LOAD BALANCED FORWARDING IN MULTIMEDIA WIRELESS SENSOR NETWORKS

by

Sinan Işık

B.S., Mathematics, Boğaziçi University, 1999

M.S., Computer Engineering, Boğaziçi University, 2003

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2011

ACKNOWLEDGEMENTS

The pursuit for a PhD degree has covered a long period of my life, which has been challenging but rather salutary for my personality. The most influential character of this episode has no doubt been my advisor, dear Cem Ersoy. I have always seen him as my academic father and I have learned and acquired a lot from him. When I get confused and demotivated, by expressing his feelings of trust on me, he made me to renovate my self-confidence and keep going on towards the target. I feel very lucky that I had the chance to work with him who thoroughly shaped my academic character and I deeply feel that he will continue to do so for the rest of my life.

I thank all of my thesis jury members, Assist. Prof. Turgay Altılar, Prof. Emin Anarım, Prof. M. Ufuk Çağlayan and Dr. Murat Zeren for their time and valuable comments. Their remarks have improved the quality of this thesis significantly.

I have very much benefitted from the discussions with my dear friend M. Yunus Dönmez. I am very thankful to him for his supportive fellowship during the journey of our academic life starting in 1995.

I also want to thank my mother Fatma Işık, my sister Sibel Işık and my parents-in-law Suna and Selahattin Özcanlı for all their physical and moral support. Apart from them, I thank my dear departed father Recep Işık whose support that I always feel in my heart.

Finally, and more than anybody, I would like to thank my one and only *love* Mehtap, who gives me the feeling that I could reach the farthest star in the universe. She suffered with me through the most difficult stages of this study and supported me unconditionally. I will never forget her grace under pressure. Besides all, she gave me the most enchanting present of my life, my *lovely* daughter Alkım. Mehtap and Alkım are the cheerful colors of my life. I devote this thesis to my *love* and *lovely*. I love you more than anything...

This work is supported by The Scientific and Technological Council of Turkey (TUBITAK) under the Grant Number 108E207 (COST ACTION IC0906-WINEMO), by Bogazici University Research Fund (BAP) under the Grant Number 5146 and by the European Community's Seventh Framework Programme (FP7-ENV-2009-1) under the grant agreement FP7-ENV-244088 FIRESENSE.

ABSTRACT

LOAD BALANCED FORWARDING IN MULTIMEDIA WIRELESS SENSOR NETWORKS

Congestion is a challenging problem in wireless sensor networks which exacerbates with the high volume of data traffic imposed by video applications that are enabled by the evolution of low-cost camera hardware. In this thesis, we propose three cross layer geographic routing schemes addressing the congestion problem in wireless video sensor networks. In Load Balanced Reliable Forwarding scheme, we introduced the notion of local load balancing where a sensor dynamically determines the next hop among the alternative neighbors providing positive advancement towards the sink by considering the balance of their buffer occupancy levels at the time of delivery. In Directional Load Balanced Spreading scheme, we combine local and direction-based (spatial) load balancing approaches to provide more reliable and faster video delivery by benefiting from the advantages of both approaches. In Multi-Sink Load Balanced Reliable Forwarding scheme, we explored the advantages of load distribution to multiple sinks. We evaluated the performance of the routing schemes via simulations in OPNET. The results show that the proposed routing schemes improve the frame delivery ratio and provide efficient video delivery in terms of energy expenditure per successfully delivered frame to the sink(s).

ÖZET

ÇOKLUORTAM KABLOSUZ ALGILAYICI AĞLARDA DENGELİ YÜK İLETİMİ

Tıkanıklık, kablosuz algılayıcı ağlarda düşük maliyetli kamera donanımlarının evrimleşmesi ile olanaklı hale gelen video uygulamalarının getirdiği yüksek hacimli veri trafiği sebebiyle kötüleşen, zorlu bir problemdir. Bu tezde, kablosuz video algılayıcı ağlarda tıkanıklık problemini hedef alan üç adet katmanlararası coğrafi iletim şeması öneriyoruz. Yük Dengeli Güvenilir İletim şemasında bir algılayıcının sonraki sıçrama noktasını, çıkış düğümüne pozitif ilerleme sağlayan alternatif komşular arasından, tampon doluluk seviyelerinin dengesini düşünerek, dinamik olarak belirlediği yerel yük dengelemesi kavramını tanıttık. Yönsel Yük Dengeli Serpme şemasında, yerel ve yön tabanlı (uzamsal) yük dengeleme yaklaşımlarını, her ikisinin de avantajlarından faydalanarak, daha güvenilir ve daha hızlı video iletimi sağlamak için birleştirdik. Çok-çıkışlı Yük Dengeli Güvenilir İletim şemasında yükün çoklu çıkışa dağıtımının avantajlarını araştırdık. İletim şemalarının başarımlarını OPNET üzerinde benzetimler vasıtasıyla değerlendirdik. Sonuçlar önerilen iletim şemalarının çerçeve iletim oranını iyileştirdiğini ve çıkış düğümlerine başarıyla iletilen çerçeve başına harcanan enerji açısından verimli video iletimi sağladığını gösterdi.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

b_{max}	Maximum of β_{s_i}s
B	Backward region of a node
c	Radio channel bitrate
c_{s_i}	Candidate relay determined by LBRF in the direction of s_i
C_n	Routing class n
d	The distance of a node to the sink node
d_{max}	Maximum of γ_{s_i}s
d_n	The distance of a neighbor node to the sink node
f_1^r	The class of α in rule r
f_2^r	The class of β in rule r
f_3^r	The class of γ in rule r
f_c^r	The class of consequence in rule r
F	Forward region of a node
n_{max}	Maximum of α_{s_i}s
r	A rule in the knowledge base in MLBRF
r_{dc}	Duty cycle of SMAC
r_t	Transmission radius of a node
s_{th}	Spreading threshold in DLBS
s_i	Sink i in MLBRF
s^*	The sink s with maximum π_s
t_{slot}	Duration of a single contention slot
t_{CTS}	Duration of a CTS packet
t_{RTS}	Duration of an RTS packet
t_{SYNC}	Duration of a SYNC packet
T_{active}	Duration of the awake period
T_{frame}	Duration of an SMAC frame
T_{listen}	Duration of the listen period
T_{sleep}	Duration of the sleep period
T_{sync}	Duration of the synchronization period

W	Contention window size for the data period
W_s	Contention window size for the synchronization period
α	Deviation angle in DLBS
α_H	Adjusted deviation angle α in the relay node in DLBS
α_r	α variable of rule r
α_{s_i}	Number of neighbors of a sensor that has frames in F in the direction of s_i
$\alpha_{s_i}^f$	Fuzzy vector obtained for α_{s_i}
β_r	β variable of rule r
β_{s_i}	Buffer occupancy level of c_{s_i}
$\beta_{s_i}^f$	Fuzzy vector obtained for β_{s_i}
γ_r	γ variable of rule r
γ_{s_i}	Distance of c_{s_i} to the sink s_i
$\gamma_{s_i}^f$	Fuzzy vector obtained for γ_{s_i}
δ_{s_i}	The routing-class of a sensor with respect to s_i
θ_r	Output value of rule r evaluated by REM
μ_c^t	Membership function of consequence for class t, $1 \leq$t≤ 9
μ_χ^H	Membership function of χ, where χ is α, β or γ, for class H
μ_χ^L	Membership function of χ, where χ is α, β or γ, for class L
μ_χ^M	Membership function of χ, where χ is α, β or γ, for class M
ν_r	Contribution of rule r
π_{s_i}	Output of the defuzzifier for s_i
ω_{s_i}	Pattern vector in a sensor for the sink s_i
$\omega_{s_i}^f$	Fuzzy pattern vector in a sensor for the sink s_i

LIST OF ACRONYMS/ABBREVIATIONS

ABS	Angle Based Spreading
ACK	Acknowledgement
AIMD	Additive Increase Multiplicative Decrease
BOC	Buffer Occupancy Condition
CODA	Congestion Detection and Avoidance
CN	Congestion Notification
CR	Characteristic Ratio
CTS	Clear-to-Send
DB	Distance Based
DGR	Directional Geographic Routing
DLBS	Directional Load Balanced Spreading
DQRP	Directional QoS-Aware Routing Protocol
DTMC	Discrete Time Markov Chain
ESRT	Event-to-Sink Reliable Transport
FEC	Forward Error Correction
FOV	Field of View
GeRaf	Geographic Random Forwarding
GPSR	Greedy Perimeter Stateless Routing
H	High
IEEE	The Institute of Electrical and Electronics Engineers
IFRC	Interference-Aware Fair Rate Control
ISO	International Organization for Standardization
JPEG	Joint Photographic Experts Group
L	Low
LBRF	Load Balanced Reliable Forwarding
M	Medium
MAC	Medium Access Control
MLBRF	Multi-Sink Load Balanced Reliable Forwarding
MPEG	Moving Pictures Experts Group

NAV	Network Allocation Vector
NCTS	Negative Clear-to-Send
NIT	Neighbor Information Table
PDE	Partial Differential Equations
QoS	Quality of Service
REM	Rule Evaluation Method
RF	Random Forwarding
RMAC	Routing-Enhanced Medium Access Control
RTS	Request-to-Send
RX	Reception
SMAC	Sensor Medium Access Control
SQCIF	Sub-Quarter Common Interchange Format
STSHC	Source-to-Sink Hop Count
SYNC	Synchronisation
TX	Transmit
VSN	Video Sensor Networks
WRF	Weighted Random Forwarding
WRRF	Weighted Round Robin Forwarding
WSN	Wireless Sensor Networks

1. INTRODUCTION

The evolution of low-cost camera hardware extended the capability of wireless sensor networks (WSNs) [1] to support multimedia adaptations for the typical WSN applications such as surveillance, target tracking, battlefield intelligence and environmental monitoring [2, 3]. Congestion is a challenging problem for (WSNs), due to the intrinsic characteristics of high node density, convergecast communication pattern and multi-hop network topology. Moreover, the necessity of providing a high Quality of Service (QoS) requirement for video traffic imposes extra difficulties besides carrying such traffic over a limited-bandwidth, error-prone wireless channel by the sensors with limited energy budget. Hence, Video Sensor Networks (VSNs) [2] emerged as a new research topic in the area of Wireless Sensor Networks (WSNs).

