Energy Efficient Collision Aware Multipath Routing for Wireless Sensor Networks

Zijian Wang, Eyuphan Bulut, and Boleslaw K. Szymanski

***Abstract*—Multipath routing can reduce the necessity for route updates, balance the traffic load and increase the data transfer rate of wireless sensor networks, improving the utilization of the limited energy of sensor nodes. However, previous methods use flooding for route discovery and transmit data with maximum power even when not needed, which leads to waste of energy. Additionally, often a serious problem of collisions among multiple paths arises. In this paper, we propose an energy efficient and collision aware (EECA) node-disjoint multipath routing algorithm for wireless sensor networks. With the aid of node position information, the EECA algorithm attempts to find two collision-free routes using constrained and power adjusted flooding and then transmits the data with minimum power needed through power control component of the protocol. Our preliminary simulation results show that ECCA algorithm results in good overall performance, saving energy and transferring data efficiently.**

***Index Terms*—energy efficiency, collision awareness, multipath routing, wireless sensor networks**

I. INTRODUCTION

WIRELESS sensors networks typically use batteries for energy supply and often these batteries are non-chargeable. Therefore, energy efficient communication is vital for prolonging the network lifetime. Several papers have addressed this issue by proposing energy efficient routing protocols. Most of them use single optimal path for every communication [1] [2]. However, any single path is vulnerable to node and link failures, especially by depletion of node batteries. In case of such failure, a new route needs to be discovered to maintain data transmission from source to destination, and such route discovery results in extra energy cost.

Using multipath in wireless sensor networks can reduce frequent routing update and enhance data transmission rates. Additionally, it can provide an even distribution of traffic load over the network. This is of great benefit to balancing the energy consumption which is necessary for extending network lifetime. Most of multipath routing protocols are based on classic on-demand single path routing methods [3] [4], such as AODV and DSR. They differ from each other on how to forward multiple route requests and how to select multiple routes. In some papers, node energy is also taken into consideration when constructing multiple paths [5] [6].

All the multipath methods mentioned above suffer from the following problems. First, they flood the route request to the whole network, which creates large communication overhead. Second, each node sends route discovery and data packets with the maximum power, which wastes energy if the recipient can receive the transmission of lower energy. Additionally, when several paths transmit data simultaneously, even if node-disjoint multipaths are used, there exist a probability of collisions, resulting in high packet loss rate and bad data transmission performance [7].

Some papers try to solve some of the above mentioned three problems. In [8], Xu et al. propose an algorithm to restrict the route request flooding to a certain area using the node's location information. Saha et al. in [9] try to find zone-disjoint multipath using directional antenna to avoid collisions between paths. Correlation and coupling metrics are used separately to calculate the relative degree of independence among a set of paths in [10] and [11]. The correlation factor between two node-disjoint paths is defined as the total number of shared links connecting the paths [10]. The coupling between two paths is calculated as the average number of nodes that are blocked from receiving data along one of the paths when a node in the other path is transmitting [11]. Choosing paths that have low correlation or coupling can improve the performance of multipath routing. The algorithm presented in [12] defines a similar correlation factor to weigh the collision probability among node-disjoint multi-paths. Then, it calculates an upper limit for correlation factor according to service requirements. Finally, it finds a minimum energy node-disjoint multipath routs that satisfy that limit. In [13], Hwang et al. define an overhearing ratio that defines the level of energy waste resulting from overhearing transmissions of one path by the other. They use this ratio to establish energy efficient multiple paths.

Manuscript received September 27, 2008. Research was sponsored by US Army Research Laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defence, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Zijian Wang. Author is with the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180 USA. (e-mail: wangz@cs.rpi.edu, phone: 518-944-1156).

Eyuphan Bulut and Boleslaw Szymanski. Co-authors are with the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180 USA. (e-mail: {bulute, szymansk}@cs.rpi.edu).

All the algorithms discussed above deal only with problem one or problem three, but they leave problem two unsolved. Furthermore, to avoid collisions between routes, they use special hardware (e.g. directional antenna) or require more information exchanges to calculate correlation between paths. Consider the algorithm in [13] as an example. The route reply messages carry residual energy and also the neighbor list of all intermediate nodes in the route back to the source node. These messages are big in size and cost energy to transmit.

