Rate-Aware Cost-Efficient Multiratecasting Routing in Wireless Sensor Networks

by

Xidong Liu

Thesis submitted to the
Faculty of Graduate and Postgraduate Studies
In partial fulfillment of the requirements
For Masters of Applied Science degree in
Electrical Engineering

School of Electrical Engineering & Computer Science
Faculty of Engineering
University of Ottawa

Abstract

In the multiratecasting problem in wireless sensor networks, the source sensor is usually required to report to multiple destinations at different rates for each of them. We present a MST-based rate-aware cost-efficient multiratecast routing protocol (MSTRC). The proposed MSTRC examines only one set partition of destinations at each forwarding step. A message split occurs when the locally-built minimum spanning tree (MST) over the current node and the set of destinations has multiple edges originated at the current node. Destinations spanned by each of these edges are grouped together, and for each of these subsets the best neighbor is selected as the next hop. We also suggested a novel face recovery mechanism to deal with void areas, when no neighbor provides positive progress toward destinations. It constructs a MST of current node and destinations without the progress via neighbors, and for each set partition of destinations corresponding to an edge e in MST, the face routing keeps going until a node that is closer to one of these destinations is found, allowing for greedy continuation, while the process repeats for the remaining destinations similarly. Our experimental results demonstrate that MSTRC is highly rate-efficient in all scenarios, and unlike existing solutions, it is adaptive to destination rate deviations.

Acknowledgements

I would give my sincerest gratitude to my supervisor Prof. Amiya Nayak and Prof. Ivan Stojmenovic for their exceptional guidance and support not only in this thesis but also in my study and research. Special thanks to Dr. Cheng Wang who gave me valuable advices, suggestions and encouragements. Without his instructions, I could not have learned so much and complete this thesis. I wish to express my cheers to my friends, Zhen Huang, Dandan Liu whom I learned from. We will be friends as we were before. Finally I would like to thank my parents who always understand and support my ideals, and this work is dedicated to them.

Contents

List of Figures

List of Acronyms

GFG	Greedy Face Greedy
GG	Gabriel Graph
GMP	Geographic Multicast Routing Protocol
GMR	Geographic Multicast Routing
GPS	Global Positioning System
LAR	Localized Geographical Routing
MANET	Mobile Ad-hoc Network
MRM	Maximum Rate Multicast
MRP	Multiple Relay Pointscription
MST	Minimum Spanning Tree
MSTRC	MST-based Multiratecast Protocol
PBM	Position Based Multicast
PID	Proportional Integral and Derivative
RCM	Optimal Rate Cost Multicast
SPBM	Scalable Position Based Multicast
UDG	Unit Disk Graph
WSN	Wireless Sensor Network

Chapter 1

Introduction

1.1 Backgroud

A wireless sensor network (WSN) is a type of self-configured, self-organized, static or mobile network that consists of a set of networked sensor nodes. Sensor nodes are usually small size, inexpensive wireless equipments with limited communication, computation and energy resources. They sense environment and communicate among each other by using wireless links. They work in a distributed way, and collaborate to perform automated tasks requiring sensing capabilities. The number of such sensors in a WSN is expected to be large, in the order of hundreds or thousands. The wireless sensor communication is usually performed through wireless channel in a multi-hop way. Data is sampled at a source node, transmitted to a neighbor node and relayed by an intermediate node until the destination node is reached. Routing protocols for wireless sensor networks are used to transmit messages from sources to destinations. They can be classified as unicast, broadcast and multicast. The unicast routing is used to send a message generated by a sensor node to a single destination or sink, the broadcasting is used to send a message from a sensor node to every other node in the network and the multicasting is used to deliver messages from a single source to a set of destinations. [32]

1.1.1 Multicast Routing

There are lots of possible applications of WSNs, including the habitat monitoring, wildfire detection, pollution monitoring, etc. In many of these scenarios, there are applications in which a single sensor needs to send the same data to multiple destinations. Those applications can benefit from the use of multicast communications to reduce bandwidth consumption in the network. Examples of those applications include the data replication, assignment of tasks or sending of commands (especially in sensor and actuator networks) to a specific group of sensors, queries to multiple sensors, etc. For instance,one real application we are using is the control of water sprinklers for water irrigation, in which, in many cases, sensors may need to send the same information to multiple actuators. Actuators are in charge of opening or closing valves depending on the need to water some areas based on information measured from sensors. Some of these applications are data-intensive; therefore, it is of paramount importance for them, to count on an efficient multicast mechanism being able to alleviate the overall consumption of resources in the network.

Multicasting is a technique used to deliver messages efficiently from a source to a set of destinations. Multicasting protocols try to minimize the consumption of network resources and take advantages of the fact that some parts of the paths from the source to the destinations can be shared by multiple destinations. The larger the path shared, the lower the overall bandwidth consumption is obtained. There have been a lot of multicast routing proposals for ad hoc networks[33], each of them based on different design decisions. Unfortunately, they cannot fulfill the unique requirements of WSNs effectively. They are mostly designed to deal with highly mobile nodes, with a higher processing and storage capacity, and a much limited amount of nodes. In addition, WSNs are characterized by their topological changes due to node failures or a duty-cycle operation. These characteristics make localized routing algorithms [12] more appropriate for WSNs. Unlike centralized ones, localized algorithms do not need to know the complete topology to take routing decisions. Furthermore, centralized algorithms introduce too much overhead to be used in WSNs.

Providing efficient multicast routing in WSNs poses special challenges compared with the unicast data delivery. In fact, the problem of computing a minimal bandwidth consumption multicast tree in wireless multihop networks was recently proven to be NP-complete in [31]. This becomes specially challenging when the overhead needs to be kept low due to the limited battery, storage capacity, bandwidth, and processing power of sensor nodes.

A number of multicast routing protocols have been proposed to enable the multicast applications. They can be divided into two main groups: multicast protocols that requires global structure such as tree based [3, 17, 29, 35, 43] or mesh based protocols [4, 10, 11, 22], and protocols that use only local information such as geographic routing or position based routing [1, 25, 32].

Multicast protocols that require a global structure need to construct and maintain a routing distribution table before and during the routing processing. These are considered very challenging for wireless sensor networks for two reasons: first, the network nodes in the sensor network normally have limited CPU power and memory so it is challenging to store the routing table in a node. Secondly, some characteristics such as the node failures, the membership changes, the topology changes all makes it very expensive for the routing table update on each node. These kind of problem are being well solved on former works.

GMR designs a multicast protocol that focuses on the localized construction of bandwidth optimal multicast trees. Thus, given that the focus is on the efficient neighbor selection function, positions of destinations are assumed to be known to multicast sources. Furthermore, given that the number of destinations is expected to be low for the multicast scenarios considered for sensor networks, the proposed scheme is mainly concerned with an enhanced neighbor selection criterion. The proposed multicast protocol GMR selects neighbors based on the cost over progress framework, which was first introduced by Kuruvila *et al.* [21] (for routing problem) integrated with a greedy neighbor selection. The cost function that considers the number of transmissions based on the results of Ruiz *et al.* [31], shows the optimality of a multicast tree, since bandwidth consumption needs to be evaluated in terms of the minimization of the number of transmissions performed. GMR avoids hard-to-tune parameters, has lower computational costs, and computes multicast paths with a lower overall cost. Moreover, the GMR scheme is general enough to be easily coupled with any scalable group management scheme.

1.1.2 Multiratecast Routing

There are WSN applications that a single sensor needs to sample data and propagate them to potential consumers at different required rate. For instance, when an environment monitoring sensor network monitors weather changes in a forest, sample temperature or earth vibration data may be reported at different rate to different sinks. Multiratecast protocols are proposed in this thesis to try to minimize the consumption of network resources by taking advantage of sharing some of the paths from the source to the destinations while each destination's specific rate requirement can still be satisfied.

One important multiratecast application is for a network with backup base stations or backup sinks. Data is collected by sensor nodes and are transmitted and stored in base station (sink) nodes for further analyze and study. The data rate for sending to a primary sink is high, for a second sink is medium and the rate is lower for a third sink node and further lower for rests of back up sinks. If the primary sink fails, the second sink node takes over the network. If the second sink fails, the third sink node takes over the network and it continues. The total backup network requires that data are captured and sent to different sink nodes by different rate range from high to low. There are scenarios that infrastructure network does not exist. The backup sink nodes may need wireless data transmission for whole processing. For example, the backup sinks can be a few laptops to capture temperature or chemical data in special circumstances. Human observers may check data on laptops for certain purposes. If primary sink failed, a human observer needs to collect information from the backup sink and can still receive useful information.

Sinks may run a separate protocol which checks if the primary sink failed. If so, other sinks may remove it from the network, select a new primary sink, and send new reporting rates requirements to all sensors. Once rates for every sink are determined, a dynamic routing protocol should be designed to support multiratecast efficiently. The protocol also should be able to adapt to rate changes quickly when the sink nodes join or leave the network or when the network topology is changed. As the following example, there are three sink nodes in the network "a","b", "c" which requires data rates of 10, 5, 8 respectively. Sink "a" is the primary node, source node S collects data and sends full information to sink "a" at rate 10. Data will also be split and sent to backup node "b" and "c" at rates of 5 and 8. When a relay node branches message to forwarding nodes at different rate, data diffusion and reassemble may be required. Paper [14] describes a data-centric routing that is related to data content processing. However, the data process part is another research topic and we focus on the routing path study in this thesis.

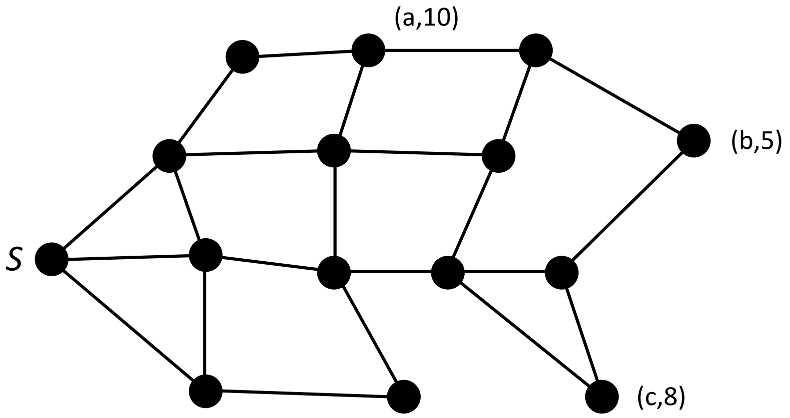
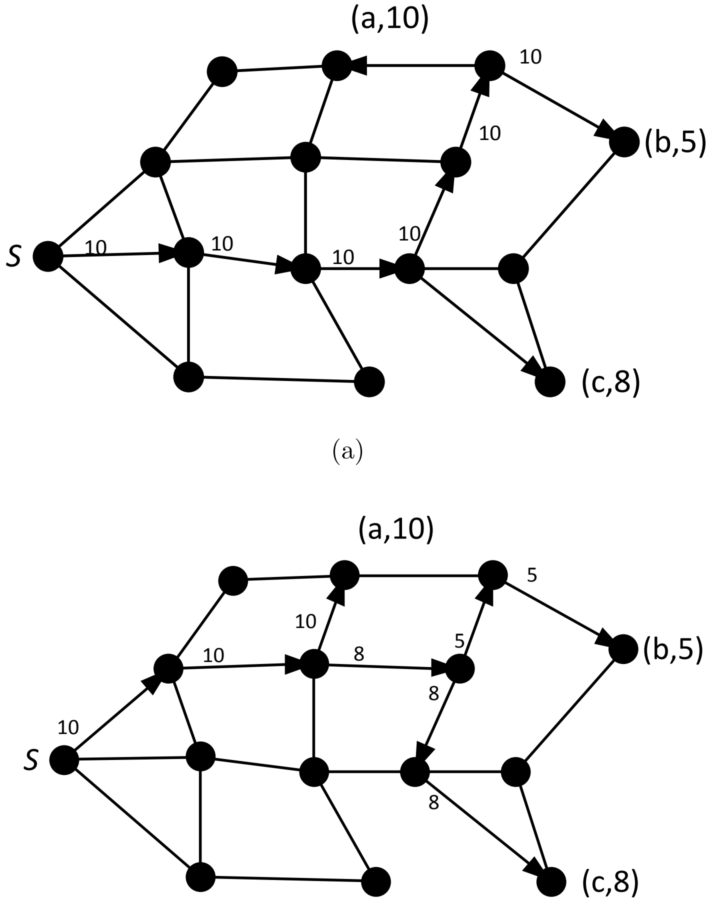


Figure 1.1: Network with Multiple rate destinations

For the network example in Figure 1.1, there can be two different routing paths as showing in Figure 1.2(a) and Figure 1.2(b). They have a same transmission number of 6 (hop count). Protocols that use hop count as a performance guidance could not tell the difference of routing paths in Figure 1.2(a) and Figure 1.2(b). However, if we consider the sum of the data rate as the routing cost, the total data transmission rate cost at routing paths in Figure 1.2(b) is 51 while the total rate cost in Figure 1.2(a) is 60. Routing path in Figure 1.2(b) is more efficient in a rate cost metric.

There are multicast protocols that use only local information for multicast routing. Existing multicast protocols include PBM [42], GMR [32], and GMP



(a)

(b)

Figure 1.2: Different rate cost paths

[25]. However, they do not satisfy multiratecast in common that they do not consider the specific multi-rate routing requirement in routing decisions. They mainly use hop count as the routing performance metrics, but it may not reflect correct routing costs for a multiple data transmission where paths with the same hop count may transmit data at different data rates at each transmission.

There are some existing rate aware multicast protocols. However, they are either protocols working on changeable rate or protocols requiring global tree construction suitable for wired sensor network. They do not fulfill the application requirement in the thesis studies. The protocol proposed in [27] is a protocol that monitors the wireless link and adapts the data transmission rate to achieve the lowest total transmission time. It does not provide a solution for the specific application requirement that each destination request multiratecast data by a different rate. Protocol in [34] is the only existing protocol that related to the multiratecast, however, it is a tree based non-localized algorithm based on the edge cost calculation. In this protocol, the source node constructs a multicast tree by sending and receiving explore/ack messages to destinations. As mentioned above, it has a large amount of overhead control messages and is not efficient in WSNs.

Routing protocols that support multiratecast should consider the data rate in a routing path selection and also considering the wireless multicast advantages. In a wireless network, a message transmitted by a node is received by all its neighbors. This means that one single message is sent at a certain rate, it is received by all the relay nodes. This property is commonly known as the wireless multicast advantage. The relay node may continue data forwarding at same rate, or it may forward selective part of coming data at a reduced rate for destinations that only require lower data rate. Splitting data and lower transmission rate according to destinations' requirement can reduce the overall routing cost. However, existing multicast protocols are not able to achieve the rate based optimization because considering only the hop count in performance metrics may not reflect the optimal path for network with multiple-rate destinations.

In the thesis, a purely localized network layer multiratecast protocols are proposed to support for dynamic multiple rate routing in WSNs. The protocol proposed is MST-based rate-aware cost-efficient multiratecast protocol (MSTRC). At each forwarding node, MSTRC applies destination set partition and then bulid a MST-like tree by using a revised prim algorithm with considering of the transmission rate. Then it calculates the cost to select the best forwarding nodes. Three different metrics to find the best next forwarding nodes are used. MSTRC ensures the minimal rate-based cost to next hops at each forwarding process. If the packet forwarded to a void area, the GFG algorithm would be applied in a multiratecast version and find the route for returning to the greedy mode.

The MSTRC protocols have three unique characteristics: Firstly, they are purely strict localized algorithms that use only node position information to make routing decisions. Secondly, they should consider the destination data rate when making routing decision, ensure paths that each destination receives data at its requested rate. Thirdly, they aim at overall multicast paths from a single source to multiple destinations that use minimum network resources; we define it as rate based cost including hop count and data rate. The Multiratecast protocols are unique from existing multicast protocol in that they adjusts routing paths according to data rate as well as hop count and distance. For a network with multiple-rate requirements in destinations, they are able to efficiently find optimal paths for all destinations with the lowest overall rate cost.

1.2 Problem Statement

We consider a request-driven, large-scale, static WSN. Nodes (sensors and sinks) are aware of their own positions by either attached GPS devices or existing localization algorithms. They share their location information with one-hop neighbors by periodic beacon messages.

Sensors are equipped with an omni-directional antenna and able to communicate directly if their distance is smaller than the maximum transmission radius R. In other words, the network is modeled as a unit disk graph (UDG). Each edge (communication link) is associated with a cost, defined as the minimum transmission power for this edge. It is computed using the first order radio model [30], $e(d) = \beta d^{\alpha} + c$, where $\beta > 0$ is the transmission amplifier, d is the transmission distance (edge length), α is a signal attenuation factor, and $c > 0$ is the energy for running electronic circuit. Sensors can adjust their transmission power so that they send reports along a selected edge using the least possible power, which equals the associated cost of the edge. Moreover, each sensor is aware of its own residual energy. If the remaining energy level falls below a threshold E_{th}, it will transmit a "quit" message to neighbors and switch off for energy saving.

The sending node generates the data packets that need to distribute to destinations. The destinations positions and rate requirements are priori knowledge to message senders. Most of the time, packets can not be sent to the destinations directly, the sender would deliver the data to neighbors that are closer to destinations.

Nodes that receive the data packets from their neighbors and are not an destination are considered as forwarding node. They perform as a bridge between the sender and destinations. The data in the packets implies to which destinations they should be sent and the forwarding nodes would send the packets to the destinations if they are connected directly or send them to neighbors that could have a good chance to find the destinations.

Destination nodes receiving the data, which are of the same size (depending on the type of data requested) and its rate may be different. They are treated equally for the transmission. So, a packet in forwarding nodes may have to be splitted into multiple data packets. In this case, the rate cost is correlated with the splitting scheme.

The research goal is to develop, based on the above network model, a localized multiratecast data propogation framework that satisfies a given set of constraints.