In VSNs, the logical unit of the communicated data is a video frame, which is composed of video packets. If no encoding schemes such as FEC [4] are utilized in the network, all packets of a video frame are required to be received by the sink node for a successful frame delivery. The performance of the video application, such as the success of object identification or the quality of tracking, highly depends on the reliable delivery of frames to the sink. The reliability concept mentioned in the context of this thesis considers the amount of video frames successfully delivered to the sink in a given sensor network.

The load on a video sensor is determined by the amount of frames created by the sensor itself and by the amount of packets that the other sensors relay over it with the aim of delivery to the sink. Since the sensing range of nodes often overlaps, the same event is usually detected by multiple sensors. The created traffic in these networks is in the form of unpredictable bursts of video frames triggered by sensed events, resulting with a sharp increase in the data input rate of a sensor. In addition, the output rate of a sensor decreases due to contention caused by many concurrent transmission attempts. In these cases, combined data input rate becomes greater than the output rate of a sensor and data packets starts to accumulate in its buffer causing local congestions.

Depending on the buffer size and the duration of the congestion, a buffer overflow is likely to occur causing video packets to be dropped by the sensor.

Since VSNs operate in a multi-hop manner, congestion taking place at a certain area may diffuse to the whole network and degrade the network performance drastically. The congestion may cause a large amount of packet loss, which in turn diminish the network throughput. As a consequence, the reliable detection of events is hampered since the desired event features sensed by many nodes could not be reliably communicated to the sink. Moreover, the congestion increases the energy expenditure of the sensor nodes which in the long run also hinders the reliable reporting of the events due to energy depletion.

A candidate solution to reduce the possibility for a sensor to be congested is to decrease the load on the sensor, by reducing the amount of data to be created or to be relayed. The amount of data created by a sensor in case of event detection is assumed to be predetermined by the application. Therefore, we focus on reducing the amount of data relayed over a sensor, which is mainly determined by the routing algorithm. In order to decrease the likelihood of congestion in a sensor, the buffer occupancy levels of each sensor should be kept as low as possible. Applying load balancing approaches in the routing decisions is a possible option to reduce the buffer occupancy levels.

A centralized routing algorithm [5–7] may calculate the optimal load balanced routing decisions for a given sensor network after deployment and may disseminate the routing information to the sensors. In each topology change, the routing decision should be recalculated and disseminated to the sensors. Hence, centralized approaches are not appropriate for the distributed and the dynamic nature of wireless sensor networks. In the distributed approach [8, 9], the routing algorithm may distribute the load in any sensor evenly among its appropriate neighbors and may decrease the buffer occupancy levels throughout the network.

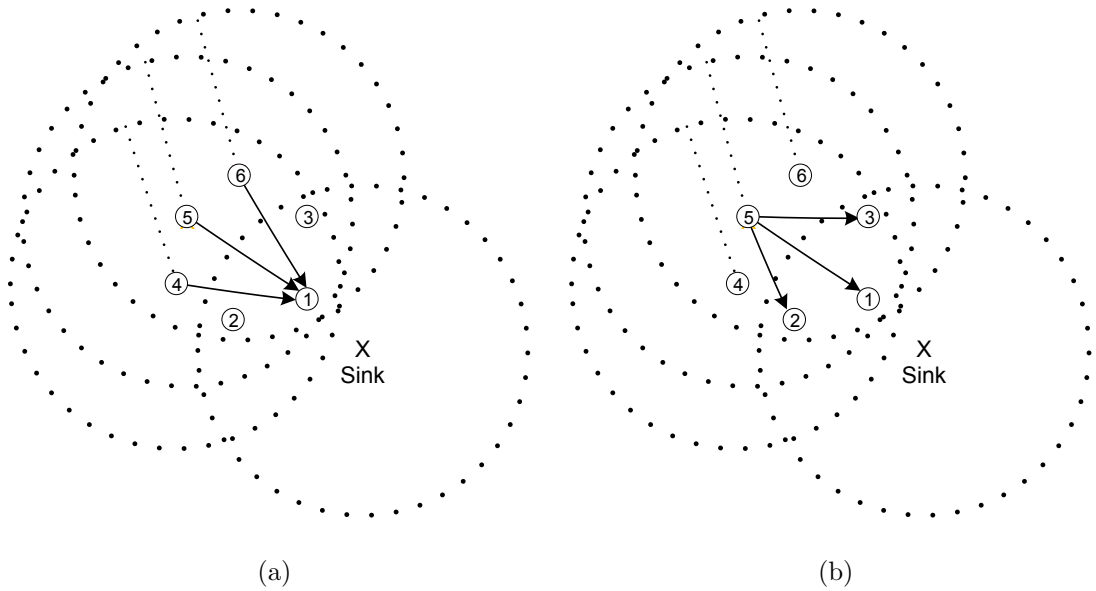
Since the location and the occurrence of an event in a sensor network are random, the load and therefore the buffer occupancy levels of the sensors in the neighborhood of

any sensor are dynamic and may not be even at the time of decision for a relay. Hence, evenly distributing the load without cross-layer assistance may result with a further degradation of the unevenness in the buffer occupancy levels in the neighborhood. Moreover, in some cases, the routing algorithm may choose a neighbor sensor with no available buffer space as a relay. In that case, the corresponding data unit is dropped at the relay sensor which causes bandwidth and energy waste. The situation is worse in terms of energy waste if the dropped data unit has traveled over many hops in the network up to this point. In order to decrease the possibility of such a routing decision and prevent from data drops in the relay sensor, the routing algorithm should use cross-layer information about the buffer occupancy levels of the sensors in the neighborhood. If the routing layer of a sensor has fresh information about the buffer occupancy levels, it can determine the relay which improves the evenness of the buffer occupancy levels in the neighborhood. However, there may be still some cases in which the buffer occupancy information about a neighbor is stale when actually there does not exist any available buffer space in that neighbor. In such a case, if the routing layer determines that neighbor as the relay, the delivery of the data results with a data drop at the relay. Hence, there should be a handshaking mechanism confirming the existence of the available buffer space in the determined relay sensor before the delivery of the data. If the existence of the available buffer space is not confirmed, the delivery of the data should be cancelled. In this thesis, we propose three cross-layer routing schemes, which integrate routing and Medium Access Control (MAC) layers in sensor networks to provide reliable event detection by means of load-balancing.

Our proposed routing schemes are based on geographic routing. Geographic routing is commonly regarded as highly scalable and energy efficient, which makes it an attractive solution for routing in wireless sensor networks [10, 11]. The most important characteristic of geographic routing is that routing decisions are based on local knowledge. Since the routing decision at each node is based on the destination's position and the position of the node's neighbors, geographic routing does not require the establishment or maintenance of routes. In addition, the nodes have neither to store routing tables nor to transmit messages to keep routing tables up to date. The problem of determining the node's position, which is beyond the scope of this thesis, is

referred to as localization. There are alternative localization techniques including global positioning system, beacon (or anchor) nodes, and proximity-based localization [12,13]. In our routing schemes, we assume that the nodes determine their locations by using one of the existing techniques.

Greedy-forwarding [14–16] is a type of geographic routing in which the relay for the data is determined in a greedy manner regarding the aim of minimizing the number of hops to reach the sink by maximizing the advancement towards the sink. The drawback of these routing strategies is that many sensors choose the same sensor as their relays and create congestion by concentrating the traffic on these preferred relays. In a sample scenario depicted in Figure 1.1a, the sender nodes (4, 5, 6) tries to route their data to the same sensor node (1), although there are two other possible relay candidates (2, 3) which can directly send their data to the sink. This generic scenario around the sink can be replicated all around the network.



(a) (b)

Figure 1.1. A comparison of routing strategies. (a) Greedy forwarding (b) LBRF.

