Honeybee-Inspired Quality Monitoring of Routing Paths in Mobile Ad Hoc Networks

Alexandros Giagkos and Myra S. Wilson

Department of Computer Science, Aberystwyth University,
Aberystwyth, SY23 3DB, UK

Abstract. This paper discusses BeeIP, a reactive multipath routing protocol inspired by honeybees, and examines its performance for both connection-oriented and connectionless traffic within mobile ad hoc networks using a new modification to the algorithm for artificial swarming. Artificial agents follow concepts borrowed from the communication and foraging activities of real honeybees to detect new routing paths and maintain successful and robust data traffic. Paths are evaluated by constantly monitoring their quality based on a list of well-defined low-level parameters. The protocol is compared with the state-of-the-art DSR, AODV and its multipath version AOMDV using four benchmark performance metrics for both TCP and UDP traffic. The results suggest that BeeIP is able to achieve high packet delivery ratio, end-to-end delay and average receiving throughput, while it is shown second best in terms of control overhead for both transport layer protocols.

1 Introduction

Mobile ad hoc networks (MANETs) [9] are prone to the mobility of the nodes and their energy constraints. The former renders centralized control impossible while the latter dramatically affects the performance of the wireless network. Under these conditions, the nodes are expected to discover routing paths and forward data in an adaptive, optimal, self-healing and robust manner.

The characteristics above can be found in nature, in particular, in insect societies such as honeybees [3]. This paper illustrates how adaptive behaviours that are met in the natural system of the hive are applied in order to solve a complex routing problem. It discusses a reactive multipath routing protocol inspired by simple yet effective principles of honeybees, and presents an experimental comparison with state-of-the-art protocols for connection-oriented and connectionless traffic in MANETs.

In the next section a short description of the routing protocols used in this study is given followed by the discussion of BeeIP, the honeybee-inspired protocol, and its important internal mechanisms. The results from the experimental comparison are presented followed by the paper's conclusion and future work.

A.P. del Pobil et al. (Eds.): SAB 2014, LNAI 8575, pp. 300–309, 2014.

2 Reactive Protocols Used for Comparison

Ad hoc On-Demand Distance-Vector protocol (AODV) [10] is a well known reactive protocol for MANETs. In AODV, when the source does not have any previous routing knowledge for a specific destination, route request packets (RREQ) are broadcast to the network. If the RREQ is received by an intermediate node, the node updates its own routing table and broadcasts the RREQ further. The process repeats until the RREQ reaches its final destination, when the destination node responds by unicasting a route reply packet (RREP). The selection of the next hop is made at each node by using the shortest path metric. The reactive nature of AODV allows it to produce less control overhead, which compromises the time it requires to set up a connection when compared to proactive protocols.

AOMDV, a multipath protocol inspired by AODV, accepts duplicate copies of RREQ messages and examines them in order to find alternative reverse paths [13]. Each intermediate node can hold a cache of alternative paths which will be used as next hop in order to reach the appropriate destination. In terms of packet forwarding, AOMDV uses a simple approach; a link is used until it breaks, at which point an alternative is found from the cache. Being a multipath routing protocol, AOMDV is able to reduce both end-to-end delay and packet loss, and is able to utilize the network topology more efficiently as the load is distributed across multiple routes.

The Dynamic Source Routing protocol (DSR) [8] is designed to eliminate the periodic update messages between nodes, thus the bandwidth consumed for this control overhead. It uses source routing described in detail in [11]. A routing entry in DSR contains all intermediate nodes to be visited by a packet, rather than just the next hop information maintained by DSDV or AODV. DSR uses a similar route request mechanism to AODV. However, to reduce the cost of route discovery in terms of control overhead, each node in the network maintains a cache of source routes it has learnt or overheard by the previously incoming RREQ messages (promiscuous mode). The cache is then used aggressively in order to limit the message propagation. However, the major disadvantage of this protocol is that its aggressive use of caches, as well as its inability to locally repair broken links, lead to stale routing information and cache pollution.

3 BeeIP: A Reactive Bee-Inspired Protocol

This section provides an overview of BeeIP, a new multipath routing protocol for MANETs based on honeybees' interactions. More detailed descriptions of the elements of the protocol can be found in [5]. Its model consists of four types of agents (packets); scouts, foragers, ack_scouts and ack_foragers. The key concepts of the protocol are briefly presented below.

