_i^P(a_i,a_{-i})+f\left(\theta_i=SU\right)\cdot u_i^S(a_i,a_{-i}) \\
&= E\left[u_i(a_i,a_{-i})\right]
\end{aligned}. \qquad (43)
$$

The potential function will be

$$
\begin{aligned}
V^{Bayesian}(a) = &\sum_{j=1}^{K}\Big(f\left(\theta_j = PU\right)\cdot\alpha_P\cdot\log_2\left(1+ p_j^P h_{jj}^P\right)+ f\left(\theta_j = SU\right)\cdot\alpha_S\cdot\log_2\left(1+ p_j^S h_{jj}^S\right)\Big)\\
-a&\sum_{j=1}^{K}\left(f\left(\theta_j = PU\right)\cdot\beta_P\cdot\sum_{k=1,k\neq j}^{K}\left(f\left(\theta_k = PU\right)\cdot p_k^P h_{kj}^P e\left(c_k^P, c_j^P\right)+ f\left(\theta_k = SU\right)\cdot p_k^S h_{kj}^S e\left(c_k^S, c_j^P\right)\right)\right)\\
-a&\sum_{j=1}^{K}\left(f\left(\theta_j = SU\right)\cdot\beta_S\cdot\sum_{k=1,k\neq j}^{K}\left(f\left(\theta_k = SU\right)\cdot p_k^P h_{kj}^P e\left(c_k^P, c_j^S\right)+ f\left(\theta_k = SU\right)\cdot p_k^S h_{kj}^S e\left(c_k^S, c_j^S\right)\right)\right)\\
-(1-a)&\sum_{j=1}^{K}\left(f\left(\theta_j = PU\right)\cdot\sum_{k=1,k\neq j}^{K}\left(\beta_P\cdot f\left(\theta_k = PU\right)\cdot p_j^P h_{jk}^P e\left(c_k^P, c_j^P\right)+\beta_S\cdot f\left(\theta_k = SU\right)\cdot p_j^P h_{jk}^P e\left(c_k^P, c_j^S\right)\right)\right)\\
-(1-a)&\sum_{j=1}^{K}\left(f\left(\theta_j = SU\right)\cdot\sum_{k=1,k\neq j}^{K}\left(\beta_P\cdot f\left(\theta_k = PU\right)\cdot p_j^S h_{ik}^S e\left(c_j^S, c_k^P\right)+\beta_S\cdot f\left(\theta_k = SU\right)\cdot p_j^S h_{jk}^S e\left(c_j^S, c_k^S\right)\right)\right)\\
-\frac{\kappa}{2}&\sum_{j=1}^{K}\left(f\left(\theta_j = PU\right)\cdot p_j^P\cdot\sum_{k=1,k\neq j}^{K}\left(f\left(\theta_k = PU\right)\cdot p_k^P + f\left(\theta_k = SU\right)\cdot p_k^S\right)\right)\\
-\frac{\kappa}{2}&\sum_{j=1}^{K}\left(f\left(\theta_j = SU\right).p_j^S\cdot\sum_{k=1,k\neq j}^{K}\left(f\left(\theta_k = PU\right)\cdot p_k^P + f\left(\theta_k = SU\right)\cdot p_k^S\right)\right)
\end{aligned}
\tag{44}
$$

Similarly, the Bayesian potential function can be rewritten as the expected value of the potential function in the game with "complete information", *i.e.*

$$V^{Bayesian}(a) = E\left[V(a)\right]. \tag{45}$$

Remark 4. In [15] it was proven that the existence of a Bayesian equilibrium is an immediate consequence of the Nash existence theorem. Theorems *5, 6* and *7* proved that the potential game converges to a Nash Equilibrium when the best response correspondence is fetched. Consequently, it can be concluded that the Bayesian potential game does have a Bayesian Nash equilibrium that maximizes the Bayesian utility function presented in (42). It must be noted that the Bayesian utility function is the expected value of the utility function in the complete information game.

