Resilient Data Collection in Mobile-assisted Wireless Sensor Networks

Baris Tas and Ali Şaman Tosun
Department of Computer Science
University of Texas at San Antonio
San Antonio, TX 78249
email{btas,tosun}@cs.utsa.edu

ABSTRACT

Mobility facilitates efficient data collection protocols improving the performance, scalability and life-time of wireless sensor networks. We propose a simple, yet effective and scalable method for resilient data collection in mobile-assisted wireless sensor networks. The mobile element covers an area using periodic long-range broadcast messages. Upon receiving a broadcast message, a sensor sends its data to the mobile element using trajectory routing in a multi-hop manner. The mobile element includes sensor acknowledgements in the broadcast using a Bloom filter. If a packet is not received by the mobile element due to an erroneous node along the trajectory, a different trajectory is used to avoid malicious nodes. Simulation results demonstrate that low number of broadcasts is enough to collect data from a large-scale network with over 99% success rate if the system parameters are set properly.

1. INTRODUCTION

To improve the scalability and performance of WSNs, there has been a flurry of work on employing a mobile node for data collection. The data mules [12] work exploit random movement of mobile node to opportunistically collect data from a sparse WSN. Here, the nodes buffer all their data locally, and upload the data only when the mobile node arrives within direct communication distance. Zebranet [4] system uses tracking collars carried by animals for wildlife tracking. Data is forwarded in a peer-to-peer manner and redundant copies are stored in other nodes. Shared wireless info-station model [13] uses radio tagged whales as part of a biological information acquisition system. Mobility of the mobile node is not controlled in these approaches. Mobile element scheduling (MES) work [14] considers controlled mobility of the mobile node in order to reduce latency and serve the varying data-rates in the WSNs effectively.

WSNs are vulnerable to various attacks due to their nature. In selective forwarding, a sensor on the path from the source to the destination drops forwarding packets [5]. Proposed solutions for detection of the attack include watchdog mechanisms where a node keeps track of its neighbors' behavior [8]. However, this solution depletes sensors' resources quickly. Another scheme which uses acknowledgement from intermediate nodes is proposed in [16]. False forwarding, where a node does not follow the forwarding mechanism precisely, is a type of misrouting attack. A malicious node falsify the routing packets to disrupt the routing tables [6]. In the wormhole attack, an adversary tunnels messages received in one part of the network over a low-latency link and replays them in a different part. An adversary could convince nodes who would normally be multiple hops from a base station that they are only one or two hops away via the wormhole [5]. To defend against wormhole attacks, a leash is added to a packet to restrict the packet's maximum allowed transmission distance [3].

The main components of our data collection protocol include trajectory routing, a cone-based topology control mechanism, and Bloom filter. Trajectory-based routing (TBR) described in [10, 9] is a generalization of source based routing, and cartesian routing. In TBR, a packet is forwarded along a curve set by the source. A cone-based distributed topology control mechanism proposed in [7] preserves the network connectivity by ensuring at least one neighbor exists in every cone of degree α around each sensor. Bloom filter is a space-efficient randomized hash-coding method for representing a set to support membership queries introduced by Burton Bloom [1]. These components fit together in harmony producing a scalable and efficient data collection mechanism.

We propose a data collection mechanism resilient to the node failures or packet dropping attacks for *large-scale* mobile-assisted wireless sensor networks. The WSN we consider consists of a mobile element (ME) and static sensors. The ME covers the area to be monitored using a pre-determined route. Since an ME is expected to have more resources than a regular sensor, its transmission range can be longer than a sensor's. On its journey, the ME broadcasts *long-range* messages. Each broadcast message triggers the sensors in the vicinity of the ME to reply back with their data in a multi-hop manner using trajectory routing. The cone-based topology control mechanism is employed to control the number of hops a packet travels towards the ME. Bloom filter is the main data structure of our system's acknowledgement mechanism. Simulation results indicate the ME is able to collect data from over 99% of the total sensors using the proposed scheme.