3.1 Scouting for Multiple Paths

A scouting process is initialized when a source requires a path to a destination and there is no sufficient routing knowledge available. A scout is broadcast and is propagated until it finally meets the destination, keeping in its header the addresses of the intermediate nodes it visits. When the scout is received by the destination an ack_scout is sent back to the source by following the reverse path of nodes already visited by the received scout. Each intermediate node that receives an ack_scout is not allowed to answer to others of the same scouting generation. Thus BeeIP is designed to find node-disjoint paths only. Node-disjoint paths are more resilient to failures than link-disjoint paths as they protect against both node and link failures. One or more ack_scouts may be sent from the destination allowing multiple paths to be found, each one marked with a unique identification number. While ack_scouts return, they collect data to measure the path's quality. This is an activity similar to the one performed by real honeybees in nature.

3.2 Artificial Foraging: Data Packet Delivery

The source node creates foragers in order to encapsulate real data, received from the transport layer. Each data packet is piggybacked to a forager which, in turn, asks for an appropriate path identification number to its destination. When it is received by the destination node, it delivers the data to the transport layer and converts to an ack_forager. Like the real honeybees, which take some time on the flower to collect the pollen or the nectar, the ack_forager stays at the destination node until some data packet needs to go back to the initial source.

While traversing a path from the destination to the source, the ack_forager collects low-level parameters that represent not only the current state of its senders, but also the network effectiveness of each intermediate link. These parameters are as follows. Firstly, the ack_forager's signal strength at the receiving node (Watts). When an ack_forager is received it carries a signal strength. A weak signal strength is an indication of long distance between the nodes and/or intermediate obstacles that affect the transmission. Secondly, the moving speed of the sender (velocity m/s). A moving node can easily go outside the transmission range and cause broken links. Thirdly, the sender's remaining energy level (Joules). Nodes with sufficient remaining energy are less vulnerable and better candidates for future packet transmissions. The forth parameter is the size of the MAC queue of the sender (bits). The queue size is an indication of how busy the sender is in terms of traffic and network congestion. Finally, the transmission delay between the sender and the receiver of a link in seconds. The use of time-stamps and synchronized clocks allows the measurement of the time an ack_forager requires to complete a transmission from the sender to the receiver of a link.

3.3 Path Quality Monitoring

A new quality value is calculated at every node visited by the ack_forager, and when it finally arrives home, the quality q of the path from the destination d to the source s can be expressed as:

$$Q_{ds} = \sum_{n=1}^{m-1} (q_{N_{n+1} \rightarrow N_n}), \qquad [d = N_m,\ s = N_1] \tag{1}$$

where m is the total number of nodes in a numerically ordered path, and $N_{n+1} \rightarrow N_n$ the pair of nodes with direction towards the source node. The quality q of a link from node j to k as traversed by an agent b is shown in the following equation:

$$\begin{aligned} q_{jk} \;=\; & sig'_b * w_{sig} + \; speed'_j * w_{speed} + \; energy'_j * w_{energy} \\ & + qd'_j * w_{qk} + \; txd'_{jk} * w_{txd} \end{aligned} \tag{2}$$

where the prime numbers are the normalized values of the parameters (sig for signal's strength, etc), and the $w's$ are the appropriate weights. A more detailed explanation of the weighting system can be found in [4].

The result, obtained by Equation (1) is a number that can be used to represent the current quality of the path, in terms of the five low-level parameters. Results from a constant number of previous flights are collected, and based on them, the source is able to investigate whether there has been an improvement or deterioration to the path performance over time. Once there is sufficient amount of data available, the last step of the methodology is to apply regression analysis using Pearson's correlation coefficient [12] to catch any strong positive or negative correlation between the two variables, in this case: time and the quality of a path Q_{ds}. If the correlation is a strong positive, the foraging capacity is increased, whereas if it is a strong negative, the capacity is decreased. The foraging capacity is defined as the number of the remaining foragers for a path, i.e., the number of foragers allowed to use a path in the future.

3.4 Path Selection

The result as calculated by Equation (1), is used to compare each path's quality with its own previous findings, thus detecting improvements and deteriorations over time. Depending on the behaviour of the routing protocol that one may want to achieve, different selection metrics can be applied. On their way back, foragers collect this information and mark each path with a selection metric value. Traditional metric values are related to the number of hops in a path, the transmission speed of its links, the expected transmission count, the energy cost, the remaining energy, etc [1]. For the experimental comparison presented in this paper, a metric related to speed is used; the summation of the (half-round) transmission delay and queueing delay for each intermediate link of the path, from the destination towards the source. This ensures that the fastest path from the list is selected.

