International Journal On Engineering Technology and Sciences – IJETS™
ISSN(P): 2349-3968, ISSN (O): 2349-3976
Volume III, Issue V, May- 2016

Threshold Based Energy Efficient Data Transmission inWireless Sensor Network

M.Sasikala Devi,
M.E(CSE), Sasurie College of Engineering, Vijaymangalam.

Abstract

Selfish node is a most complaining problem in Wireless network. It delays the data transmission, holds the data and denies the resources to be utilized. In order to handle this types of issues we propose a technique which pinpoint the selfish node, divert data transmission and deliver packets. We assure to have the below mentioned features to be implemented using our COCOWA technique. A quality guaranteed selfish node eliminated neighbor selection to meet the transmission delay requirement, A distributed packet scheduling algorithm to further reduce transmission delay, a mobility-based segmentation technique that adaptively adjusts segment size according to node mobility in order to reduce transmission time, A traffic elimination to increase the transmission throughput, and A data redundancy elimination-based transmission algorithm to eliminate the redundant data to further improve the transmission Quality. This paper also proposes a novel adaptive routing technique called SNRP (Selfish Node Recovery Protocol), which does not use a dedicated SINR but instead leverages free bits in head flits of existing packets to carry and propagate rich congestion information without introducing additional wires or flits. In order to balance the network load, SNRPadopts a novel three-stage strategy of output link selection, which adequately utilizes the propagated information to make routing decisions. Experimental results on both synthetic traffic patterns and application traces show that SNRP achieves more efficient and data transmission by diverting data from reaching the selfish node and reach destination successfully.

Introduction

A wireless network consists of many number of selfish nodes which are partiallydistributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location. Provide a bridge between the real physical and virtual worlds. Allow the ability to observe the previously unobservable at a fine resolution over large spatio-temporal scales. It has a wide range of potential applications to industry, science, transportation, civil infrastructure, and security. The more modern networks are bi-directional, also enabling control of sensor activity. The development of Hydrid Networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The wireless network is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth.

The topology of the wireless networks can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routingor flooding. Hydrid Networks are more difficult to implement than PC or Web-based applications. There are three reasons for this that we found. First, the connection of computing to real world workflows makes such applications complex in the sense that current software development focuses on digital workflows. Second there is the lack of software support for distributed Hydrid Networksystems[6]. Third, programming and managing wireless sensor nodes are difficult and complex tasks. Some initial solution ideas have been published for these problems. Complex integration of multiple Hydrid Networks has been addressed by Wireless, which focuses on the networking aspect. Another proposal named FLOW focuses on the abstract software generation aspect. Both proposals have in common that they expect certain technical properties from their sensor nodes and thus show example implementations for one type of sensor network only. Also, both systems are complex in themselves, requiring a developer to learn a complex technical software system. Attempts to lower complexity in sensor networks by providing more abstract approaches have been developed in the context of TinyOS.abstraction patterns and interfaces were developed for the design of sensor node software, but this approach only focuses on one sensor node.

In areas in which there is little or no communication infrastructure or the existing infrastructure is expensive or inconvenient to use, wireless mobile users may still be able to communicate through the formation of an ad hoc network. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may not be within direct wireless transmission range of each other. Each node participates in an ad hoc plus infrastructure routing protocol that allows it to discover "Three-hop" paths through the network to any other node is introduced in this work. The idea of ad hoc networking is sometimes also called infrastructure less networking, since the mobile nodes in the network dynamically establish routing among themselves to form their own network "on the fly". Most Wi-Fi networks function in infrastructure mode. Devices on the network all communicate through a single access point, which is generally the wireless router. For example, let's say you have two laptops sitting next to each other, each connected to the same wireless network. Even when sitting right next to each other, they're not communicating directly. Some examples of the possible uses of this networking include students using laptop computers to participate in an interactive lecture ,business associates and sharing information during a meeting, soldiers relaying information for situation awareness on the emergency disaster relief and battlefield personnel coordinating efforts after a hurricane or earthquake[3].