In this paper, we propose an energy efficient and collision aware (EECA) node-disjoint multipath routing algorithm. The route discovery flooding is restricted within the neighbors of nodes along the discovered route. Each node transmits route discovery messages and data using proper power with the aid of node position information. Additionally, we use the broadcast nature of wireless communication to avoid transmission collisions between two discovered routes.

The remainder of the paper is organized as follows. We describe the network model and our assumptions in Section II. In Section III, we introduce our energy efficient and collision aware node-disjoint multipath routing algorithm. Section IV presents the simulation results. Finally, we provide conclusions and outline the future work in section V.

II. NETWORK MODEL AND ASSUMPTIONS

The sensor network consists of N randomly deployed nodes with uniform distribution over a finite, two-dimensional planar region. Each node can adjust radio transmit power to vary its communication range from 0 to the maximum transmit range, denoted as R. We assume that each node knows its position and also the positions of its neighbors within its transmit range R. These assumptions are satisfied if nodes have access to a low energy GPS and exchange their position information at the network deployment stage. Additionally, we assume that each node knows the position of the destination node. This assumption is immediately satisfied in applications in which the unique sink node's position is known to every node in the network. In other cases, the destination's position information can be obtained through some energy efficient location update methods [14].

III. ENERGY EFFICIENT COLLISION AWARE NODE-DISJOINT MULTIPATH ROUTING ALGORITHM

The EECA is an on-demand routing protocol that builds multiple paths using request/reply cycles. Instead of flooding the route request message to the whole network, it restricts the route discovery flooding to the neighbors of the nodes iteratively added to the route being discovered.

To guarantee no collisions between two routes, each pair of nodes from the two constructed routes have to be apart a certain distance from each other. Of course, if these two routes are apart a distance R, there will be no collisions at all, as is the case for the two green routes shown in Fig. 1. However, these two routes should not be too far away from each other, otherwise long hops and unnecessary energy cost will be incurred, as in the case of the gray route in Fig. 1.

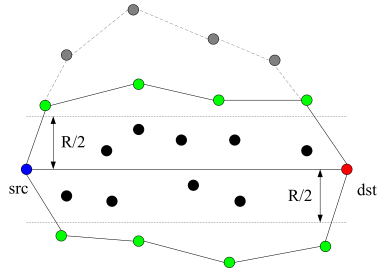

Fig.1 An illustration of collision avoiding routes

We use the broadcast nature of wireless communication to detect potential collisions. If a node overhears a message from a node on other route, it means that there is a potential for collisions between packets sent by these two nodes. Therefore, the overhearing node should not be in any route. By properly adjusting the transmit power of each node on the route, EECA reduces the potential collision area of each node and saves energy. Using adjusted transmit power relaxes also the restriction on distance between two routes allowing it to be smaller than R, which then results in collision-free short hop routes.

A. Route Request

Initial route request at source node

When the source has data to transmit but no route to destination has been established yet, it will start the route request procedure. The source will first check its neighbor list to find out whether there are two groups of neighbors satisfying the following three conditions: 1) all those nodes are closer to the destination; 2) nodes in each group lay at one side of the source-destination line, opposite to the side of the other group; 3) each node is distanced more than R/2 from the source-destination line. If such neighbors are found (the green nodes shown in Fig. 1 are the examples), the source will conclude that there are two potential routes which will at least avoid collisions between the first two nodes. From all eligible nodes, the source will choose the pair of nodes resulting in smallest transmit power which is then used to broadcast route request. The route request message also carries the position information of source and destination nodes and the route request type (an attempt to discover two routes, not just one, in this case).