3.5 Broken Links Detection

Since BeeIP is designed to evaluate routing based on a 'path' level instead of 'link' level, link breakage within a path is detected when no foragers return to the source node within a period of time. In such a case, the source node sets the path's foraging capacity to 0 and marks the path as unacknowledged. The first ensures that no future foragers will be given the broken path's unique identification number, whereas the latter allows the path to become available again, if a forager eventually returns. Furthermore, internal timers are used to prune any unacknowledged paths. This mechanism ensures that the control overhead remains low as no special messages need to be sent to confirm nodes existence. As the protocol is multipath, being able to switch quickly between routing alternatives allows it to be robust and resilient to bottlenecks and congestion.

3.6 Connectionless Transport Protocols

In TCP, acknowledgement packets, i.e., ack_scouts and ack_foragers are exploited for free, as they piggyback the TCP acknowledgements and fly back. In UDP this mechanism is not available. This problem is addressed by generating swarm packets. A swarm is a control packet that carries the number of foragers that return to the source and is released when one of two criteria is met. Either when the number of waiting foragers has reached a predefined threshold (empirically set to one third of initial colony population) or when a swarming timer has expired. A swarm acts like an ack_forager in that it collects quality information of the traversed path, which eventually affects its the foraging capacity at the source.

4 Experiments and Results

BeeIP is compared with AODV, AOMDV, and DSR. Four performance metrics are used for the comparison of the routing protocols; the packet delivery ratio (PDR), the control overhead (CO), the average end-to-end delay (EED) and the average receiving throughput (RTP) of the communication sessions during the simulation. All experiments have been repeated 10 times using NS2 [7] and the average results are presented. The set up of NS2 for both connection-oriented (TCP) and connectionless (UDP) traffic experiments is given in Table 1.

Table 1. Summary of NS2 configuration

Number of nodes:	50
Terrain size:	1500 x 300 m^2
Simulation time and runs:	900 seconds, 10 runs
Initial energy:	100 to 1500 Joules
Movement model:	Random Waypoint (1 to 10 m/s)
Traffic generators:	FTP/TCP and CBR/UDP (packet size: 512 bytes)
Sending rate (UDP only):	3 packets per second
Active sources:	10 (both), 20 (UDP), and 30 (UDP)
Pause times:	0, 30, 60, 120, 300, 600, and 900
MAC layer:	IEEE 802.11b DCF (queue size 50 packets)
PHY layer:	914MHz Lucent WaveLAN
Transmission range:	150 metres

4.1 Connection-oriented Traffic

Figure 1 shows the PDR (%) as a function of the varying pause times. The error bars show the standard error from the mean. The first observation is that BeeIP shows the best PDR for highly dynamic networks and is rather insensitive to the pause time variations, whereas AODV's performance is slightly decreased as the network loses its dynamic characteristic. AOMDV is found to perform better than its single-path equivalent. On the other hand, due to its aggressive caching, DSR is able to perform better under less stressful situations, i.e., when the topology is less dynamic (pause time increases).

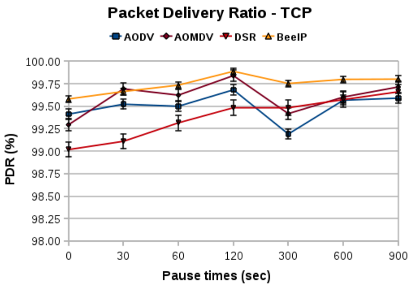

Fig. 1. PDR wrt. pause times

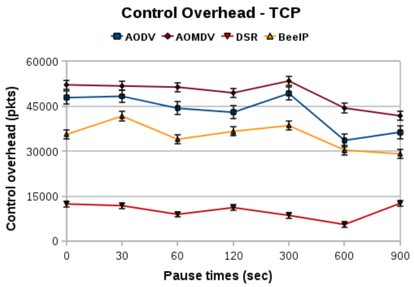

Fig. 2. CO wrt. pause times

Table 2. Route requests for different pause times, between multipath protocols AOMDV and BeeIP