Incomplete information about the channel gain. In this case, it is assumed that there is no complete information about the channel gains. Instead it is assumed that the probability distribution function of the channel gain is provided. This can be the case for fading channels; for instance in Rayleigh fading channels, the channel gain has a Rayleigh distribution, but the gain varies over time, and depends on several factors that might make it difficult for the nodes to measure it. Then the utility function for a PU node will be

$$\begin{aligned} u_i^{P,Bayesian}(a_i,a_{-i}) = {} & \alpha_P \log_2\left(1+p_i^P \int h f_{h_{ii}^P}(h)dh\right) \\ & -\beta_P\left(\sum_{k\neq i}^{M} p_k^P\left(\int h f_{h_{ki}^P}(h)dh\right)e\left(c_k^P,c_i^P\right)+\sum_{k=1}^{N} p_k^S\left(\int h f_{h_{ki}^S}(h)dh\right)e\left(c_k^S,c_i^P\right)\right) \\ & -p_i^P\left(\beta_P\sum_{k\neq i}^{M}\left(\int h f_{h_{ik}^P}(h)dh\right)e\left(c_k^P,c_i^P\right)+\beta_S\sum_{k=1}^{N}\left(\int h f_{h_{ik}^P}(h)dh\right)e\left(c_k^P,c_i^S\right)\right) \\ & -\kappa\, p_i^P\left(\sum_{j=1,j\neq i}^{M} p_j^P+\sum_{k=1}^{N} p_k^S\right) \end{aligned} \tag{46}$$

where $f_{h_{ij}^P}(h)$ and $f_{h_{ij}^S}(h)$ are the *pdf*s of the channel gains between primary node *i* and node *j*, and secondary node *i* and node *j*, respectively. Recalling that $E[x]=\int x f(x)dx$, (46) can be rewritten as

$$\begin{aligned} u_i^{P,Bayesian}(a_i,a_{-i}) = {} & \alpha_P \log_2\left(1+p_i^P E\left[h_{ii}^P\right]\right) \\ & -\beta_P\left(\sum_{k\neq i}^{M} p_k^P\left(E\left[h_{ki}^P\right]\right)e\left(c_k^P,c_i^P\right)+\sum_{k=1}^{N} p_k^S\left(E\left[h_{ki}^S\right]\right)e\left(c_k^S,c_i^P\right)\right) \\ & -p_i^P\left(\beta_P\sum_{k\neq i}^{M}\left(E\left[h_{ik}^P\right]\right)e\left(c_k^P,c_i^P\right)+\beta_S\sum_{k=1}^{N}\left(E\left[h_{ik}^P\right]\right)e\left(c_k^P,c_i^S\right)\right) \end{aligned}$$

$$-\kappa\ p_i^P\left(\sum_{j=1,\,j\neq i}^{M} p_j^P+\sum_{k=1}^{N} p_k^S\right)$$
$$=E\left[u_i^P(a_i,a_{-i})\right] \tag{47}$$

Similarly, the Bayesian utility function for the secondary users, and the Bayesian Potential function can be derived as

$$\begin{aligned}u_i^{S,Bayesian}(a_i,a_{-i}) &= \alpha_S\log_2\left(1+p_i^S E\left[h_{ii}^S\right]\right)\\ &-\beta_S\left(\sum_{k=1}^{M} p_k^P\left(E\left[h_{ki}^P\right]\right)e\left(c_k^P,c_i^S\right)+\sum_{k\neq i}^{N} p_k^S\left(E\left[h_{ki}^S\right]\right)e\left(c_k^S,c_i^S\right)\right)\\ &-p_i^S\left(\beta_P\sum_{k=1}^{M}\left(E\left[h_{ik}^S\right]\right)e\left(c_k^P,c_i^S\right)+\beta_S\sum_{k\neq i}^{N}\left(E\left[h_{ik}^S\right]\right)e\left(c_k^S,c_i^S\right)\right)\\ &-\kappa\ p_i^S\left(\sum_{j=1}^{M} p_j^P+\sum_{k=1\,j\backslash k\neq i}^{N} p_k^S\right)\\ &=E\left[u_i^S(a_i,a_{-i})\right]\end{aligned} \tag{48}$$