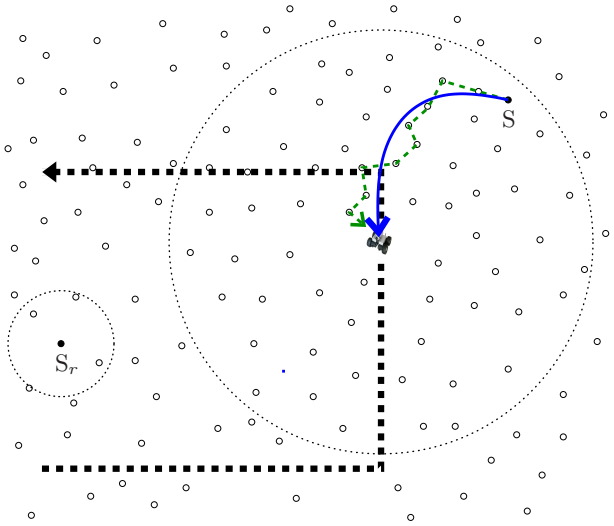


Figure 1: System Model

2. SYSTEM MODEL

The WSN in this work consists of an ME and n sensors (s_i is a sensor where i is the id of the sensor and $1 \leq i \leq n$). The sensors are statically deployed in a bounded region of $A \times A$. We assign the transmission range of a sensor according to the distances between the sensor and its close neighbors with a cone-based topology control mechanism. This approach reduces the transmission interference which is a bottleneck for the performance of an application designed for a dense WSN. Also, it extends the life-time of the sensors. r_i is the range of the sensor with the id i; whereas r_{ME} is the range of the ME. The range of the ME is the same throughout the application except for a few initial transmissions. Also, It is greater than the range of any sensors. The ME covers the area of interest with periodic broadcasts, and speed, V_{ME}. Each broadcast message is associated with a sequence number (SN_i: i^{th} sequence number). Sensors within the transmission range of a broadcast reply back to the ME with their data in a multi-hop manner. The ME follows a space-filling curve as its route. Once the ME completes its tour, it reports all the sensor readings it has collected to the base station (BS).

The communication from the sensors to the ME is based on trajectory routing. Trajectory routing requires a dense network, and the nodes know the locations of their neighbors, at least approximately. Therefore, the sensors are assumed to know their locations and the locations of their neighbors. Also, the ME knows the sensor locations approximately. These can be achieved using mobile-assisted localization techniques such as the one proposed in [11], or an expensive option for localization would be attaching a GPS device for each node. In Trajectory routing, the source node embeds a curve into the packet, and the intermediate nodes forward the packet as close as possible to the curve using greedy techniques.

The system model is shown in Figure 1. The ME collects data from the static sensors deployed in an area of interest. The circles represent the sensors. The thick dashed line represents the route of the ME. The transmission range of the ME is larger than the transmission range of a sensor. The transmission range of a sensor (S_r), and the ME is depicted in the figure. The ME transmits long-range broadcasts. The broadcast message triggers the sensors within the broadcast range to send their data. An acknowledgement mechanism bypasses unnecessary replies. For example, when the sensor, S, receives the broadcast message, it sends its data along the curve as shown in the figure if S has not received its acknowledgement yet. The intermediate nodes forward the packet originated from S to the ME along the trajectory shown as the dashed line.

3. PROPOSED SCHEME

The ME covers the area to be monitored following a space-filling curve. It transmits broadcast messages. A broadcast message has two functionalities. It triggers the sensors within the vicinity of the ME to reply back with their data, and it contains a Bloom filter carrying the acknowledgements for the sensors whose data have been successfully received after the previous broadcast. Trajectory routing is used when the sensors send their data to the ME. Collisions are mitigated using the acknowledgement mechanism, the cone-based topology control algorithm, and a back-off mechanism which adds proper delays to the reply messages from the sensors to the ME. The communications are secured using symmetric-asymmetric keys and cryptographic functions. The proposed scheme is designed for dense networks, and it is resilient to node failures and packet dropping attacks.