In our first routing scheme, namely LBRF (Load Balanced Reliable Forwarding), the relay node is dynamically determined according to the current buffer occupancy levels of the neighbors that provide positive advancement towards the sink. The buffer occupancy information in the neighborhood is obtained by a piggybacking mechanism embedded in the underlying MAC protocol. Our scheme tries to perform load balancing

and avoid congestion by preferring the neighbor with the smallest buffer occupancy level in the determination of the relay. Instead of just maximizing the advancement, the primary aim of LBRF strategy is to minimize the average buffer occupancy level in the local neighborhood and establish a balance between the advancement towards the sink and the degree of congestion at the chosen relay. As shown in Figure 1.1b, the sender node 5 distributes its load to its neighbor nodes (1, 2, 3) which provides positive advancement towards the sink. We call the approach applied by LBRF as local load balancing.

In our second routing scheme, namely DLBS (Directional Load Balanced Spreading), we combine the ideas in LBRF and another geographic routing protocol, DGR (Directional Geographic Routing) [17], which is a multipath routing scheme for real time video communications in wireless sensor networks. In LBRF, each sensor tries to provide load balance locally in its neighborhood, whereas DGR aims to provide load balancing by spatially distributing the load on a sensor in predetermined directions (Figure 1.2). For this purpose, DGR divides a single video stream into multiple sub-

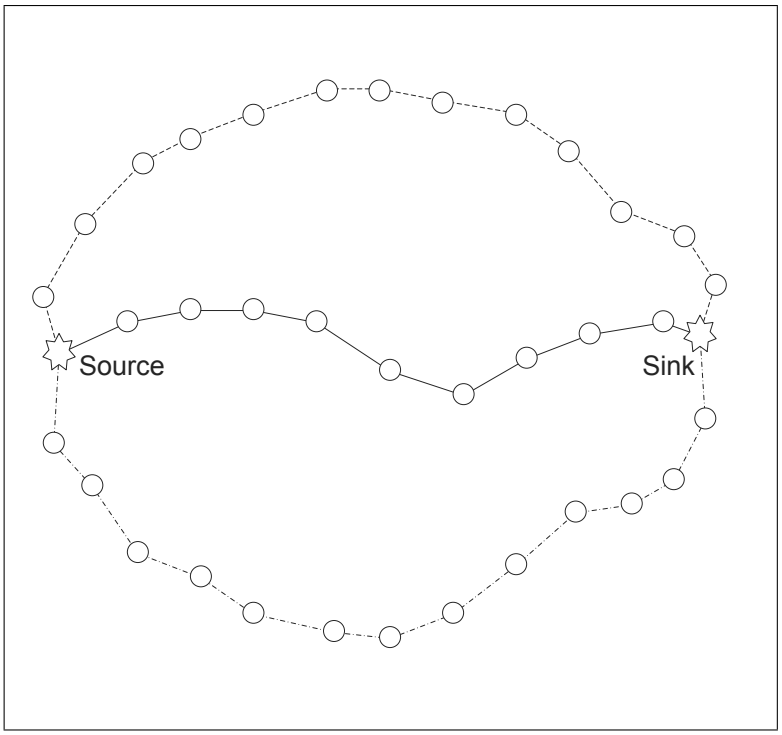


Figure 1.2. Direction-based (spatial) load balancing.

streams and transmits these sub-streams in multiple disjoint paths in parallel. These disjoint paths are constructed by using an angle-based decision scheme. In this scheme, a sensor assumes that the sink is located in the direction of the given angle, and aims to route its packets towards the virtual coordinate of the sink. The DLBS scheme aims to disseminate the data spatially adopting the DGR's angle-based directional routing scheme, and diverts portions of data flow towards different regions of the network. The load balance in each direction is attained using the LBRF scheme. By spatially distributing the load, the DLBS scheme obtains better performance in terms of packet loss and latency with respect to LBRF by utilizing extra buffers in less congested regions in the network.

Load distribution to multiple sinks (Figure 1.3) is another approach for load balancing and is a candidate solution for the congestion problem in video sensor networks (VSNs) which also provides extra benefits in terms of energy-efficiency and reliability. Extra sinks in the environment relieves the burden around any sink. In addition, the average path length, consequently, the number of hops that a frame has to travel between a sensor and a sink decreases due to shorter geographic distances. Traveling

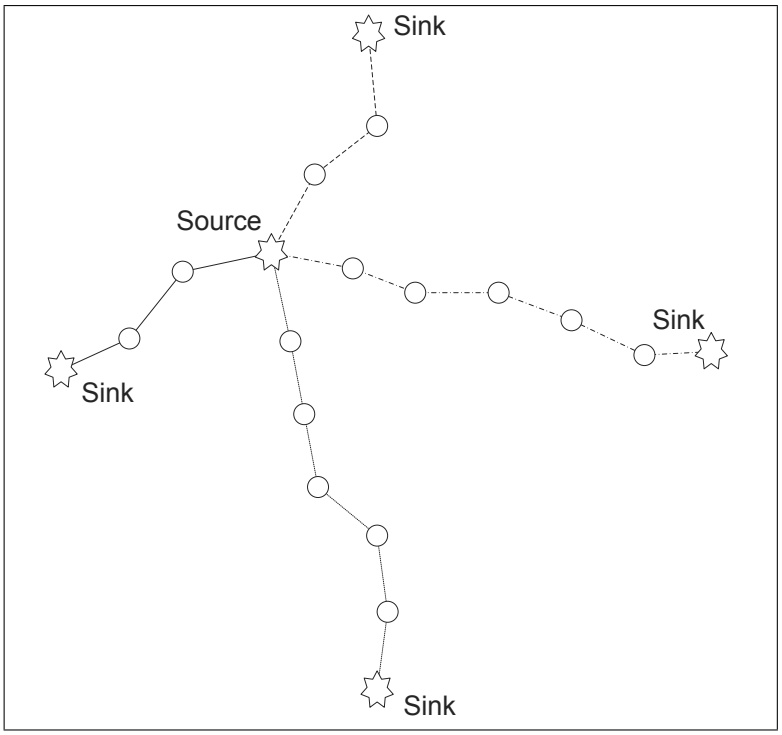


Figure 1.3. Load distibution to multiple sinks.

shorter hops results with less energy consumption on the whole network due to the degradation in the number of sensors employed as a relay node. Moreover, usage of extra sinks relieves the unbalanced energy consumption among the sensors and improves the lifetime of a deployed sensor network. A multi-sink sensor network also becomes more robust against the inaccessibility of a sink node due to single point of failures such as node failures (energy exhaustion), node destructions (fire) or communication destructions (jamming).

In our third routing scheme, namely MLBRF (Multi-Sink Load Balanced Reliable Forwarding), we combine load distribution to multiple sinks and local load balancing approach to provide reliable and energy efficient data delivery in a multi-sinked VSN for target tracking. In order to gain the maximum benefit from the deployment of multiple sinks, it is essential to distribute the load among the sinks evenly. In a case where multiple targets roam in the sensor field and the majority or all of the targets are closer to one of the sinks, the provision of an even distribution becomes a challenging problem. The focus in this thesis is the selection of the destination sink for a frame to be delivered by a sensor node with the aim of load balancing among the deployed sinks.

MLBRF proposes a sink selection mechanism based on fuzzy logic. In order to evaluate the traffic density in the direction of a sink, we use a fuzzy inference system to combine two dynamic criteria which are the number of contenders and the buffer occupancy levels in the neighborhood with a static criterion which is the distance of the candidate relay sensor to the sink. The result of the fuzzy engine is the current membership value of the sensor for a sink. Subsequent to the evaluation of the current membership values of the sensor for each sink, the destination sink for a frame is determined as the one with the greatest membership value. MLBRF uses LBRF both in the sink selection process and in the delivery of a frame to the selected sink.

The contribution of this thesis includes the proposal of three new cross layer geographic routing schemes, which address the problem of congestion in VSN. These routing schemes uses the buffer occupancy information of the candidate next hop neigh-

bors to provide load balancing. The modification in the operation of SMAC to support accurate monitoring of the buffer occupancy conditions of the neighbors is another contribution of this thesis. In addition, we also provide a sender based version of an originally receiver based dead-end handling mechanism. Local and spatial load balancing approaches, individually, are shown to be substantially more reliable than a static greedy forwarding scheme. We also show that the directional load balanced spreading technique, which is the combination of both approaches, provides a better congestion handling and a more reliable frame delivery as compared to each of these load balancing schemes. In a multi-sink network structure, by directing the video frames to the closest sink statically in a target tracking scenario, network operates with the worst performance in terms of reliable frame delivery, frame latency and energy efficiency. Our dynamic multi-criteria fuzzy selection mechanism achieves the best performance in terms of each metric by combining local indicators of the congestion level in the neighborhood towards a sink with the distance criterion.

In Chapter 2, the literature reviews on congestion management, load balancing, multi-sinked sensor networks and application of fuzzy logic for sensor networks are provided. A detailed description of LBRF routing scheme including the relay selection and dead-end handling mechanisms together with the modifications in the operation of the underlying SMAC protocol required for the functionality of the routing schemes are explained in Chapter 3. In Chapter 4, the detailed operation of DLBS routing scheme is provided. A detailed description of MLBRF strategy is given in Chapter 5 which includes descriptions for the fuzzy sink selection mechanism and the dead-end handling mechanism in multi-sink network structures. Thesis is concluded in Chapter 6 with future research directions for the presented works.