$$\begin{aligned}V^{Bayesian}(a) &= \alpha_P\sum_{i=1}^{M}\log_2\left(1+p_i^P E\left[h_{ii}^P\right]\right)+\alpha_S\sum_{i=1}^{N}\log_2\left(1+p_i^S E\left[h_{ii}^S\right]\right)\\ &-a\beta_P\sum_{j=1}^{M}\left(\sum_{k\neq j}^{M} p_k^P\left(E\left[h_{kj}^P\right]\right)e\left(c_k^P,c_j^P\right)+\sum_{k=1}^{N} p_k^S\left(E\left[h_{kj}^S\right]\right)e\left(c_k^S,c_j^P\right)\right)\\ &-a\beta_S\sum_{j=1}^{N}\left(\sum_{k=1}^{M} p_k^P\left(E\left[h_{kj}^P\right]\right)e\left(c_k^P,c_j^S\right)+\sum_{k\neq j}^{N} p_k^S\left(E\left[h_{kj}^S\right]\right)e\left(c_k^S,c_j^S\right)\right)\\ &-(1-a)\sum_{j=1}^{M}\left(\beta_P\sum_{k\neq j}^{M} p_j^P\left(E\left[h_{jk}^P\right]\right)e\left(c_k^P,c_j^P\right)+\beta_S\sum_{k=1}^{N} p_j^P\left(E\left[h_{jk}^P\right]\right)e\left(c_k^P,c_j^S\right)\right)\\ &-(1-a)\sum_{j=1}^{N}\left(\beta_P\sum_{k=1}^{M} p_j^S\left(E\left[h_{jk}^S\right]\right)e\left(c_k^S,c_j^P\right)+\beta_S\sum_{k\neq j}^{N} p_j^S\left(E\left[h_{jk}^S\right]\right)e\left(c_k^S,c_j^S\right)\right)\\ &-\frac{\kappa}{2}\sum_{j=1}^{M} p_j^P\left(\sum_{k\neq j}^{M} p_k^P+\sum_{k=1}^{N} p_k^S\right)-\frac{\kappa}{2}\sum_{j=1}^{N} p_j^S\left(\sum_{k=1}^{M} p_k^P+\sum_{k\neq j}^{N} p_k^S\right)\\ &=E\left[V(a)\right]\end{aligned} \tag{49}$$

Recalling *Remark 4*, it can be concluded that the Bayesian game for the case of incomplete information about channel gains have a unique NE and the Bayesian potential game converges to a Nash Equilibrium when the best response correspondence for the PU and SU nodes are fetched.

IV. Simulation Results

In this section, simulation results for a network of 5 PU nodes and 30 SU nodes (Fig. 2) are presented. We assume that the transmission power at the nodes can vary in the range [0,P_{max}] mW. The initial transmission power at the PU and SU nodes were selected as uniformly random numbers between 0 and P_{max}. The weighting factors were considered as following: $\alpha_p = 400$, $\beta_P = 10$, $\alpha_S = 370$, $\beta_S = 1$, $\kappa = 1/(N+M)\cdot P_{\max}$, P_{max} = 1 mW (or 0 dBm), and $a = 2$.

Fig. 3 illustrates the initial utility at each of the PU and SU nodes, based on the randomly assigned initial transmission powers.

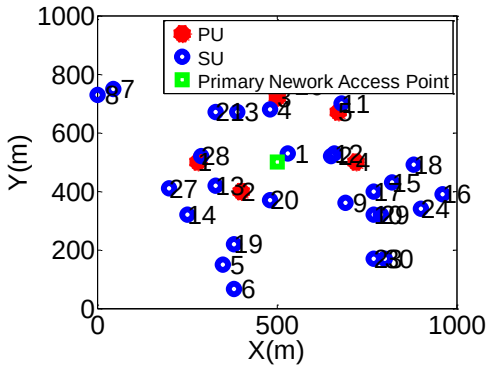

Fig. 2. Network topology.

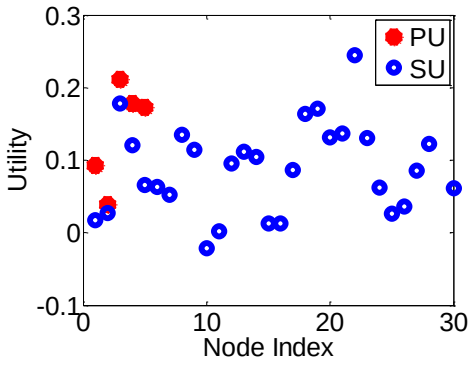

Fig. 3. Initial utility at each node.

Fig. 4 shows the transmission power and utility at the PU nodes and five SU nodes as the game is played, and the players update their transmission powers using (10). Though there are 30 SU nodes, for the sake of brevity, the results for 5 nodes are presented. It can be noticed how each PU and SU player enhances its own utility function by selecting the proper transmission power. Note that since it is chosen $\beta_P = 10\beta_S$, the PU nodes care less about the interference that they cause on the SU nodes.

