Energy efficient routing protocols for sensor nodes in wireless sensor networks

Stefan Gaschler
Institute of Computer Engineering
University of Heidelberg
Computer Architecture Group
Heidelberg, Germany 691117
Email: stefan.gaschler.sg@gmail.com

***Abstract*—This paper summarizes four routing techniques with low power consumption for wireless sensor networks. First, the motivation for the usage of these techniques is shown. After introducing the idea of the respective technique, the function is described and each of them is analyzed for the required transmissions.**

I. Introduction

Sensors are the contact point between the real and the virtual world. They collect data from the environment and give this data to a processing system which can respond to it. A Wireless Sensor Network (WSN) can contain hundreds or thousands of sensors. The sensors may be arranged to a larger area so that the sensors are not able to communicate directly with the base station (BS) or the available energy is limited so it is no option to establish a long range communication, then the sensors have to communicate among each other and an efficient routing protocol is necessary. The figure 1 shows a sensor network architecture and the structure of the schematic drawing of a sensor node. Each sensor consist of a Power Unit, a Processor Unit, a Sensor Unit and a Transmission Unit.

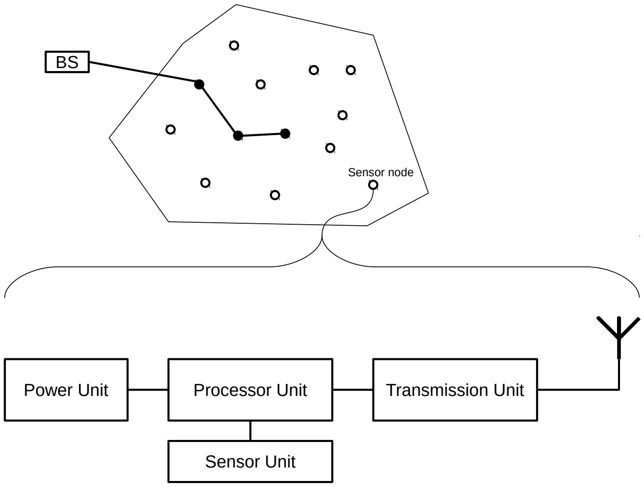

Fig. 1. sensor node and wireless network architecture

The Processor Unit has to find a good routing path and is mainly responsible for the routing decisions. Due to a limited transmission and reception range, each sensor has only limited information of the surrounding network so the main problem is to transport the data to the BS with a minimum of transmissions.

II. competitive Protocols

The four protocols are compared to similar routing ideas to clarify the reduction of the necessary transmissions with these protocols.

1) Query Flooding: When a sink node needs certain data it requests this data with a query telegram. With Query Flooding it broadcasts the query to each neighbor within the sending range. A received query is broadcasted if it is the first time of reception else it is gonna be dropped. This ensures that the query is routed to the hole network including the node which can resolve the query with the requested interest.

2) Event Flooding: If a node detects an Event, it sends an event telegram. An event can be defined as a transgression of a limit or any other occurrence. With Event Flooding this telegram is sent to each neighbor as with Query Flooding. So the event information is spread to the hole network including a sink node for this interest.

3) Random Walk: Random Walk is a query based mechanism. A query is send to a random neighbor until it reaches the node which can resolve the requested data in the query. This mechanism uses complex queries with more than one interests, so it is possible to collect more data with one query.

4) Greedy Forwarding: Greedy Forwarding uses a location based forwarding. It calculates the distance to the destination node and forwards to the node which makes the best progress. A node must know the coordinates of it self, its neighbors and the destination. So the telegram will automatically find the shortest way to the sink node.

III. criteria of a wireless network

The used protocol leads to different network types with different criteria. Each of this criteria is used to determine if the respective protocol is suitable for the application.

1) Classification: The classification can be divided for the most networks in flat, hierarchical or location based. In a flat network each node has the same rights while in hierarchical networks some nodes have special functions. Location based networks use the location information of each other nodes to send the data towards this direction.