Selfish Node Recovery Protocol

Selfish Node is commonly used for insecured data transmission in wireless communication as a way to measure the quality of wireless connections. Typically the energy of a signal fade having distance. In wireless networks, this is commonly defined by path loss. But not like wired networks that where the existence of a wired path between the receiver an sender are determines the correct reception of a message, the wireless communication network has to take a lot of environmental parameters to account the examples are background noise and interfering strength of other simultaneous transmission. SINR attempts to create a representation of this aspect.

So we have implemented the TAS protocol by maintaining the details about the sender and receiver and the communication media in the network. We implement this through overhearing concept. This TAS implements grouping of nodes depending on the threshold value so that the communication will be easy.

In overhearing, the data will be clustered relevancy to it is node and the data transferred according to it. The cluster will be formed in such a way that, cluster head and gateway will be appointed to each cluster. The message will be transmitted from sender top receiver.

So the basic idea is to autonomously learn unknown and possibly random mobility parameters and to the group of mobile node with similar mobility pattern to the same cluster. The nodes in a cluster can then interchangeably share their resources for load balancing and overhead reduction , aiming to achieve scalable routing and efficient.

In our protocol, a secured code called threshold for on-line updating nodal contact probability with it is mean proven to converge to the true contacts probability. Subsequently, a set of functions are devised to form clusters and select gateway nodes based on nodal contact probabilities. Finally gateway nodes exchange the network information and perform routing. The results show that it is achieve higher delivery ratio and significantly lower overhead and end-to-end delay, compared with its non-clustering counterpart.

Overview of Selfish ResistedRouting

Since BSes are connected with a wired backbone, we assume that there are no power constraints and bandwidth on transmissions between BSes. We use intermediate nodes to denote relay nodes that function as gateways connecting an infrastructure wireless network and a mobile ad hoc network. We assume every mobile node is dual-mode; that is, it has ad-hoc network interface such as a WLAN radio interface and infrastructure network interface such as a 3G cellular interface[1]. DTR aims to shift the routing burden from the adhoc network to the infrastructure network by taking advantage of widespread base stations in a wirelesswirelessnetwork[2]. Rather than using one multi-hop path to forward a message to one BS, DTR uses at most two hops to relay the segments of a message to different BSes in a distributed manner, and relies on BSes to combine the segments. Demonstrates the process of DTR in a wirelesswireless network.

We simplify the routings in the infrastructure network for clarity. when a source node wants to transmit a message stream to a destination node, it divides the message stream into a number of partial streams called segments and transmits each segment to a neighbor node. Upon receiving a segment from the source node, a neighbor node locally decides between direct transmission and relay transmission based on the QoS requirement of the application. The neighbor nodes forward these segments in a distributed manner to nearby BSes. Relying on the infrastructure network routing, the BSes further transmit the segments to the BS where the destination node resides.

The final BS rearranges the segments into the original order and forwards the segments to the destination. It uses the cellular IP transmission method to send segments to the destination if the destination moves to another BS during segment transmission. Our DTR algorithm avoids the shortcomings of ad hoc transmission in the previous routing algorithms that directly combine an ad-hoc transmission mode and a cellular transmission mode[7]. Rather than using the multi hop ad-hoc transmission, DTR uses two hop forwarding by relying on node movement and widespread base stations. All other aspects remain the same as those in the previous routing algorithms (including the interaction with the TCP layer). DTR works on the Internet layer. It receives packets from the TCP layer and routes it to the destination node, where DTR forwards the packet to the TCP layer. The data routing process in DTR can be divided into two steps: uplink from a source node to the first BS and downlink from the final BS to the data's destination. Critical problems that need to be solved include how a source node or relay node chooses nodes for efficient segment forwarding, and how to ensure that the final BS sends segments in the right order so that a destination node receives the correct data. Also, since traffic is not evenly distributed in the network, how to avoid overloading BSes is another problem. First section will present the details for forwarding node selection in uplink transmission and Second section will present the segment structure that helps ensure the correct final order of segments in a message, and DTR's strategy for downlink transmission. The other will present the congestion control algorithm for balancing a load between BSes.