3.1 Sensor-ME Communication

Trajectory routing is used for the communication from the sensors to the ME. When a sensor receives a broadcast message, it either drops the packet, or replies back to the ME depending on whether the sensor had previously sent its data to the ME *successfully*, or not. If the ME has not received the sensor data yet, the sensor picks a trajectory where the starting point of the trajectory is the sensor location, and the final point of the trajectory is the ME position. Then, the sensor embeds the trajectory into the packet, and the packet is sent to the ME along the trajectory in a multi-hop manner.

Although a broad range of curves can be defined, we pick the upper or lower arc of the major axis of an ellipse as the trajectory. When a sensor is ready to send its data to the ME, it picks one side of the ellipse, embeds that trajectory into its packet, and sends the packet. If it receives an acknowledgement at the next broadcast, the sensor is done with sending its own data to the ME; and drops the succeeding broadcast messages. However, it continues to forward the packets which were originated by the other sensors if the sensor is along the curve of those packets. If the sensor does not receive the corresponding acknowledgement at the next broadcast due to node failures, collisions, or insufficient density of the network, it picks the *other* side of the ellipse as its trajectory; and sends its data along this trajectory. This approach increases the probability of the data packets to be received by the ME. The major axis of the ellipse divides the plane in half, and we guarantee that a node does not forward its packet if its only forwarding choice is a node on the other half plane. Since the half planes can not contain a sensor in common, the hops along both curves are different.

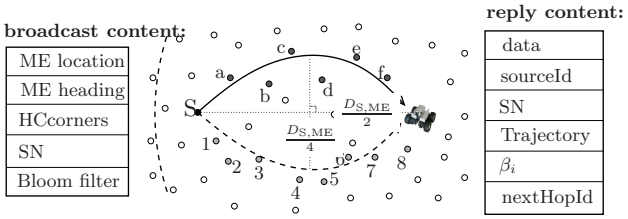


Figure 2: Switching between curves

Figure 2 shows the interaction between an ME and sensor nodes. In the figure, circles represent sensor nodes, the two arcs (solid and dashed) represent two different trajectories. The dashed arc on the left shows the long-range broadcast of the ME since the figure shows a tiny portion of the whole network. The rectangles show the broadcast and reply message contents. The source node, S, embeds the solid trajectory into its packet and sends the packet to the ME in a hop-by-hop manner. The packet follows the trajectory, and in the ideal case, the packet is transmitted to the ME through the sensor nodes a, b, c, d, e, and f. If the acknowledgement for this packet is not received at the next broadcast, the sensor, S, sends its data along the dashed trajectory. The packet is expected to be forwarded through the sensors 1, 2, 3, 4, 5, 6, 7, 8.

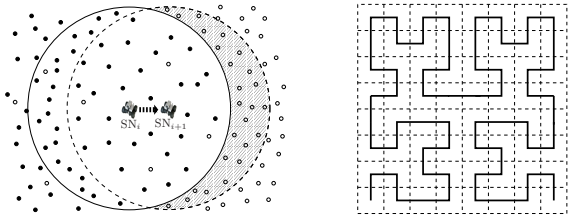
An *acknowledgement* mechanism is required to prevent a sensor from replying to every broadcast message it receives. Each broadcast message has a Bloom filter containing the acknowledgements corresponding to successful sensor data receptions triggered by the previous broadcast message. In this way, sensors become aware of their successful data transmissions. With the acknowledgement mechanism, node failures on the trajectories can also be detected. Moreover, if a sensor does not receive its acknowledgement due to a node failure on the trajectory, it uses a different trajectory increasing the resilience to the node failures. Finally, the acknowledgement mechanism increases the overall quality of the network bypassing unnecessary transmissions.