1.3 Motivation

Sensors are normally powered by low-energy batteries. Manual replacement or recharge of sensor batteries is often infeasible due to operational factors such as inaccessibility to the sensory field or tight maintenance budget. It is highly desirable to prolong the lifetime of the network as a whole by minimizing and balancing energy usage among individual sensors. In critical real-time scenarios such as disaster management, emergency rescue and battlefield surveillance, information is often required to arrive at a certain number of destinations at different ratio.

Existing data routing algorithms usually require centralized control and emphasize mostly energy efficiency. They seldom consider the multiple rate requirement. The few known localized, rate-aware algorithms have major drawbacks in energy saving as well as in rate arrangement and thus have limited effect both on prolonging the network lifetime and on meeting the cost and rate requirement. Motivated by the insufficiency and incompleteness of previous works, we address the problem of multiratecast routing in data propagation with respecting to given computation requirements.

1.4 Contributions

The objectives of this thesis are: (i) to understand the limitations of routing in wireless sensor networks, (ii) to investigate existing multiratecast routing solutions and find the problems which affect the performance of efficient data propogation and (iii) to propose solutions to improve the multiratecast routing in wireless sensor network.

The contributions are the following:

1. We review the existing unicast, multicast and multiratecast schemes and point out their limitations. The existing unicast and multicast protocols are meet with the most of requirements of different applications. But the do not take the transmission rate into consideration. It is very hard and low efficient to simply use these protocols to transmit data with multiple rate. Current multiratecast protocols are proposed by following normal multicast schemes and do not take advantages of the multirate and have not too much improvement on the cost.

2. To improve transmit efficiency in multiratecast routing, we proposed a multiratecast protocol that employed the MST algorithm. The MST problem is similar to Steiner tree problem but such trees may be efficiently computed in time $O(nlogn)$. We improved the algorithm to adapt the rate assumption. This protocol is also cost aware because we used three kinds of metrics to measure the cost in the neighbor selection procedure. It ensures us to find the most efficient forwarding nodes.

3. We also did a bunch of experiments to exam the the efficiency of our algorithm. MSTRC protocol and Two similar protocols are implemented in our simulation platform. We compare them in different network environment and dimensions. Simulation results indicate that the MSTRC protocol is dramatically more efficient and achieves a higher throughput under different network configurations.

1.5 Organization

This thesis is organized as follows:

- Chapter 2 is a background on the routing technology in wireless sensor and ad hoc network. This chapter presents the related work which has been done, it provides a comprehensive study on the unicast,multicast and multiratecast in wireless sensor network. A brief description of current issues and the corresponding solutions are described in this chapter.We discuss the solutions of multiratecast problems.
- Chapter 3 introduces the concept of multiratecast, which commonly arise in data propagating in wireless sensor network. A novel and simple algorithm is presented in this chapter to increase the efficiency. Then we propose a new multiratecast data routing protocol for wireless sensor networks.
- The corresponding simulation results, comparison, analysis and limitation are discussed in chapter 4.
- Chapter 5 discusses the conclusion and future work.

Chapter 2

Literature Review

There have been a lot of routing protocols developed for wireless sensor networks. Based on a different data distribution, they can be divided to unicast, multicast, and Broadcast protocols. Unicast routing is a one-to-one routing that a message generated by a sensor node is sent to a single destination or sink. Broadcasting is one-to-all routing that is used to send a message from a sensor node to every other node in the network. Multicast routing is used to deliver messages from a single node to a set of destinations. Due to the limitation of wireless sensor networks such as limited energy resources, high cost of transmission, limited processing capabilities and restricted lifetime, it is inefficient to use the broadcast (flooding) as a routing scheme in sensor networks. Multicasting protocols try to minimize the consumption of network resources by taking advantage of the fact that some parts of the paths from the source to the destinations can be shared by multiple destinations. There is also a scenary that the destinations required different transmission rate in multicasting routing. The proposed multiratecasting protocols not only need to solve the problems in multicasting routing but also have to handle the rate arrangement.

2.1 Unicast Protocols in WSN

Based on how a routing path is found from a source node to a destination, unicast protocols can be divided to two main categories: non-geographical and geographical routing.

Most unicast protocols are non-geographical routing protocols. They do not directly use geographical information in a routing decision. They can be further divided into three approaches: the proactive approaches, the reactive approaches and hybrid approaches. The proactive routing protocols have characteristic that each node in the network maintains a route to every other node in the network at all times. Periodic routing information update and maintenance is required for each node. Examples of proactive routing protocols are DSDV [28] and OLSR [15]. The reactive routing protocols are also called on-demand routing. A routing path is discovered and computed based on the routing demand. It has less overhead compared to proactive approaches but have longer route acquisition latency. Examples of reactive routing protocols are AODV [18] and DSR [8]. Hybrid routing approaches combine the proactive and reactive routing protocols in various ways to allow flexible routing based on network characteristics.

Geographical unicast routing protocols [1, 5, 12, 19, 33, 36, 38, 40] had characteristics that network nodes are able to continually obtain geographical information through a type of Location Service such as GPS and the geographical information is used in a routing path selection. Geographical routing can be further divided into non-localized protocols and localized protocols. Non-localized protocols utilizes geographical coordinate to perform global routing discovery from the source node to destination node such as LAR [33]. Localized geographical routing protocols [1, 5, 38] have proven to be very effective in providing unicast routing in the common resource-constrained scenarios [12, 36, 38] that WSNs present.

They work with local information, require a low computational cost, adapt very fast to changing network conditions, and are able to route messages with a very low control overhead. Each node taking part in a routing process takes decisions about which neighbor is the best one to be selected as the forwarder in order to carry the message as nearer to the position of the destination as possible. Thus, the information about position is fundamental. Although the use of hardware-based positioning systems such as GPS might be possible, there are scenarios (e.g., indoor) in which they cannot be effectively used. However, they can also work based on virtual coordinates, as shown in [6] and [2]. Similarly, the proposed multiratecast protocols in this thesis can also work based on these virtual coordinates. Greedy Face Greedy [1] is an example of localized geographical routing that is duplicated as in GPSR [19]. The routing in GFG composes two schemes: greedy forwarding mode for nodes that have neighbors closer to destination than itself and recovery mode for nodes do not have closer neighbors to destination. In recovery mode, GFG applies face routing until greedy mode routing can be resumed.

2.2 Greedy Face Greedy

Greedy Face Greedy is a location-based Unicast routing protocol for wireless network. It guarantees packet delivery in a connected planar graph. It consists two routing modes: greedy routing mode and recovery mode. Packet routing starts from greedy mode. Each GFG packet contains destination's location. Every node in network periodically broadcasts a beacon packet within its own radio range which contains its current location information. Every node stores its neighbors' location information after receiving beacon packets. Starting from the original node, each greedy mode forwarding node calculates the distances from every neighbor node to the destination node. The neighbor node located closest to the destination node is selected as the next hop. When a forwarding node cannot find any node that is closer to destination than itself, the routing is known as the local optimum situation and is changed from greedy mode to recovery mode. Packet routing in recovery mode traverses along faces of a planar sub-graph of original network graph. The most common planar sub-graph used is the Gabriel Graph (GG), which contains an edge CA if the disk with diameter CA contains no other nodes inside it. Note that node C may decide which of its edges belong to GG based on the position of itself and its neighbors, without sending any message for the purpose of constructing GG. Face routing continues along faces until the packet reaches a node that is closer to the destination than the node where greedy mode failed. Then the routing resumes to greedy mode. Packet is routing in either greedy mode or recovery mode until it reaches destination.

2.3 Multicast Protocols in WSN

Providing efficient multicast routing in wireless sensor networks has specific challenge compared to the Unicast routing. Besides the common resource limitations of sensor network, multicast routing need compute efficient multicast distribution paths that making use of a minimal amount of control information, sending as few as possible duplicate packets, and consuming the minimal overall network resources to all destinations.

There have been a lot of multicast routing proposals for wireless sensor networks; each of them based on different design decisions. Most of them are non-geographical routing protocols and are not designed to work in a localized way.

Based on the routing paths structure they employ, they can be divided into tree-based protocols, mesh-based protocols and hybrid protocols [35, 43]. Tree based protocols have one shared path from the source node to each destination. They use lower number of relay nodes compared to mesh based protocols but the tree needs to be reconstructed when links break due to node mobility. In addition, they also rely on periodic flooding that is a costly operation for sensor network. Examples of tree based protocols are MAODV [29], ADMR [41], and AMRIS [26]. Mesh-based protocols expand a multicast tree with additional paths so that they have multiple paths from the source node to each destination. They have more redundancy in routing structure compared to tree-based protocols and the additional paths can be used to forward multicast data packets when some of the links break. The mesh-based protocols have proven to be particularly suited for scenarios with high mobility rates. However, the maintenance of these structures through periodic broadcasts and the large amount of duplicate forwarding makes them impractical for sensor network. Examples of mesh-based protocols are CAMP[10], ODMRP [11], NSMP [22], and DCMP [4]. Hybrid protocols are mix of tree and mesh protocols such as AMRoute [43] and MCEDAR [35]. For all above three types of protocols, their non-localized operation produces an excessive control overhead for wireless sensor network.

There have been multicast protocols for wireless sensor networks that take geographical information to perform a multicast routing. They are also called Geo-Multicast protocols. However, applying geographical routing in multicast faces specific challenges such as how to select next forwarding nodes and perform efficient routing based on position information for multiple destinations. Various geographical multicast protocols have been proposed such as [16, 17, 25, 32, 42, 46, 47] to explore different ways to perform efficient Multicast. DDM [17] combines Unicast data tables for multicast data forwarding which requires additional overhead and makes it best suitable only for small multicast groups. LAM [46] makes use of broadcast so that it is impractical for wireless sensor networks. Protocol in [16] uses position information to build a multicast tree that aims at minimize the number of links. However, its cost calculation is not optimal because the wireless medium is one-to-many communication and cost should be better characterized by number of transmission nodes instead of number of links. Because of the design challenges of multicast, localized protocols are believed to be better suitable for wireless sensor networks [37]. PBM [25], GMR [32] and GMP [42] are examples of localized geographical multicast protocols.

2.3.1 Localized Geographical Multicast

Localized multicast protocols usually use local geographical-position information of nodes to make routing decisions. The position information can be acquired by location service such as GPS or virtual coordinates as in [2] and [6]. The position information is included in the message header and is used at each forwarding node for the next routing nodes selection.

Position Based Multicast (PBM)[42] is one of the localized geographical multicast protocols. Although it is not initially thought for sensor networks, it fulfils most of the desired design criteria of localness and limited network overhead. PBM is a generalization of Greedy-Face-Greedy routing to operate over multiple destinations.

It builds a multicast tree, whose shape can vary from the shortest path tree, to an approximation of a minimum cost multicast tree depending on a parameter denoted as λ. Authors in try to find a good trade-off between the total number of nodes forwarding the message and the optimality of individual paths towards the destinations. Each node evaluates all possible subsets of neighbors W using a function to evaluate each $w \in W$. $f(w) = \lambda N + (1 - \lambda)S$, $0 \leq \lambda \leq 1$, where N is the number of neighbors in the considered subset $(|w|)$ divided by the total number of neighbors n, and $S$is the summation of the minimal distances from nodes in W to destinations, normalized by the summation of distances from the current node to all destinations. From all possible subsets of neighbors W, the current node selects the one with optimal $f(W)$. If the best subset of neighbors is a single node, then that node will be the only relay for all the destinations. If a subset of neighbor nodes is selected, then each of the nodes in the subset will take care of routing the data messages to part of the destinations. If at some node there are no nodes providing advantages towards one or more destinations, the authors use, only toward those destinations, a variant of face routing like the one we describe in the previous section. The main problem with this approach is that determining the optimal value for λ is not a trivial task.

In fact, the authors evaluated different values of λ but they never came out with a determination of an optimal value. An additional issue is the fact that the algorithm is computationally expensive. Evaluating all possible neighbor subsets has an exponential computational cost as the number of neighbor increases.

For networks with a very large number of multicast receivers, PBM may not scale well due to the need to include all destinations in multicast data packets. To improve the scalability, another protocol called scalable position based multicast for mobile ad hoc networks (SPBM)[39] was designed. It uses the geographic position of nodes to provide a scalable group membership scheme and to forward data packets. SPBM is mainly focused on the task of managing multicast groups in a scalable way. However, they fail to provide efficient multicast forwarding, because they use one separate unicast geographic routing for each destination.

GMR[42] is another localized geographical multicast protocol proposed by Shibo Wu and K. Selcuk Candan. Starting from the source node, it tries to build a virtual multicast Steiner tree by applying a stateless reduction ratio heuristic calculation. It traverses all destinations and tries to find a pair of destinations with highest reduction ratio and then creates a virtual destination node to represent the two destinations. It then repeats the calculation for new destination set including new virtual node and all rest of destinations to create next best virtual node until all destinations are represented by one final virtual node. After that, the multicast routing is simplified as unicast from the current node to final virtual node. GMR tries to find a neighbor node closest to the final virtual node as a next hop and sends messages to the neighbor node. After the neighbor node receives the messages, it applies same reduction ratio heuristic calculation again to construct a virtual tree and forwards message to the next final virtual nodes. Same calculation continues until message reaches destinations. If no neighbor can provides forwarding to final virtual node or if it is not beneficial to create a virtual destination node, message is duplicated and same algorithm applies on each of the message routing.

The advantage of GMR is that it simplifies the multicast routing to unicast by a linear virtual node calculation at each forwarding node. The disadvantage of GMR is that it calculates virtual destination by two destination nodes at a time. The created new virtual node may change the virtual destination topology. It will be less efficient routing if destinations are scatted around the source node and may end of a virtual node on opposite direction of some destination nodes.

2.3.2 Multirate Multicast Protocols

All the above routing protocols discussed in previous sections focus on single rate multicast routing where they consider all packets transfer at a constant rate. However, there are some protocols that perform routings at variable rates [13, 23, 45].

In [44],the authors propose a distributed flow control scheme for multirate multicast, based on the Proportional Integral and Derivative (PID) controllers. The PID controller at each router computes its expected incoming rate and feeds back this rate to its upstream router, such that the local buffer occupancy can be stabilized at an appropriate value. The proposed MRM controller achieves the fairness in two aspects: 1) the intrasession fairness, i.e., the receivers from the same source within the same multicast session can receive data at different rates, if they subscribe networks with different capacities; 2) the intersession fairness, i.e., the link bandwidth is fairly shared among multiple multicast sessions from different sources.

Uyen and Xiong [27] proposed a rate-adaptive multicast protocol for MANET. It is based on a simple routing metric: among several paths between a sender and a receiver, the routing protocol selects the path with the lowest total transmission time. The authors claim that low transmission time helps increase the throughput and reduce the energy consumption. The protocol monitors the quality of wireless links and suggests optimal transmission rate that helps reach its goal. This protocol considers data rate adaption in the routing path selection, but as mentioned above, does not consider the multiratecast requirement and is quite different from the problem we are trying to solve.

The paper [48] addresses the performance optimization for scalable video coding and multicast over networks. Multipath video streaming, network coding based routing, and network flow control are jointly optimized to maximize a network utility function defined over heterogeneous receivers. Contextual priors of scalable video layers are imposed on the flow routing optimization problem, seeking to guarantee the transmission cost for each layer in an incremental order and find jointly optimal multicast paths and associated rates. Through a primal decomposition and the primal-dual approach, a decentralized algorithm with two-level optimization update is developed to solve the target convex optimization problem.

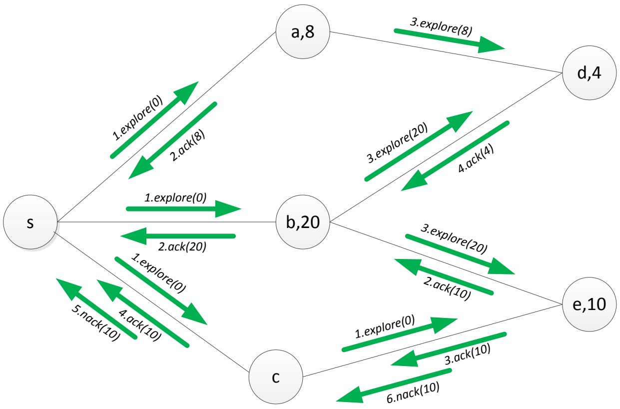
Gurdip, Sandeep and Sanjoy [34] proposed a rate based propagation protocol for sensor networks. In this paper, they tried to solve the similar problem that multiple destinations require data at different data rates. However, it is not a localized algorithm and does not have the assumption that network nodes know the position and rate information of destinations. It tries to construct a tree that has optimal total cost. The cost is calculated as sum of all edge cost that is the rate of sending data multiplied by the length of the edge. The path selection is based on a tree that is constructed by flooding explore message and calculating the Ack messages at each network node. Starting from a source node, explore messages that contain data of current node are broadcasted to all current node's neighbors. Because destinations have multiratecast requirements, each node in the multicast tree must receive data at the rate which is equal or larger than the rate it need to forward to its children. The initial explore message from the source node has rate 0. When the explore messages reach destinations, Ack messages are sent back including the required rate. When a relay node first receives an Ack message, it will set the rate in Ack as its planned rate and accepts the node that sent explore message as its parent node and broadcasts the same explore message to all its neighbors. A relay node waits for all responses of its received explore messages before sending back to its parent to update its rate. When a node receives more than one explore message, it performs a cost comparison calculation to decide if it needs to switch the parent node. It will calculate the cost of adding the new rate for current child and the cost of removing existing rate. By comparing the two costs of adding and removing parent results, the node can decide if it needs to switch parents to the low cost node.

Figure 2.1 shows an example of the rate-based data propagation algorithm on unweighted graph. The data is sent from a source node s to destination node *a, b, d, e* that request data at rate of 8, 20, 4, 10,respectively.