2. LITERATURE REVIEW

In this chapter, we provide brief overviews of the literature on the subjects that constitutes the basis of the thesis.

2.1. Congestion Management in WSNs

Congestion is one of the major problems affecting the application performance in WSNs and has appealed the interests of the WSN researchers. The techniques developed to solve the problem of congestion in WSNs can be divided into two groups as congestion avoidance and congestion control [18]. Congestion control techniques try to remove congestion upon its occurrence, whereas congestion avoidance techniques aim to prevent congestion from happening.

Congestion control studies in WSNs focuses on the regulation of the amount of traffic offered to the network which concurrently satisfy the requirements of the application and conforms to the limits of the network capacity preserving from the adverse effects of congestion. These studies employ congestion detection mechanisms based on either monitoring the buffer occupancy or monitoring channel activity or both. Well-known studies in this subject are ESRT [19], CODA [20] and Fusion [21].

ESRT [19] provides congestion control in the context of reliable delivery with the assumption that the sink can reach all source nodes with a high-powered one-hop broadcast reducing the overall network capacity. ESRT keeps a network operating near its optimal load by broadcasting the network-wide reporting rate to the source nodes. In case of congestion, a sensor reports the congestion state by setting a congestion notification (CN) bit in the packet header. The sink periodically computes and broadcasts a new reporting rate based on a reliability measurement, the received CN bits, and the previous reporting rate. In this approach, the reporting rate has to be set according to the most congested region in the network. In that case, the noncongested sources will be constrained by a conservative reporting rate.

CODA [20] proposes an open-loop hop-by-hop backpressure mechanism for transient congestion and a close-loop multisource regulation mechanism for persistent congestion. In the backpressure mechanism, each sensor detects congestion by monitoring the channel utilization and the buffer-occupancy level. In case of congestion, a backpressure message is sent to neighbors, which may drop packets, reduce its sending rate, and further propagate the backpressure. For multisource regulation, if the reporting rate of a sensor is above a preconfigured threshold, the reporting rates of the sensor nodes are regulated in terms of the received stream of ACKs transmitted by the sink. In contrast to ESRT, CODA only regulates the sources associated with a data event that have contributed to the congestion and uses a hop-by-hop signaling.

Fusion [21] consists of three congestion mitigation techniques. Hop-by-hop flow control resembles the backpressure mechanism of CODA. In this technique, a congestion bit is piggybacked to all transmitted packets. The upstream neighbors, which overhear the packets with the congestion bit set, stop transmitting until the congestion bit is unset. In the source rate limiting, each sensor tries to determine the number of distinct sources routing through the parent by monitoring the transit traffic and limits its own traffic to provide per-node fairness. The prioritized MAC technique gives a higher opportunity for the congested nodes in accessing to the channel.

The work in [22] also tries to provide per-source fairness in a tree routing structure using a rate limiting approach similar to the one in [21]. In this proposal, each sensor shares its effective available bandwidth equally amongst all upstream nodes which in turn ensures fairness. Interference aware fair rate control (IFRC) [23] aims to dynamically allocate fair and efficient transmission rates to each node. For this purpose, IFRC detects incipient congestion at a node by monitoring the average queue length and communicates the congestion state to the set of potential interferers using a congestion sharing mechanism by piggybacking the relevant information to each outgoing packet. IFRC converges to a fair and efficient rate using an additive increase multiplicative decrease (AIMD) control law. In addition to works mentioned above, [24] proposes a node priority based congestion control scheme for wireless sensor networks. It assumes that the nodes located in a WSN have different bandwidth requirements

for data transmission. Each node has a node priority index generated on the basis of packet inter arrival time and service time to regulate the access to the transmission media in favor of the nodes with heavy traffic.

One of the problems of rate-based congestion control in sensor networks is the difficulty to determine the right amount of rate regulation in response to congestion. Due to the dynamic nature of WSN, the bandwidth available to sensors changes all the time, which causes sensors to perform continuous rate regulation resulting with high overhead. However, it is highly likely that the traffic bursts are transient in sensor network applications. Hence, the rate regulation approach is inefficient to resolve congestions arising from these transient bursts. In the schemes which uses a tree based routing structure, many upstream sensors are restricted to route their traffic through a single downstream sensor as shown in Figure 1.1a. The traffic rate of an upstream node is throttled by the downstream node and it is determined by sharing the downstream capacity among the upstream neighbors. This approach results with an aggressive source rate reduction, which leads to substantial loss to the applications due the failure to deliver data of great importance sensed by a small number of sources.

Congestion avoidance studies in WSNs focuses on the preventive mechanisms that can be applied prior to the occurrence of a possible congestion. These mechanisms include the regulation of traffic and the diversion of the data routes. The congestion avoidance scheme proposed in [18] is based on the information about buffer occupancy information advertized in data packets. Each node announces $1/k$ of its residual buffer size as its current available buffer space as a solution to the hidden terminal problem. The upstream nodes determine the number of packets to be forwarded to a sensor according to these advertisements. In a multipath routing scheme, a list of downstream neighbors is created using these advertisements in the order of preference. If the buffer of a downstream sensor is determined to be full then a sensor chooses the next relay from the list.

In [25], a MAC protocol for sensor networks is proposed which gives proportionate access to the medium based on the total number of source nodes for which it is

forwarding data. In this protocol, downstream nodes have higher source counts and gets higher access to the medium than their upstream nodes. Weighted round robin forwarding (WRRF) is proposed which is based on the source count. Using this forwarding scheme, each node delivers its weighted-share amount of packets. In addition, an upstream node is refrained from transmissions when its downstream node does not have sufficient amount of buffer space.

In [26], a congestion avoidance mechanism is proposed which is based on the ratio of the number of downstream and upstream nodes of a sensor, which is called as the characteristic ratio (CR). In addition, the available queue sizes of the downstream nodes, which are determined by the underlying multipath routing protocol, are utilized to detect incipient congestion. Depending on the CR value, a node makes a number of decisions, including the rate control, fair queuing and buffer based fair queuing in data routing.

The majority of the studies mentioned in this section assume that packets are routed along a predetermined tree rooted at the sink node. There are exceptions which concentrate on modifying the routing rules in order to avoid [27] or mitigate congestion [28, 29]. An alternative technique to avoid congestion in WSNs is the usage of mobile sinks [30, 31].

2.2. Load Balancing in WSNs

Load balancing is an important issue in sensor networks since if too much traffic is routed through any particular node, then it will deplete its energy quicker than the other nodes, potentially partitioning the network [32]. Thus it is important to try and keep nodes alive as long as possible. A discussion on the lifetime benefits of load balancing in homogeneous sensor networks is provided in [33].

A node-centric load balancing strategy is proposed in [6], which centrally calculates the cumulative load of data traffic from child nodes in a routing tree on their parent nodes. Using these calculations, it constructs a load-balanced tree which iter-

atively grows outwards from the base station root. At each step, the algorithm first selects the branch with the lightest load, and then connects the unassigned node generating the heaviest load to this branch. Load balance is achieved by absorbing the nodes generating the greatest load to the branches with the lightest load.

In [9], a distributed load balancing scheme is proposed, which works with any geographic routing protocol. This scheme aims to prevent a node to be heavily involved in the routing decisions. Each node beacons its traffic levels and its distance to the base station to its neighbors. Traffic is calculated based upon the number of packets a node has contributed in routing during a period of time. Using the information gathered from the beacons, it decides where to forward the packet based on the distance to the base station and the level of traffic at its neighbors.

In [34] the Wireless Minimum Cost Problem is defined as a network optimization problem used for performing load balancing in wireless networks with a single type of traffic. The load balancing in this study is assumed to be the efficient distribution of the traffic so that the overall cost to the network is minimized. Necessary and sufficient conditions are developed for a flow to be optimal. A distributed algorithm is introduced for finding such flows, which can be thought of as trying to satisfy Kirchhoff's Voltage and Current Laws in a properly defined electric circuit.

In [35] an adaptive receiver-oriented load balancing strategy for geographic forwarding for WSNs is proposed. It works with a MAC protocol where sensors have asymmetric sleep-awake schedules. The region providing positive advancement towards the sink is partitioned into several zones according to the distance to the sink. When a sensor has packets to be forwarded, it broadcasts an RTS message which contains the geographic position of the sensor, the number of packets in the requested data burst and an index starting from 0. The awake sensors which receive the RTS message compute their index based on the requested data burst, their queue occupancy levels and the the expected length of their own successful burst. The sensors with the announced index responds immediately with a CTS message. If no sensor responds to RTS, the transmitting node iteratively searches for sensors with higher indexes by further RTS

messages. If multiple sensors of the same index reply, their CTS messages collide. Then a similar search is applied according to the zone starting from the nearest one to the sink. Although this strategy considers the transmission characteristics of both the sender and the receiver in load balancing, the determination of a suitable relay may require many RTS-CTS exchanges which are time and energy consuming. Another receiver-oriented load-balanced routing strategy which is based on the residual energy and the expected hop count of sensor nodes to the sink is discussed in [36].