2) Data aggregation: With the data aggregation it is possible to combine several measurements to one data packet to reduce the number of telegrams which must be sent. The aggregation can be applied on other data than only the measurements.

3) Overhead: The overhead can have a massive impact on the efficiency of the protocol. This overhead is necessary for the information exchange between the nodes. The information which has to be sent depends on the classification of the network. So in hierarchical networks the most important information are ID and hop count of the ambient nodes.

4) Data delivery model: The data delivery model describes mainly the trigger type of a data transmission. Some of them are started by a query of another node and some are started by an event and data is send from the measuring node to the data sink or data are send cyclically.

5) Scalability: The scalability is mainly the ability to update the information in each node so that he can operate with this node. This depends on the scope of the ambient information. If only a few information from the neighbors are needed, it can be updated easily.

6) Quality of service: The quality of service is a criteria to ensure that the transmitted data are usable for the receiving node. In some applications it is necessary to guarantee a certain maximum delay for the data transmission. If a data transmission exceeds a time limit, it might be possible that the data are useless for the node.

IV. Routing Protocols

A. Rumor Routing

1) Idea: The idea behind the rumor routing protocol is to close the gab between a protocol which floods the network with queries and one which floods it with event messages.

2) Mechanism: Each node that recognizes an event sends an agent telegram to a randomly chosen neighbor. This agent contains the information about the event and the distance to this event which is incremented at each hop. Every node which receives the agent stores this information and the ID of the previous node. An agent can take the information of previous handled agents and carry it with him. In figure 2 node $N1$ generates the agent $A1$ on event $E1$. After this node $N2$ recognizes the event $E2$ and does the same. When the agent $A2$ hits the crossing point, it takes the information from the first agent and takes it with him.

A sink node which starts a query has to send a query message. This message is also forwarded in a random manner. If a node has a route information about the targeted event, it forwards the query along this route. This can be seen in figure 2. Node $N3$ starts a query $Q1$ and forward it to one of his neighbors. As soon as the query reaches a node with the necessary information, it walks along the route to the event node $N1$.

If a node receives an agent with a event information which is already known in this node, he stores the ID of the node with the smaller distance. In figure 3 the agent $A1$ was created first, but the Agent $A2$ found a way with a minor hop count. So it can the route for $A1$ and insert its own route.

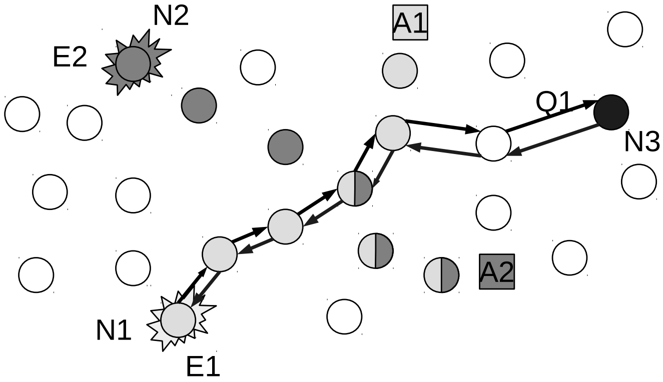

Fig. 2. Agent creation and information walk

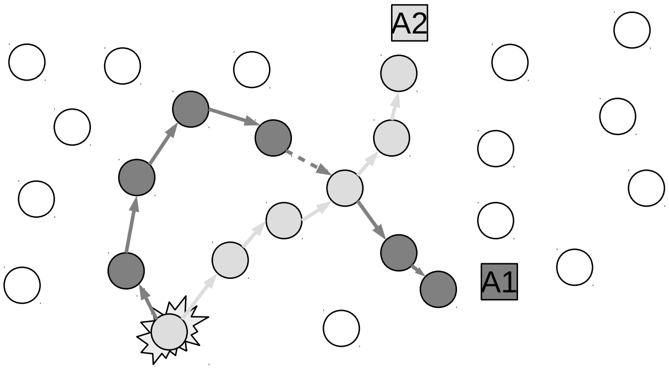

Fig. 3. Agent creation and information walk

3) Analysis: The number of transmissions T for query flooding can be calculated with

$$T = Q \cdot N$$

Where Q is the Number of queries and N is the number of nodes in the system. If the system uses an event flooding algorithm it can be calculated by

$$T = E \cdot N$$

where E is the number of events.