Since the ME uses long-range broadcast messages, it is possible that a sensor receives the broadcast message multiple times with different ME locations. As a result, the sensor uses a trajectory having different destination location. In this way, different intermediate nodes are used for the sensor-ME communication when the ME is at different regions increasing the resilience to the node failures. As a result, A sensor will have many chances to transmit its data to the ME thanks to the long-range broadcasts and separate forwarding paths increasing the resilience to node failures.

3.2 ME-Sensor Communication

The ME transmits long-range broadcast messages periodically. The broadcast period, τ, is an important factor for the performance of our system. A significant issue regarding the broadcast period is the limitation on the packet size for WSNs. Remember that we use an acknowledgement mechanism to prevent further replies from the sensors whose data had been received successfully by the ME previously. To achieve this, the acknowledgements are embedded to the broadcast packets. Because of the packet size limitation, there is a limit on the number of the acknowledgements that can be embedded into the broadcast packet. We use a probabilistic space-efficient data structure, Bloom filter, to increase the number of acknowledgements that can be embedded into a broadcast message.

Bloom filter is used to test whether an element is a member of a set or not. It is a one-bit vector array of size m, initially all bits set to 0. Whenever an element is inserted to the Bloom filter, the element is first hashed by k independent and uniformly distributed hash functions. Each of the k hash values is used as the bit index of the Bloom filter which is set to 1. In our protocol, when the ME receives a packet originating from a sensor, s_i, with the sequence number SN_j, the ME applies the one-way *sha1* function on $\mathrm{id}_{\mathrm{s_i}}|\mathrm{SN}_j$. The Bloom filter is reset for each broadcast message. *sha1* produces a 20 byte message digest of which we use each of the first k bytes as the index values to the Bloom filter. This is repeated for all the sensors which can transmit their data successfully to the ME. The Bloom filter is included in the next broadcast message (SN_{j+1}). When the sensor, s_i, receives the broadcast, it queries the Bloom filter using the result of $sha1$ applied on $\mathrm{id}_{\mathrm{s_i}}|\mathrm{SN}_j$. The sensor either drops the broadcast or replies back to the ME depending on the membership in the Bloom filter. If the sensor does not see its id in the Bloom filter, it replies with its data. In the meantime, the ME resets the Bloom filter for the new broadcast, and inserts the acknowledgement for the sensors from which the ME received their data. The bloom filter containing the acknowledgement corresponding to the broadcast message with SN_{j+1} is included in the new broadcast message with SN_{j+2}. The same mechanism is repeated for all broadcasts. In this way, a sensor knows whether the sensor data is received by the ME or not.



(a) Consecutive broadcast messages

(b) Hilbert curve

Figure 3: Data collection protocol.

The false positives rate (*fpr*) of a Bloom filter can be controlled through the parameters n_{bf}, m, and k; since the *fpr* is approximately $(1-e^{\frac{-kn}{m}})^k$. Because of the packet size limitation in WSNs, we pick m as 256 bits (32 bytes). Moreover, we only let 1% or less than 1% of *fpr*. Under these constraints, n_{bf} is found as 25, and k is found as 5 after trying various values for n_{bf} and k for our data collection application. As a result, our Bloom filter is capable of holding n_{bf} acknowledgements for n_{bf} sensors at each broadcast message. n_{bf}, the number of total sensors in the network (n), the speed (v_{ME}) and the range of the ME (r_{ME}) are the key factors for determining the broadcast period. Assuming most of the sensors within the range of the previous broadcast message are acknowledged, the next broadcast is transmitted when there are n_{bf} expected unacknowledged sensors within the range of the ME. In Figure 3(a), the solid circles represent the sensors which have received their ac-