The conventional single-path routing schemes typically based on shortest-paths are not very effective to support video transmission in VSNs due to the degradation in the perceived quality of video caused by buffer overflows and large queuing delays. To address this problem, DGR [17] is proposed as a multipath geographic routing scheme for real-time video communications in wireless sensor networks. The assumption in DGR is the existence of a single video sensor generating video streams. It aims to divide these streams into multiple sub-streams, which are transmitted over multiple disjointed paths in parallel by considering the constraints in bandwidth and energy in VSNs and delay in video delivery. The routing through multiple paths in DGR facilitate load balancing, bandwidth aggregation, and fast packet delivery. The DGR scheme itself does not concentrate on reliable data delivery in the routing layer, which is established by the use of FEC coding. A similar multipath routing protocol for multimedia sensor networks utilizing the load partitioning strategy to control the congestion is proposed in [37].

2.3. Multi-Sink WSNs

In the deployment of multiple sinks in a sensor network, the number of the sinks and their optimal positions are of primary concern in the maximization of the network lifetime. In [38], the optimal number of sinks and their optimal positions are determined using an iterative clustering technique using k-means clustering. Another solution [39] is based on integer linear programming which also concurrently deals with the routing problem in a given network. As opposed to these methods, in [40] a solution called *1 hop* is proposed which uses only local information for sink placement. In addition, *1*

hop relocation algorithm is proposed for the coordinated relocation of multiple sinks. In a recent study [41], a novel technique based on gene expression programming is proposed for the solution of optimal multiple sinks deployment problem.

Beyond the optimal positioning of multiple sinks within the network, the concentration is on the balancing of the total load among these sinks to minimize the problem of packet loss in the convergecast process in wireless sensor networks due to congestion and collisions near the sinks. In [42], a cross-layered partition-based network-load balancing protocol is proposed for efficient data dissemination in multi-sink WSNs which basically combines network wide load balancing and clustering techniques with a local metric-based routing for optimized routing tree building. In [43], a routing algorithm to balance the load among the neighbors of sink nodes (deputies) is proposed in which each packet can randomly select a deputy as its destination. To find the next hop during the routing, a forward factor is used which is the quotient of dividing its neighbors residual energy by their shortest hops to the destination. The proposal in [44] is based on Directed Diffusion [45] and it addresses the overhead data delivery problem in multi-sinked networks by forwarding data toward the nearest sink. This protocol implements a kind of load-balancing by selecting the next nearest sink after the energy level of nodes in the original path falls below a certain threshold.

A Gradient-based load balanced data forwarding is proposed in [46] which allows the nodes in forwarding path can dynamically select their next hop node according to the residual energies of the next hop candidates. The proposed method primarily uses the shortest path to route the data from a certain source to the nearest sink for saving energy, and dynamically switches to a different path to balance the load of nodes. Another gradient based routing protocol is proposed in [47], which addresses the drawback of existing protocols, which construct the gradient using the cumulative traffic load of a path, that a sensor node cannot efficiently avoid the usage of the path containing the most overloaded node. As a solution, this study introduces a new gradient-based routing protocol for load balancing which selects the least-loaded path for forwarding where the gradient of a sensor is a weighted average of the cumulative path load and the traffic load of the most overloaded node over the path. Similar to

gradient-approaches are swarm-based [48] and ant colony optimization based [49] multi-sink routing approaches. In [48], each node selects the next hop among its neighbors with the highest pheromone level stored in the routing tables. Pheromone levels are also used in [49] to transfer the data to multiple sinks in different directions. Other multi-sink routing approaches are based on three coloring, reinforcement learning and set-covering respectively [50–52].

In [53], a vector field formulation is introduced to model the communication load in a sensor network. The routing is performed in the direction of this vector field at any location of the network, the magnitude of which represents the density of the amount of data that is being transited through that location. The total communication cost in the network is defined as the integral of a quadratic form of the vector field over the network area, which leads to the solution in the form of partial differential equations (PDEs) analogous to Maxwell's equations in electrostatic theory. The vector field model is used to solve the optimal routing problem in the multiple-sink case. In [54], they extended their work to multi-commodity sensor networks and proposed a routing method based on PDEs in [53]. In [55], by pointing out the disadvantages of PDEs, authors proposed a distributed routing method in multi-sink sensor networks based on field theory which abstracts the sensor networks to a gravitational field generated by sinks. Similar to electrostatic and gravitational fields, a distributed and scalable potential field estimation algorithm and a probabilistic forwarding scheme are proposed to ensure low overhead and high resilience to network dynamics in [56]. Other multi-commodity formulations of the optimal routing problem to multiple sinks are proposed in [57–59]. In addition, in [59] a heuristic is proposed to approximate the optimal solution in a distributed setting.

In [60] and [61], a multi-sink version M-GeRaf of a geographic forwarding algorithm GeRaf [62] and a performance improvement are proposed respectively. M-GeRaf exploits a random geographic routing approach in order to achieve a reliable data delivery in networks with an aggressive power-off strategies. The reliability is achieved by exploiting the intrinsic broadcast characteristics of the wireless channel and the possibility of learning information from the neighbor transmissions. In [63], three geographic

localized anycast algorithms for multi-sink sensor networks are proposed, which consist of greedy and recovery phases to provide guaranteed delivery. One of the algorithms uses hop count as the metric, while others apply power consumption. All algorithms construct a path from the source sensor node to one of sinks/actors. During the path construction, there exists a single destination to reach. The main feature is that this destination may change along the path according to the network topology. Another variation of geographic forwarding to multiple sinks is proposed in [64].

In [65], a geographic routing scheme is proposed for providing QoS for multi-target tracking in multi-sink WSNs. The concept of event ordering is introduced, by virtue of which a priority-based buffer management scheme is applied to achieve QoS. In addition, a directional angle restricted QoS-aware routing protocol DQRP is proposed for the dissemination of the event ordering list. The buffer management scheme works in conjunction with the DQRP to ensure accurate as well as energy-efficient detection in the presence of multiple targets. Regarding target detection and tracking applications, the study in [41] aims to solve the problem of optimal multiple sink deployment.

2.4. Fuzzy Application in WSNs

In [66], a fuzzy diffusion scheme is proposed for single-sink networks which is an energy optimization on the directed diffusion scheme. Fuzzy diffusion shifts the energy cost of data forwarding to nodes having high residual energy or less data, while still achieving an energy-balance in the network. Low energy nodes with heavy traffic have reduced data forwarding burden and spend most of their power in sensing and communicating their sensor data, thus seeking to achieve network longevity. In [67], a fuzzy-based decision-making mechanism is proposed for selecting data dissemination protocols in WSNs, which aims to select the most efficient protocol considering the network performance and application-specific requirements. The mechanism relies on performing simulations and on defining and executing a two-tier fuzzy system. First, well-known WSN protocols are simulated over different scenarios and application requirements are fed to a knowledge base. Then, a set of fuzzy systems are built based on the simulation results. In [68] fuzzy logic is used to assign weights to edges using

lifetime membership and minimum energy membership values in a routing request. A fuzzy multi-objective aggregation function is used to compute the weight of an edge and the data is sent along the minimum weight path. This work is improved in [69] by adding source-to-sink delay membership value for a path. A similar approach is followed in [70] where a a group of nodes is managed by a gateway node in a cluster-based architecture. Gateway periodically invokes a fuzzy routine to determine the energy cost of a link between any two sensor nodes. Once the costs of all possible links to the gateway are computed, the route is determined by a shortest path algorithm. In [71], the routing is optimized using a fuzzy inference system considering different cost functions such as distance, remaining battery power, and link usage in selecting the next hop node among multiple candidates. A second two-layer inference system is proposed for hierarchical networks. The studies in [72] and [73] are similar to the proposals in [66] and [68] respectively.

In [74], new metrics are developed to detect congestion in each node by considering the queue lengths and channel conditions observed in the one-hop neighborhood. Based on the estimated level of congestion, each node applies dynamic rate adaptation via fuzzy control and probabilistic load balancing among the one-hop neighbors to avoid the creation of congestion and bottleneck nodes.

In [75], an algorithm for route selection is proposed for WSNs containing multiple sinks where the sensor nodes use a fuzzy inference machine in order to select the most appropriate sink. The algorithm tries to extend the network lifetime by avoiding bad routes during packet delivery by accounting the lowest energy level in the path and the number of hops.

MLBRF decides locally the most appropriate sink for data delivery by combining local dynamic congestion indicators in the neighborhood of a sensor such as the number of contenders and the buffer occupancy levels with the distances of candidate relays using a low-complexity fuzzy inference system. The aim of MLBRF is the effective distribution of the load among the sinks generated upon detection of a target. Contrary to the tree based static load balancing or pre-delivery path constructing alternatives,

MLBRF makes per-hop decisions on determining the next hop by using cross layer buffer-occupancy information to provide reliable, fast and energy efficient video delivery via load balancing.