Rumor routing does generates a number of agents A on each event with a lifetime L_a and a number of query messages with a lifetime L_q. With this the number of transmissions is calculated as followed.

$$T = E \cdot A \cdot L_a + Q \cdot L_q$$

In figure 4 it can be seen that the rumor routing does less increase as the query flooding function and is until some point below the line of event flooding.

B. ACQUIRE

1) Idea: The idea behind ACQUIRE is to create complex queries for several variables. Each node which receives a query tries to resolve the query complete or partially.

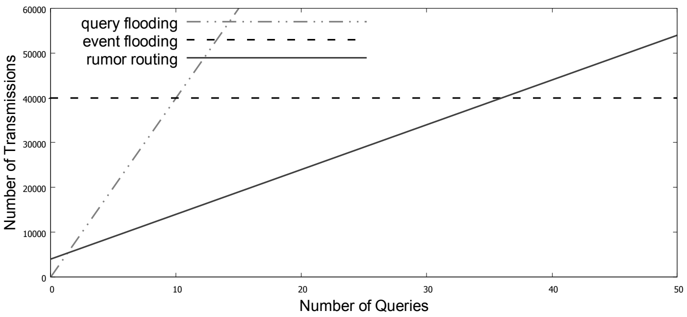

Fig. 4. Number of Transmissions by Number of Queries

2) Mechanism: A requesting node starts a query which can consist out of sub-queries with different interests. A node which becomes an active node, like node A in figure 5, tries to resolve the query completely or partially with its own information and those of its neighbors in a look-ahead range of d nodes. If a node doesn't have valid information of its neighbors to resolve the query, it sends an update message to its neighbors to get valid data again. These information are stored in the node with a time to live which marks when a new update has to be done. A unresolved query is forwarded to the next node in a distance of d hops. So the query is getting smaller as peaces of it gets resolved until a node can resolve it completely. This node routes the query back as a complete response.

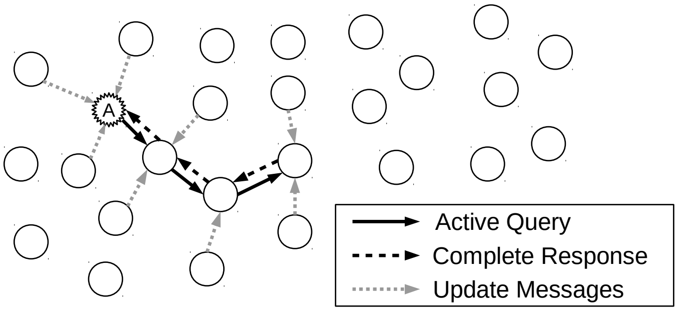

Fig. 5. resolving a Active Query

3) Analysis: For the analysis two parameters are very important, the look-ahead range d and the data dynamics c or in other words c is the probability that a information has to be updated. For example, if a node has to update every 50 queries $c = 0.02$. The higher c is the lower is the number in which d can be chosen. When $d = 1$ the system behaves like with the random walk protocol. The problem is that a highly changing system has a shorter information time-to-live and the more update messages must be send. The number of messages which are necessary to update a node can be seen in following formula.

$$T_{Update} = F(d-1) + \sum_{i=0}^{d} i \cdot N(i)$$

Where function $F(d)$ is the number of nodes within a range d and $N(i)$ is the number of nodes at the actual hop i. A average number of transmissions can be determined by

$$T_{average} = (c \cdot T_{Update} + 2d) \cdot S_M$$

where S_M is the average number of steps to answer a query of size M depending on the look-ahead range.