knowledgements from the ME; and the void circles represent the sensors which could not receive their acknowledgements yet. The figure shows two consecutive broadcasts. Since the Bloom filter is able to carry $n_{\rm bf}$ sensor acknowledgements, the next broadcast is transmitted when the shaded area in the figure holds $n_{\rm bf}$ expected number of sensors. Let the shaded area be $\rm A_s$. Then, $\rm A_s = \frac{n_{bf}}{n} A^2$ where A^2 is the total area of the monitored field. A_s is actually the difference of the two circles with radius r_{ME}: $A_s = C_{SN_{i+1}} - C_{SN_i}$ where C_{SN_i} is the transmission area of the broadcast message with SN_i, and $C_{SN_{i+1}}$ is the transmission area of the consecutive broadcast message. From geometry, we know that the area of the difference of any two circles with the same radius is $\rm Circ_{diff} = \pi r_{ME}^2 - 2(\frac{\theta r_{ME}^2}{2} - \frac{d}{2}\sqrt{r_{ME}^2 - (\frac{d}{2})^2})$ where $\theta = 2\arccos(\frac{d}{2r_{ME}})$ is the angle of the arc between the intersection points of the two circles, and d is the distance between the two circles. If $A_s == \rm Circ_{diff}$, then the centers of the circles are the locations of two consecutive broadcast messages. Let d_c be the d value guaranteeing the equality, $A_s == \rm Circ_{diff}$. d_c becomes the expected distance between two consecutive broadcast messages. Since all the variables other than d is known, d_c is computed using a binary search. Then, the expected broadcast period, τ, is calculated as $\tau = \frac{d_c}{V_{ME}}$.

3.3 Controlling Collisions

Upon receiving a broadcast message, if the sensors in the vicinity of the ME reply back to the ME all at the same time, *collision*s occur inevitably. Therefore, a *back-off mechanism* is required to mitigate the number of collisions. After finding the delay, a sensor *predict*s the location of the ME at the time (current time + delay) the sensor will be sending its data, and uses this location as the destination point of its trajectory.

The aim of the back-off mechanism is to assign different periods of delays to the sensors. After receiving a broadcast message, a sensor, s_i, calculates a delay of period ($Delay_{s_i}$) which is less than the broadcast period, τ. $Delay_{s_i}$ depends on the orientation of the sensor with respect to the ME, and the heading of the ME. Recall that two consecutive broadcasts intersect because of the limitation on the packet size. This limitation favors the mitigation of the collisions since fewer sensors have to send their data within the period between two consecutive broadcasts.

The location of the ME at the time a sensor is sending its reply message needs to be predicted. The predicted ME location is set as the destination point of the trajectory. The broadcast message includes the ME location, and a few corners of the space filling curve being used. For example, if Hilbert Curve (HC) [2] is used as the route of the ME, the next two HC corners the ME will visit are included in the broadcast. The speed of the ME is also known by the sensor (can be included in the broadcast messages). Using this information together with the $Delay_{s_i}$, predicting the location of the ME at the time s_i sends its reply message becomes trivial.

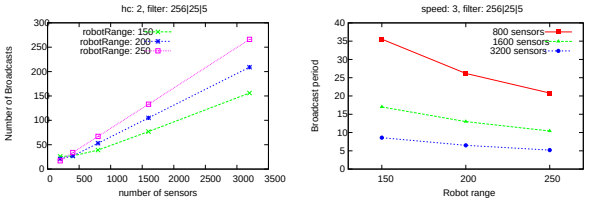
4. SIMULATIONS

We conducted extensive simulations to support the proposed scheme. We implemented the simulations using ns2 wireless network simulator. Our protocol is implemented and added as a new protocol to ns2 core code. We used a modified version of *802.11* MAC layer. The RTS/CTS mechanism is disabled to provide the communication between two elements which have different transmission ranges as our protocol requires, and to mimic *802.15.4* standards. Disabling the RTS/CTS protocol also mitigates the collisions since there will be relatively less transmissions. In addition to the simulation results presented in this paper, supplementary simulation results can be found on the project web page [15]. Videos are available for some of the individual sample runs of the simulations to give an idea about the mechanics of the system.