	0	30	60	120	300	600	900
AOMDV:	5022	5219	5231	3854	5690	3542	2566
BeeIP:	848	1555	785	891	1164	746	732

Looking at the CO caused by the four routing protocols in figure 2, it is understood that as nodes lose their mobility, the number of control packets required to maintain routing is also reduced, especially with BeeIP, AODV and AOMDV. Between AODV and AOMDV, the latter is found to produce more CO due to its multiple RREP packets that are sent to support multipath discovery. Nonetheless, BeeIP is still able to keep the CO low, as it looks for node-disjoint paths, utilizing unicast scouting as well as the traditional broadcast. Also, compared to AOMDV, BeeIP is able to use the multiple paths in parallel, which not only distributes the traffic load across the alternative paths, it also mandates less number of route requests (scouting processes) for each communication session. Table 2 summarises the number of route requests incurred during the experiment between the multipath protocols. To conclude, DSR is shown to outperform all other protocols due to its lack of local repair mechanism, which reduces the CO dramatically. However, this strength's trade-off is to be prone to inactive

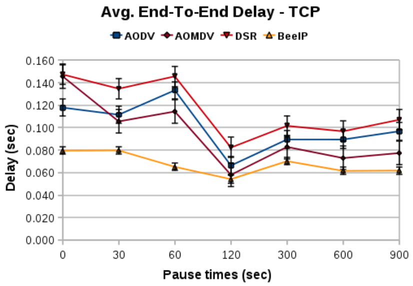

Fig. 3. EED wrt. pause times

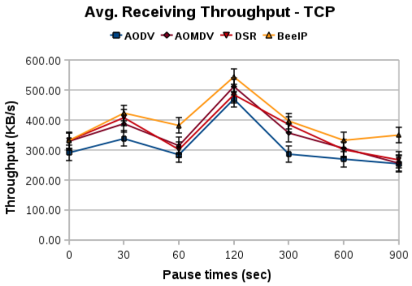

Fig. 4. RTP wrt. pause times

and out-of-date routing entries in cache, which decreases the protocol's PDR (figure 1) and EED (figure 3) for high mobility rates.

Figures 3 and 4 illustrate the two strengths of BeeIP, namely the low EED and average RTP. It is found that BeeIP is able to transfer more KB per second than the others, allowing each data packet to experience lower delay while being forwarded from the source to the destination node. Additionally, the protocol's small error from the mean is another indication that it is not sensitive to the increase of pause times when handling TCP traffic. This is reasoned due to the fact that BeeIP's buffering is always kept to a minimum. Firstly, being multipath the protocol uses more than one path to transmit packets for each session, spreading the data over the topology. Secondly, BeeIP shows balanced CO as fewer (as shown in Table 2) scouts are released. Thus, less packets flood the network and occupy the MAC interface queue of each of the nodes.

4.2 Connectionless Traffic

In this section, the performance of the protocols using connectionless traffic is under investigation. In order to provide a comprehensive comparative study between the protocols, two sets of experiments are presented; altering the pause times of the Random Waypoint movement model, and introducing low, medium and high traffic load to the network by changing the number of CBR active sources. Similar to the connection-oriented set up, 50 mobile nodes move within a 1500 x 300 m^2 terrain for 900 seconds. Their initial energy level is a random number between 100 and 1500. In terms of traffic, 10 CBR active sources are constantly sending data packets of 512 bytes size, with a sending rate of 3 packets per second. Again, pause times of 0, 30, 60, 120, 300, 600 and 900 seconds are used.

Due to the constant bit rate traffic generator the numbers are lower. The results are similar to TCP. In figure 5, BeeIP is shown to achieve better PDR than the other protocols for all pause times. The CO (figure 6) is also shown to be better than AODV and AOMDV, with a deteriorated performance as the nodes tend to fixed positions. DSR is found to have the better CO score.