3. LOAD BALANCED RELIABLE FORWARDING

In this chapter, we propose a new cross-layer geographic routing scheme called LBRF (Load Balanced Reliable Forwarding) which integrates network and MAC layers to provide reliable and energy efficient data delivery in WSNs. The main idea of LBRF is to consider the buffer occupancy level information of candidate relay sensors while routing a video frame with the aim of delivery to the sink. In LBRF, each sensor passively monitors and stores the buffer occupancy condition of its candidate relay sensors by a simple reporting mechanism. Sensors report their current buffer occupancy condition to their neighbors by simply piggybacking the related information to each MAC layer packet. Neighbors overhearing the packets, stores the related information for further use in a possible frame delivery. In each decision of a relay selection, sensors try to provide dynamic load balancing among their candidate relays by evaluating their stored buffer occupancy information. In this way, buffer occupancy levels of the sensors are kept in a lower level and more available buffer space is left for video frames which will be created upon detection of an unpredictable event.

The primary objective of LBRF is the reliable transportation of the generated video frames in the network which are important for the performance of the application that the video sensors are deployed for. However, energy expenditure and the latency are also taken into consideration in the design of LBRF. For improving the latency, LBRF tries to maximize the advancement towards the sink by selecting the candidate relay among the neighbors providing the best contribution to the load balance in the neighborhood. For improving the energy expenditure, LBRF utilizes SMAC [76] as the underlying MAC protocol, which primarily aims to reduce the energy consumption in wireless sensor networks . For further improving reliable delivery and energy expenditure, LBRF applies a cancellation mechanism which is a kind of handshake embedded in SMAC. In a sensor, stored buffer occupancy information related to a neighbor may be stale, and that neighbor, which actually has no available buffer space, may be selected as the load balancing sensor in the decision of a frame delivery. When the sender node issues an RTS to that neighbor and receives the corresponding CTS, it infers from

the piggybacked buffer occupancy information that the relay has no buffer space and it cancels the delivery of the frame. In addition to the cancellation mechanism, a sensor that has no candidate relay with available buffer space is refrained from delivering its packets which relieves the contention of the neighbors in accessing the medium.

In LBRF, the important factors determining the quality of the relay selection decision in terms of load balancing, is the time of the decision and the accuracy of the stored buffer occupancy information belonging to the neighbors. When the relay for a frame is decided in the routing layer, the conditions in the neighborhood may have totally changed at the time of delivery due to possible waiting in the buffers. Hence, LBRF decides the relay for a frame in the MAC layer just before the contention for the medium to enhance the quality of the decision. On the other hand, SMAC creates some difficulties which negatively affects the accuracy of the monitoring the buffer occupancy conditions of the neighbors. In the design of SMAC, overhearing is identified as one of the major sources of energy waste, which is the reception of packets that are destined to other nodes. SMAC utilizes an overhearing avoidance mechanism to reduce the energy waste. However, the accuracy of the reporting mechanism of LBRF decreases because of the overhearing avoidance mechanism which prevents neighbors to acquire some of the packets containing the current buffer occupancy conditions of the senders and to update their records with the fresh information. Hence, we modified the operation of SMAC, especially the overhearing avoidance mechanism, in order to enable the neighbor nodes to acquire the required packets and to increase the accuracy of the monitoring. In Section 3.3, we will give a detailed explanation of the modified SMAC together with a summary of the original SMAC.

As a geographic routing scheme, LBRF uses the location information of the sensors in relay selection. LBRF assumes that each sensor knows its location and acquires its neighbors' positions through a neighbor discovery process. It also assumes that the sensors possess the location information of the sink. The location information of the neighbors is stored in a table called Neighbor Information Table (NIT). This table also stores the buffer occupancy level information of neighbor nodes. In order to report the current buffer occupancy condition, nodes piggyback their current buffer occupancy

information to each SMAC packet. On hearing these packets, each neighboring sensor updates the corresponding entry in its NIT with the fresh information, even if the received packets are not destined to it. In this way, each node passively monitors the buffer occupancy levels of its neighbors. In addition, the reception time of the last received packet from a neighbor is also recorded and updated in NIT following the receptions. If for a sufficiently long time no packet is received from a neighbor, than that node is accounted to be unreachable and it is not considered as a candidate in the further relay selection processes.

3.1. Relay Selection in LBRF

The routing strategy works by classifying the neighbors of a node into two distinct groups with respect to their distances to the sink. The forward region of a node is the area in the transmission region of a node which provides positive advancement towards the sink. The remaining area is called as the backward region. If the distance of a node to the sink node is d, the distance of a neighbor node to the sink node is d_n and the transmission radius of a node is r_t; then the forward region of a node contains the neighbors satisfying $0 < (d - d_n) \leq r_t$ and the backward region contains the neighbors satisfying $-r_t < (d - d_n) \leq 0$ (Figure 3.1a).

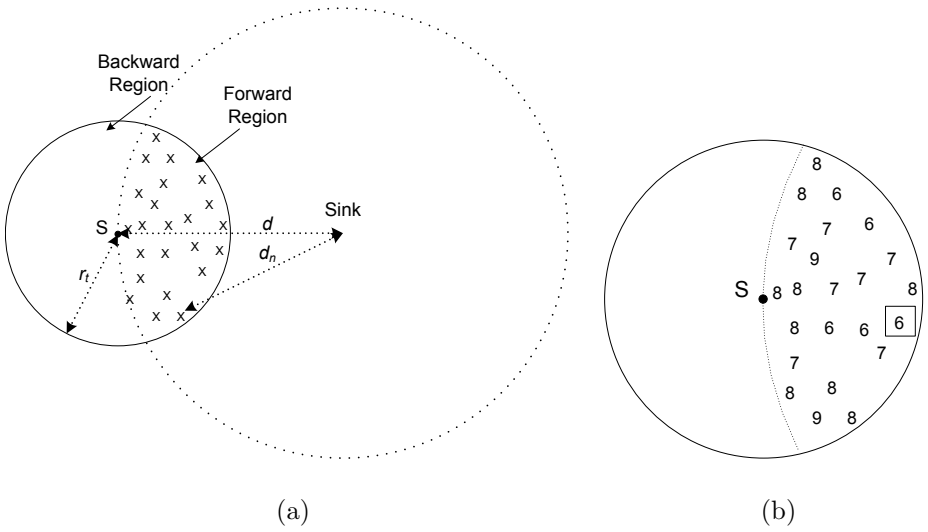


Figure 3.1. Neighbor classification and a sample relay selection in LBRF.

```
1: procedure SearchForCandidateRelay(N,NIT,class,region,relay)
2:     N_c ← ∅
3:     for all n ∈ N do
4:         if (NIT(n).class = class ∧ NIT(n).region = region) then
5:             N_c ← {n} ∪ N_c
6:         end if
7:     end for
8:     boc_min ← ∅
9:     boc_min ← argmin_{n∈N_c} (NIT(n).boc)
10:    relay ← argmin_{n∈ boc_min} (NIT(n).disttosink)
11:    return relay
12: end procedure
13: procedure DetermineRelay(selfclass,relay)
14:    relay ← 0
15:    if (selfclass = 1 ∨ selfclass = 3) then
16:        region ← FORWARD
17:    else if (selfclass = 2 ∨ selfclass = 4) then
18:        region ← BACKWARD
19:    end if
20:    if (selfclass = 1) then
21:        SearchForCandidateRelay(N, NIT, 1, region, relay)
22:    else
23:        SearchForCandidateRelay(N, NIT, selfclass − 1, region, relay)
24:        if (relay = 0) then
25:            SearchForCandidateRelay(N, NIT, selfclass, region, relay)
26:        end if
27:    end if
28:    return relay
29: end procedure
```

Figure 3.2. Pseudo-code for the relay selection mechanism of LBRF.

If a node does not have any neighbor in the forward region, it is called as a dead-end. The handling of the dead-end problem will be explained in the following subsection. In case of a frame delivery, a proper relay node is searched in the first group of neighbors. The relay node for a frame is chosen with respect to the buffer occupancy levels of the nodes and the distances of the nodes to the sink node. The candidate nodes are primarily sorted according to their buffer occupancy levels from the smallest to the largest. A sensor is excluded from this group if its buffer occupancy level is not enough to store the frame to be delivered. The relay node is chosen as the one with the minimum distance to the sink among the neighbors with the lowest buffer occupancy level at the time of delivery (Figure 3.1b). In this way, while minimizing the average buffer occupancy and providing load balancing in the neighborhood, LBRF tries to maximize the advancement towards the sink. A pseudo-code of relay selection in LBRF is presented in Figure 3.2. Routing classes are introduced to provide a solution for the problem of routing around dead ends, which will be discussed in the following section.

3.2. Routing Around Dead-Ends in LBRF

Geographic routing strategies should deal with possible dead-ends caused by connectivity holes (Figure 3.3). There are some solutions [15, 77, 78] which are effective in bypassing dead ends. For example, GPSR [15] deals with dead-ends by means of making the network topology graph planar and by using "face routing". In [79], an efficient non-planar routing technique around dead-ends in sparse topologies is proposed. The authors also proved that their technique finds loop-free routes. In our routing scheme, we provide a sender-based adaptation of this receiver-based technique to deal with possible dead-ends.