A long routing path will lead to high overhead, hot spots and low reliability. Thus, DTR tries to limit the path length. It uses one hop to forward the segments of a message in a distributed manner and uses another hop to find high-capacity forwarder for high performance routing. As a result, DTR limits the path length of uplink routing to two hops in order to avoid the problems of long-path multi-hop routing in the ad-hoc networks. Specifically, in the uplink routing, a source node initially divides its message stream into a number of segments, then transmits the segments to its neighbor nodes.

The neighbor nodes forward segments to BSes, which will forward the segments to the BS where the destination resides. Below, we first explain how to define capacity, then introduce the way for a node to collect the capacity information from its neighbors, and finally present the details of the DTR routing algorithm. Different applications may have different QoS requirements, such as efficiency, throughput, and routing speed. For example, delay-tolerant applications (e.g. voice mail, e-mail and text messaging) do not necessarily need fast real-time transmission and may make throughput the highest consideration to ensure successful data transmission.

Some applications may take high mobility as their priority to avoid hot spots and blank spots. Hot spots are areas where BS channels are congested, while blank spots are areas without signals or with very weak signals[4]. High-mobility nodes can quickly move out of a hot spot or blank spot and enter a cell with high bandwidth to a BS, thus providing efficient data transmission. Throughput can be measured by bandwidth, mobility can be measured by the speed of node movement, and routing speed can be measured by the speed of data forwarding. Bandwidth can be estimated using the nonintrusive technique proposed[10]. In this work, we take throughput and routing speed as examples for the QoS requirement. We use a bandwidth/queue metric to reflect node capacity in throughput and fast data forwarding. The metric is the ratio of a node's channel bandwidth to its message queue size[5]. A larger bandwidth/queue value means higher throughput and message forwarding speed, and vice versa. When choosing neighbors for data forwarding, a node needs the capacity information (i.e., queue size and bandwidth) of its neighbors. Also, a selected neighbor should have enough storage space for a segment.

To keep track of the capacity and storage space of its neighbors, each node periodically exchanges its current capacity and storage information with its neighbors. In the ad hoc network component, every node needs to periodically send "hello" messages to identify its neighbors. Taking advantage of this policy, nodes piggyback the capacity and storage information onto the "hello" messages in order to reduce the overhead caused by the information Exchanges. If a node's capacity and storage space are changed after its last "hello" message sending when it receives a segment, it sends its current capacity and storage information to the segment forwarder. Then, the segment forwarder will choose the highest capacity nodes in its neighbors based on the most updated information. When a source node sends out message segments, it chooses the neighbors that have enough space for store a segment to choose neighbors that have the highest capacity[8]. In order to find higher capacity forwarders in a larger neighborhood around the source, each segment receiver further forwards its received segment to its neighbor with the highest capacity. That is, after a neighbor node mi receives a segment from the source, it uses either direct transmission or relay transmission. If the capacity of each of its neighbors is no greater than itself, relay node mi uses direct transmission. Otherwise, it uses relay transmission. In direct transmission, the relay node sends the segment to a BS if it is in a BS's region. Otherwise, it stores the segment while moving until it enters a BS's region. In relay transmission, relay node mi chooses its highest-capacity neighbor as the second relay node based on the QoS requirement.

The second relay node will use direct transmission to forward the segment directly to a BS. As a result, the number of transmission hops in the ad-hoc network component is confined to no more than two. The small number of hops help to increase the capacity of the network and reduce channel contention in ad-hoc transmission. Algorithm 1 shows the pseudo-code for neighbor node selection and message forwarding in DTR. The purpose of the second hop selection is to find a higher capacity node as the message forwarder in order to improve the performance of the QoS requirement. As the neighborhood scope of a node for high capacity node searching grows, the probability of finding higher capacity nodes increases. Thus, a source node's neighbors are more likely to find neighbors with higher capacities than the source node. Therefore, transmitting data segments to neighbors and enabling them to choose the second relays help to find higher capacity nodes to forward data.