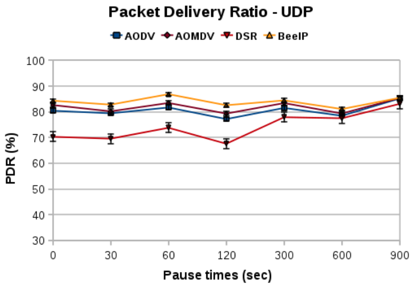

Fig. 5. PDR wrt. pause times

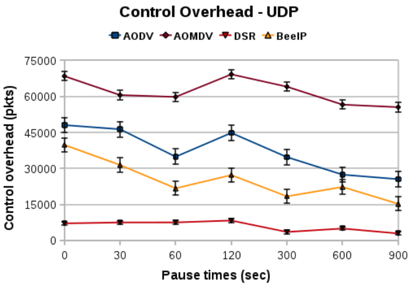

Fig. 6. CO wrt. pause times

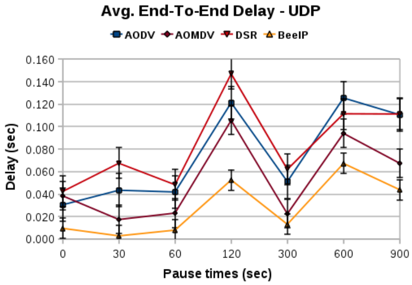

Fig. 7. EED wrt. pause times

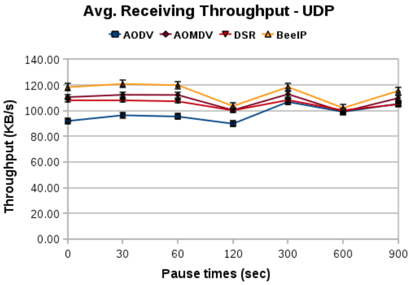

Fig. 8. RTP wrt. pause times

The average EED is proved to be the strength of BeeIP, as illustrated in figure 7 where, due to its fast packet switching mechanism, the proposed protocol outperforms the others. Interestingly, compared to the average EED for FTP/TCP data, the numbers are again very low. The reason for that is the extra overhead and delays TCP experiences because of its internal mechanisms at the transport layer [14]. Moreover, figure 8 summarises the performance of the protocols in terms of the average RTP for different pause times. The results again indicate that BeeIP has a better throughput performance than others, under static or moving nodes.

As mentioned before, in order to evaluate the performance of the protocols under various traffic loads the experiments have been repeated by using 10, 20 and 30 active CBR sources at a time, forming 15, 30 and 50 connections respectively. The pause time is kept to 30 seconds. Affecting the traffic load of the network emphasises the ability of the protocols in handling network congestion.

An initial observation from figure 9 is that as the number of active sources increases, the PDR is dramatically reduced. All protocols face a deterioration achieving higher ~83% (BeeIP) and lower ~69% (DSR) for 10 sources and higher ~39% (BeeIP) and lower ~34% (DSR) for 30 sources, as a result of the high traffic and congestions caused by bottlenecks. BeeIP is found to be competent especially when the number of active sources is 30. Increasing the number of active sources causes the performance of the protocols to follow a reversed pace

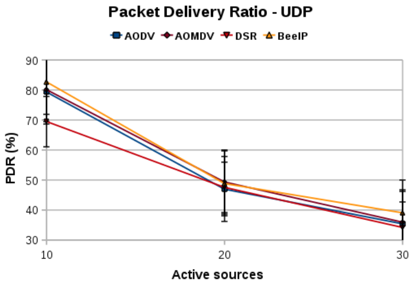

Fig. 9. PDR wrt. no of active sources

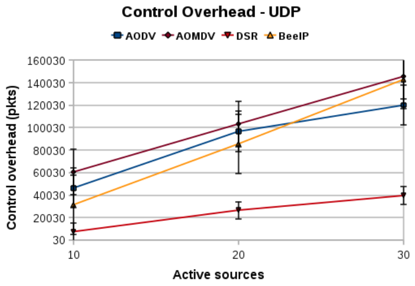

Fig. 10. CO wrt. no of active sources

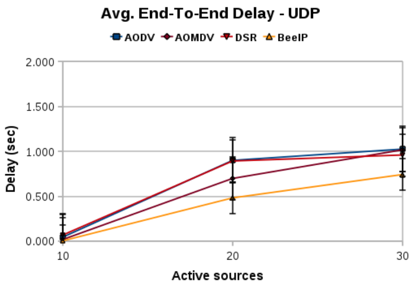

Fig. 11. EED wrt. no of active sources

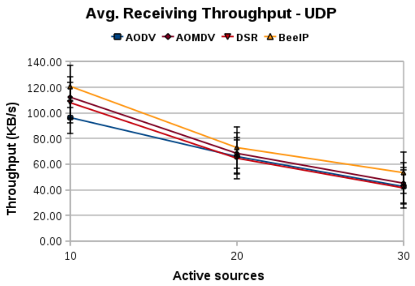

Fig. 12. RTP wrt. no of active sources

in terms of CO; more control packets are sent as more sources send data packets to destinations. This is shown in figure 10, where DSR's design to reduce control packets proves fruitful. Finally, figures 11 and 12 present the average EED and RTP for all protocols. BeeIP is found to outperform the others in terms of delay and transfer more KB per second successfully.