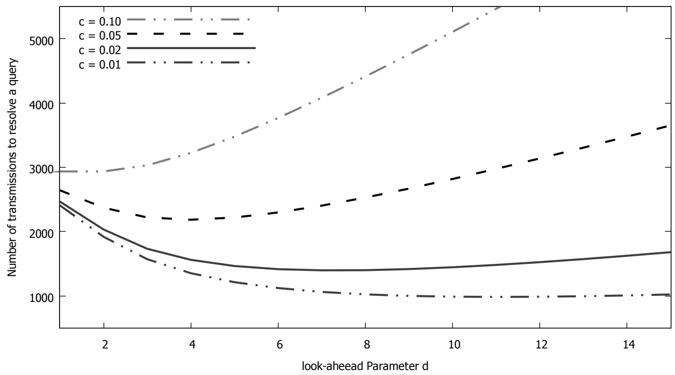

Fig. 6. consequence of d and c on the number of transmissions
Here, N = 1000 and M = 200

The optimal value for d in dependence of c can be seen in figure 6. This minimum point is independent on the number of nodes in the system and the size M of the complex queries. It can be seen that the ACQUIRE protocol needs less transmissions when the value for the data dynamics gets smaller. Above this value it is better to use a random walk protocol.

C. SOFROP

1) Idea: The Self Organizing and Fair Routing Protocol (SOFROP) is used for sensor networks with highly mobile sensor and stationary base stations. Because of the fast changing topology it tries to reorganize its topology continuously.

2) Mechanism: At first the base stations tries to establish the shortest route to the central data sink. For that the data sink starts to flood the base station network with telegrams including its ID and an initial hop count of one. Each base station stores the hop count and the ID of the received telegram and forwards a new telegram containing an incremented hop count and its own ID. The data are only updated when the hop count is lower than the stored. This forwarding is done until the hop count reaches the number of base station.

This method is only done once whereas the update of the neighbor list in the sensor nodes is updated periodically with a frequency f. This update is started by the base stations only which generates a area configuration packet (ACP) including its ID and the hop count. Every sensor receiving this telegram ignores this packet when the hop count is equal or greater than the stored one, else it stores the hop count and the ID and forwards it with an incremented hop count and its own ID. So it finds always the shortest way to the base station and even nodes which are outside the range of the base station can be connected to the network.

If a sensor looses the connection to the previous node or base station and receives only ACPs with a equal or higher weight, it increases its own weight by one and waits until it receives an ACP with an hop count one less than the actual. This is done until a maximum weight k is reached. This maximum weight limits the depth of the tree of the network.