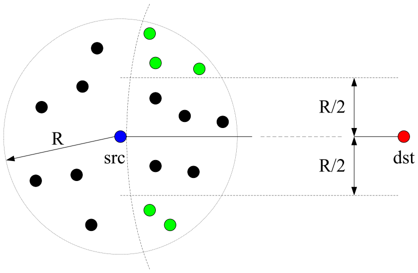

Fig.2 The route request procedure at the source

Route request at intermediate nodes

If an intermediate node receives a route request message from a previous node which is on the same side of the source-destination line as itself and which is further to the destination than itself, it will start a back-off timer for that source-destination route request (we will explain the calculation of the back-off timer later). When this timer expires, the node will broadcast a “local reply” message for that route request with the sufficient power to just reach the previous node. After receiving one “local reply” message for the route request just sent, the previous node will immediately broadcast a “shut up” message with the power used to broadcast route request (if the previous node is the source node, it will wait until receiving two “local reply” messages to broadcast “shut up” message). In the style of computing with time [15], any other node receiving “local reply” or “shut up” message before its timer is over will cancel its timer. Moreover, any node receiving a “local reply” or a “shut up” message from a node on the other side of the source-destination line will not respond to any future requests for this source-destination route. As a result, only one neighbor will win the competition while the route discovery flooding is restricted. Additionally, we use broadcast “local reply” and “shut up” messages to avoid collisions without incurring any additional overhead.

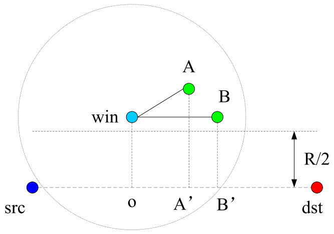

Fig.3 The route request procedure at an intermediate node

```
Algorithm 1 FindProperTransmitPower
  Input: Sorted neighbor list
  Output: Transmit power
  if neighbor_num = 1 then
    return Power(Distance(win_node, list[0].node))
  end if
  for each $0 \le i <$ neighbor_num do
    ratio = $\frac{(list[i+1].pro\_len - list[i].pro\_len)/R}{(list[i+1].tx\_p - list[i].tx\_p)/Power(R)}$
    // i+1 has larger pro_len but uses less power
    if ratio < 0 continue;
    // i+1 increment of power is smaller than its pro_len gain
    if ratio > 1 continue;
    // i+1 increment of power is larger than its pro_len gain
    if 0 < ratio > 1 break;
  end for
  return Power(Distance(win_node, list[i].node))
```

When the winner receives the “shut up” message from the previous node, it will rebroadcast route request message with carefully designed transmit power established as follows. First, the winner refines the neighbor by removing neighbors that are not closer to the destination than itself or not further from the source-destination line than either R/2 or the distance of the winner from this line. Then, for each neighbor on the refined list, the winner computes the distance between the projections of the winner and the neighbor onto the source-destination line. This distance, oA in Fig. 3, is called progress length and abbreviated *pro_len* measures the progress of bringing a packet to the destination, if that neighbor is chosen as the next intermediate node on the route. The refined neighbor list is then sorted in the increasing order of this distance. Finally, the winner calculates also the corresponding transmit power (abbreviated as tx_p) needed to reach each neighbor in the list. The proper transmit power for the rebroadcast route request is determined by Algorithm 1, where function Distance(node$_1$, node$_2$) returns the distance between node$_1$ and node$_2$ and function Power(d) returns the transmit power required for distance d.

Back-off time calculation

Three factors are considered when neighbors receiving route request calculate their back-off times, namely: progress length, distance to the source-destination line and residual energy, as shown in Algorithm 2. “init_energy” is the initial energy of each node. “energy_threshold” is the threshold predefined for the entire network. k_1, k_2 and k_3 are three parameters to balancing the weights of three above mentioned factors and we require that $k_1+k_2+k_3=1$. L is a scaling factor that defines the stretch of back-off time. Hence, the closer the neighbor’s distance to the source-destination line is to R/2, the larger its progress length is, and the higher its residual energy is, the shorter its back-off timer will be, increasing its chances to win the competition for being on the route.

```
Algorithm 2 CalculateBackoffTime
  Input: pro_len, dist_to_line, node_energy
  Output: back-off time
  pro_fac = 1 - pro_len/R
  if dist_to_line < R/2 then
    dist_fac = R/2/dist_to_line*(1 - dist_to_line /(R/2))
  else
    dist_fac = dist_to_line/(R/2)-1
  end if
  energy_fac = 1 - node_energy/init_energy
  if node_energy < energy_threshold then
    energy_fac = init_energy/node_energy*energy_fac
  end if
  $T = (k_1 * pro\_fac)^2 + (k_2 * dist\_fac)^2 + (k_3 * energy\_fac)^2$
  return $\sqrt{T}/L$
```

Route requests failure

The source may fail to discover two collision-free routes for the following reasons:

(1) there is no neighbor pair around the source that satisfies the three requirement, as shown in Fig. 4(a);

(2) a potential winner on one route is shut up by the node in another route, as shown in Fig. 4(b).