5 Conclusion and Future Work

Adaptive behaviour inspired by honeybees is applied to dynamically discover and maintain routing solutions and forward data packets over wireless paths. BeeIP is discussed and compared to the state-of-the-art routing protocols for both connection-oriented and connectionless traffic. The latter is achieved by artificial swarming, the protocol's latest feature that allows foragers to aggregate and return to the source without the need of acknowledgements. The results indicate that a honeybee-inspired approach is able to outperform the other protocols, in terms of better packet delivery ratio (PDR), the average end-to-end delay (EED) and the average receiving throughput (RTP). The comparison with another nature-inspired and, in particular, bee-inspired routing protocol such as BeeAdHoc [2], would offer extra depth to the merits to BeeIP's design. However, no approved source code has been available in the public domain for the network simulator that was used in this project. Thus, these protocols are not included in this study and are instead seen as future work. Nevertheless, a qualitative comparison with BeeAdHoc is available in [6].

This research was funded by Aberystwyth University Postgraduate Research Fund.

References

1. Campista, M.E.M., Passos, D.G., Esposito, P.M., Moraes, I.M., de Albuquerque, C.V.N., Saade, D.C.M., Rubinstein, M.G., Costa, L.H.M.K., Duarte, O.C.M.B.: Routing metrics and protocols for wireless mesh networks. Computer Communications 22(1), 6–12 (2008)
2. Farooq, M., Pannenbaecker, T., Vogel, B., Mueller, C., Meth, J., Jeruschkat, R.: BeeAdHoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior. In: Proc. GECCO Genetic and evolutionary computation (GECCO 2005), pp. 137–172. ACM, New York (2005)
3. von Frisch, K.: The Dance Language and Orientation of Bees. Oxford University Press (1967)
4. Giagkos, A.: Protocol Design and Implementation for Bee-Inspired Routing in Mobile Ad hoc Networks. Ph.D. thesis, Aberystwyth University (2012)
5. Giagkos, A., Wilson, M.S.: Swarm intelligence to wireless ad hoc networks: adaptive honeybee foraging during communication sessions. Adaptive Behavior 21(6), 501–515 (2013), `http://dx.doi.org/10.1177/1059712313500797`
6. Giagkos, A., Wilson, M.S.: BeeIP - A Swarm Intelligence Based Routing for Wireless Ad Hoc Networks. Information Sciences (2014), `http://dx.doi.org/10.1016/j.ins.2013.12.038`
7. Issariyakul, T., Hossain, E.: Introduction to Network Simulator NS2. Springer (2009)
8. Johnson, D.B., Maltz, D.A.: Dynamic Source Routing in Ad-Hoc Wireless Networks. In: Imielinski, T., Korth, H. (eds.) Mobile Computing, pp. 153–181. Kluwer (1996)
9. Murthy, C.S.R., Manoj, B.S.: Ad Hoc Wireless Networks Architectures and Protocols. Prentice Hall, Upper Saddle River (2004)
10. Perkins, C.E., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing IETF RFC3561 (July 2003), `http://www.ietf.org/rfc/rfc3561.txt`
11. Postel, J.: Internet Protocol - DARPA Inernet Programm, Protocol Specification, RFC 791 (September 1981), `http://www.ietf.org/rfc/rfc791.txt`
12. Read, T.R.C., Cressie, N.: Goodness-of-fit statistics for discrete multivariate data. Springer (1988)
13. Tarique, M., Tepe, K.E., Adibi, S., Erfani, S.: Survey of multipath routing protocols for mobile ad hoc networks. Journal of Network and Computer Applications 32(6), 1125–1143 (2009)
14. Xiao, H., Zhang, Y., Malcolm, J.A., Christianson, B., Chua, K.C.: Modelling and analysis of tcp performance in wireless multihop networks. Wireless Sensor Network 2(7), 493–503 (2010)