n sensors are deployed using uniform distribution in a region of 1000m $\times$ 1000m. The values of the system parameters are the following: number of sensors (n: 200, 400, 800, 1600, 3200), maximum sensor range (r_{max}: 10m, 20m, 30m, ..., 150m), the range of the ME (r_{ME}: 150m, 200m, 250m), cone alpha values (α: 60°, 90°,120°, 150°), curve levels (2, 3), and the ME speed (V_{ME}: 3m/s, 6m/s). Also, the constant parameters for the Bloom filter are $m = 256$, $n_{bf} = 25$, and $k = 5$. For each setting, we generated 100 different sensor configurations. Our main performance criterion is the *success rate* which is described as the percentage rate of the number of the sensors whose data is received by the ME over the number of all sensors considering that the ME tours the area to be monitored once.

Although any space-filling curve can be used as the route of the ME, we used two different space-filling curves for comparison purposes: Hilbert curve(HC), and snake-scan curve (SC) which is a non-recursive space-filling curve. A level-3 HC is shown in Figure 3(b). A HC is defined recursively. 4 level k curves are combined to have a level $k+1$ curve as follows. A square is initially divided into 4 ordered quadrants and a *first-order curve* is drawn by connecting the center of the quadrants. For the next level of HC, each of the quadrants is divided into 4 and 4 scaled-down level 1 HCs are connected by changing the level 1 HCs' orientations preserving their order. SC produces better results since it contains less number of turns compared to HC. Although the use of SC outperforms the use of HC, their success rates are almost identical. We focus on HC for the route of the ME as we present the simulation results.

The required number of broadcasts is feasible although the system achieves high success rates. The number of broadcast messages depends on r_{ME}, n, and the length of the ME route (HC_{level}). The relation among these variables are shown in Figure 4(a) where $HC_{level} = 2$ is used. As n increases, the number of broadcasts increases since the required area of the difference of two consecutive broadcast messages gets smaller and the broadcast frequency increases. Also, the number of broadcasts is directly proportional to r_{ME}. The broadcast period, τ, depends on r_{ME}, n, and V_{ME}. The relation among τ, r_{ME}, and n is demonstrated in Figure 4(a) when $V_{ME} = 3$. As r_{ME} increases, τ decreases since the area of the difference of two consecutive broadcast messages increases faster when r_{ME} is longer. When the network gets denser, the required area of the difference of two consecutive broadcast messages gets smaller since dense network has more sensors per unit of area; and τ decreases.

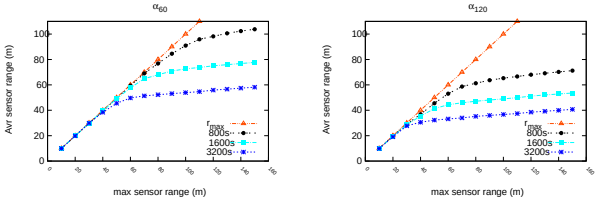


(a) (b)

Figure 4: Number of broadcasts and broadcast period

The ranges of the sensors are arranged using the cone-based topology algorithm. The technology can only allow limited number of range levels for the transmission range of the commercial sensors [7]. In our simulations, the cone-based topology mechanism is applied assuming the sensors are able to set one of eight range levels to their transmission range. r_{max} is divided into eight and the result is set to range level 1. The other range levels are computed by adding the division result each time as the eighth level being r_{max}.

Most of the simulation results we present discuss the cases where the sensor range is increasing. The graphs use the maximum sensor range (r_{max}) as the sensor values for clear representation. However, the real values for the sensor ranges are different depending on the α cone value, and the network density since we are deploying the cone-based topology algorithm. We use $\alpha = 60°$, and $120°$ in our simulations. To give an idea regarding the relation between the r_{max} values and the average of real sensor ranges for different settings, Figures 5(a), and 5(b) are used respectively for α values $60°$, and $120°$. The figures include the deployments with 800, 1600, and 3200 sensors. As seen in the figures, for low r_{max} values, the averages of the real sensor ranges are the same as the r_{max} values. As the maximum sensor range increases, the effect of cone-based topology control algorithm can be seen since a sensor now has more neighbors around it. Also, the average of the real sensor range values for dense networks is less than the average of the ones for scarce networks because of the same reason. Finally, the average of the real sensor ranges when $\alpha = 120°$ is less than the average of the real sensor ranges when $\alpha = 60°$ as expected.