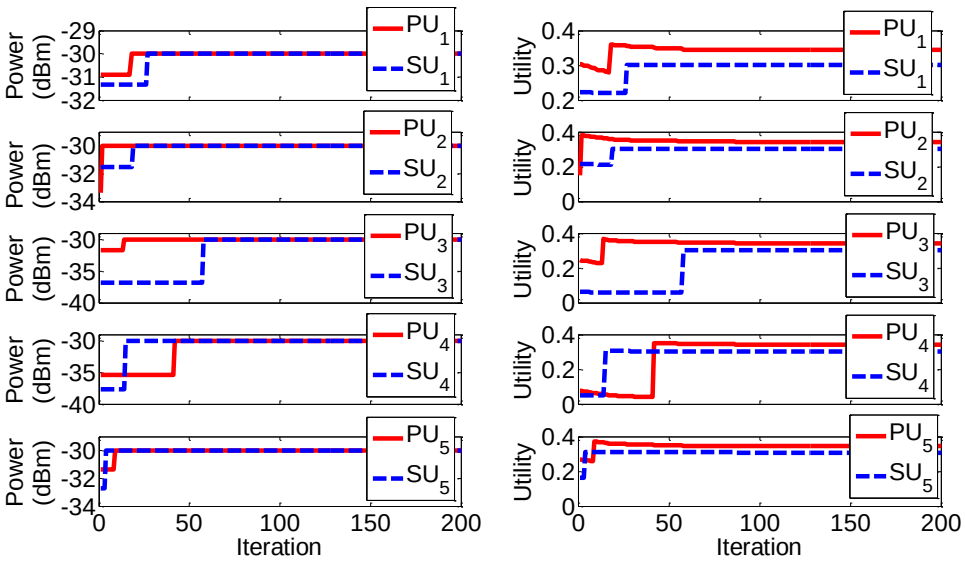

Fig. 4. a) Transmission power and b) utility at PU nodes and five SU nodes.

Fig. 5 illustrates final transmission power and the final utility at each PU and SU node (after 200 iterations, when the nodes have achieved their steady-state transmission power and utility).

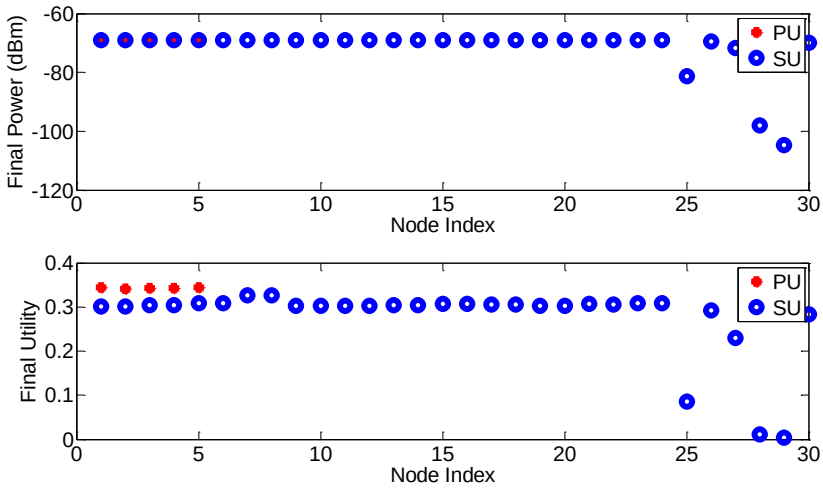

Fig. 5. Final transmission power and final utility at PU and SU nodes ($\beta_P = 10\beta_S$).

Fig. 6 the potential function of the combined PU and SU game. It can be noticed how the potential function converges to the Nash equilibrium as the PU and SU games (Fig. 4) converge to their Nash equilibria.

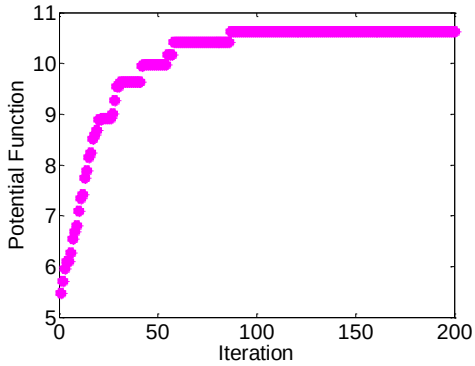

Fig. 6. Potential function, *V*, of the combined PU and SU game.