If a source node has the highest capacity in its region, the segments will be forwarded back to the source node according to the DTR protocol. The source node then forwards the segments to the BSes directly due to the three-hop limit. Though sending data back and forth leads to latency and bandwidth wastage, this case occurs only when the source nodes is the highest capacity node within its two-hop neighborhood[9]. Also, this step is necessary for finding the highest capacity nodes within the source's two-hop neighborhood, and ensures that the highest capacity nodes are always selected as the message forwarders. If the source node does not distribute segments to its neighbors, the higher capacity node searching cannot be conducted. Note that the data transmission rate of the ad hoc interface is more than 10 times faster than the cellular interface example 3G and GSM. Thus, the transmission delay for sending the data back and forth in the ad-hoc transmission is negligible in the total routing latency. By distributing a message's segments to different nodes to be forwarded in different directions, our algorithm reduces the congestion in the previous routing algorithms in the wirelesswireless networks.

When a node selects a relay to forward a segment, it checks the capacity of the node. Only when a node, say node mi, has enough capacity, the node will forward a segment to node mi. Therefore, even though the paths are not node-disjoint, there will be no congestion in the common sub-paths. In which the source node is in the transmission range of a BS. The value in the node represents its capacity. There exist nodes that have higher capacity than the source node within the source's two-hop neighborhood. If a routing algorithm directly let a source node transmit a message to its BS, the high routing performance cannot be guaranteed since the source node may have very low capacity. In DTR, the source node sends segments to its neighbors, which further forward the segments to nodes with higher capacities. The source node has the highest capacity among the nodes in its two-hop neighborhood. After receiving segments from the source node, some neighbors forward the segments back to the source node, which sends the message to its BS. Thus, DTR always arranges data to be forwarded by nodes with high capacity to their

BSes. DTR achieves higher throughput and faster data forwarding speed by taking into account node capacity in data forwarding.

System Model

Network Configuration

We consider networks with 35 nodes. The mobile nodes are randomly and uniformly deployed in a square area of size 1000×800 m. The node transmission range is set to 250 m. Nodes move at the speed of 5m/ms across in the network field among Access Points. Data traffic is generated for 100 packets per millisecond over the whole network. Each packet is randomly and uniformly assigned to a source, excluding nodes that are one hop from the sink. The chosen source queues the assigned packets and transmits them as soon as possible. The maximum queue length per node is set to 50 packets. A newly generated packet is accepted by the source only if its buffer is not full.

Neighbor selection

In this module, an intermediate node assigns the highest priority to the packet with the closest deadline and forwards the packet with the highest priority first. Queue length threshold is set to avoid queuing congestion, we set up a space utility threshold TUs for each node as a safety line to make the queue scheduling feasible. In CoCoWa, after receiving a forward request from a source node, an intermediate node Ni with space utility less than threshold e TUs replies the source node. The replied node Ni informs the source node about its available workload rate, and the necessary information to calculate the queuing delay of the packets from the source node. The source node selects the replied neighbor nodes that can meet its deadline for packet forwarding based on the calculated queuing delay.

Packet Scheduling

In this module, we further reduce the stream transmission time, a distributed packet scheduling algorithm is proposed for packet routing. This algorithm assigns earlier generated packets to forwarders with higher queuing delays and scheduling feasibility, while assigns more recently generated packets to forwarders with lower queuing delays and scheduling feasibility, so that the transmission delay of an entire packet stream can be reduced.

We use t to denote the time when a packet is generated, and use CoCoWa technique to denote the delay requirement. Let WS and WI denote the bandwidth of a source node and an intermediate node respectively, we use TS→I - Sp /WS to denote the transmission delay between a source node and an intermediate node, and TI→D - Sp/WI to denote the transmission delay between an intermediate node and an AP. Let Tw denote the packet queuing time and Tw(i) denote the packet queuing time of ni. The source node needs to calculate Tw of each intermediate node to select intermediate nodes that can send its packets by the deadline.