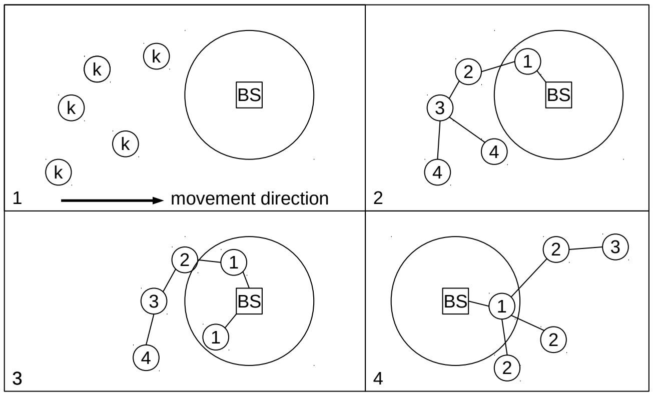

Fig. 7. Topology generation and update

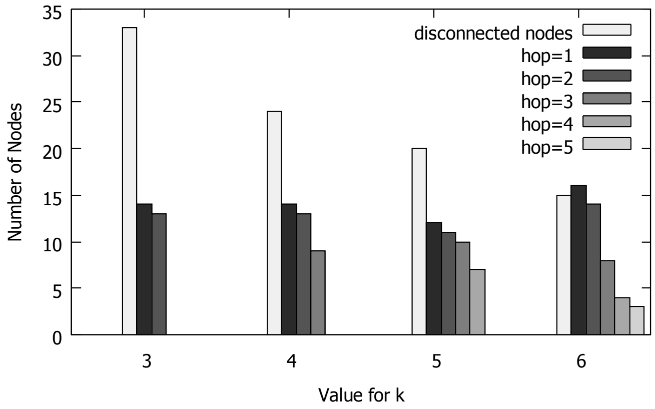

Fig. 8. Number of nodes and their hop count for different k values

If a sensor is out of range of a base station or another node so it does not receive any ACPs, it returns to an idle mode after a predefined timeout. In this idle mode the sensor sets its weight to the maximum value k and does not send any telegrams.

A node which has established a connection to a base station or to another node starts data transmission. Now it send its collected data and the packets which it receives from other nodes which are defined as interest. By virtue of the topology it is possible that a node becomes a bottleneck so it has to drop some packets. A node does not drop any packets as long as the remaining output capacity $C_r \geq 0$. Each interest has an packet rate α_p. The remaining output capacity is calculated as following

$$C_r = C_o - \sum_{i=0}^{n} \alpha_p$$

where C_o is the output capacity of the node and n the number of active interests which are forwarded by the node. If C_r becomes negative the node compares the packet rate of each interest with the fair rate α_f:

$$\alpha_f = \frac{C_o}{N_i}$$

Packets with a smaller rate than α_f are always forwarded whereas packets with a rate bigger than the actual fair rate the node has to decide if the packet should be dropped. first the node increases the shared capacity C_s by the packet rate α_p. The node calculates a probability to drop a packet P_d with the number of interest N_s that are shared, the shared capacity C_s and the package rate of the Interest α_p.

3) Analysis: The analysis is based on a simulation in [1]. In this scenario 60 sensors are thrown in a river to collect different data. Due to the strong variability of the river a constantly topology change is the consequence. There is also a number of not connected sensor nodes which data are never collected during a simulation run. In figure 8 it can be seen how many nodes are not connected to any base station and as far the others are. With a increasing value for k the number of not connected nodes decreases.

Figure 9 shows the transmissions to build the topology and collect the data and the number of nodes which are not connected to a base station. The data transmission and the transmissions for the ACPs can be calculated with

$$T_{data} = \sum_{i=1}^{k-1} i \cdot N(i) \text{ and } T_{ACP} = \sum_{i=1}^{k-2} 1 + i \cdot N(i)$$

where $N(i)$ is the number of nodes with $hopcount = i$. So the optimal value for k is a compromise between the number of transmissions and the number of disconnected nodes. Smaller values will result in a longer lifetime but in less data coverage whereas higher values collect more data but the higher workload will result in a shorter lifetime of the nodes which are near to a base station.

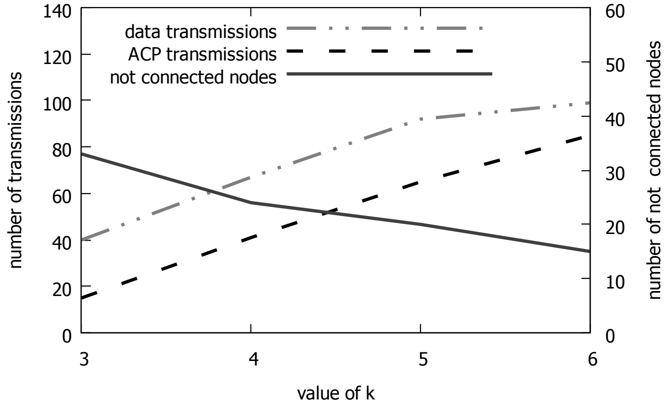

Fig. 9. Number of transmissions and not connected nodes for different values for k

D. SPEED

1) Idea: The SPEED protocol is a stateless protocol for Real-Time communication in wireless networks. Its benefit is that it tries to avoid congested areas in a wireless network and routes the packages around this area. The protocol is location bases so the data are send to a x and y coordinates.