In the following the effect of the β_P/β_S ratio on the results will be examined. Fig. 7 shows the final utility of the PU and SU nodes when $\beta_P = \beta_S$, $\beta_P = 10\beta_S$ and $\beta_P = 100\beta_S$. The thick markers show the center of the cluster for each case, and the the thin markers represent the utility value for each individual node. The PU nodes in each case achieve very similar values of utility, so the individual points cannot be detected in the figure. Note that $\beta_P = \beta_S$ means that the primary and secondary networks have equal rights in using the spectrum and resources. In other words, the combined game will treat the nodes equally, and the interference that they cause on each other will be penalized equally. It can be noticed that when $\beta_P = \beta_S$, the utility of the PU and SU nodes have achieved the mean steady-state values (center of the cluster) of the same range. As β_P/β_S ratio increases, the gap between the PU and SU utility clusters increases. At $\beta_P = 100\beta_S$, the gap between the SU and PU utility clusters is even more prominent where the PU network has a much greater priority than SU network.

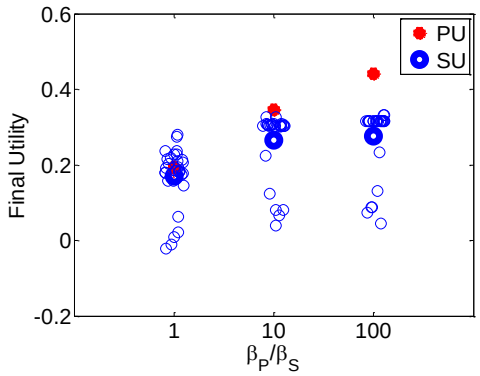

Fig. 7. Final transmission power and final utility at PU and SU nodes for various β_P/β_S.

In the following a case with incomplete information about the channel gains is studied. The same network topology is considered with 5 PU nodes and 30 SU nodes (Fig. 2). It is assumed that the transmission power at the nodes can vary in the range [0,P_{max}] mW. The initial transmission power at the PU and SU nodes were selected as uniformly random numbers between 0 and P_{max}. The weighting factors were considered as following: $\alpha_p = 400$, $\beta_P = 10$, $\alpha_S = 370$, $\beta_S = 1$, $\kappa = 1/\left(N+M\right)\cdot P_{\max}$, P_{max} = 1 mW (or 0 dBm), and $a = 2$. In addition, the channels are assumed to be Rayleigh fading channels with the following distribution

$$f_h(h) = \frac{2h}{\Omega} e^{-h^2/\Omega}, \quad h \geq 0, \tag{50}$$

where $\Omega = E\left[h^2\right]$ is the average of the fading envelope of the Rayleigh distribution. Fig. 8 illustrates the transmission power and utility function of the PU nodes and five SU nodes for the Bayesian game with incomplete information and Rayleigh fading channels. The transmission powers at the nodes are calculated based on the expected value of the channel gain (and not the actual value of the channel gain).

Fig. 9 illustrates the Rayleigh fading channel gain and how it changes over time. The Bayesian potential function is shown in Fig. 10. Note how the Bayesian potential function converges – though it is bursty compared to the potential function in the game with complete information due to the channel uncertainties.