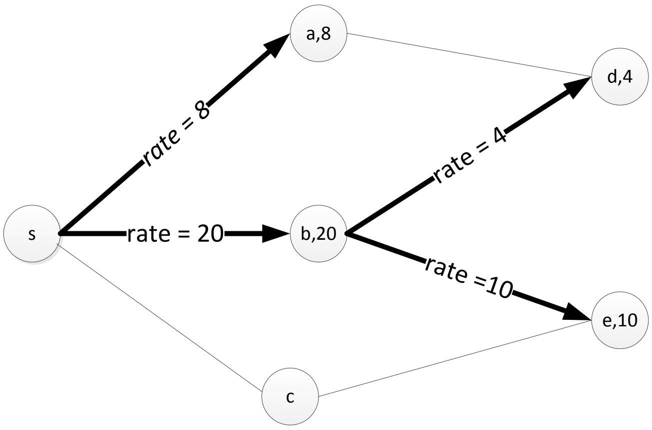
From the Figure 2.1(a), we can see that the source node s sends out explore message with rate 0 to all its neighbors. The numbers next to the nodes in the figure denote the rate requirements of the nodes, and the index before the message denotes the sequence in which the messages are being sent.[34] Nodes a, b reply *ack(8)* and *ack(20)* message to node s and continue send out *explore(8)* and *explore(20)* message to all their neighbors. Node c is not a destination node, it simply continue the flooding with *explore(0)* message. Node d receives explore message from $a$and b, it selects the node with higher rate as parent node and sends back *ask(4)* message to b. Node e first receives *explore(0)* message from node c and accepts c as parent node by sending back *ack(10)* to node c. Later, after node c receives *explore (20)* message from node b, it finds switching parent node from c to b will lower the cost of tree. Node c then sends *ack(10)* to b to accept b as parent. Node c also sends out *Nack (10)* message to c to remove it as parent node. The Figure 2.1(b) shows the rate-based data propagation tree construction results. The edges in bold belong to data propagation tree and the rates listed beside those edges are the data rates need to be sent on the edges. The rate based propagation protocol defines cost of the tree as the sum of cost of all tree edges, where the cost of the tree edge is rate of sending the data multiplied by the weight of the edge which can be listed *as* where $w(i, j) * re$, where $w(i, j)$ is the edge from *i to j, re* is the rate of sending data on the edge. For an unweighted graph as the example, if we use all edge weight as 1, the tree cost can be sum of data rate on all tree edges. Figure 2.1(c) shows the final routing paths and the cost based on all weight equal to 1 scenario. "In the case of weighted networks, the cost of using an edge is the weight of the edge multiplied by the rate at which data is sent over the edge." [34] The tree construction algorithm will be modified when accepting and switching parents because it needs new edge cost calculation to decide which operation takes lower cost. However, the paper [34] did not give a clear definition of the edge weight or the detail description of tree construction process on weighted graph.

As illustrated in Figure 2.1, the rate-based data propagation protocol needs globally broadcast explore/ack messages to build the data propagation tree. It needs complex procedure to complete the tree construction. The complexity of tree calculation increases rapidly when there is large number of network nodes or large number of destinations. The tree structure is also fragile to maintain. It needs to be built before routing. If network is broken anywhere in the tree, it needs to be fully rebuilt before routing can be performed again. Mesh based solutions also need to be rebuild reporting tree when anything breaks and tree is extracted. Compared to tree or mesh based protocols, the proposed localized protocols have advantage that if there is any link broken, they need only locally update neighbor position information and routing process should be able to dynamically do different neighbor selection based the position of current neighbors.

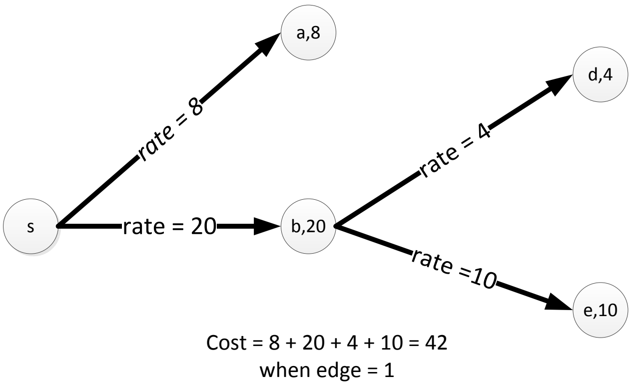
Article [24] proposed the RCM and MRM algorithms. For RCM there are three variants of the algorithm, which differ in the actual cost-over-progress for-



(a) Tree construction process



(b) Tree construction result



(c) Final routing paths and cost

Figure 2.1: Illustration of the rate-based data propagation algorithm

mula being applied. Here we describe (and use in our comparison in simulations) only the best version from [24]. RCM investigates only set partitions of destinations that have, for each set, a neighbor with progress toward each destination in the set. Destinations without having neighbor with progress are not considered. If the number of destinations is large (investigating all set partitions requires exponential time), RCM finds the best neighbor for each subset from the partition. Otherwise, it starts from single destination in each set, and iterates a merge process to optimize it. The quality of a set partition $P = \{M_1, M_2, ..., M_{|P|}\}$ of destinations assigned to the current node c, is measured by this cost over progress ratio

$$ratio(P) = \frac{\sum_{i=1}^{|P|} rate(M_i)}{\sum_{i=1}^{|P|} max(prog(c,n,M_i):n \in N(s))}.$$

The nominator shows the sum of allocated rates (the maximum of requested rates for destinations in the set) for each set. The denominator sums the progresses made by selected neighbor n for each set M_i. Progress provided by n is $prog(c, n, M_i) = \sum_{k=1}^{|M_i|}(|cd_k| - |nd_k|)$, where the sum is taken for each destination d_k from M_i. Selected neighbor n of s maximizes this expression. In a merging step, all pairs M_i and M_j of sets in the current partition are tested if a neighbor n closer to all destinations in both sets exist to provide progress toward them, and whether this neighbor could improve above quality measure. If so for a given pair, the neighbor with best such improvement is selected, and two tested pairs are merged. Pairing process continues until one particular set partition cannot be improved by merging any pair.

Compared to RCM, MRM is simpler when calculating the next forwarding nodes in greedy mode. It is still a general Greedy Face Greedy Algorithm, the Greedy routing selecting gives priority to the destination of maximum data rate. In MRM greedy routing part, when a node S wants to find a routing node to multiple destinations with multiple rates, it applies multiple relay point (MRP) strategy. The strategy works as follows: For all the destinations in greedy list, Find one destination that requires highest rate, say it is D_m. Figure out the neighbor nodes that can provide most distance progress towards the just picked destination D_m , mark this neighbor node as N_m. For rest of the destinations in greedy routing list, as long as node N_m can provide greedy routing to the destinations, use N_m as forwarding nodes for those destinations. If there are still destinations that cannot be forwarded by N_m, repeat step 1 to 3 to find next forwarding node until all the destinations are covered.

If the face routing is involved, it is similar to RCM routing, the greedy routing can be resumed whenever the forwarding node is closer to destination than the face routing starting node. Since there is no destination set partition required, this algorithm requires much less calculation. It is very suitable for the situation where they are high number of destinations or sensor nodes have very limited power consumption requirement.

Chapter 3

The Rate-Aware Cost-Efficient Multiratecasting Routing

3.1 Preliminaries

3.1.1 Network Model

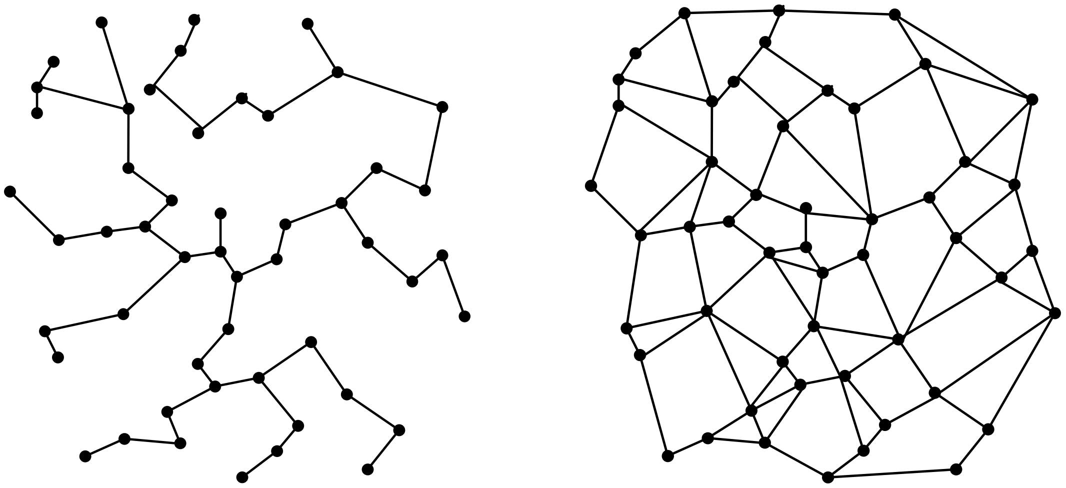
We model a wireless network by a graph $G = (V, E)$, where V is the set of vertices (the network nodes) and $E \subseteq V^2$ is the set of edges that gives the available communications: there exists a pair $(u, v) \in E$ if the node u is physically able to communicate with v. The neighborhood set $N(u)$ of a node u is defined as:

$$N(u) = \{v \in V \mid v \neq u \wedge (u, v) \in E\} \tag{3.1}$$

The density d of the network is the average value of neighbors per node. The construction of the set of edges E depends on the considered underlying physical model. The most well-known one is the *unit disk graph* model. Given a set of nodes V and a maximum communication range R, E is defined as:

$$E = \{(u, v) \in V^2 \mid u \neq v \wedge \mid uv \mid \leq R\} \tag{3.2}$$

$|uv|$ being the Euclidean distance between nodes u and v. Figure 3.1 provides an example of a unit disk graph. For a given multicast task, the set of destinations is denoted as $D = \{d_1, ..., d_k\}$. We assume that nodes are able to adjust their transmitting power, i.e., sending a message from a node u to a neighbor v is done by using the smallest possible power for that. We also assume that nodes regularly collect 1-hop neighborhood information by using beacon messages. This is a fairly common assumption in literature.



(a) The Euclidean MST $\Delta(V)$ (b) The Gabriel graph of G

Figure 3.1: Graph $G = (V, E)$

3.1.2 MAC Layer Model

Whenever a message is duplicated during a multicast, the current forwarding node has to send the packets to more than one immediate neighbor. In this work, we consider two simplified MAC layer models. In the unicast MAC layer, sending a message to i next hop neighbors is performed by i independent reliable unicast transmissions. In the multicast MAC layer, sending a message is done by only one single transmission. In other words, the latter refers to a MAC layer implementation which exploits the broadcast capabilities of the wireless communication media. A detailed investigation on how such a reliable communication is achieved either in the unicast or the multicast MAC layer is beyond the scope of this work.

3.1.3 Geometric Concepts

The minimum spanning tree (MST) is a well-known graph construction: a tree $\Delta(u_1, ..., u_n)$ is a MST if its weight $\mid \Delta(u_1, ..., u_n) \mid$ is minimal. At this time, the weight of the tree denotes the sum of the weight over all tree edges. In a Euclidean MST, illustrated by Figure 3.2, the weight of an edge is equal to its Euclidean length. One example would be a telecommunications company laying cable to a new neighborhood. If it is constrained to bury the cable only along certain paths, then there would be a graph representing which points are connected by those paths. Some of those paths might be more expensive, because they are longer, or require the cable to be buried deeper; these paths would be represented by edges with larger weights. A spanning tree for that graph would be a subset of those paths that has no cycles but still connects to every node. There might be several spanning trees possible. A minimum spanning tree would be one with the lowest total cost.

There are now two algorithms commonly used, Prim's algorithm and Kruskal's algorithm. Both are greedy algorithms that run in polynomial time, so the problem of finding such trees is in function problems, and related decision problems

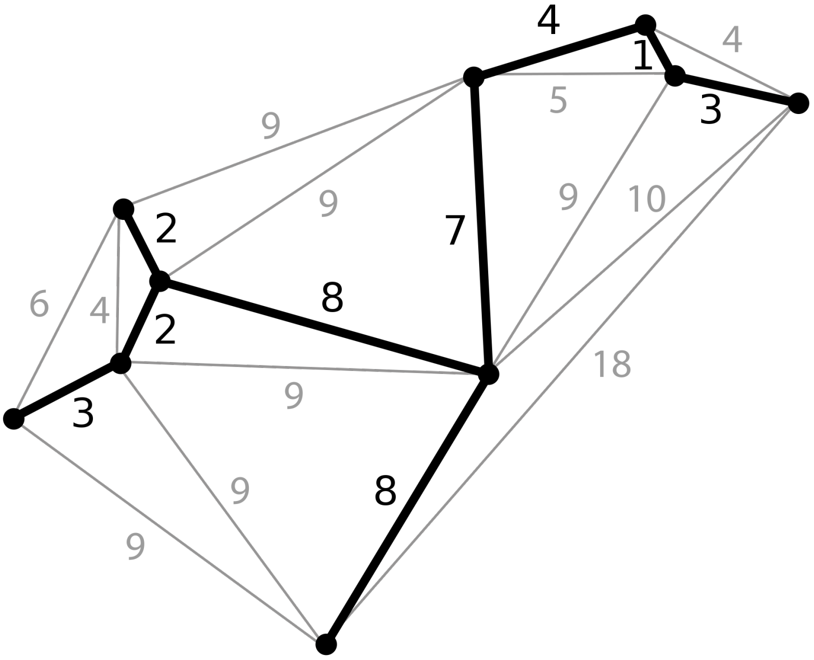


Figure 3.2: Minimum Spanning Tree

such as determining whether a particular edge is in the MST or determining if the minimum total weight exceeds a certain value are in P. Such trees may be efficiently computed in time $O(n \log n)$. Note that we use throughout this paper the notation $\Delta(s)$, which is equivalent to $\Delta(u_1, ..., u_n)$ for any set $S = \{(u_1, ..., u_n)\}$.

The Steiner tree problem is somewhat similar to the MST problem: the goal is to construct a tree $\Gamma(u_1, ..., u_n)$ with the minimal weight, while allowing the insertion of additional intermediate vertices (called Steiner points) in order to reduce the weight of the resulting spanning tree. This problem is known to be NP-complete [20]. In practice, heuristics are used. The difference between the Steiner tree problem and the minimum spanning tree problem is that, in the Steiner tree problem, extra intermediate vertices and edges may be added to the graph in order to reduce the length of the spanning tree. These new vertices introduced to decrease the total length of connection are known as Steiner points or Steiner vertices. It has been proved that the resulted connection is a tree, known as the Steiner tree. The Steiner tree problem has applications in circuit layout or network design.

A planar graph is a graph in which no edges intersect. Gabriel graph construction is a prominent localized construction method which is based on a geometric concept which was introduced by Gabriel and Sokal in [9]. Starting from a unit disk graph $G = (V, E)$, each edge $(v, e) \in E$ is considered and removed if there exists a vertex w located inside the circle $U(u, v)$ of diameter $\mid uv \mid$ centered at the midpoint of the segment $[uv]$. This graph is very interesting for decentralized networks since this removal strategy may be applied independently by each node, and does not require any message exchange. Refer to Figure 3.3 for an illustration of a Gabriel graph constructed over an entire network.

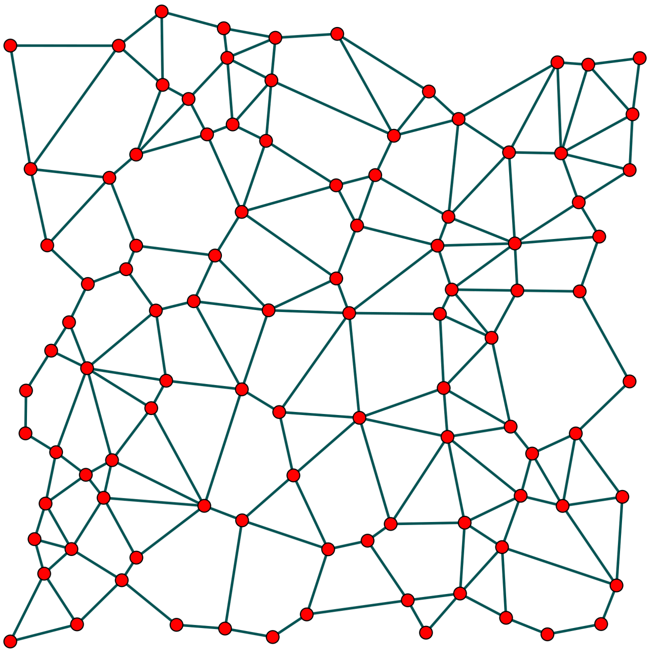


Figure 3.3: Gabriel graph

3.2 Problem Formulation

In the previous study, considering unicast routing scenario, when the destination requires data at specific rate, the total routing network resource consumption is proportional to the total transmission number and the data rate in a routing path. Combining the two factors: hop count and rate, the routing cost for single rate unicast routing can be evaluated by the following equation:

$$C = kr \tag{3.3}$$

where C is th total single rate cost, r is the data rate required by the destination, k is the total hop count number (transmission number).

Considering a network with multirate destinations, different from unicast routing, the multiratecast message may split and the transmission rate may change at relay nodes when routing path branches. Multiratecast require that each destination receives data at its request rate, so that message forwarding has to be at the maximum rate among those destinations it covers, it has to fulfill following rate requirements:

$$r_p \geq r_c, r_p = max(r_{p^1}, ..., r_{p^n}) r_c = max(r_{c^1}, ..., r_{c^m}) \tag{3.4}$$

where r_p is the parent node, $r_{p^1}, ..., r_{p^n}$ is data rates for destinations that parent node covers. And r_c is the child node, $r_{c^1}, ..., r_{c^m}$ is data rates for destinations that child node covers. The start rate at source node S is the maximum rate for all destinations. When message reaches next forwarding nodes, the next forwarding nodes will retransmit message at the equal or less rates depends on their covered destinations. The total routing consumption can be evaluated as following equation:

$$C = \sum_{j=1}^{k} r_j \tag{3.5}$$

where C is the total multirate cost, r_j is the transmission rate at forwarding node j, j is the forwarding node number, k is the total hop count. A performance metric by considering both transmission number and transmission rate will better reflect the network resource consumption for multiratecast routing.