2) Method: Each node gathers information about its neighbors with tree kinds of beacons. One periodically send beacon with the location information and two on demand beacons which one of them contains the delay estimation and the other is a back-pressure message. The set of neighbors NS_i contains the nodes within the radio range of node i. This set contains the forwarding set FS_i which contains every node which is closer to the destination D. If FS_i is empty, the node will drop the packet and sends back an back-pressure beacon to signalize that there is no route toward the destination for this direction to prevent against further forwarded packets in this direction.

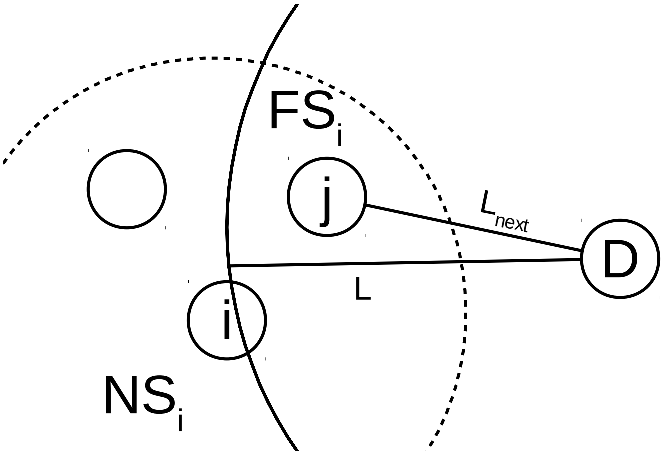

Fig. 10. Neighbor set and Forwarding set of node i

The FS_i is divided in two groups. The first group with a relay speed greater a certain desired speed s which is calculated as following

$$Speed_i^j(Dest) = \frac{L - L_{next}}{HopDelay_i^j} > S_{setpoint}$$

where L is the distance to the Destination and L_{next} is the distance to the next hop node. The second group are the nodes with a speed less than $S_{setpoint}$. A forwarding candidate is only chosen from the first group and a node with the highest relay speed has a higher probability to be chosen.

If no node can maintain the desired single hop relay speed S, the node calculates a new relay ratio u with

$$u = 1 - K \cdot \frac{\sum e_i}{N}$$

where K is a proportional gain, e_i is the miss ratio of the nodes within FS and N is the number of nodes within FS.

So the node 2 in figure 11 has only node 3 in the FS and this node has a relay speed less then the needed. Now it sends a back-pressure message with the ID, destination and a average time-to-send delay. Every node which receives this beacon can determine if this node is in the FS of the containing destination. If it is, the node updates its delay table. In figure 11 only node 1 would update its list. If a node does not have any node in the FS it becomes a dead end for the data. This node will send a back-pressure beacon with an infinite time-to-send delay.