(3) there is no candidate neighbor around the winner, as shown in Fig. 4(c).

In case (1), the source will re-broadcast a route request changing the type of request to an attempt to discover one energy efficient route. In case (2) and (3), if at least one route is established, the source will use it. If no route is discovered after waiting for a certain time, the source will try to discover single energy efficient route. In all the above three cases, there may exist two routes with no collisions, but these routes will require many hops and thus cost more energy than a single, efficient route would require. Therefore, we choose to find single energy efficient route in such a case.

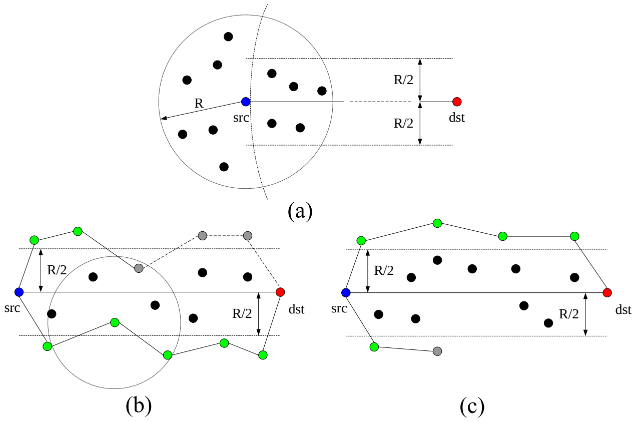

Fig.4 Instances for route request failure

The discovery of a single energy efficient route differs slightly from the discovery of two routes with no collisions. The differences can be summarized as follows. First, the refined neighbor list includes all neighbors that are closer to the destination than the winner and which are closer to the source-destination line than R/2. Moreover, each node receiving the route request, including the source, uses Algorithm 1 to calculate the proper transmit power. Finally, the *dist_factor* computed by Algorithm 2 is now defined as:

dist_factor= dist_to_line /R.

Consequently, the closer to the source-destination line the neighbor is, the larger its progress length is and the higher its residual energy is, the more likely it is to win the competition for being on the constructed route.

B. Route Reply and Data Transmission

When the destination receives the route request, it unicasts the route reply message to the source using the backward route that has been constructed through the route request procedure. The route reply message carries the total residual energy of the route and the identity and the residual energy of the node with the least residual energy on that route. When the route replay arrives at the source, the forward route table is established and the source also stores the total residual energy of the route and the node with the least energy. The source will start transmitting data when once the route is discovered. If there are two routes, the source will use these two routes simultaneously (sending two packets at a time, if available) and adjust the traffic load according to the energy situation of each route. The power for each data transmission is adjusted to the level just sufficient to reach the next hop in the route to save energy and reduce collisions and interference.

IV. SIMULATION

We used NS-2.33 simulator to evaluate the proposed scheme in terms of the average packet delivery ratio, the average end-to-end delay, the average residual energy and the number of nodes alive.

The simulated network is composed of 100 static nodes deployed uniformly randomly within a 1000 m by 1000 m area. IEEE 802.11 is used as the MAC and physical layer protocol. We used two-ray-ground propagation model and each node's maximum transmit range is set to 250 m. The power drained for each transmission is 1.6 W for omni-directional transmit range of 250 m and varies with the transmit range. The power drained for reception is constant and equal to 1.2 W. Sources send CBR (continuous bit-rate) traffic over the connections that are spread randomly over the network. All packets are of the same size of 512 bytes. The number of connections varies from 10 to 30 with the increment of 10 and each connection stays up for a duration needed for sending of 300 packets by the source. For each traffic model, 10 network topologies are generated randomly. Each node has the initial energy of 30 joules and the "energy_threshold" is set to 5 joules through the network. k_1, k_2 and k_3 are all set to 1/3 and L equals to 20.