The problem of multiratecast algorithms can be described as follows: given a certain message generated by a source node S and a group of destination nodes $\{d_1, ..., d_m\}$ which each destination node requests to receive data at a specific rate, MSTRC needs to find a set of relay nodes in the network so that the message is delivered to all destinations with a minimum consumption of network resources and at the same time, assures that each destination receives data at its required rate.

3.3 MST-based Multiratecasting

The problem that multiratecast algorithms are facing and possible solutions are described in previous sections. To solve the problem efficiently, the number of messages should be minimized. Then multiple branches of paths are needed so that the path with higher rate can be shared as much as possible to reduce the network bandwidth usage.

Based on this assumption, our MST-based algorithm is proposed as follows, we divide our algorithm into three parts.

- First, we introduce a replication method to build a tree-based backbone. It decides how many replications current node should split, and arranges the transmission ratio of each forwarding node in the backbone construction.

- Second, based on the constructed backbone, the algorithm will choose the forwarding nodes for each destination. The neighbor selection step also takes the rate consumption into consideration. We use three different metrics for the selection of forwarding nodes. According to the experimental results, we find that different measurements will fit into different conditions.
- The last part of our algorithm is the recovery strategy when messages are sent to a void area by greedy algorithm. The algorithm will quit from the greedy mode and run into the face mode (by GFG algorithm) until a better node for forwarding is found.

We would like to introduce the details in the following sections one by one.

3.3.1 Rate-aware Message Replication

As we know, the main difference between routing in unicast and multicast is the neighbor selection function. In unicast routing, only a single node among all the n neighbors is selected as the relay. While in multicast scenario, current node may split the message into multiple copies and send them to multiple destinations. It has to be decided which neighbor is suitable for which destination(s).

The number of neighbors selected is of vital importance because each relay will start a new and separate path toward certain destination that the node is responsible for. There are many ways of selecting neighbor subsets for forwarding. However, choosing too many neighbors means that there will be different routes for different destinations which losing the advantages of sharing routes in multicasting. On the other hand, choosing too little relays implies that some of them will be in charge of routing toward more than one destination. This seems to be a good decision for energy savings as several messages can share the same route.

However, from another point of view, it can also lead to a common longer path which increases the energy consumption significantly. The problem now is how to select the subset of relays, at the same time arrange an optimal transmission rate to each origin Location.

In [24], this problem is solved by using two methods MRM,RCM. If the number of neighbor is not large (less or equal to 5), they would enumerate all the possible combination of neighbors as relay node. If the number of neighbor is large(more than 5), they would choose some of the combination that are more likely to be relay nodes to measure. The transmission rate is arranged to fit the need of all the destinations.

Our proposed message splitting strategy is one of the most important aspects of our algorithm. To improve such selection, we use MST based backbones and taking the rate arrangement into consideration. The general idea is to build a MST tree with current node and the destination nodes. In the process of building the tree, we consider the rate that the destination nodes need as a fact.

Suppose S being the source node and $D = \{d_1, ..., d_k\}$ being the message destinations. Furthermore, let $C(u, v)$ denote the weight of the shortest weighted path from u to v.

Under the unicast MAC assumption, a weighted Steiner tree $\Gamma(S, d_1, ..., d_k)$, using $C(u, v)$ as the cost function, defines the optimal multiratecast backbone. In this work we do not assume that a node is able to request all network nodes for computing such a Steiner tree. Moreover, we do not assume that the cost function $C(u, v)$ is even known to the nodes. Thus computing a Steiner tree as a optimal multiratecast backbone is not possible in this general multiratecast setting.

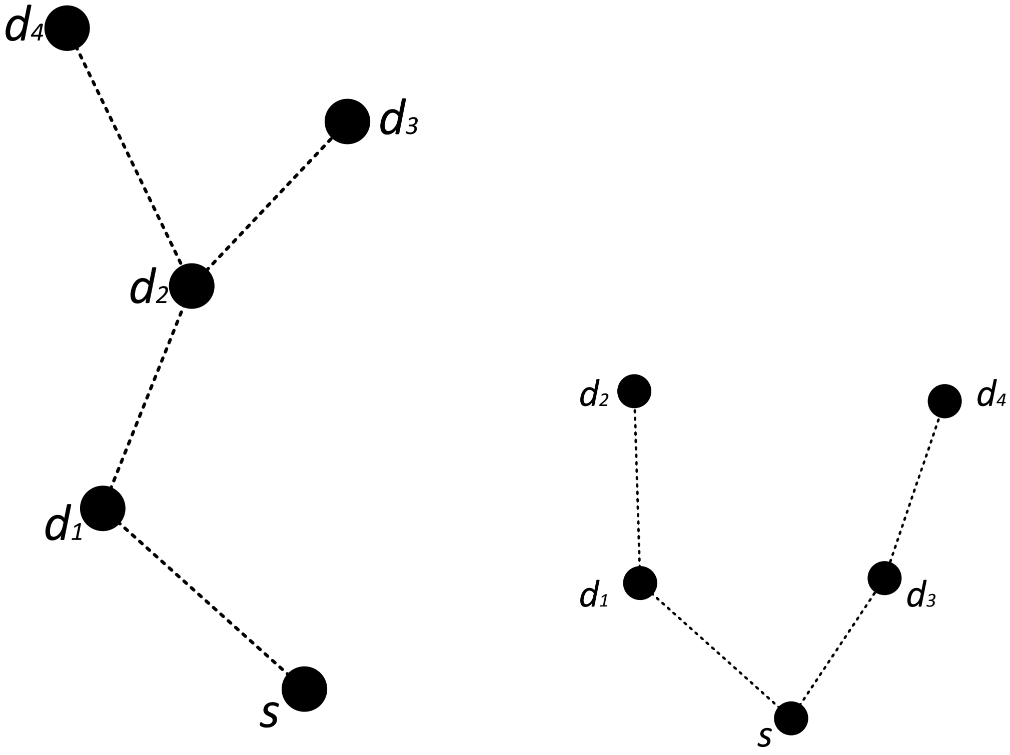
As an approximation, we utilize the concept of weighted MST instead which may be efficiently computed even by constrained devices. Since the exact energy model is not known, we approximate the routing cost by the simplified assumption that $|uv| < |uw|$ always implies $C(u, v) < C(u, w)$. In this case, the weighted MST is equivalent to the Euclidean MST, for which only locations of nodes are needed. Under the multicast MAC assumption, energy savings are possible at nodes where the message is split. At this point, any set of next hop nodes might produce the same routing cost as addressing a single next hop node. In a small scale multiratecast, it might thus be more energy-efficient to perform a single direct large broadcast transmission instead of performing many small transmissions in order to reach the destinations. In a large scale multiratecast, however, we expect that the energy savings which are possible at the nodes where the message is split will be outweighed by the routing costs which are required in order to route the message between those split points. Thus, we use the same MST approximation even under the multiratecast MAC assumption.

The following describes the rule which is performed at each forwarding node in order to decide if a message has to be kept as one single packet or if it has to be split and sent toward different destination subsets. Let c be the current forwarding node and let $d_i \subseteq D$ be the set of multicast destinations which have to be handled by c. Node c has to calculate the MST $\Delta(\{c\} \bigcup d_i)$ over itself and d_i. This tree provides the backbone which is to be used to reach all destination nodes in D_i from c. The message thus has to be routed along the edges of this tree, and must be split at node c if multiple paths start from this node. Actually, each of these paths is represented by an edge which originates at node s and spans a subset of destination nodes. These are forming exactly a destination subset to which c has to send an individual message copy. Assume each node defines rate that it wants to receive from source. In the localized algorithm all nodes except source will have rates (all rates are equal to 1, this is same as MST in [7]).

Let $R(i)$ be desired rate for node i. During the execution, each node maintains rate provided so far, $P(i)$. Initially $P(i) = 0$ for all nodes (including source). Edge weights are not needed in algorithm. Consider candidate edges (u, v). Each node u from V_{new} has its $P(u)$. Candidates are those nodes v which are excluded in V_{new} but have some neighbors included in V_{new}. For each v, considering all existing neighbors u from V_{new}, and calculate the cost of adding v to the already constructed tree via node u.

If $R(v) <= P(u)$ then there is no need to increase rate on the whole path from source to u, only to define rate at u. The cost then is only $R(v)$. In fact the cost of communicating in the last hop to v is $P(v)$ no matter which neighbor u is used. If $R(v) > P(u)$ then the rate needs to be increased on the whole path from source to u. Assume that path has h hops, and the path from source S is $S, A_1, A_2, ..., A_h$. For each hop calculate rate increase, if needed, which is 0 if no increase needed, and $R(v) - P(A_i)$ if there was an increase. Then sum up all these increases on the path. This sum is the overall cost for this candidate edge, and the one with minimal cost is selected in the iteration. After selecting, all nodes A_i on the path from source to v will update their provided rate $P(A_i)$ to $P(v)$ if it was smaller than that.

This strategy is illustrated by Figure 3.4, where node S has to handle the destination nodes d_1, d_2, d_3 and d_4. In Figure 3.4(a), the resulting MST $\Delta(s, d_1, d_2, d_3, d_4)$ has only one edge originated at node S, so all destinations are grouped together. In this case, the message will not be split and will have to be routed along the edge (S, d_1). In Figure 3.4(b), there are two edges originated at node c: the first one spans d_1 and d_2, while the second one spans d_3 and d_4. The message will thus be duplicated into two packets. The first one will be routed along the edge (S, d_1) toward the set of destinations $\{d_1, d_2\}$, and the second one will be routed



(a) All destinations are kept in the same set

(b) Two subsets d_1, d_2 and d_3, d_4 are created

Figure 3.4: The message splitting strategy used by MSTRC

along (S, d_3) toward the set of destinations $\{d_3, d_4\}$. Energy-efficient message forwarding along an edge of the multiratecast backbone will select the best neighbor with respect to the metric being applied (e.g., based on hop count, Euclidean distance, energy consumption, which we would discuss in next section) and the destination nodes which are reachable along this backbone edge. For instance, in Figure 3.4(a) next hop selection along the multiratecast backbone edge Sd_1 will consider the best node with respect to the metric and the destination node d_1. Next hop selection along the multiratecast backbone in Figure 3.4(b) will consider the best node with respect to the metric and the destination nodes d_1 and d_2.

3.3.2 Energy-efficient Neighbor Selection

In the previous section, we explained how the MST backbone be constructed, and gave detailed explanations of the different aspects of its operation. However, we did not explain into detail how is the neighbor selection algorithm implemented. That is, what is the concrete algorithm used by the current node to select neighbors for certain destinations. In this section we describe that part of the protocol, showing the benefit in terms of computational cost compared to previous works.

Considering the multiratecasting problem, where a source node wishes to send a packet to a number of destinations with known positions at known rate. Assume that a node c, after receiving a multiratecast message is responsible for destinations $d_1, ..., d_n$, and that it evaluates neighbors n_1,...,n_n as possible candidates for forwarding. The task can be regarded as to decide which neighbors should be selected based on the cost of the selected nodes as forwarding nodes.

The cost of each subset is a value to reflect the local routing efficiency for multiratecast when current node split message to the next forwarding nodes with different data rate. After set partition, for each given set, the next forwarding data rate for every subset is a defined value. The value is the maximum data rate among destinations in the subset. The neighbor selection method tries to find a best neighbor to cover all destinations for each subset. A parameter rate cost factor is introduced to reflect the local rate cost efficiency towards selected destinations. The rate cost factor rcf is as of distance progress ratio $R/\Delta d$. R is the data rate cost towards the selected destinations, Δd is the distance progress towards selected destinations.

The rate cost factor can be calculated based on three ways, on each individual destination, on each subset or on the whole set. The routing path and message split may be different when apply different rate cost factor calculation thus result in different overall routing performance. Because there is no existing study show which rate cost factor calculation method is most suitable for multiratecast routing, three different methods to calculate set cost are explored. For a given destination set, the neighbor selection method applies three types rate cost factors to get the set cost.

3.3.2.1 Method A: Cost Focus on Subset

This type focuses on finding minimum rate cost factor for each subset. The rate cost factor in method A is calculated as the maximum rate in subset divided by total distance towards all destinations in the subset. Because the maximum rate for subset is same value for all neighbors, so that among all neighbor nodes that provide the distance progress to subset destinations, method A tries to find a best neighbor node that provide maximum distance advance to all destinations in the subset. This will result in a best neighbor with minimum rate cost factor. The method A uses the minimum rate cost factor as subset cost, and then calculates set cost as sum of all subset costs. Suppose we have a set partition DS_i with total k destination set that $DS_i = \{M_1, ..., M_k\}$. For each destination subset M_j suppose we have p destinations $M_j = d_{j1}, ..., d_{jp}$. Method A calculates the rate cost factor for each neighbor node as following equation:

$$rcf_{subset} = \frac{Rmax_{subset}}{\Delta d} = \frac{Rmax_{subset}}{\sum_{l=1}^{p} \mid\mid DS_l \mid - \mid nd_l \mid\mid} \tag{3.6}$$

The best forwarding neighbor node n should have lowest subset rate cost factor, or the maximum distance progress to all destinations in subset because the maximum rate of the subset is a constant value. The best forwarding node n should fullfill following equations:

$$\frac{Rmax_{subsetj}}{\sum_{l=1}^{p} ||DS_l| - |nd_l||} \leq \frac{Rmax_{subset}}{\sum_{l=1}^{p} ||DS_l| - |nd_l||} \tag{3.7}$$

After each subset M_j has a best forward node, the method A calculates its set cost for destination set DS_i by sum of all subset rate cost factors as following equation:

$$RC_1 = \frac{\sum_{j-1}^{k} R_j}{\sum_{l-1}^{jp}(||DS_{jl}| - |n_j d_{jl}||)} \tag{3.8}$$

Based on calculation of method A, the rate set cost listed in Figure 3.5 is:

$$RC_1 = \frac{max(R_1, R_2)}{||DS_1| - |n_1 d_1|| + ||DS_2| - |n_1 d_2||} + \frac{max(R_3, R_4)}{||DS_3| - |n_2 d_3|| + ||DS_4| - |n_2 d_4||} \tag{3.9}$$

Figure 3.6 and Figure 3.7 list two possible destination sets for a given network with 4 destinations. Forwarding node S has two neighbor nodes n_1 and n_2. By calculating the set cost for Figure 3.6 and Figure 3.7, we may get different next forwarding set cost by method A.

For the above Figure 3.6, we have destination set with two subset $DS =$

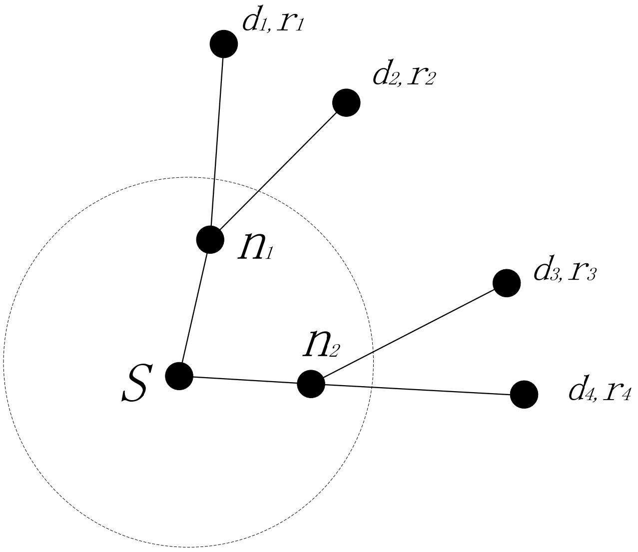


Figure 3.5: Example of MSTRC Forwarding

d_1, d_2, d_3, d_4. Method A calculates set cost as following steps:

For subset M_1, it calculates the distance advance from n_1 to $\{d_1, d_2\}$, and then n_2 to $\{d_1, d_2\}$, and then compares the results to find out the best neighbor for subset M_1.

$$\Delta dn_1(subset1) = ||Sd_1| - |n_1d_1| + ||Sd_2| - |n_1d_2|| = 3.89 \tag{3.10}$$

$$\Delta dn_2(subset1) = ||Sd_1| - |n_2d_1| + ||Sd_2| - |n_2d_2|| = 3.21 \tag{3.11}$$

Best neighbor node for subset M_1 is n_1. For subset M_2 it calculates the distance advance from n_1 to $\{d_3, d_4\}$, and then n_2 to $\{d_3, d_4\}$, and then compares the results to find out the best neighbor for subset M_2.

$$\Delta dn_1(subset2) = ||Sd_3| - |n_1d_3| + ||Sd_4| - |n_1d_4|| = 2.37 \tag{3.12}$$

$$\Delta dn_2(subset2) = ||Sd_3| - |n_2d_3| + ||Sd_4| - |n_2d_4|| = 4.31 \tag{3.13}$$

Best Neighbor node for subset M_2 is n_2. Set cost for Figure 3.6 by method A is:

$$RC_1 = \frac{max(R_1, R_2)}{||Sd_1| - |n_1 d_1|| + ||Sd_2| - |n_1 d_2||} + \frac{max(R_3, R_4)}{||Sd_3| - |n_2 d_3|| + ||Sd_4| - |n_2 d_{4p}||} = 2.90 \tag{3.14}$$

3.3.2.2 Method B: Set Cost Focus on Individual Destination

In calculation method B , we focus more on individual destination rate cost factor. The individual destination rate cost factor is defined as forwarding rate to a destination divided by the distance progress from a neighbor node to the destination as following equation:

$$rcf_{individual} = \frac{R_{forwarding}}{\Delta d} = \frac{R_{forwarding}}{||Sd_l| - |nd_l||} \tag{3.15}$$

where $R_{forwarding}$ is the forwarding rate that the destination requires $|Sd_l|$ is the distance between current node S to the destination d_l. $|nd_l|$ is the distance between neighbor node n to the destination d_l.