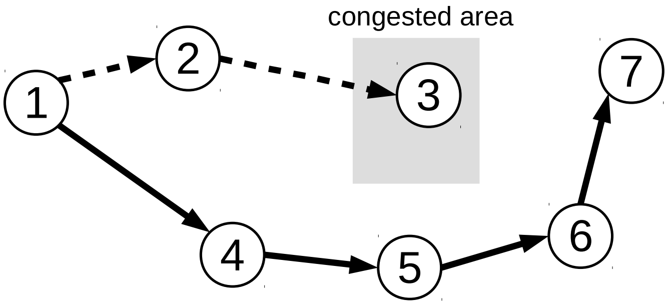

Fig. 11. rerouting around a congested area

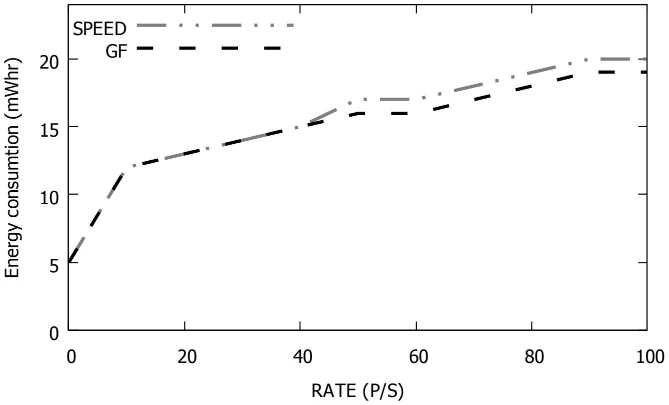

Fig. 12. energy consumption in dependence of the malicious data rate

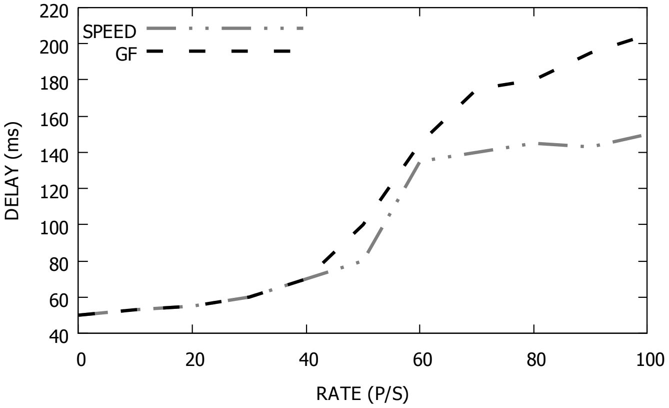

Fig. 13. delay of the data packets in dependence of the malicious data rate

3) Analysis: The analysis for the SPEED protocol is based on the simulation in [2]. The simulation results are compared with the results of the Greedy Forwarding. This protocol forwards the packets to the node that make the most progress towards the destination. The Figure 12 shows that the energy consumption of SPEED is slightly higher than the one of GF. This is because of the additionally back-pressure messages which are send.

The great benefit of the SPEED protocol is the delay time in a higher loaded network and the distribution of the telegrams over more nodes. Figure 13 shows that SPEED and GF have a equal delay time until a malicious rate reaches a certain point. After this point the advantage of the distributed load leads to a shorter delay.

The figure 14 shows the distribution of the telegrams on the different nodes. With GF all messages are send along one route and so only a few nodes have to forward all the data whereas with SPEED the telegrams are distributed over all nodes. The nodes in the direct line to the destination are utilized the most

TABLE I. PROTOCOL OVERVIEW

Routing Protocols	Classification	Data Aggregation	Overhead	Data delivery model	Scalability	QoS	Query Based
RR	Flat	Yes	Low	Demand driven	Good	No	Yes
ACQUIRE	Flat	Yes	Low	Complex query	Good	No	Yes
SOFROP	Hierarchical	No	High	Continously	Good	No	No
SPEED	Location/Data centric	Ltd	Low	Geographic	No	Yes	Yes

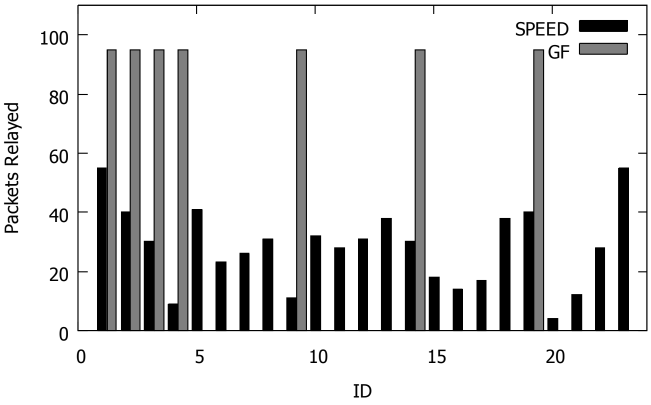

Fig. 14. distribution of the transmitted packets

but even the on a longer route have a workload. In GF the high loaded nodes will fail at first.

