ABSTRACT
When an event occurs in an area observed by a Wireless Sensor Network, a vast amount of information is transferred towards the sink(s) of the network. This phenomenon may sometimes result into network congestion and as a consequence packets may start dropping due to queue overflows. To our knowledge most proposed reactions to this phenomenon utilize only static nodes whereas a few utilize mobile sink(s). Our approach is focusing on using multiple mobile nodes to build Hard Alternate Disjoint Paths between some of the sources of the congested nodes and the sink(s) to mitigate congestion. Mobile Congestion Control (Mobile-CC) algorithm is applicable in areas where congestion happens repeatedly, or is of high duration or even permanent. If congestion happens for a very small period of time there will be no big benefit (in terms of congestion mitigation) when moving any nodes to that area, unless there is great probability of the event happening again at the same place.

Categories and Subject Descriptors

General Terms
Algorithms, Performance, Design, Reliability

Keywords
Wireless Sensor Networks, Mobility, Multipath Routing, Congestion Mitigation

1. INTRODUCTION
Congestion control in Wireless Sensor Networks is a field that attracts a lot of attention. Most of the existing congestion control algorithms approach the issue using fixed sensor nodes. Some notable exception like Karence et al proposed COngestion avoidance for Sensors with a MOible Sink (CoS-MoS)[2], in which they prove that sink movement may result to congestion mitigation. In this work we attempt to introduce the notion of utilizing multiple mobile nodes for congestion mitigation. Mobile nodes initially reside behind the sink and move on demand to build “Hard Alternate Disjoint Paths” (HADPs) between congested area and sink. These paths are called “Hard Alternative Paths” because only mobile nodes are involved in the implementation of these paths and no other existing node, beside, of course, the forwarder nodes. This approach does not replace existing control algorithms but instead cooperates with them. In this work Mobile-Congestion Control algorithm (Mobile-CC) it is implemented over Contiki Collect Protocol a low power collection protocol that runs over Rime communication stack[1]. When congestion is detected in the network, using the proposed Congestion Mechanism, a Congested Node Selection Mechanism is executed at the congested node in order to determine for which forwarder nodes to request support for. When the decision is taken, Congestion Notification Mechanism informs the sink node in order to take action based on the information gathered from the congestion scene. The final mechanism involved is the mechanism who is responsible for HADP building and joining of mobile nodes to the network. In the following section a more detailed explanation of Mobile-CC mechanisms is given.

2. IMPLEMENTATION
The first mechanism implemented by Mobile-CC is the Congestion Detection Mechanism. Several parameters are being monitored by existing congestion detection mechanisms in order to infer that congestion exists in the network. These include, buffer occupancy, received and generated packet rates and number of retransmissions of sent packets. The proposed mechanism continuously monitors buffer occupancy and when this raises above a threshold λ, rate monitoring is initiated. Rate monitoring is performed for a period of τ seconds and when this elapses, the sum of total received rate and generated rate is checked. If this sum is greater than a threshold µ we assume that congestion is happening and Congestion Notification Mechanism takes the baton. In case it is lower, buffer occupancy is checked again and if it drops lower than a threshold λ, rate monitoring stops (otherwise rate monitoring continues). When congestion is detected the Congestion Detection Mechanism is disabled for a period of δ seconds in order to prevent sending...
repeated notifications for the same congestion event. This time is provided to Mobile-CC to react and if it is necessary after the reaction to send a new notification. In case a node receives a congestion notification from another node it disables the Congestion Detection Mechanism for $\delta$ seconds in order to prevent sending multiple notifications to the sink for the same event.

Congested Node Selection Mechanism is responsible for enabling the congested node to decide which nodes to continue serve and which to request support for. The congested node aims to eliminate the number of handoffs and maximize the number of nodes to continue serve. The properties taken into consideration by this mechanism are the sending rate per node of each of the forwarder nodes who send information through the congested area, the link qualities between the congested node and the forwarders nodes and the time the forwarders nodes joined the congested nodes. The nodes firstly identifies the set(s) of nodes who have maximum total sending rate below a threshold $\sigma$. If the result is only one set, then the node request support for the nodes within the set. If the result is more than one sets, then the sum of link qualities and join times of the two sets are taken into account to decide which set worths more to request support for.

When the selection process is completed by the congested node, Congestion Notification Mechanism kicks in to deliver the information gathered from the congestion scene to the sink. A Congestion Detection Message (CDM) is created and forwarded with the highest priority. This message includes the ID, position and sending rate of each of the nodes to be healed. Here an assumption is done that both mobile and fixed nodes are aware of their location (this assumption is beyond the scope of our research). At the point of time a node joins another node, it informs the parent node of its location and in this way all nodes are aware of the locations of their sender (children) nodes.

The final mechanism involved in Mobile-CC is the Reaction Mechanism. In this mechanism, sink decides the number of HADPs that have to be built as well as the number and target locations of the mobile nodes and informs them to move to the specific locations. Mobile nodes introduce themselves to the network, healing the predefined, by the sink, nodes (node ID of sender nodes to healed is given). The mobile nodes turn their radio off, move to the target locations and switch their radio on again and introduce themselves to the network.

A number of simulations were performed in COOJA simulator [3] and the results are shown in Figure 1 and 2. Results prove that the number of dropped packets is dropped and throughput is increased dramatically after mobile nodes move in at around 100 seconds.

3. CONCLUSIONS AND FUTURE WORK

In this work we introduce the notion of using mobile nodes in order to mitigate congestion for heavily congested or repeatedly congested areas of the network. This solution can be performed by implementing hard alternative paths (paths that include just mobile nodes and not any other existing network nodes). Mobile-CC algorithm has been implemented in COOJA network simulator and results prove that it can significantly contribute in limiting packets drops and increasing congestion. The next steps are concentrated in implementing Mobile-CC in real nodes in order to check its efficiency under real conditions.

Acknowledgments.

This work has been conducted under the European Union Project GINSENG funded under the FP7 Program (FP7/2007-2013) grant agreement no 224282.

4. REFERENCES