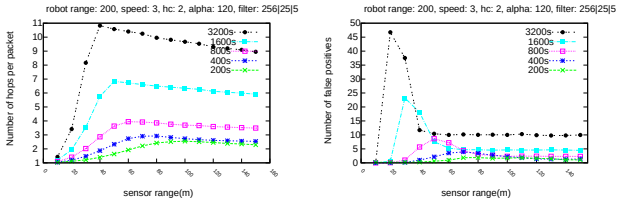


(a) α_{60} (b) α_{120}

Figure 5: Cone-based topology.

The success rate of our system is promising collecting data from over 99% of the sensors deployed. The success rate values as the maximum sensor range increases are depicted in Figure 6(a) for the configuration where $r_{ME} = 200$m, $HC_{level} = 2$, $\alpha = 120°$, and $V_{ME} = 3$m/s. In the figure, the x-axis represents the maximum sensor range. The actual sensor ranges are different than the presented since the cone-based topology is used. For example, a setting with 3200 sensors achieve the optimum success rate when $r_{max} = 40$m. The corresponding average sensor range is found to be about 30m when Figure 5(b) is analyzed. In the beginning, as the sensor range increases, the success rate also increases. The reason is when the sensor range is small, a sensor does not have enough neighbors to apply the trajectory routing. Also, dense networks achieve their optimum success rates earlier than the scarce networks because dense networks have more number of neighbors which results in better routing performance.

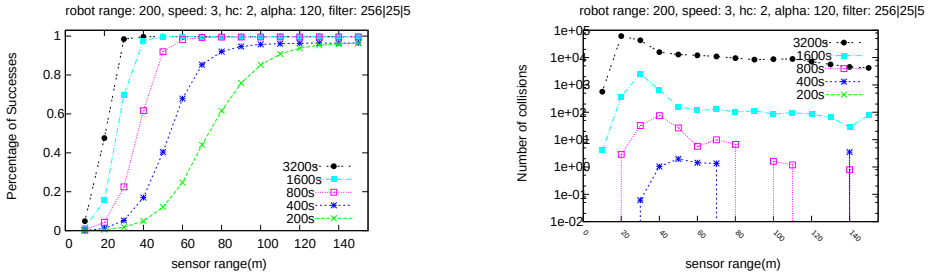
The number of total collisions as the sensor range increases for the same configuration is shown in log-scale in Figure 6(b). For all sensor range values, more collisions occur in dense networks compared to scarce networks as expected. In the beginning, the sensor range is small, so the collisions are less likely to occur. As the sensor range increases, the number of collisions also increases until sensors start to send their data to the ME successfully. When more packets are being received by the ME, the number of collisions start to decrease since a sensor is not required to re-send its data once it receives its acknowledgement. The number of collisions continue decreasing sharply until optimum success rates are achieved. Further increase in sensor range provides less collisions since the number of hops in the communications between the sensors and the ME will decrease. We also analyzed the distribution of the collisions. The maximum number of collisions per sensor is as low as 1 for individual simulation optimum settings with 1600 or less sensors. The maximum number of collisions per sensor is about 40 for the simulations resulting optimum success rate with 3200 sensors. Also, more packets collide on the sensors closer to the ME route. When the ME changes its direction, more collisions occur. Furthermore, when the range of the ME increases, the number of collisions also increases since more sensors are triggered to send their replies.



(a) (b)

Figure 7: Average number of hops per packet and total no of false positives.