Let us call the forward region of a sensor's coverage area as F and, and the backward region of a sensor's coverage area as B. In addition, let $C_1, \ldots, C_n$ be a set of n routing-classes that sensors assign themselves according to their routing abilities. The routing class of a node determines the region (F or B) to search candidate relays, and the routing classes that these candidate relays should belong to. The relay is determined

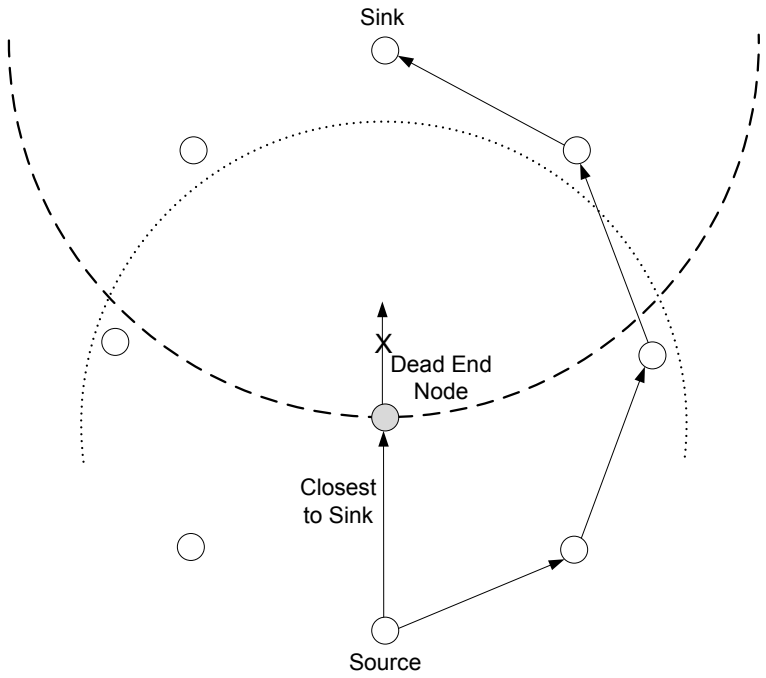


Figure 3.3. Illustration of the dead-end problem.

among the set of candidate relays using the buffer occupancy and distance based relay selection mechanism as described in Figure 3.2. The routing-class information of the neighbors of a sensor is acquired by a simple reporting mechanism as before and stored in NIT. In order to report the current routing-class, nodes piggyback routing-class information to each MAC layer packet. On hearing the packet, each neighboring sensor updates the related routing class information with the fresh one, even if the received packet is not destined to it. In this way each node has information about the routing-classes of its neighbors.

The routing class of a node is determined based on the gathered routing class information. In case of any change in the routing class information from a neighbor, a node checks and updates its routing class accordingly (Figure 3.4). In that way, these changes iteratively diffuse in the network over time and the routing classes in the network reach to a steady state. The routing class of a node is updated either in the initialization phase or in case of a topological change in the network such as energy depletion, sensor destruction or a long-term communication disruption. If for a sufficiently long time no packet is received from a neighbor, than the routing class

```
1: procedure UpdateSelfRoutingClass(N,NIT,selfclass)
2:     selfclass ← 1
3:     for all n ∈ N do
4:         if (NIT(n).class = 1 ∧ NIT(n).region = FORWARD) then
5:             return selfclass
6:         end if
7:     end for
8:     selfclass ← 2
9:     for all n ∈ N do
10:        if ((NIT(n).class = 1 ∨ 2) ∧ NIT(n).region = BACKWARD) then
11:            return selfclass
12:        end if
13:    end for
14:    selfclass ← 3
15:    for all n ∈ N do
16:        if ((NIT(n).class = 2 ∨ 3) ∧ NIT(n).region = FORWARD) then
17:            return selfclass
18:        end if
19:    end for
20:    selfclass ← 4
21:    return selfclass
22: end procedure
23: procedure ProcessPacketFromNeighbor(packet)
24:    NIT(packet.nodeid).boc ← packet.boc
25:    if (NIT(packet.nodeid).class ≠ packet.class) then
26:        NIT(packet.nodeid).class ← packet.class
27:        UpdateSelfRoutingClass(N, NIT, selfclass)
28:    end if
29: end procedure
```

Figure 3.4. Pseudo-code for the routing class updates upon packet reception.

of that node is stored as C_0 in NIT. The neighbors of class C_0 are assumed to be dead and are not involved in any routing decision. A node determines its neighbors that belong to C_0 at the time of each frame delivery by controlling the last packet reception times in the NIT and reevaluates its routing class.

The initialization phase is started after deployment. At the beginning, all nodes that have neighbors in F assign themselves to class C_1 and all nodes that do not have neighbors in F assign themselves to class C_2 (Figure 3.5). The sink node also assigns itself to class C_1. A C_1 node has a route to the sink involving only C_1 nodes, and all nodes in other classes try to find a route to a C_1 node. All nodes of class C_1 search for candidate relays among the C_1 nodes in F. If a C_1 node determines that there remains no C_1 neighbors in F, it updates its routing class as C_2. The nodes of class C_2 handle the packets that they generate or receive according to a different rule: the packet is sent away from the sink selecting a node located in B, preferably of class C_1 or of class C_2 (if a C_1 node does not exist in B).

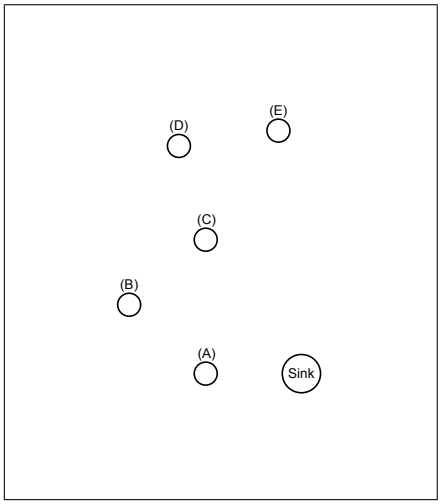
```
1: procedure InitializeClassInformation(N,NIT,selfdisttosink,selfclass)
2:     selfclass ← 2
3:     for all n ∈ N do
4:         distdiff ← (selfdisttosink − NIT(n).disttosink)
5:         if (distdiff > 0) then
6:             NIT(n).region ← FORWARD
7:             selfclass ← 1
8:         else
9:             NIT(n).region ← BACKWARD
10:        end if
11:        NIT(n).class ← 1
12:    end for
13:    return selfclass
14: end procedure
```

Figure 3.5. Pseudo-code for the initialization of routing classes.

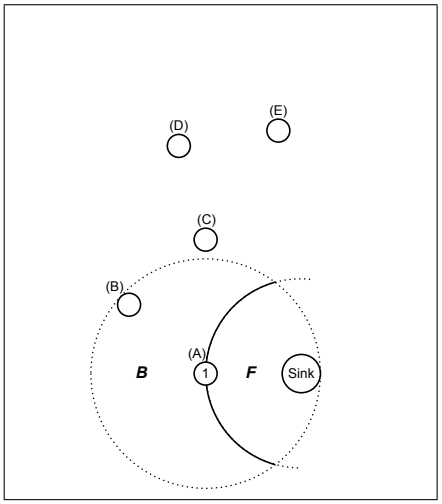
Now, if a C_2 node determines that there remains no neighbor in B that is of class C_1 or of C_2, than it updates its routing class as C_3. In this case, nodes of class C_3 do not have C_1 neighbors but may have a C_2 neighbor which is located in F. The nodes of class C_3 try to find a route for their packets by selecting a node located in F, preferably of class C_2 or of class C_3 (if a C_2 node does not exist in F). If even a node of class C_3 determines that there remains no neighbor in F that is of class C_2 or C_3, than it updates its routing class as C_4. Similar to the nodes of class C_2, the nodes of class C_4 try to route their packets by selecting a node located in B, preferably of class C_3 or of class C_4 (if a C_3 node does not exist in B).

This mechanism may be generalized for an arbitrary number of routing classes n. Any node of class C_k, $k \leq n$ always searches a candidate relay preferably of class C_{k-1} or of class C_k which is located in B for k even, and in F for k odd, with the exception of C_1 nodes, which always look for other nodes of class C_1 which are located in F. As a general rule, a greater number of routing classes allows more nodes to find a route toward the sink. However, it is shown in [78] that four routing classes are sufficient to handle most situations.