Selfish node elimination

In CoCOWa, the selfish nodes overhear and cache packets. From the overhearing, the nodes know who have received the packets. When a source node begins to send out packets, it scans the content for duplicated chunks in its cache. If the sender finds a duplicated chunk and it knows that the selfish has sent this chunk before, it replaces this chunk with its signature.

Performance evaluation

All mobile nodes are randomly scattered with a uniform distribution. Randomly select one of the deployed nodes as the source node.

We evaluate our proposed method with respect to the following metrics: Throughput, E2E latency, Packet loss ratio.

CoCoWa Throughput:is the ratio of the number of report messages the sink receives to the total number of report messages the source node sends.

Packet loss ratio:measures the ratio of packets have been dropped during transmission time

End to end latency: It refers to the time taken for a packet to be transmitted across a network from source to destination.

These parameter values are recorded in the trace file during the simulation by using record procedure. The recorded details are stored in the trace file. The trace file is executed by using the Xgraph to get graph as the output.

Conclusion

wirelesswireless networks have been receiving increasing attention in recent years.Awirelesswireless network combining an infrastructure wireless network and a mobile ad-hoc network leverages their advantages to increase the throughput capacity of the system. However, current wirelesswireless networks simply combine the routing protocols in the two types of networks for data transmission, which prevents them from achieving higher system capacity. In this paper, we propose a Distributed Three-hop Routing (DTR) data routing protocol that integrates the dual features of wirelesswireless networks in the data transmission process.

In DTR, a source node divides a message stream into segments and transmits them to its mobile neighbors, which further forward the segments to their destination through an infrastructure network. DTR limits the routing path length to three, and always arranges for high-capacity nodes to forward data. It is not like most existing routing protocols, DTR produces significantly lower overhead by eliminating route discovery and maintenance. It has distinguishing characteristics of short path length, short-distance transmission, and balanced load distribution provides high routing reliability with high efficiency. DTR also has a congestion control algorithm to avoid load congestion in BSes in the case of unbalanced traffic distributions in networks. Theoretical analysis and simulation results show that DTR can dramatically improve the throughput capacity and scalability of wirelesswireless networks due to its high scalability, efficiency, and reliability and low overhead.

Reference

[1] G. N. Aggelou and R. Tafazolli. On the relaying capacity of next generation gsm cellular networks. IEEE Personal Communications Magazine, 2001.
[2] B. Bengfort, W. Zhang, and X. Du. Efficient resource allocation in wirelesswireless networks. In Proc. of WCNC, 2011.
[3] L. M. Feeney, B. Cetin, D. Hollos, M. Kubisch, S. Mengesha, and H. Karl. Multi-rate relaying for performance improvement in ieee 802.11 wlans. In Proc. of WWIC, 2007.
[4] J. Cho and Z. J. Haas. On the throughput enhancement of the downstream channel in cellular radio networks through multihop relaying. IEEE JSAC, 2004.
[5] C. Sarr, C. Chaudet, G. Chelius, and I. G. Lassous. A Node- Based Available Bandwidth Evaluation In IEEE 802.11 Ad Hoc Networks. IJPEDS, 00(00):1–21, 2005.
[6] P. Gupta and P. R. Kumar. The capacity of wireless networks. IEEE TIT, 2000.
[7] H. Y. Hsieh and R. Sivakumar. On Using the Ad-hoc Network Model in Wireless Packet Data Networks. In Proc. of MOBIHOC, 2002.
[8] B. Liu, Z. Liu, and D. Towsley. On the capacity of wirelesswireless networks. In Proc. of INFOCOM, 2003.
[9] T. Liu, M. Rong, H. Shi, D. Yu, Y. Xue, and E. Schulz. Reuse partitioning in fixed two-hop cellular relaying network. In Proc. of WCNC, 2006.
[10] C. Sarr, C. Chaudet, G. Chelius, and I. G. Lassous. A Node- Based Available Bandwidth Evaluation In IEEE 802.11 Ad Hoc Networks. IJPEDS, 00(00):1–21, 2005.