V. COMPARISON OF THE PROTOCOLS

The table I shows the criteria for each protocol as described in section III. For the classification we have three different types. Rumor Routing and ACQUIRE are flat network. Each of the sensor nodes can start a query to every node. SOFROP is an hierarchical protocol where the sensor nodes sends its data to the next node or base station when it is connected to the network. The base stations do not have to start a query. The SPEED protocol is a location based protocol where data are send to an explicit position.

Rumor Routing and ACQUIRE are both able to aggregate data, but in two different ways. The aggregation in Rumor Routing is done by the Agents which carry the event information from one node to another whereas in ACQUIRE the aggregation is done by the query telegrams. SOFROP does not perform any data aggregation it only forwards its data. The data aggregation in SPEED is not included in the basic protocol but there are extensions to insert this functionality.

The Overhead for the three protocols Rumor Routing, ACQUIRE and SPEED is low. Only a few information beacons have to be sent, provided it is optimally adjusted. SOFROP has a quite big overhead because of the cyclically sent ACPs. The number of the ACPs is almost the half ot the number for the data transmission what is quite high in comparison to the other protocols.

The data delivery model of these four protocols is very different. The Rumor Routing protocol only acts on demand. If there is no Event, a query will not answered. In ACQUIRE protocol the data are collected with a complex query what reduces the number of queries in the network. The data collection in SOFROP is happens in a continuous manner. Each sensor starts to send as soon as it has a connection to another node. The Data in SPEED are sent to a certain position in the network consisting of x and y coordinates.

If scalability is a important criteria, Rumor Routing, ACQUIRE and SOFROP would be the best choice. SOPFROP only has a limit if a node becomes a bottleneck and is not able to forward all incoming packets. In SPEED every node has to know where the data sink is before the network starts. If we would like to insert a new node, we would have to update every other node with its position.

The quality of service criteria is only given in the SPEED protocol. It can redirect the telegrams if a node is a dead end. The other protocols do not check if the data arrive at the targeted node so a packet loss might be a problem in this networks.

Only the SOFROP protocol is not query based. Every sensor sends it data continuously to the next node or base station. A node of one of the other protocols will send its data when it gets a query from the sink.

VI. SUMMARY AND CONCLUSION

In conclusion it can be said that every routing protocol has its own application what it is invented for. For flat networks with low timing requirements it is a good choice to take Rumor Routing or ACQUIRE depending on the data dynamics and the number of events and queries. SOFROP is ideal for systems with a very fast changing topology but it a higher overhead for this must be taken into account. If the delay or the distribution of the energy consumption is a important criteria, SPEED could deliver the best results.

REFERENCES

[1] M. Akba, M. R. Brust, and D. Turgut, "Sofrop: Self-organizing and fair routing protocol for wireless networks with mobile sensors and stationary actors," *Computer Communications*, vol. 34, no. 18, pp. 2135–2146, 2011.

[2] T. He, J. Stankovic, C. Lu, T. Abdelzaher *et al.*, "Speed: A stateless protocol for real-time communication in sensor networks," in *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on.* IEEE, 2003, pp. 46–55.

[3] D. Braginsky and D. Estrin, "Rumor routing algorthim for sensor networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications.* ACM, 2002, pp. 22–31.

[4] G. Raghunandan and B. Lakshmi, "A comparative analysis of routing techniques for wireless sensor networks," in *Innovations in Emerging Technology (NCOIET), 2011 National Conference on.* IEEE, 2011, pp. 17–22.

[5] N. Sadagopan, B. Krishnamachari, and A. Helmy, "The acquire mechanism for efficient querying in sensor networks," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on.* IEEE, 2003, pp. 149–155.