For every destination in a subset, $R_{forwardingis}$ is a constant value equal to the maximum data rate among all destinations in the subset $Rmax_{subset}$ because method B tries to find a best forwarding neighbor node that covers the subset. The best node should provide minimum total individual destination rate cost factors for all destinations in the subset. Method B then calculates subset cost as the sum of individual costs of the best forwarding node, and then calculate set cost for destination set as the sum of all subset cost. For subset M_j, the best forwarding node n fulfills following equation:

$$\sum_{l=1}^{p} \frac{Rmax_{subsetj}}{||Sd_{jl}| - |n_j d_{jl}||} \leq \sum_{l=1}^{p} \frac{Rmax_{subsetj}}{||Sd_{l}| - |n_j d_{jl}||} \tag{3.16}$$

For the whole destination set DS_i, method B calculates the set cost as the sum of all best individual rate cost factors of every subset. Equation is as below:

$$RC_2 = \sum_{j=1}^{k} \sum_{l=1}^{jp} \frac{R_j}{||Sd_{jl}| - |n_j d_{jl}||} \tag{3.17}$$

As for example listed in Figure 3.5, method B has the rate cost as following:

$$RC_2 = \frac{max(R_1, R_2)}{||Sd_1| - |n_1 d_1||} + \frac{max(R_1, R_2)}{||Sd_2| - |n_1 d_2||} + \frac{max(R_3, R_4)}{||Sd_3| - |n_2 d_3||} + \frac{max(R_3, R_4)}{||Sd_4| - |n_2 d_4||} \tag{3.18}$$

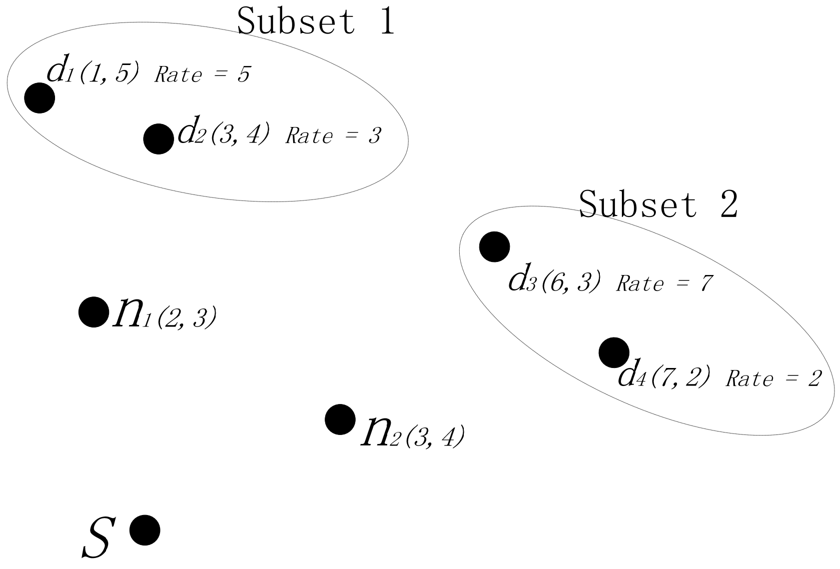


Figure 3.6: Example of MSTRC Set Cost Calculation

For the destination set in Figure 3.6, method B calculates the set cost as following steps. For subset M_1, it calculates the rate cost factor from n_1 to

$\{d_1, d_2\}$, and then n_2 to $\{d_1, d_2\}$, and selects the neighbor with a lower rate cost factor.

$$rcf(n_1, subset_1) = \frac{max(R_1, R_2)}{||Sd_1| - |n_1d_1||} + \frac{max(R_1, R_2)}{||Sd_2| - |n_1d_2||} = 5.12 \tag{3.19}$$

$$rcf(n_2, subset_1) = \frac{max(R_1, R_2)}{||Sd_1| - |n_2d_1||} + \frac{max(R_1, R_2)}{||Sd_2| - |n_2d_2||} = 14.59 \tag{3.20}$$

The best neighbor for subset M_1 is n_1. For subset M_2, it calculates the rate cost factor from n_1 to $\{d_3, d_4\}$, and then n_2 to $\{d_3, d_4\}$, and selects the neighbor with a lower total rate cost factor.

$$rcf(n_1, subset_2) = \frac{max(R_3, R_4)}{||Sd_3| - |n_1d_3||} + \frac{max(R_3, R_4)}{||Sd_4| - |n_1d_4||} = 12.1 \tag{3.21}$$

$$rcf(n_2, subset_1) = \frac{max(R_3, R_4)}{||Sd_3| - |n_2d_3||} + \frac{max(R_3, R_4)}{||Sd_4| - |n_2d_4||} = 6.52 \tag{3.22}$$

The best neighbor for subset M_2 is n_2. The set cost I by method B for Figure 3.6 is:

$$RC_2 = \frac{max(R_1, R_2)}{||Sd_1| - |n_1d_1||} + \frac{max(R_1, R_2)}{||Sd_2| - |n_1d_2||} + \frac{max(R_3, R_4)}{||Sd_3| - |n_2d_3||} + \frac{max(R_3, R_4)}{||Sd_4| - |n_2d_4||} = 11.64 \tag{3.23}$$

3.3.2.3 Method C: Rate Cost Based on Destination Set

In method C , the rate cost factor is based on the whole destination set. The set rate cost factor is calculated as the sum of all subset rate divided by the sum of all the maximum subset distance progress. The rate cost factor is as in following equation:

$$rcf_{set} = \frac{\sum\limits_{j=1}^{k} R_j}{\sum\limits_{j=1}^{k} \Delta D_j} = \frac{\sum\limits_{j=1}^{k} R_j}{\sum\limits_{j=1}^{k} (\sum\limits_{l=1}^{jp} ||Sd_{jl}| - |n_j d_{jl}||)} \tag{3.24}$$

where k is the number of subsets in DS_i, $DS_i, = \{M_1, ..., M_k\}$, j is the subset number, R_j is the max rate of destinations within the subset M_j, jp is the number of destinations in subset M_j, $M_j = \{d_{j1}, d_{j2}, ..., d_{jp}\}$, d_{j1}, d_{j2},...,d_{jp} are destinations in subset M_j, $|Sd_{jl}|$ is the distance between current forwarding node S to destination d_{jl} in M_j, $|n_j d_{jl}|$ is the distance between neighbor node n to destination d_{jl} in M_j.

Because given a destination set, the sum of subset rate is a fixed value that is equal to the sum of maximum data rate of all subsets, so in order to lowest the set rate cost factor , method C needs to select best neighbor node for each subset that provide the maximum distance progress for all destinations in each subset. The best forwarding node for subset M_j should fulfill the following equation:

$$\sum_{l=1}^{p} ||Sd_{jl}| - |n_j d_{jl}|| \geq \sum_{l=1}^{p} ||Sd_l| - |n_j d_{jl}|| \tag{3.25}$$

After selecting best forwarding node for each subset, The rate cost of the destination set DS_i that contains k destinations $DS_i, = \{M_1, M_2, ..., M_k\}$ can be gotten by following method:

$$RC_3 = \frac{\sum\limits_{j=1}^{k} R_j}{\sum\limits_{j=1}^{k} (\sum\limits_{l=1}^{jp} |Sd_{kl}| - |n_j d_{jl}|)} \tag{3.26}$$

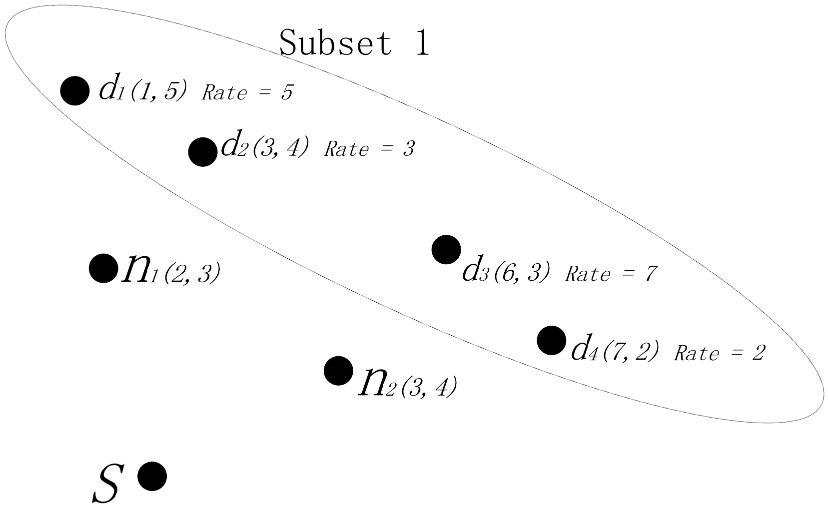


Figure 3.7: Example of MSTRC Set Cost Calculation

As example listed in Figure 3.5, method C calculates the rate cost as following:

$$RC_3 = \frac{max(R_1, R_2) + max(R_3, R_4)}{||Sd_1| - |n_1d_1|| + ||Sd_2| - |n_1d_2|| + ||Sd_3| - |n_2d_3|| + ||Sd_4| - |n_2d_4||} \tag{3.27}$$

For the set example in Figure 3.6, method C selects the best neighbor node in the same way of method A. It calculates the distance progress from n_1, n_2 to the destinations in the two subsets, and selects n_1 for subset M_1 and n_2 for subset M_2. The set cost I by method B for Figure 3.6 is :

$$RC_3 = \frac{max(R_1, R_2) + max(R_3, R_4)}{||Sd_1| - |n_1d_1|| + ||Sd_2| - |n_1d_2|| + ||Sd_3| - |n_2d_3|| + ||Sd_4| - |n_2d_4||} = 1.14 \tag{3.28}$$

3.3.3 Example of Best Set Selection

In this section, we will give an example, a forwarding node in greedy mode will have to compute a rate-aware MST for the message splitting strategy, which has a time complexity in $O(k \log k)$, k being the number of destination in the method of

Table 3.1: Example of set cost results compare

	method A	method B	method C
Set cost A	2.90	11.64	1.14
Set cost B	0.93	12.19	1.60
Final selection (forwarding nodes)	(n_2 for all destinations)	(n_1 for subset 1, n_2 for subset 2)	(n_2 for all destinations)

the best set selection process. As mentioned before, current node would calculate the set cost for every set and choose the neighbor with the lowest cost as the forwarding node.

Suppose we only consider 4 destination nodes as in Figure 3.6 and Figure 3.7. After set cost calculation, the method A, B and C may select better set and forwarding node.

The method A and method C both select set in Figure 3.7 with n_2 as the forwarding node to cover all destinations while method B select set in Figure 3.6 with n_1 cover subset M_1 and n_2 cover subset M_2.

This example only shows routing forwarding difference with 4 destinations and 2 nodes. For network with a larger number of destinations and more neighbor nodes, the routing selection and branch can be different for method A, B and C.

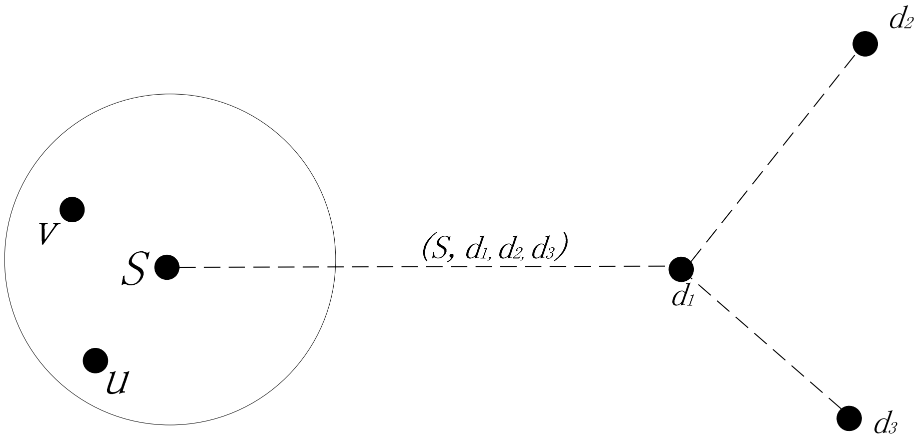
3.4 Recovery Strategy

As previously stated, the message may arrive at a void area, i.e.,the set of possible next hop nodes might be empty for a given destination subset. For instance, suppose that in Figure 3.8(a) node S has to send a multiratecast message toward the destination set $\{d_1, d_2, d_3\}$. All destinations are connected over the link (S, d_1) of $\Delta(S, d_1, d_2, d_3)$. However, the source may not select any of its two neighbors u and v as the next hop, since they both satisfy $|\Delta(u, d_1, d_2, d_3)|$, $|\Delta(v, d_1, d_2, d_3)| > |\Delta(s, d_1, d_2, d_3)|$. Without any further provision, multiratecast routing toward these destination nodes will be stopped at S. This happens independently whether any of these destination nodes are reachable or not reachable from S. In accordance to the notion for unicast greedy algorithms, such a node can be denoted as concave with respect to the destination subset.

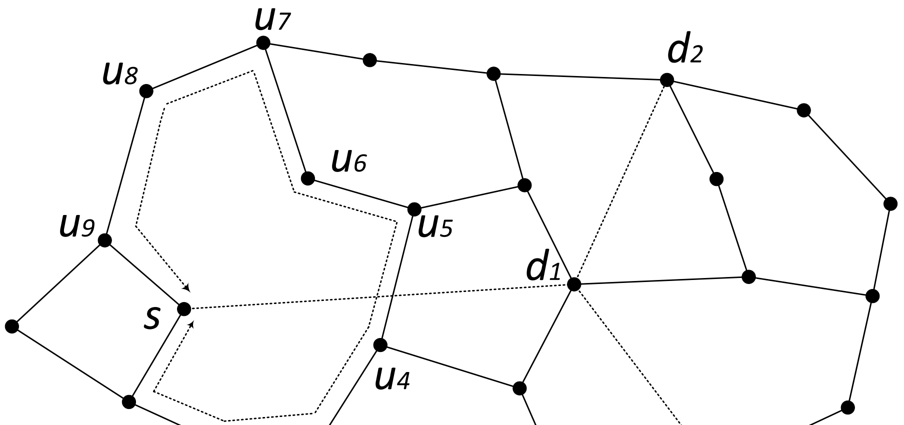
Face routing that can be used in order to handle greedy routing failures for each destination individually is a well-known unicast routing scheme. In the following we describe a multiratecast extension of face routing which can handle the greedy multiratecast situations. Similar to unicast face routing, the multiratecast scheme requires a localized topology control mechanism which transforms the underlying wireless network into a planar graph. In this work, we employ the previously described common Gabriel graph construction which requires the wireless network to comply with the unit disk graph model.

As depicted in Figure 3.8(b), a planar graph partitions the plane into faces which can be traversed in a localized way by employing the left/right hand rule; a receiver node sends the message along the edge which is lying next in clockwise/counterclockwise direction of the edge it was received from. For instance, when starting at node S in Figure 3.8(b), the face F will be traversed along the path $Su_1, ..., u_9$ when using the right hand rule and along the path $Su_9, ..., u_1$ when using the left hand rule.

Unicast face recovery has been described in different variants. In this work we employ the variant which transmits the message along the sequence of faces which are intersected by the straight line Sd connecting source node S and destination node d. Whenever the message arrives at a node which is closer to d than the start node d, face recovery is switched back into greedy mode again. Note that under



(a) Situation in recovery strategy



(b) Constructed face mode

Figure 3.8: Recovery from concave nodes

the Gabriel graph assumption, this unicast face recovery mechanism simplifies to traversing the very first face only.

The idea of multiratecast face recovery is as follows. Suppose that the current forwarding node d has computed a destination subset d_i for which no better greedy neighbor node exists. Let Sd be the edge connecting node S with $\Delta(\{S\} \bigcup d_i)$. By using any of the two rules, $RightHandRule$ or $LeftHandRule$, node S starts traversal of the face which is intersected by the outgoing minimum spanning tree edge Sd. Face traversal continues until the message arrives at a node u which satisfies $\mid \Delta(\{u\} \bigcup d_i) \mid < \mid \Delta(\{S\} \bigcup d_i) \mid$. At this node, the destination subset d_i is handled in greedy mode again. A special case occurs when no such node u is found during face traversal. In this case, in order to avoid a message loop, the message is dropped if it is about to be sent again over the first face traversal edge in the same direction.

Refer to Figure 3.8(b) for an example. The edge Sd_1 connects node S with $\Delta(S, d_1, d_2, d_3)$. Since node S is concave with respect to $\{d_1, d_2, d_3\}$, it will start traversal of face F, i.e., the face which is intersected by Sd_1. Assuming the right hand rule, face traversal will visit the nodes u_1, u_2, and u_3. Since node u_3 is the first one satisfying $\mid \Delta(v_3, d_1, d_2, d_3) \mid < \mid \Delta(s, d_1, d_2, d_3) \mid$, it will handle the destination subset $\{d_1, d_2, d_3\}$ in greedy mode again.

3.5 The MSTRC Protocol

3.5.1 Message Format

After independently execute the greedy and face part routing calculation, forwarding node sends out one message at the required rate among the destinations it covers to all sits neighbors. The message can include both greedy and face forwarding information in the message head. A next forwarding neighbor may be selected as the greedy forwarding node for some destination and face routing node for another destination. All related routing information is contained in the message head. The selected next forwarding neighbors will capture the message and continue the MSTRC routing calculation and forwarding until message reaches destinations.

Message Head

- Source Node: The source node position information.
- Current Node: The current forwarding node information.
- Data rate: the data sending rate from current node to next node.
- Next Node: The next forwarding nodes information.
- Greedy Information: Greedy routing information include a list of all neighbor nodes that will forward message in Greedy mode, the destinations that each neighbor node covers, their positions and required data rates.
- Face Information: Face routing information include a list of neighbor nodes that will forward message in Face node, the destination that each neighbor node covers, their positions, data rates, face start points.

Message Data

- Data that need to be forwarded to destinations.

3.5.2 Data Transmission

The protocol may be described as follows, the source node S, which initiates the multiratecasting task toward the destination set $D = \{d_1, ..., d_k\}$, first has to decide if a message split should be performed. It thus computes the rate-aware MST $\Delta(\{S\} \bigcup D)$, and groups together all destinations spanned by edges originated at S.