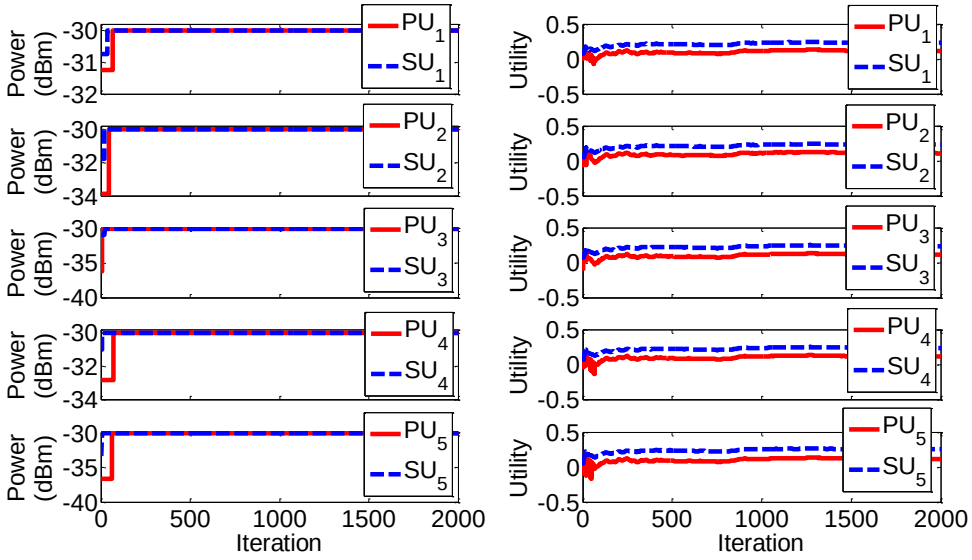

Fig. 8. a) Transmission power and b) utility at PU nodes and five SU nodes for the Bayesian game with incomplete information and Rayleigh fading channel.

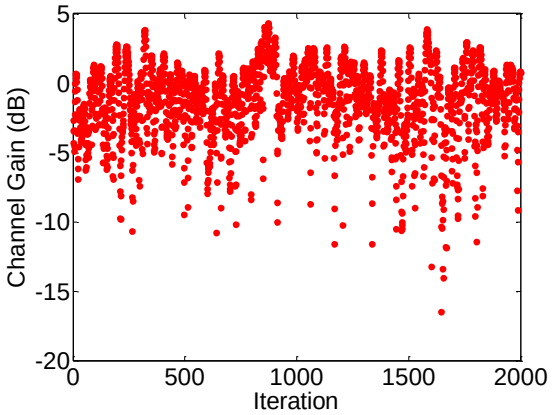

Fig. 9. Samples of the Rayleigh fading channel gain over time.

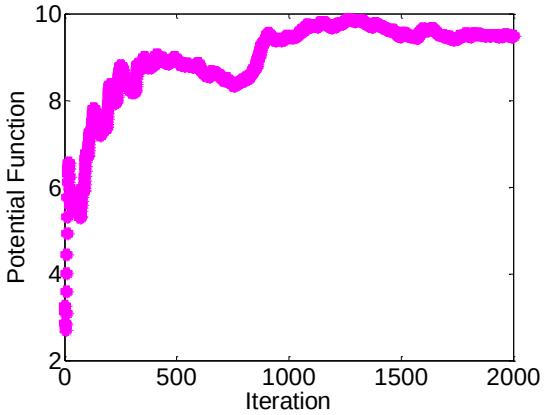

Fig. 10. Bayesian potential function of the combined PU and SU game with incomplete information.

V. CONCLUSION

In this paper we presented a game theoretic approach for resource allocation in primary and secondary users in cognitive networks. The solution takes into account the

priority of the networks in their access to the resources. The existence and uniqueness of the Nash equilibrium for the PU and SU games were proved and then a potential function was presented for the combined PU and SU game. Consequently the existence of the Nash equilibrium for the potential game was shown.

The proposed approach provides the flexibility of adjusting to various priorities among the several co-existing networks, while guaranteeing the existence and uniqueness of Nash equilibrium and consequently achieving it. The simulation results show the convergence of the transmission power and utility for the PU and SU users, and also the convergence of the potential function. They also provide results for various cases of network priorities.

REFERENCES

[1] J. Mitola III., "Cognitive radio: an integrated agent architecture for software defined radio." PhD thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2000.

[2] B. Wang, Y. Wu, and K.J. R. Liu, "Game theory for cognitive radio networks: An overview," *Computer Networks,* (2010), doi:10.1016/j.comnet.2010.04.004.

[3] L. Giupponi and C. Ibars, "Distributed cooperation among cognitive radios with complete and incomplete information," *EURASIP Journal on Advances in Signal Processing*, Article ID 905185, 13 pages, 2009.

[4] R. Menon, A.B. MacKenzie, R.M. Buehrer, and J.H. Reed, "WSN15-4: A game-theoretic framework for interference avoidance in ad hoc networks," *Global Telecommunications Conference, GLOBECOM '06. IEEE* , vol., no., pp.1-6, 2006.

[5] Z. Zhang; J. Shi, H.-H. Chen, M. Guizani, and P. Qiu; , "A cooperation strategy based on Nash bargaining solution in cooperative relay networks," *Vehicular Technology, IEEE Transactions on* , vol.57, no.4, pp.2570-2577, July 2008.