The average number of hops a packet visits (Hop_{avr}) is shown in Figure 7(a). As the sensor range increases, the packets are more likely to be received by the ME and Hop_{avr} increases. Hop_{avr} reaches its maximum value when the success rate is maximum. After this point, the increase in sensor range results in less Hop_{avr} values since the packet can reach the ME using less many hops. The sensor ranges are shorter when denser networks are considered as a result of the cone-based topology algorithm. Therefore, Hop_{avr} values for dense networks are higher than Hop_{avr} values for sparse networks. The total number of false positives caused by the Bloom filter is depicted in Figure 7(b). The number



(a) Success ratio (b) Collisions

Figure 6: Success ratio and collisions.

of false positives reaches its local minimum value when the success rate is maximum. In terms of speed, the success rate decreases as the ME travels faster since the performance of the back-off mechanism decreases causing more collisions.

5. CONCLUSION

In this paper, we propose an efficient, scalable, and resilient data collection protocol for large-scale mobile assisted WSNs. The system is built on components including trajectory routing, cone-based topology mechanism, and Bloom filter which are seamlessly integrated. Performance of the proposed scheme depends on many tunable parameters including sensor range, density and topology of the network. Using simulations, we show that the system parameters can be tuned for a high rate of successful data collection. It is possible to cover a large area using limited number of broadcasts for networks with sufficiently high density ideal for trajectory routing. Collisions play an important role on the success rate, and they are minimized with a back-off mechanism. As sensor range is increased, success rate initially increases and starts to decrease after reaching its maximum value. This is due to the higher number of collisions that occur with increased range. Finally, future work includes identifying the misbehaving or malicious nodes.

6. REFERENCES

[1] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13:422–426, 1970.

[2] D. Hilbert. *Uber die stetige Abbildung einer Linie auf Flachenstuck*, volume 38, pages 459–460. Math. Ann, 1891.

[3] Y.-C. Hu, A. Perrig, and D. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1976–1986 vol.3, 2003.

[4] P. Juang, H. Oki, and Y. Wang. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In 10^{th} *International Conference on Architectural Support for Programming languages and Operating Systems.*, October 2002.

[5] C. Karlof and D. Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2–3):293–315, September 2003.

[6] I. Khalil. Mimi: Mitigating packet misrouting in locally-monitored multi-hop wireless ad hoc networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5, 2008.

[7] L. Li, J. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer. A cone-based distributed topology-control algorithm for wireless multi-hop networks. *Networking, IEEE/ACM Transactions on*, 13(1):147 – 159, feb. 2005.

[8] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 255–265, New York, NY, USA, 2000. ACM.

[9] B. Nath and D. Niculescu. Routing on a curve. *SIGCOMM Comput. Commun. Rev.*, 33(1):155–160, 2003.

[10] D. Niculescu and B. Nath. Trajectory based forwarding and its applications. In *Mobicom 03*, New York, NY, USA, 2003. ACM.

[11] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Mobile-assisted localization in wireless sensor networks. In *INFOCOM 2005*, volume 1, pages 172 – 183 vol. 1, march 2005.

[12] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: modeling a three-tier architecture for sparse sensor networks. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 30–41, 2003.

[13] T. Small and Z. Haas. The shared wireless infostation model - a new ad hoc networking paradigm (or where there is a whale, there is a way). In *ACM MobiHoc*, pages 233–244, 2003.

[14] A. Somasundara, A. Ramamoorthy, and M. Srivastava. Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In *Proceedings of the 25th IEEE International Real-Time Systems Symposium*, pages 296–305, 2004.

[15] B. Tas and A. S. Tosun. Project web page for simulation results. `www.cs.utsa.edu/~tosun/trajectory/trajectory.html`. Accessed: 31 January 2015.

[16] B. Yu and B. Xiao. Detecting selective forwarding attacks in wireless sensor networks. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, pages 8 pp.–, 2006.