For each subset $D_i \subseteq D$ obtained in this way, S computes a subset $n_i(S) \subseteq N(S)$, which contains all neighbors $v \in N(S)$ such that $|\Delta(\{v\} \bigcup D_i)| < |\Delta(\{S\} \bigcup D_i)|$ (these are the neighbors providing positive progress toward D_i). If $n_i(S)$ is not empty, S computes the cost over progress ratio $Q(S, v, D_i)$ for each neighbor $v \in n_i(S)$. The neighbor providing the best ratio is chosen as the next hop toward D_i. If $n_i(S)$ is empty, then S is concave with respect to D_i, and face recovery must be used to escape from this void area. Node S thus applies the recovery strategy presented in previous to select the face node v as the router toward D_i. The whole process is repeatedly done until all subsets D_i have been considered.

If unicast MAC is considered, a packet is sent for each subset D_i. Each of them contains the set of destination nodes, the selected router and the mode (greedy or face) that must be used. In the case of face routing, the packet also contains the very first edge traversed by the packet in face mode, and the weight of the MST at the starting node ($|\Delta(\{S\} \bigcup D_i)|$ in this example). If multicast MAC is considered, all this information is aggregated into the same packet. This means that the latter will contain a list of the selected next hops and for each of them, the set of destinations they have to serve, the mode to use and the additional face information if needed. In both cases, the packet is sent using the minimum energy needed for successful transmission to the next hop(s).

When a node u receives a packet, it needs to check if it has been designated as the next hop by the previous transmitter. If not, the packet is simply ignored. If so, it checks the routing mode currently used for the given set of destinations

$D_i \subseteq D$. In greedy mode, u repeats the same process followed by S. In face mode, it checks whether it is closer to the set of destinations (e.g., $|\Delta(\{u\}D_i)|$ is less than the weight written in the packet). If so, it handles D_i in greedy mode. If not, the face recovery is applied once again. Of course, if u was one of the destinations, it removed itself from D_i and stopped the process if the latter became empty.

Figure 3.9 illustrates a sample run of MSTRC over a randomly deployed network. In Figure 3.9(a) is given the MST $\Delta(\{S\} \bigcup T)$, while Figure 3.9(b) provides the multicast tree produced by MSTRC. The MST spanning all destinations was used at the source node s. Since two edges originate at S, the message was split into two packets at this node. The first one was sent toward d_0 and d_1 along the edge (S, d_0), while the second one was sent toward the rest of the destination nodes along the edge (s, d_2). One can observe in Figure 3.9(b) that MSTRC was able to follow these edges in an effective way. One can also observe that the message splitting strategy correctly works by looking at the path followed to reach d_4 and d_9 from the node lying close to d_5. Instead of following the edge (d_5, d_9), MSTRC routed the message along a common path among d_4 and d_9, and then split it at the end of this path. The same observation applies to nodes d_7, d_8 and d_9.

Regarding the complexity of MSTRC, a forwarding node in greedy mode will have to compute a rate-aware MST for the message splitting strategy, which thus has a time complexity in $O(k \log k)$, k being the number of destination nodes. In the worst case, all destination nodes are handled separately. For each of them and for each neighbor, a new MST must be computed. In this case, the complexity in time of MSTRC is thus $O(mk^2 \log k)$ for the greedy mode, m being the number of neighbor nodes. This complexity may actually be better estimated since a MST

(a) The MST $\Delta(s \bigcup T)$

(b) The multicast tree produced by MSTRC

Figure 3.9: A sample run of MSTRC for a set D of 10 destinations and a density d = 35

has a maximum degree of 6, regardless of the value of k. Since the face mode has a complexity in $O(k \log k)$, the final complexity of MSTRC in the worst case is $O(mk\ \log k)$, which is lower than the complexity of GMREE ($O(mk \min(m,k)^3)$, still considering the worst case).

Algorithm 1 Handle message in greedy mode

```
$s \leftarrow$ current node
if $s$ is a multicast destination then
    pass message to upper protocol layer
    remove this node from multicast destinations
end if
$d_1, ..., d_k \leftarrow$ multiratecast destinations
$r_1, ..., r_k \leftarrow$ rate requirement of destinations
$T \leftarrow \Delta(s, d_1, ..., d_k, r_1, ..., r_k)$
for all edges $st$ in $T$ do
    $T_i \leftarrow$ destinations reachable over $st$ in $T$
    $S \leftarrow \Delta(\{s\} \bigcup T_i)$
    if $\exists v \in N(s)$ with $|\Delta(\{v\} \bigcup T_i)| < |S|$ then
        $v \leftarrow$ neighbor which minimizes $Q(s, v, T_i)$
        greedy forward message to $v$
    else
        start face recovery
    end if
end for
```

Algorithm 2 Start face recovery

```
$s \leftarrow$ current node
$n_{GG}(s) \leftarrow$ Gabriel graph neighbors of $s$
$v \leftarrow$ node in $n_{GG}(s)$ lying next in $cw$ direction from $st$
face forward message to $v$
```

Algorithm 3 Handle message in face mode

```
$s \leftarrow$ face traversal start node
$e \leftarrow$ face traversal start edge
$v \leftarrow$ previous node
$u \leftarrow$ current node
$d_1, ..., d_k \leftarrow$ multicast destinations
if then $|\Delta(u, d_1, ..., d_k)| < |\Delta(s, d_1, ..., d_k)|$
    handle message in greedy mode
else
    $n_{GG}(u) \leftarrow$ Gabriel graph neighbors of $u$
    $w \leftarrow$ node in $n_{GG}(u)$ lying next in $cw$ direction from $uv$
    if $vw = e$ then
        drop message
    else
        face forward message to $w$
    end if
end if
```

Chapter 4

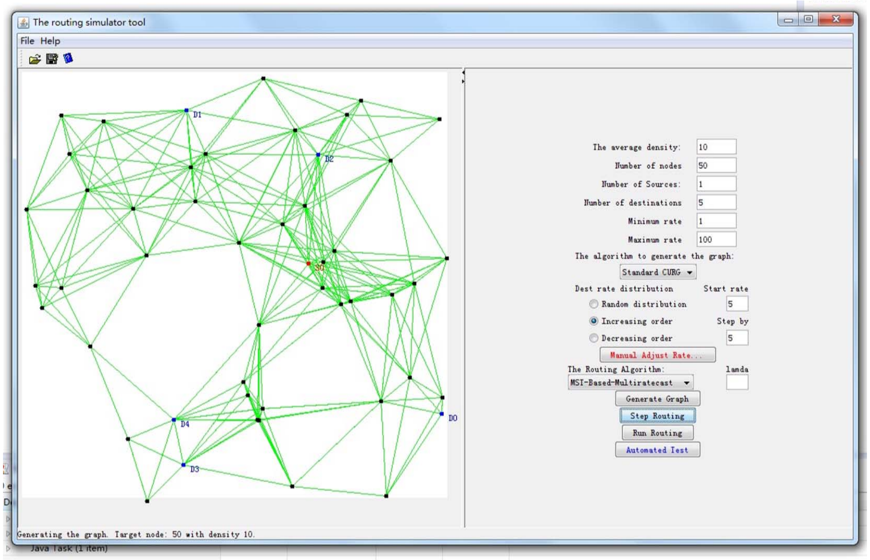
Performance Evaluation

4.1 Simulation Network

We simulated and compared all the multiratecast algorithms in simulation networks by using Java. The networks use Unit Graph Model and are generated by the CRUG (Connected Random Unit Graph) algorithm. The algorithm requires the number of nodes (N) and the average density degree (d) as input, and runs as follows:

For a given interval, each node is placed randomly first. There are total $\frac{N\times(N-1)}{2}$ edges, among all the N nodes, sort which are by their length in increasing order. The radius R that corresponds to a chosen average degree d can be set to equal to the length of the $\frac{N\times d}{2} - th$ edge in the sorted order. Any edge that is not longer than R will remain in the graph, while other edges are eliminated. The Figure 4.1 and Figure 4.2 show the generated network with certain parameters.

Once the graph is generated, Dijkstra's shortest path algorithm (from one node to all other nodes) is used to check the connectivity of the graph. Only connected graphs are saved and used for simulation. The unconnected graphs



The average density: 10
Number of nodes 50
Number of Sources: 1
Number of destinations 5
Minimum rate 1
Maximum rate 100
The algorithm to generate the graph: Standard CURG
Dest rate distribution
○ Random distribution
● Increasing order
○ Decreasing order
Start rate 5
Step by 5
The Routing Algorithm: MSI-Based-Multiratecast
lamda

Generating the graph. Target node: 50 with density 10.

Figure 4.1: Random network with 50 nodes, 5 destination

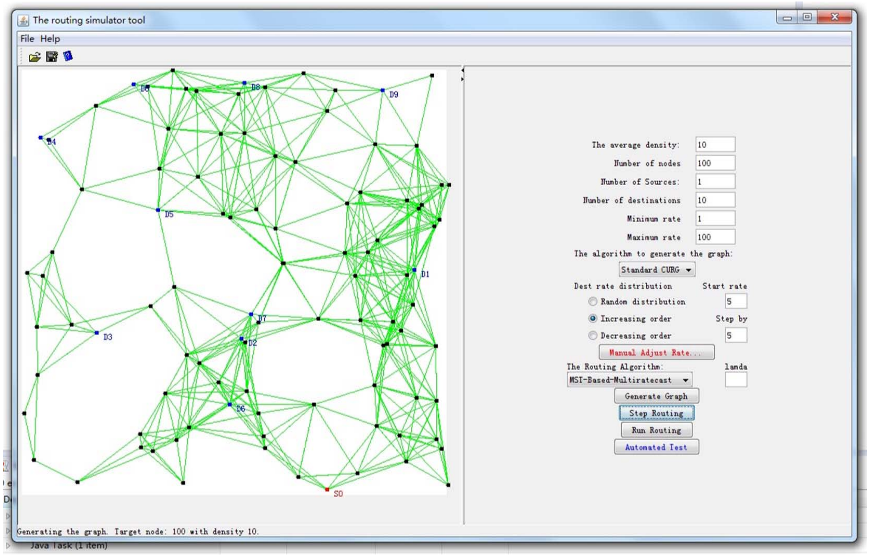
are discarded. The source and destination nodes are randomly selected in the connected network. We test the network routing from one single source node to multiple destinations because multiratecast algorithms are designed to base on this one-way, point-to-multipoint transmission. The data rate required by each destination node is able to be changed when user tests simulation network.

4.2 Testing Parameters

In order to observe the routing algorithm performance in different scenarios, we simulated different network by alternating following parameters:

- D (Average Degree): The efficiency of the multiratecast may depend on different network density. In order to test the average degree impact on the routing performance, we tested the network from sparse to dense network by using average degree: 8, 12, 16, 20, 24, 28, 32.

- DN (Number of Destinations): In order to test the destination number



The average density: 10
Number of nodes 100
Number of Sources: 1
Number of destinations 10
Minimum rate 1
Maximum rate 100
The algorithm to generate the graph: Standard CURG
Dest rate distribution
○ Random distribution
● Increasing order
○ Decreasing order
Start rate 5
Step by 5
The Routing Algorithm: MSI-Based-Multiratecast
lamda

Generating the graph. Target node: 100 with density 10.

Figure 4.2: Random network with 100 nodes, 10 destination

impact on the routing performance, we choose number 2, 3, 4, and 5 to study the routing performance for network with small number of destinations and destination number of 10, 20, 30, 40, 50 to study performance for networks with large number of destinations.

- N (Number of Nodes): We choose network with 100, 300, 500 nodes to study the protocols performance with small number of destinations (2-5 destinations). We choose network with 500, 800, 1000 nodes to study the protocols performance with larger number of destinations (10-50 destinations).

- SD (Rate Distribution): To study the impact of destination rate distribution on routing performance, we studied network with same average destination rate but with different rate standard deviation. For networks with only 5 destinations, two data rate groups are applied. One has the average rate 1000 and the rate standard deviation of 0, 7.14, 16.9, 14.3, 33.9 as listed in table. For networks with larger number of destinations, we studied the network with 10, 20, 30, 40, 50 destinations. We applied a destination rate with a average rate 1000 and standard deviation of 100,300,500,700. To study the impact of more drastic rate distribution, we applied a group of more drastic rate distribution for network with 6 destinations. The average rate for all 6 destinations is 18518.5. The rate standard deviation is 403.73, 7567.17, 14066.17, 20894.2, 40108.36.

- T (Test Execution Time): For each test scenario with given protocol, given network nodes, average degree, given destination nodes and rate, we generate 50 network graphs and perform routing on every one of them. We save and use the average routing results of 50 executions as final result to do analysis.
- P (Protocols): We tested protocol MSTRC,RCM and MRM. For MSTRC protocol, we applied the third rate cost factors method as described in previous chapter and compare results to find out which rate cost factor is the most efficient in an overall multiratecast routing performance.

Table 4.1: Destination rate group for network with 5 destinations

Mild Destination Rate Distribution					Average Rate	SD
D_1	D_2	D_3	D_4	D_5		
100	100	100	100	100	100	
97	103	111	96	93	100	7.141428
90	80	115	120	95	100	16.95582
70	110	127	79	114	100	24.32077
130	120	50	120	80	100	33.91165

Table 4.2: Rate distribution for network with large number of destinations

Destination	Average Rate	SD			
D_1-D_{50}	1000	100	300	500	700

4.3 Performance Metrics

To assess the performance of the proposed algorithms, we used the total rate cost as a performance metric that is described previously as the total transmission rate in routing paths. As mentioned, the proposed multiratecast protocols are novel multicast protocols designed for routing in wireless sensor networks where each destination requests data at different rate.

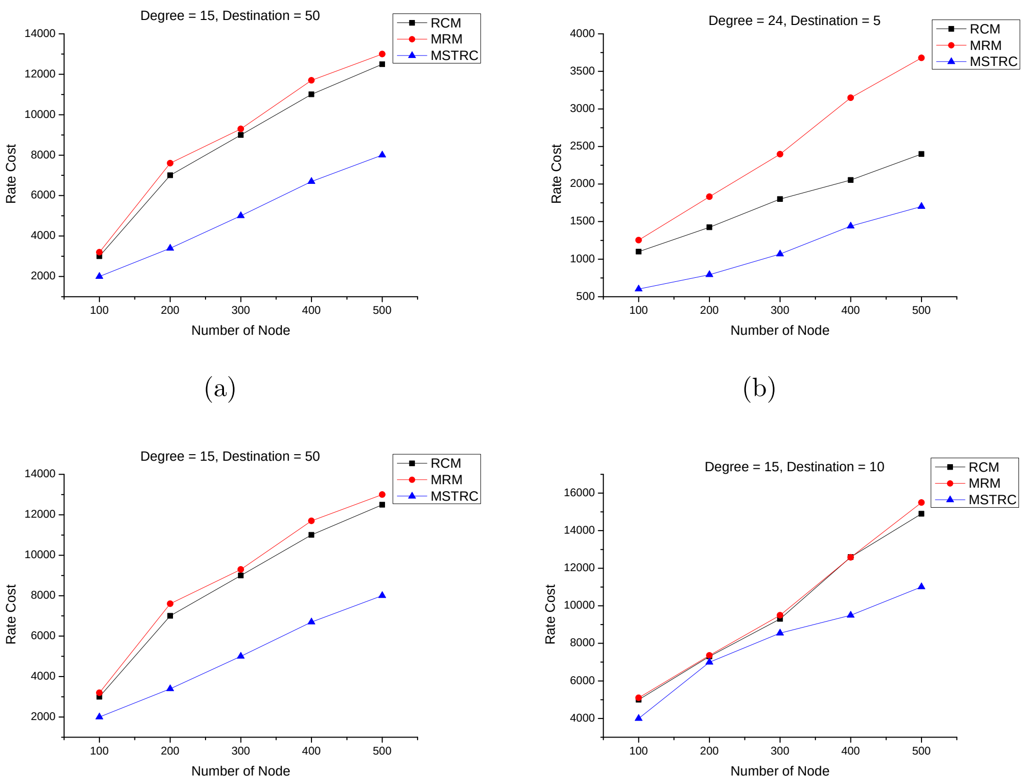
Existing common performance metric like hop count does not reflect the efficiency for multiratecast routing. In simulation, we record the transmission rate at each forwarding node and use the sum of transmission rate as performance metric to evaluate different protocols.

4.4 Simulation Results

4.4.1 The Impact of Number of Nodes

To evaluate the impact of network node number on the routing protocol, we studied the routing data based on the networks that had same average degree, same destination number and same rates but only differ in the number of network nodes. The simulation results show that as the increases of network nodes, the routing cost increase and the increase ratio is similar to all multirate multicast protocols. We select some of the results listed in below as examples.