[6] L. Tianming and S.K. Jayaweera, "A novel primary-secondary user power control game for cognitive radios with linear receivers," *Military Communications Conference, 2008. MILCOM 2008. IEEE* , vol., no., pp.1-7, 16-19 Nov. 2008.

[7] C. Ghosh, D.P. Agrawal, and M.B. Rao, "Channel capacity maximization in cooperative cognitive radio networks using game theory," *SIGMOBILE Mob. Comput. Commun. Rev.* 13, 2 (Sep. 2009), 2-13.

[8] C.U. Saraydar, N.B. Mandayam, and D.J. Goodman, "Efficient power control via pricing in wireless data networks," *Communications, IEEE Transactions on* , vol.50, no.2, pp.291-303, Feb 2002.

[9] M.J. Osborne and A. Rubinstein, *A Course in Game Theory*. The MIT Press, 1994.

[10] F. Meshkati, H.V. Poor, S.C. Schwartz, and R.V. Balan, "Energy-efficient resource allocation in wireless networks with quality-of-service constraints," *Communications, IEEE Transactions on* , vol.57, no.11, pp.3406-3414, Nov. 2009.

[11] D. Monderer and L. S. Shapley, "Potential games," Games and Economic Behavior, No. 14, 124-143 (1996).

[12] J. Hofbauerand and W. Sandholm, "On the global convergence of stochasticfictitious play," *Econometrica*, vol. 70, pp. 2265–2294, 2002.

[13] R.D. Yates, "A framework for uplink power control in cellular radio systems," *Selected Areas in Communications, IEEE Journal on* , vol.13, no.7, pp.1341-1347, Sep 1995.

[14] C.W. Sung and K.-K. Leung, "A generalized framework for distributed power control in wireless networks," *Information Theory, IEEE Transactions on* , vol.51, no.7, pp.2625-2635, July 2005.

[15] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, Cambridge, Mass, USA, 1991.

[16] W. Stallings, *Wireless Comunications & Networks.* Prentice Hall, USA, 2005.

2. CONCLUSIONS AND FUTURE WORK

In this dissertation, dynamic approaches were utilized to perform resource allocation in cognitive networks. First, dynamic channel allocation in wireless ad hoc networks was studies and, and then a dynamic adaptive routing protocol for wireless ad hoc networks was proposed. Consequently, the work was expanded to the multi-channel multi-interface networks in the presence of fading channels and channel uncertainties. Finally, the resource allocation in multi-priority co-existing cognitive networks was addressed by proposing a distributed cooperative game-theoretic approach.

In the first paper a distributed dynamic channel allocation algorithm was proposed for wireless networks whose nodes are equipped with single radio interface. The single-radio assumption was made for the sake of simplicity of the network, making it suitable to apply to wireless ad-hoc sensor networks. The periodic nature of the algorithm made it dynamic and enabled the channel allocation to adapt to the topographic changes, possible loss of some channels, mobility of the nodes, and the traffic flow changes. By selecting realistic desired performance metric, the convergence of the algorithm was guaranteed, and analytical proof of convergence was presented. The simulation results for networks of different densities and data channels were provided and showed a significant improvement in throughput, drop rate, energy consumption per packet, fairness index when compared to the single-channel 802.11 and multi-channel 802.11 with randomly allocated multiple channels.

In the second paper, the Bellman's principle of optimality and dynamic programming techniques along with the OLA were utilized to find the routes reactively

and forward-in-time. A model for the network, as well as the state equation and cost function was presented. After finding the initial cost to Base Station at each node (by propagating a beacon from Base Station), the link cost to the one-hop neighbors are updated frequently, and OLA was used to estimate the cost of the next node to Base Station. The boundedness of the OLA error and route cost estimation, and also the convergence of the decision signal to the optimal solution was proven. The advantage of using such a reactive routing protocol that estimates the remaining route cost and selects the next node in real-time is that in case of broken links, node failure or changes in the link quality, there is no need to flood the network with route request messages and/or to wait for a route reply message from the destination (or other nodes that have a route to the destination).