We compared EECA with AODV protocol and with runs simulating 300 sec of network life. All the results are based on 10 runs with the identical traffic model and network topology.

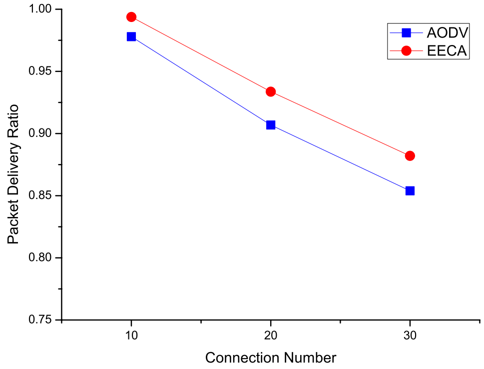

Fig.5 The packet delivery ratio

Fig. 5 shows the packet delivery ratio of EECA and AODV for 10, 20 and 30 CBR connections. As expected, the packet delivery ratio goes down for both protocols when the number of CBR connections increases. However, EECA loses fewer packets than AODV (1.5% to 3% less) in all the cases. That is because EECA tries to decrease the possibility of data collisions by varying the transmit power and splitting data into two collision-free routes.

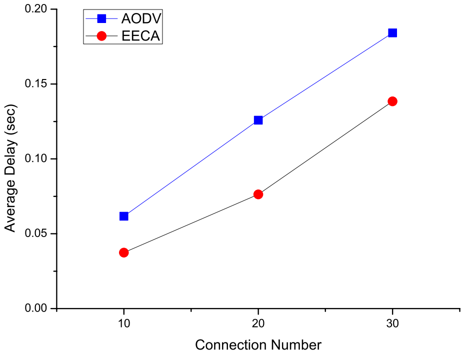

Fig.6 The average end-to-end delay

The average end-to-end delays for both EECA and AODV are shown in Fig. 6. The EECA gets better result thanks to using two routes simultaneously which increases the bandwidth of the route. Additionally, EECA also benefits from reduction of the probability of data retransmission caused by collisions.

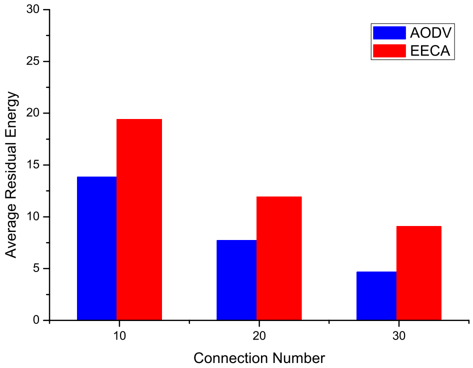

Fig.7 The average residual energy

Fig. 7 shows the average residual energy of nodes in the network after simulation. It is clear that EECA leaves the network with much higher residual energy than AODV does (around 40% to 90%). This result demonstrates that EECA leads to lower total energy consumption in communication than AODV and most likely other traditional protocols do. This benefit is achieved from two EECA properties. First, EECA transmits data using the minimum power needed to reach the next hop. Second, EECA also restricts the route request procedure in the route discovery phase which can be very costly in terms of energy used.

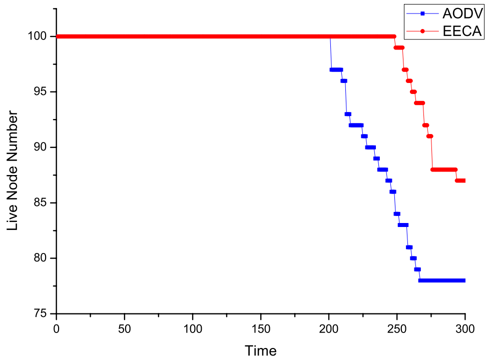

Fig.8 An instance of the number of nodes alive

We also compared the number of nodes alive at the end of execution for these two protocols. Fig. 8 shows the result for one instance of simulation with 30 CBR connections under certain network topology. Obviously, EECA can prolong the network connectivity longer than AODV. There are two reasons for this outcome. First, the total energy consumption is reduced in EECA. Second, EECA distributes the traffic load to multiple paths.