In Figure 4.3, the first two sub-figures show an example of routing cost for networks with 5 destinations. The destination rate distribution is equal to Ta-



(a)

(b)

(c)

(d)

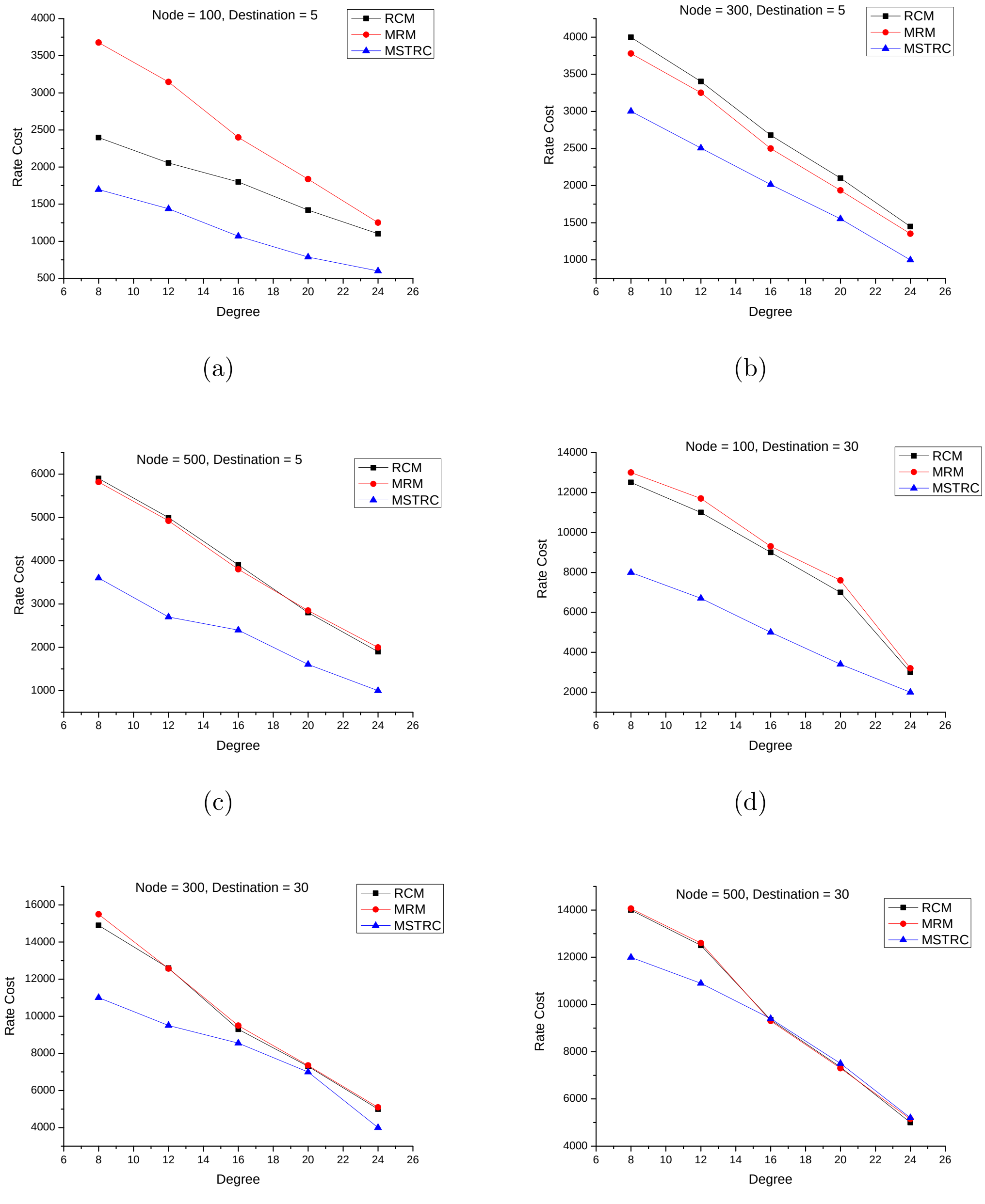
Figure 4.3: Rate Cost at Varying Node Number

ble 4.2. The average degrees for the simulation network are equal to 12 and 24. The left sub-figure shows routing results in network with 12 degree. The right sub-figure shows results in network with 24 degree. MSTRC, RCM and MRM had similar costs but RCM obviously has more routing costs. We can see that MSTRC slightly outperforms others. The increase ratio is similar to all the protocols. For network with a large number of destinations, the routing cost also increases along with network nodes. The other two sub-figures show examples of two network routing results. One is for a network with 10 destinations and average degree 15, the other is for a network with 50 destinations. The results show that no matter the network is sparse or dense, the rate cost increases as the node number increases, MRM has the worst performance while MSTRC protocol outperforms RCM. With the network node number increases, the difference of performance between MRM and MSTRC increases. MSTRC shows more advantages in network with more destination numbers.

4.4.2 The Impact of Average Degree

To measure the performance with different network density, we analyzed the routing cost from sparse network to dense network by increasing average degree. For network with same destination number, same destination rate, same number of network nodes, we studied routing results with average degrees of 8, 12, 16, 20, 24, respectively.

Simulation results in Figure 4.4 show that as the increase of network degree, the cost of multiratecast routing decreases dramatically. We can see that the proposed MSTRC protocols are more efficient and suitable for high density networks because they have more chances to select the best next forwarding node in greedy and less chances in face routing.



(a) (b) (c) (d) (e) (f)

Figure 4.4: Rate Cost at Varying Degree

Results of routing cost for small network with 5 destinations and average rate 100, distribution standard deviation 7.14 are show in the sub-figures. For network with 100, 300 and 500 nodes, as the network degree increases, the cost of all protocols routing decreases. MRM is the worst performer that has the largest cost than others while the other two method RCM and MSTRC show similar performance, and MSTRC slightly outperforms others.

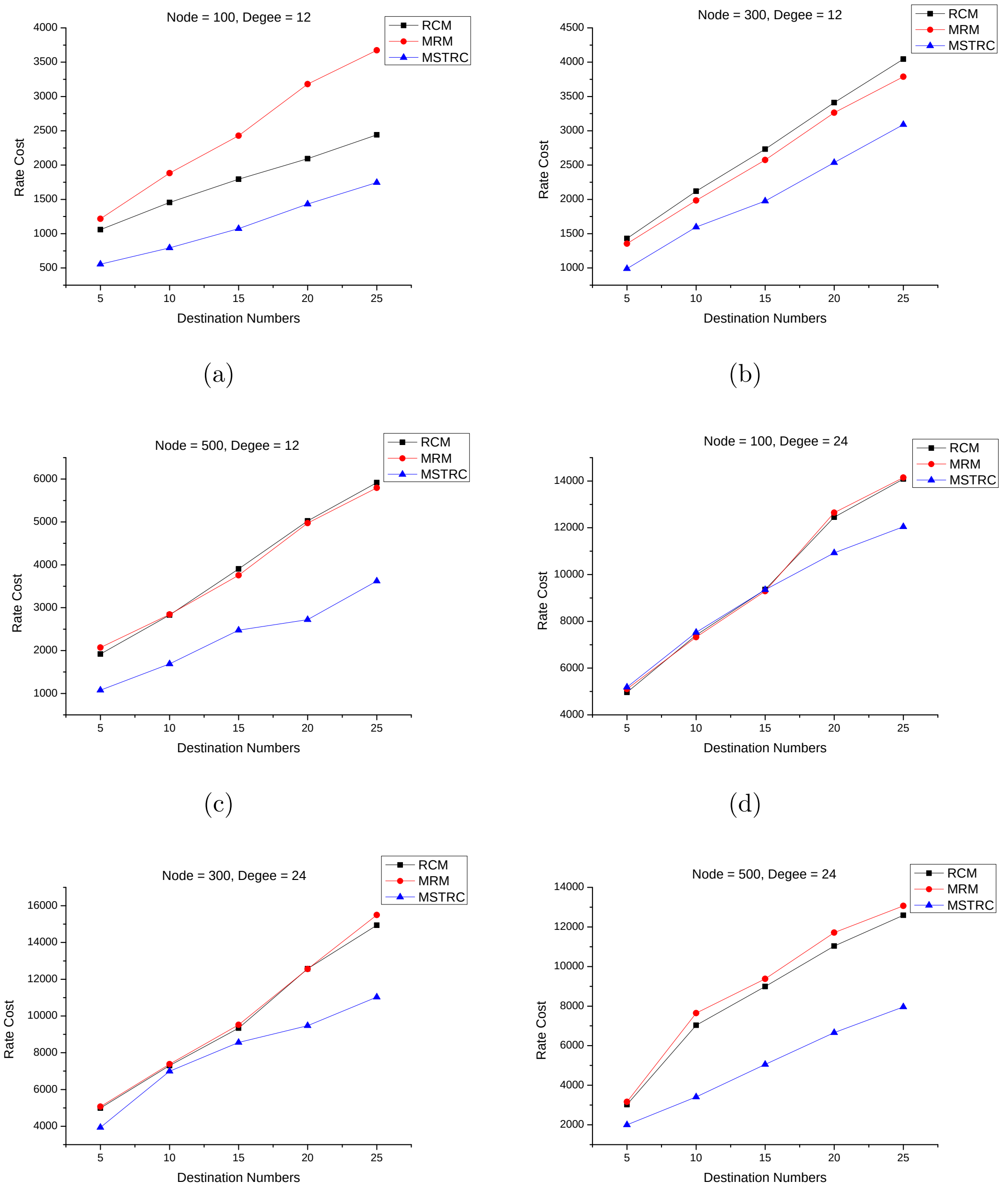
We can observe from the Figure 4.4 that the routing cost for a network with 30 destinations and an average rate of 300, a distribution standard deviation of 300. For networks with 100 nodes, 300 nodes and 500 nodes, as the network average degree increases, the routing costs of all the protocols decrease. MRM is still the worst performer, however its rate cost is not so large as compared to that in smaller number of destinations.

From the simulation results, we can conclude that both MSTRC and RCM have a good routing performance for a large and dense network while MSTRC is superior. This partly because we apply the greedy set partition for large network. The routing with the greedy set partition may depend on a initial set partition. A non-optimal set partition may result in more routing costs.

4.4.3 The Impact of Numbers of Destinations

To study the effect of the number of destinations on the routing performance, we compared routing results in the same network scenario except the destination number differs. As expected, the routing costs increase as the increasing of destination numbers in both a small and a large number of networks. However, the speed of ratio increase is slower than that of the destination number increase.

We have two networks with small number of destinations. One is a network with 300 nodes, average degree 20, average rate 100 and rate distribution 16.9.



(a) (b) (c) (d) (e) (f)

Figure 4.5: Rate Cost at Varying Destination Number

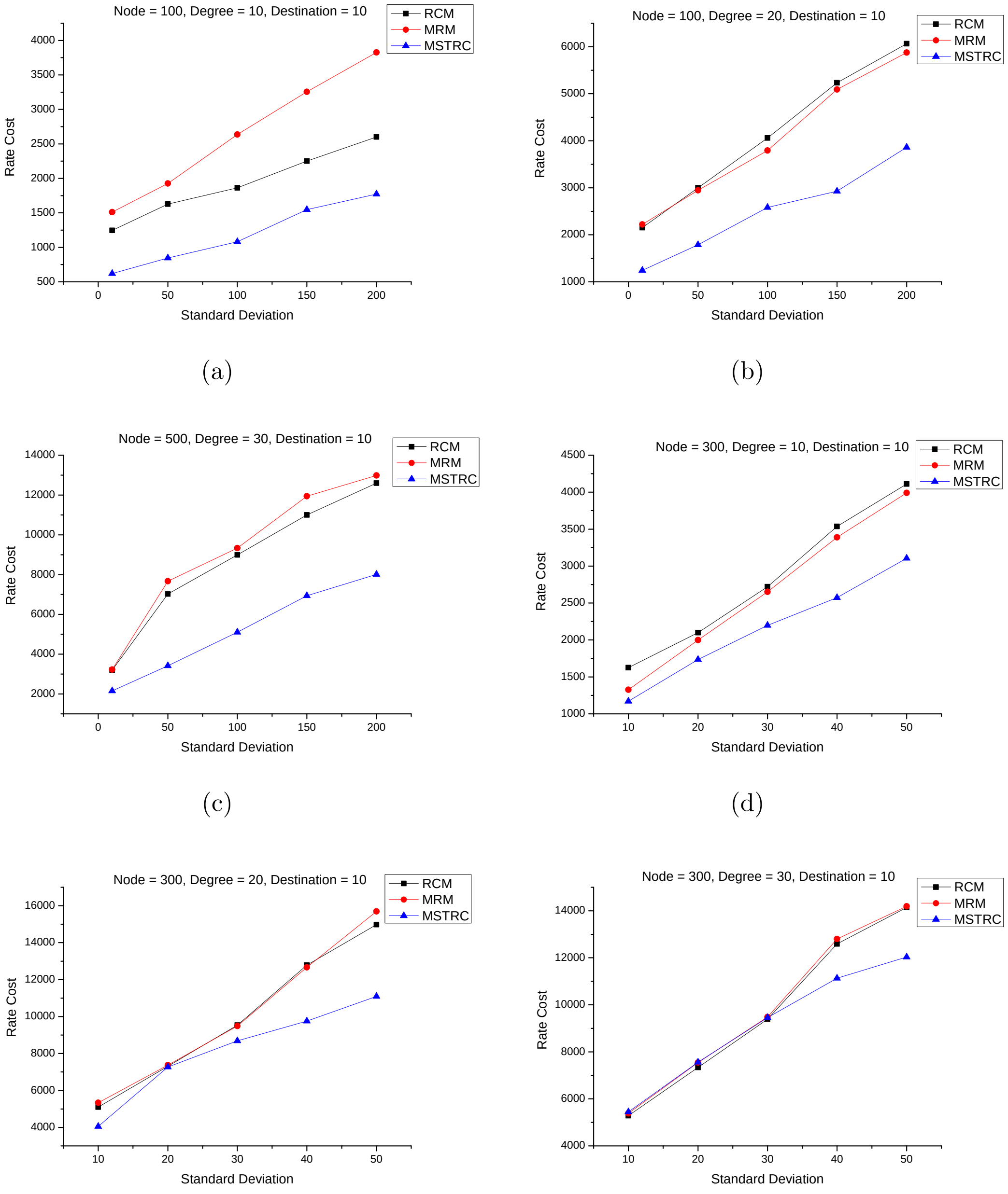
The other is a network with 500 nodes, average degree 28, average rate 100 and rate distribution 24.3. By increasing the destination number from 2 to 5, which is increased by 150% , the average cost only increase from 900 to 1500 which is 66% increase.

For networks with a large number of destinations, we list two examples. The first network is with 300 nodes, an average degree of 12, an average rate of 100 and a rate distribution of 500. The second network is with 500 nodes, an average degree of 24, an average rate of 100 and a rate distribution of 300. As we can see from Figure 4.5, when destination number increases 5 times from 10 to 50, the rate cost only increases less than 3 times.

The MSTRC has a better performance in the network with a large number of destinations. It is reasonable; the RCM applies the greedy destination set partition instead of a complete destination set partition. However, the rate cost calculation of the RCM is partly based on the selection of a initial destination set. The results show that multiratecast protocols reduce overall routing cost by sharing paths. From the results we can conclude that both MSTRC and MRM are efficient multicast protocols and MRM has a better performance than RCM with the greedy set partition and is a practical protocol suitable for a large network with a large number of destinations.

4.4.4 The Impact of Rate Distribution

Multiratecast protocols are specially designed to fulfill the requirement that destinations request data at different rate. To evaluate the impact of destination rate changes to routing performance, we studied networks with the same degree, same network node numbers, same destination number and same average destination rate but only different rate distributions.



(a) (b)

(c) (d)

(e) (f)

Figure 4.6: Rate Cost at Varying Rate Distribution

We first study the network with a small number of destinations and mild rate changes. MSTRC routing in this scenario applies a complete destination set partition. Figure 4.6(a), 4.6(b) show routing results for a network with 5 destinations with a mild rate distribution. Routing results for networks with small network nodes (100), medium network nodes (300) and large network nodes (500) are displayed. The degrees for these networks are low(8), medium(24) and high(32). For each simulation network graph, 5 groups of different data rates are used. The average data rate for all destinations are set to the same number 100, the rate distribution standard deviation is set in increasing order from 10 to 50. It is observed that MSTRC has the lowest total rate cost and RCM has the highest cost. With the increase of the rate standard deviation, the costs slightly increase for three protocols. The cost for RCM keeps flat or slightly decreases although its cost is always larger than the other two algorithms.

Secondly, we study the same networks with 5 destinations but more drastic rate distribution. Simulation results in Figure 4.6(c) and 4.6(d) show that when the destination's data rate differences are more drastic, the data rate distribution of destinations affects the routing performance for RCM and MRM.

The performance of the algorithms with increasing rate distribution in three network configurations similar as we studied before but as the average data rate changed to 222.2 and rate standard deviation increased from 26.11 to 468.04. It is observed that as the data rate difference increase, all the algorithms' rate costs increase. However, the difference of routing performance between RCM and MRM, and the difference between three RCM methods are decreased. When the destinations have most data rate differences, RCM and MRM have similar performance. MSTRC is still the best among all the algorithms, but the cost differences among the three algorithms are not as big as in previous mild rate distribution scenario.

RCM costs increase more as the rate distribution increasing and it becomes the worst performer in the network with the largest rate distribution. MRM keeps flat rate cost and is still the worst algorithm compare to others. MSTRC slightly outperforms MRM. The results show that when data rates have relatively more drastic differences among destinations, the routing algorithms that consider the destination with maximum data rate is most important in routing decision. We can conclude from the results that MSTRC with a complete set partition is the best for network with small number of destinations but MRM is also very competitive.

Thirdly, we study the network with a large number of destinations. RCM apply the greedy set partition in this scenario. We ran simulation on networks with the same network nodes (500,800,1000), degree (8,12,16,20,24,28,32) and destination number (10,20,30,40,50), with only destination rate changing. Simulation results show that the rate distribution for each destination has a direct impact on the routing cost. Figure 4.6(e), 4.6(f) show results for networks with 500 and 1000 nodes, with the destination number of 10 and 50, and with the degree of 24 and 32. The average rate for both of the networks is 1000 and rate distribution standard deviation is from 100 to 700. It is observed that as the rate distribution increases, all protocols have obvious rate cost increase. MSTRC outperforms MRM and RCM.

Although different protocols show different rate cost and the costs increase along with the increasing of standard deviation, when the standard deviation is big enough, or in other words, the rate distribution is drastic enough, all the routing protocols have similar rate cost. The average rate for all 6 destinations is 1858.5. The rate standard deviation is 0, 403.73, 7567.17, 1066.17, 2094.2, 4108.36. From the figures, we can see that MRM outperforms RCM protocol, however, the difference of their performance gets smaller when the rate standard deviation increases. When the standard deviation is big enough, all the protocols have similar rate cost. These are reasonable results because when the rate distribution is extremely unbalanced, the one destination that has much higher rate decides the total routing path selection. Compared to the highest rate destinations, the rate cost of paths to other destinations have little impacts to the overall routing rate cost.

We also can see from the simulation results that for network with a large number of destinations where RCM apply the greedy set partition, MRM has less rate cost than RCM no matter at which destination rate distribution. This is different from the results for network with a small number of destinations where MSTRC with the complete set partition has less rate cost than MRM. We believe it is because the greedy set partition is not as optimal as the complete set partition.