Simulation results for various packet size, data rate, number of flows and average distance of the source node to Base Station verify that the proposed algorithm can improve the throughput and energy efficiency significantly in static and mobile scenarios (compared to AODV). The proposed algorithm appears to outperform multi-channel version of the AODV in higher congestions and longer routes, also for larger packets. As the moving speed of the nodes increases, throughput and energy efficiency of the network decreases. However, the proposed protocol outperforms multi-channel version of AODV in throughput and energy efficiency even in the higher node movement speeds. Network topology changes faster in higher speeds, causing more frequent link failures. In case of link failures, multi-channel AODV sends out new route requests and waits to receive route reply message in order to complete the route repair process. However, the proposed dynamic routing algorithm adapts the routes with respect to the link conditions and finds

the routes in real-time; hence when network topology changes, it responds to the link failures more quickly and more efficiently.

The third paper proposed an adaptive routing protocol for multi-channel multi-interface wireless ad hoc networks in the presence of channel uncertainties. Time-varying outage probability for the channel links were determined by modeling the channel as a Markov model, and subsequently used to estimate the channel condition by MLE. Simulation results in the presence of Rayleigh fading over the wireless channels, and distortion over one channel showed that the effective bandwidth as a function of outage probability can be a more precise estimation of the available bandwidth of the channel.

The time-varying outage probability and channel estimation were utilized in routing and load balancing over multiple links. The fact that time-varying outage probability and channel uncertainties were taken into account in the routing and load balancing, made the protocol sensible to the dynamic nature of the network and channel. Approximate dynamic programming (ADP) techniques are utilized to find dynamic routes, while solving discrete-time Hamilton-Jacobi-Bellman (HJB) equation forward-in-time for route cost. It was shown analytically that when the number of hops increases, the proposed scheme results in near optimal route.

Simulation results in the presence of Rayleigh fading over the wireless channels, and distortion over one channel show that the effective bandwidth as a function of outage probability can be a more precise estimation of the available bandwidth of the channel. Simulation results for various packet size, data rate, number of flows and average distance of the source node to Base Station verified that the proposed algorithm can

significantly improve the throughput and energy efficiency in static and mobile scenarios (compared to AODV with multi-channel multi-interface implementation). In the proposed algorithm, in case of link failure (due to node movements and changes in network topology) or congestion, there is no need to re-discover a new route. Instead, the next node is selected in real time. This specification saves the network from the flooding overhead that can happen in route re-discovery. The proposed algorithm appears to outperform multi-channel AODV in higher congestions, larger packets, and longer routes.

The fourth paper presented a game theoretic approach for resource allocation in primary and secondary users in cognitive networks. The solution takes into account the priority of the networks in their access to the resources. The existence and uniqueness of the Nash equilibrium for the Primary User (PU) and Secondary User (SU) games were proven and then a potential function was presented for the combined PU and SU game. Consequently the existence of the Nash equilibrium for the potential game was shown. Proving that the combined game of PU and SU games is a potential game itself guarantees that as the PU and SU games achieve their NE based on different utility functions, the potential game (which is the combined PU and SU games) achieves its NE. Furthermore, the proposed approach provides the flexibility of adjusting to various priorities among the several co-existing networks. The simulation results show the convergence of the transmission power and utility function for the PU and SU nodes, and also the convergence of the potential function. They also provide results for various cases of network priorities.

As part of the future work, the resource allocation problem in co-existing networks can be extended to a resource allocation game with incomplete information. Although the incomplete information on node type and channel gain were studied in the fourth paper, and addressed by converting the game to Bayesian game, it was assumed that the knowledge of the network nodes about each other was symmetrical. For instance, if a node *i* was aware of the transmission power of node *j*, then node *j*, too, was aware of the transmission power of node *i*. Definition of the potential function was based on this assumption. However, some nodes may deny sharing their information with the other nodes or networks. In this case, the problem can be approached by non-cooperative games in which the nodes try to enhance their QoS and spectrum resources in a selfish manner.

VITA

Behdis Eslamnour was born in Tehran, Iran, on September 14, 1973. In September 1996, she received her B.S. in Electrical Engineering from Sharif University of Technology, Tehran, Iran. In June 2000, she received her M.S. degree in Electrical Engineering from Amirkabir University of Technology, Tehran, Iran. In December 2005, she received her M.S. degree in Computer Engineering from University of Missouri-Rolla (currently Missouri University of Science and Technology), Rolla, Missouri, USA. Behdis received her Ph.D. degree in Computer Engineering from Missouri University of Science and Technology in December 2010.