V. CONCLUSION AND FUTURE WORK

In this paper, we introduce an energy efficient and collision aware (EECA) node-disjoint multipath routing algorithm. The main idea of EECA is to use the broadcast nature of wireless communication to avoid collisions between two discovered routes without extra overhead. Additionally, EECA restricts the route discovery flooding and adjusts node transmit power with the aid of node position information, resulting in energy efficiency and good performance of communication. We have studied the performance of EECA protocol relative to AODV under a group of network topologies and traffic scenarios. We observed that EECA achieved better performance in energy conservation and data transfer efficiency in all cases.

In the future, we will further study setting of some parameters in EECA to understand their influence on the protocol performance. We also plan to apply EECA to mobile wireless networks and try to relax the assumption that all nodes know the destination position.

REFERENCES

[1] I. Stojmenovic and X. Lin, “Power aware localized routing in wireless networks,” IEEE Trans. Parallel and Distrib. Sys., 2001, 12(11): 1122-33.
[2] C.K. Toh, “Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks,” IEEE Commun. Mag., 2001, 39(6): 138-147.
[3] M.K. Marina, and S.R. Das, “On-demand multipath distance vector routing in ad hoc networks,” Proc. 9th IEEE Int. Conf. Network Protocols (ICNP), 2001, pp. 14-23.
[4] S.J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," Proc. IEEE Int. Con. Communications (ICC), 2001, vol.10, pp. 3201-3205.
[5] A.P. Subramanian, A.J. Anto, J. Vasudevan, and P. Narayanasamy , “Multipath power sensitive routing protocol for mobile ad hoc networks”, Proc. Conf. Wireless on Demand Network Systems, 2004, LNCS 2928, 2004, pp. 171-183.
[6] L. He, “Energy-efficient multi-path routing with short latency and low overhead for wireless sensor networks,” Proc. ACIS Int. Conf. Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007, Volume 3, pp. 161 - 167.
[7] M.R. Pearlman, Z.J. Haas, P. Sholander, and S.S. Tabrizi, "On the impact of alternate path routing for load balancing in mobile ad hoc networks," Proc. ACM Int. Symp. Mobile Ad Hoc Networking & Computing, 2000, pp. 3-10.
[8] X. Lin and I. Stojmenovic, "Location-based localized alternate, disjoint and multi-path routing algorithms for wireless networks," J. Parallel and Distributed Computing, 2003, 63(1):22-32.
[9] D. Saha, S. Toy, S. Bandyopadhyay, T. Ueda, and S. Tanaka, "An adaptive framework for multipath routing via maximally zone-disjoint shortest paths in ad hoc wireless networks with directional antenna," Proc. Global Telecommunications Conf., 2003.
[10] K. Wu, and J. Harms, "Performance study of a multipath routing method for wireless mobile ad hoc networks," Proc. Symp. Modeling, Analysis and Simulation on Computer and Telecommunication Systems (MASCOTS), 2001, pp. 99-107.
[11] M.R. Pearlman and Z.J. Haas, P. Sholander, and S.S. Tabrizi, "On the impact of alternate path routing for load balancing in mobile ad hoc networks," Proc. ACM MobiHoc, 2000, pp. 3-10.
[12] M. Liu, Z. Xu, J. Yang, and J. Ye, "Collision-constrained minimum energy node-disjoint multipath routing in ad hoc networks," Proc. Int. Conf. Wireless Communications, Networking and Mobile Computing, 2006, pp. 1-5.
[13] D.-Y. Hwang, E.-H. Kwon, and J.-S. Lim, "EASR: an energy aware source routing with disjoint multipath selection for energy-efficient multihop wireless ad hoc networks," Proc. Conf. Networking 2006, LNCS 3976, 2006, pp. 41-50.
[14] I. Stojmenovic, “A Routing Strategy and Quorum Based Location Update Scheme for Ad Hoc Wireless Networks,” Technical Report TR-99-09, Computer Science Dept., SITE, Univ. of Ottawa, Sept. 1999.
[15] B.K. Szymanski and G. Chen, “Computing with Time: from neural networks to wireless networks,” Computer Journal, 2008, 51(4):511-522.