4.4.5 Comparison of Multiratecast and Unicast

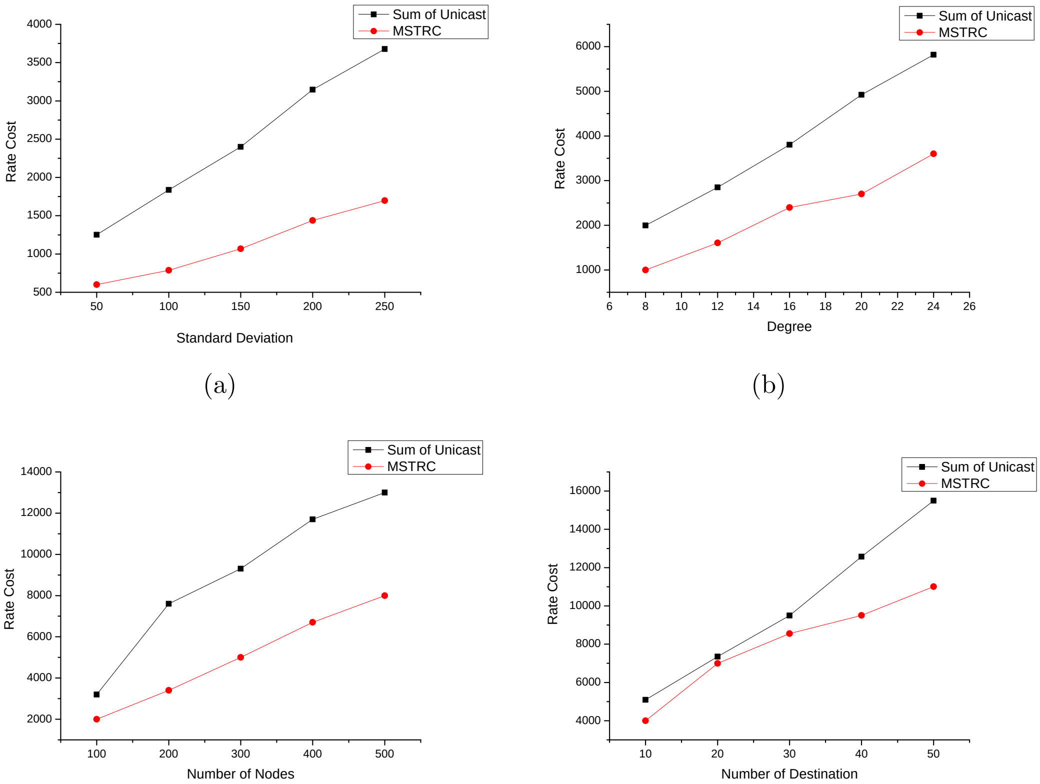
To study the routing efficiency of the multiratecast protocols, we compared the MSTRC multiratecast rate cost to the sum of unicast rate cost in the same network configuration.

The reason to select MSTRC is that: based on previous simulation data analysis, we found among the protocols we study, MSTRC is the best routing protocol for network with a large number of destinations and it is better than the MRM and RCM with the greedy set partition. Besides, it is also a very competitive protocol for networks with a small number of destinations. We believe the comparison between the MSTRC and unicast rate cost will provide a proper representation of the efficiency routing multiratecast.

The following sub-figures in Figure 4.7 give examples of MSTRC and the sum of unicast cost based on networks with one variable parameter. The unicast routing is performed from a source node to each destination node individually. During the forwarding procedure, the neighbor node that is closest to destination node will be selected as next relay node until message reaches the destination node. The sum of transmission rate at each relay node on the unicast path to the destination is the rate cost for the destination. The total rate cost for all destinations is the sum of unicast rate cost that we use to compare to MSTRC. As we can see from the figures that MSTRC has a lower rate cost than the sum of unicast rate cost in all simulation scenarios.

Figure 4.7(a) shows the impact of the destination rate standard deviation on the rate cost. It shows a network with 500 nodes, an average degree of 24, 10 destinations and an average destination rate of 1000. It has a relatively mild standard deviation of 100, 300,500,700. The rate cost of MSTRC slightly increases as the SD increasing, however, the sum of unicast cost almost remains same. From the figure, we see the sum of unicast cost only slightly increases but there is big increase in MSTRC. At the highest standard deviation, the MSTRC cost is almost the same as the sum of unicast cost. This is because the destination with the highest rate has the most impact for the overall routing cost, the cost of other destination with lower rate cost is not so significant compared to the one with highest rate, thus the multiratecast has similar rate cost as the unicast.

Figure 4.7(b) shows rate cost changes as only the average degree changing. The network is with 5 destinations, 500 nodes, and an average rate of 2222.2. Both MSTRC rate cost and the sum of unicast cost decrease as the increasing of degree. This is because as the average degree increases in the same 500 nodes network, there are more directly connected nodes in the network which results in



(a) (b)

(c) (d)

Figure 4.7: Comparisons of MSTRC and sum of Unicast

less chance of message relay and face routing.We can conclude that the MSTRC shows more efficiency in network with a large number of network nodes.

Figure 4.7(c) shows rate cost varies as only number of network nodes increase. We studied network with 5 destinations, an average degree of 20, and an average rate of 1858.5. As the network nodes increase from 100 to 300 and 500, the rate cost increases in the simulation results. However, the increase of MSTRC rate cost is slower than the sum of unicast cost. MSTRC shows a larger superiority in network with a larger number of destinations.

Figure 4.7(d) shows rate costs changes as destination number changing. We studies the network with 500 nodes, an average degree of 20, and an average rate of 1000. As the destination number increases from 10 to 50, we can see that the sum of unicast cost increases sharply while MSTRC only has a mild increase. The larger the destination number, the bigger the difference between MSTRC and unicast rate cost. We can conclude that the destination number has a quite significant impact for the routing.

Chapter 5

Conclusions

We introduced the multiratecast protocols MSTRC for a generic multirate multicast routing purpose in WSNs. MSTRC is a purely localized multicast protocol. It does not require the use of an extensive broadcast to route messages to a set of destinations. The MSTRC protocol aims to resolve the efficient multiratecast routing problem where destinations request data at different rate. It uses a rate-aware MST construction process to form the backbone for routing and cost-based neighbor selection at each routing step, allowing it to find a good tradeoff between the optimality of the normal multicast tree and the efficiency of data delivery.

The algorithm calculates the rate cost at each relay node to select the best forwarding nodes with the lowest rate cost so that the overall routing cost can be optimal. The locally-built MST that also fits the rate requirements from different sink nodes in an efficient way is used as an efficient approximation of the optimal routing backbone. Using a MST is highly relevant in the context of dynamic wireless networks since its computation has a low time complexity $O(n \log n)$. After constructing the MST, a message duplicate occurs and the set of destinations has multiple edges originated at the current node. Destinations spanned by each of these edges are grouped together, and for each of these subsets the best neighbor is selected as the next hop. We introduced three measurement methods to select the next hop from neighbors. Since such greedy localized scheme may lead the message to a void area (i.e., there is no neighbor providing a positive progress toward the destinations), we also proposed a new multiratecast generalization of the well-known face recovery mechanism.

Computing the optimal multicast tree in terms of bandwidth consumption is similar to finding the multicast tree with the minimum number of forwarding nodes, and was shown to be NP-complete. Thus, the use of the greedy forwarding scheme is fully justified. The rate-aware MST construction process was first introduced in our work, it allows the transmission achieving an optimal rate cost and building a backbone minimum spanning tree to guide the data forwarding. In addition, using the greedy packet splitting is also a good idea for sensor networks because the routing decision at each single node is performed based on the current network topology. Thus, MSTRC is able to adapt to topological changes due to the sensor operating in a duty cycle. One of the other key aspects of MSTRC is the cost-aware neighbor selection function. The reason is that the shape of the tree is one of the key parameters governing the overall optimality of the tree. Given that the complexity of testing all possible neighbor subsets grows exponentially, we proposed a heuristic algorithm which has shown to be efficient compared with other protocols.

For MSTRC, we did the simulations for networks with a small and large number of destinations. We analyzed routing results based on the variation of the numbers of destination , network node, average degree and rate distribution. We also compared rate costs for different protocols and the rate costs of the sum of unicast. The above studies are based on a list of assumptions, including the position of destinations is known by the source, an ideal MAC layer, no packet loss and every node is capable of transmit data at destination required rates.

The MSTRC have common limitations that the destinations' position information is included in the message head. The encoding overhead will be significant if the number of destination is increasing to a large number. We do not compare the message overhead in our simulation because the overhead does not change from one to another position. The overhead can cause heavy computation time at a forwarding node. A new recently proposed location-based protocol optimizes the coding efficiency by destination grouping and hierarchy. Reduce the message overhead and computation time will be future research topics related to the position-based multiratecast.

Future work related to geographical multicast and multiratecast can be doing in the following areas:

1. We plan to analyze different possible solutions to optimize the multiratecast face routing.
2. The cost-aware functions can be changed to deal with energy efficiency and realistic physical layers.
3. MSTRC works perfectly when the number of receivers is not very large. Dealing with a larger number of receivers needs us to further develop new approaches.
4. The movement or joining/leaving of the network of sensor node may cause routing loop, we may study the delivery rate in this situation.
5. Power and energy efficiency metrics can be used to compare different routing protocols.
6. For the assumption that every node can provide maximum data rate for destinations is not valid or some relay node has lower transmission rate, how to adapt the multiratecast routing path selection.
7. Multiple sources to multiple destinations data routing may require a data merge. Multiratecast protocols need to support the many-to-many routing request.
8. If several destinations request the same rate, destinations can be grouped according to specific rates and the multiratecast routing protocols can be changed based on rate groups.

References

[1] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[2] A. Caruso, S. Chessa, S. De, and A. Urpi. Gps free coordinate assignment and routing in wireless sensor networks. In *Proc. of IEEE INFOCOM*, pages 150–160, 2005.

[3] K. Chen and K. Nahrstedt. Effective location-guided overlay multicast in mobile ad hoc networks. In *International Journal of Wireless and Mobile Computing*, volume 3, pages 1–15, 2005.

[4] S.K. Das, BS BS Manoj, and C.S. Ram Murthy. A dynamic core based multicast routing protocol for ad hoc wireless networks. In *Proc. of ACM Int. Symp. on Mobile Ad Hoc Networking & Computing*, pages 24–35, 2002.

[5] G.G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical report, DTIC Document, 1987.

[6] R. Fonseca, S. Ratnasamy, J. Zhao, C.T. Ee, D. Culler, S. Shenker, and I. Stoica. Beacon vector routing: Scalable point-to-point routing in wireless sensornets. In *Proc. of Symp. on Networked Systems Design & Implementation*, pages 329–342, 2005.

[7] H. Frey, F. Ingelrest, and D. Simplot-Ryl. Localized minimum spanning tree based multicast routing with energy-efficient guaranteed delivery in ad hoc and sensor networks. In *Proc. of Int. Symp. on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–8, 2008.

[8] H. Frey and I. Stojmenovic. On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks. In *Proc. of 12th Int. Conf. on Mobile Computing and Networking*, pages 390–401, 2006.

[9] K.R. Gabriel and R.R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Biology*, 18(3):259–278, 1969.

[10] JJ Garcia-Luna-Aceves and E.L. Madruga. The core-assisted mesh protocol. *IEEE Journal on Selected Areas in Communications*, 17(8):1380–1394, 1999.

[11] M. Gerla, CC Chiang, and SJ Lee. On-demand multicast routing protocol. Technical report, Internet Draft, http://www. ietf. org/internet-drafts/draft-ietf-manet-odmrp-00. txt, 1998.

[12] S. Giordano, I. Stojmenovic, and L. Blazevic. Position based routing algorithms for ad hoc networks: A taxonomy. *Ad Hoc Wireless Networking*, pages 103–136, 2004.

[13] Z. Guo and Y. Yang. On nonblocking multirate multicast fat-tree data center networks with server redundancy. In *Proc. of IEEE IPDPS*, pages 1034–1044, 2012.

[14] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. Impact of network density on data aggregation in wireless sensor networks. In *Proc. of IEEE ICDCS*, pages 457–458, 2002.

[15] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Proc. of IEEE Int. Conf. on Technology for the 21st Century*, pages 62–68, 2001.

[16] L. Ji and M.S. Corson. A lightweight adaptive multicast algorithm. In *Proc. of IEEE GLOBECOM*, pages 1036–1042, 1998.

[17] L. Ji and M.S. Corson. Differential destination multicast-a manet multicast routing protocol for small groups. In *Proc. of IEEE INFOCOM*, pages 1192–1201, 2001.

[18] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353:153–181, 1996.

[19] B. Karp and H.T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proc. of Int. Conf. on Mobile Computing and Networking*, pages 243–254, 2000.

[20] R.M. Karp. Reducibility among combinatorial problems. *50 Years of Integer Programming 1958-2008*, pages 219–241, 2010.

[21] J. Kuruvila, A. Nayak, and I. Stojmenovic. Hop count optimal position-based packet routing algorithms for ad hoc wireless networks with a realistic physical layer. *IEEE Journal on Selected Areas in Communications*, 23(6):1267–1275, 2005.

[22] S. Lee and C. Kim. Neighbor supporting ad hoc multicast routing protocol. In *Proc. of Mobile and Ad Hoc Networking and Computing (MobiHOC)*, pages 37–44, 2000.

[23] H. Li, C. Wu, Z. Li, W. Huang, and F.C.M. Lau. Stochastic optimal multirate multicast in socially selfish wireless networks. In *Proc. of IEEE INFOCOM*, pages 172–180, 2012.

[24] X. Liu, A. Casteigts, N. Goel, A. Nayak, and I. Stojmenovic. Multiratecast in wireless fault-tolerant sensor and actuator networks. In *Proc. of 2nd IEEE Int. Conf. on Computer Science and its Applications*, 2009.

[25] M. Mauve, H. Füßler, J. Widmer, and T. Lang. Position-based multicast routing for mobile ad-hoc networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):53–55, 2003.

[26] A. Mizumoto, H. Yamaguchi, and K. Taniguchi. Cost-conscious geographic multicast on manet. In *Proc. of IEEE SECON*, pages 44–53, 2004.

[27] U.T. Nguyen, A. Asif, and X. Xiong. Multirate-aware multicast routing in manets. In *Proc. of IEEE MASS*, pages 554–557, 2006.

[28] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *ACM SIGCOMM Computer Communication Review*, 24(4):234–244, 1994.

[29] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[30] V. Rodoplu and T.H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333–1344, 1999.

[31] Pedro M. Ruiz and Antonio F. Gomez-Skarmeta. Approximating optimal multicast trees in wireless multihop networks. In *Proc. of IEEE Symposium on Computers and Communications*, pages 686–691, 2005.

[32] J.A. Sanchez, P.M. Ruiz, J. Liu, and I. Stojmenovic. Bandwidth-efficient geographic multicast routing protocol for wireless sensor networks. *IEEE Sensors Journal*, 7(5):627–636, 2007.

[33] J.A. Sanchez, P.M. Ruiz, J. Liu, and I. Stojmenovic. Bandwidth-efficient geographic multicast routing protocol for wireless sensor networks. *Sensors Journal, IEEE*, 7(5):627–636, 2007.

[34] G. Singh, S. Pujar, and S. Das. Rate-based data propagation in sensor networks. In *Proc. of IEEE Wireless Communications and Networking Conference*, pages 2480–2485, 2004.

[35] P. Sinha, R. Sivakumar, and V. Bharghavan. Mcedar: Multicast core-extraction distributed ad hoc routing. In *Proc. of IEEE Wireless Communications and Networking Conference*, pages 1313–1317, 1999.

[36] I. Stojmenovic. Position-based routing in ad hoc networks. *IEEE Communications Magazine*, 40(7):128–134, 2002.

[37] I. Stojmenovic. Localized network layer protocols in wireless sensor networks based on optimizing cost over progress ratio. *IEEE Network*, 20(1):21–27, 2006.

[38] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.

[39] M. Transier, H. Füßler, J. Widmer, M. Mauve, and W. Effelsberg. Scalable position-based multicast for mobile ad-hoc networks. In *Proc. of Int. Workshop on Broadband Wireless Multimedia: Algorithms, Architectures and Applications*, 2004.

[40] J. Widmer, M. Mauve, H. Hartenstein, and H. Füβler. Position-based routing in ad hoc wireless networks. In *The Handbook of Ad Hoc Wireless Networks*, pages 219–232. CRC Press, Inc., 2003.

[41] C.W. Wu and YC Tay. Amris: A multicast protocol for ad hoc wireless networks. In *Proc. of IEEE Military Communications Conference*, pages 25–29, 1999.

[42] S. Wu and K.S. Candan. Gmp: Distributed geographic multicast routing in wireless sensor networks. In *Proc. of IEEE ICDCS*, pages 49–54, 2006.

[43] J. Xie, R.R. Talpade, A. Mcauley, and M. Liu. Amroute: ad hoc multicast routing protocol. *Mobile Networks and Applications*, 7(6):429–439, 2002.

[44] N. Xiong, X. Jia, L.T. Yang, A.V. Vasilakos, Y. Li, and Y. Pan. A distributed efficient flow control scheme for multirate multicast networks. *IEEE Transactions on Parallel and Distributed Systems*, 21(9):1254–1266, 2010.

[45] N. Xiong, A.V. Vasilakos, L.T. Yang, and E. Hossain. An adaptive and predictive approach for autonomic multirate multicast networks. *ACM Transactions on Autonomous and Adaptive Systems*, 6(3), 2011.

[46] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical report, Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, 2001.

[47] Gang Zhao, Xiangqian Liu, and Anup Kumar. Destination clustering geographic multicast forwireless sensor networks. In *Proc. of Int. Parallel Processing Workshops*, pages 48–, 2007.

[48] J. Zou, H. Xiong, L. Song, Z. He, and T. Chen. Prioritized flow optimization with generalized routing for scalable multirate multicasting. In *Proc. of IEEE ICC*, pages 1–6, 2009.