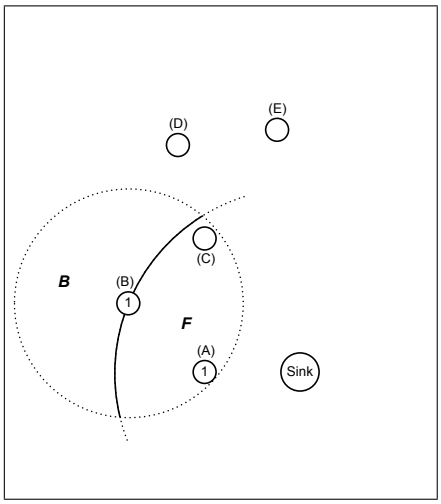
The routing classes in a sample scenario with five sensors is illustrated in Figure 3.6. In this scenario, the node A determines its class as C_1 since the sink is located in its forward region (Figure 3.6b). The node B also determines its class as C_1 since node A, which also belongs to class C_1, is located in its forward region (Figure 3.6c). The node C has no neighbors of class C_1 in its forward region. It determines its routing class as C_2, since node B, which is a C_1 node, resides in its backward region (Figure 3.6d). The node D has no C_1 neighbors in its forward region and has no C_1 or C_2 neighbors in its backward region. It determines its routing class as C_3, since node C, which is a C_2 node, resides in its forward region (Figure 3.6e). Similarly, the node E has no C_1 or C_2 neighbors in its forward and backward regions. It determines its routing class as C_4, since node D, which is a C_3 node, resides in its backward region (Figure 3.6f).



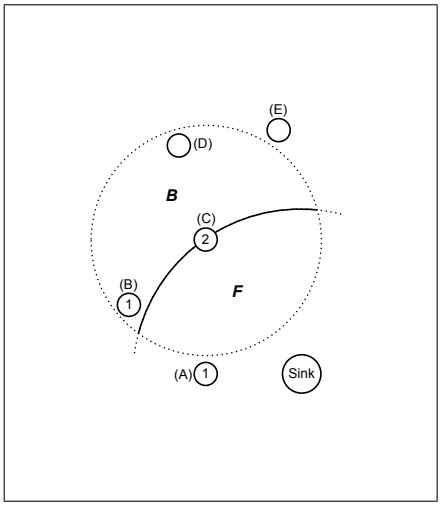
(a) A sample network topology



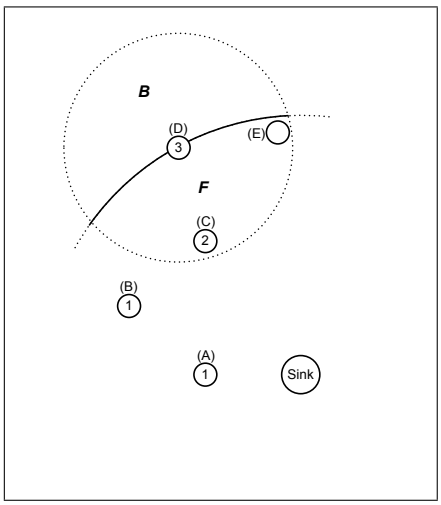
(b) Node A belongs to class C_1



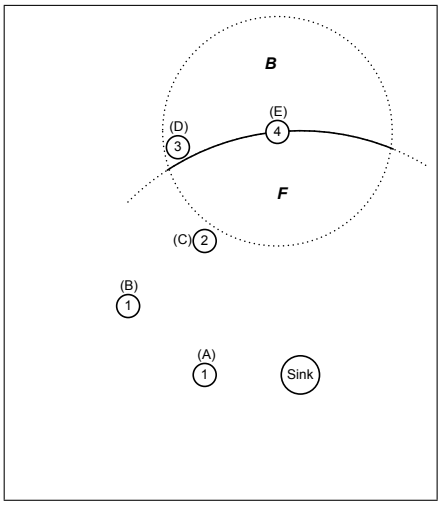
(c) Node B belongs to class C_1



(d) Node C belongs to class C_2



(e) Node D belongs to class C_3



(f) Node E belongs to class C_4

Figure 3.6. Illustration of the routing classes of LBRF in a sample scenario.

3.3. Modification of SMAC

The functionality of LBRF depends on the accurate monitoring of the buffer occupancy conditions of the sensors in the neighborhood. In order to report the current buffer occupancy condition of a sensor via SMAC packets, the structure of SMAC packets (RTS, CTS, DATA, ACK, SYNC) is modified to contain the BOC (Buffer Occupancy Condition) entry. At the time of each delivery, the current buffer occupancy condition is embedded by SMAC in the BOC entry of the packet. The sensors that receive any of these packets from their neighbors update their NIT with the BOC information extracted from these packets.

In addition, the operations of SMAC in the sender and the receiver of a data delivery are also modified. Upon reception of any packet destined to it, a node updates the BOC information extracted from the packet in NIT. If the destination of an RTS packet has available buffer space to receive the amount of data indicated in the RTS packet, the normal SMAC operation is performed. Otherwise, the node receiving the RTS packet sends the CTS packet and switches to the SLEEP mode for the current cycle. Such CTS message is referred as NCTS (Negative CTS) indicating that the frame delivery should be cancelled. The destination of the CTS message infers that the current buffer space of the source of CTS message is not adequate to store the data requested to be delivered. Accordingly, it cancels the delivery of data and switches to the SLEEP mode. This type of RTS-CTS exchange is called as an unsuccessful RTS-CTS exchange. The sender of RTS message retries delivery in the subsequent cycle by starting the RTS-CTS exchange with a new relay for the corresponding data. The operation of SMAC in response to an unsuccessful RTS-CTS exchange is called as the cancellation mechanism. Instead of cancelling the transmission, either the sender or the receiver can retry transmission in the current cycle. However, such modifications increase the complexity of the original SMAC transmission scheme which allows only one contention per cycle, but provide a slight increase in the efficiency where the observation of NCTS messages is infrequent.

As we previously stated, the overhearing avoidance mechanism of SMAC prevents

neighbors to receive packets from ongoing communication activity that they are not involved in and to update their NIT with the fresh information. Since it prevents the neighbors to receive the NCTS message, they also cannot infer that an RTS-CTS exchange is unsuccessful, which causes these neighbors to sleep till the expected end of communication as a side effect of the virtual carrier sensing mechanism of SMAC. The problem is crucial especially when the message passing feature of SMAC is enabled. In that case, the neighbor nodes which have not realized an unsuccessful RTS-CTS exchange, record that the medium will be busy till all the fragments of a message is delivered. In order to overcome these problems, the overhearing avoidance mechanism should be relaxed and the virtual carrier sensing mechanism should be modified. In addition, when the message passing feature enabled, SMAC should be modified for the overhearing nodes to estimate the final buffer occupancy conditions of the sender and the receiver of a message without overhearing all the fragments.

The knowledge about the buffer occupancy information of some neighbors may still be stale in cases where the node cannot overhear some packets from those neighbors or where new packets are created at those neighbors after their last successful transmission. The fresh BOC information synchronizes the NIT with the buffer occupancy condition of the source of the received packet.

In the following subsections, we will summarize the original SMAC [76] and then describe the modifications in overhearing avoidance and virtual carrier sensing mechanisms of SMAC together with the estimation mechanism of overhearing nodes when the message passing feature is enabled.

3.3.1. Original SMAC

SMAC protocol is a medium access control (MAC) protocol for wireless sensor networks that enables low-duty-cycle operation on nodes in a multi-hop network by periodically putting nodes into sleep state. The basic scheme is that each node sleeps for some time, and then wakes up and transmits its buffered packets or listens to see if any other node wants to communicate with it. Nodes exchange their schedules

by periodically broadcasting a SYNC packet to their immediate neighbors and form virtual clusters.

SMAC adopts virtual and physical carrier sense, and the RTS/CTS exchange for the hidden terminal problem. The duration of the remaining transmission is indicated in each transmitted packet. If a node receives a packet destined to another node, it determines the duration to keep silent from this field. The node records this value in a variable called the network allocation vector (NAV) and sets a timer for it. Before initiating a transmission, a node first checks its NAV. If its value is not zero, the node determines that the medium is busy. This is called virtual carrier sense. Physical carrier sense is performed by listening to the channel for possible transmissions, by selecting a random contention slot within a contention window to avoid collisions and starvations. The medium is determined as free if both virtual and physical carrier sense indicates that it is free. If a node fails to access the medium, it goes to sleep and wakes up when the receiver is free and listens again. Broadcast packets are sent without using RTS/CTS. Unicast packets follow the sequence of RTS/CTS/DATA/ACK between the sender and the receiver. After the successful exchange of RTS and CTS, the two nodes will use their normal sleep time for data packet transmission. They do not follow their sleep schedules until they finish the transmission.

Overhearing is the reception of packets that are destined to other nodes. SMAC tries to avoid overhearing by letting interfering nodes go to sleep after they hear an RTS or CTS packet. Since DATA packets are normally much longer than the control packets, the approach prevents neighboring nodes from overhearing long DATA packets and following ACKs.

SMAC applies message passing to reduce application-perceived latency and control overhead. A message is the collection of meaningful, interrelated units of data. SMAC fragments the long message into many small fragments, and transmit them in a burst to decrease the penalty of accumulated contention latency for individual fragments. Only one RTS packet and one CTS packet are used. They reserve the medium for transmitting all the fragments. Every time a data fragment is transmitted, the

sender waits for an ACK from the receiver. If it fails to receive the ACK, it will extend the reserved transmission time for one more fragment, and re-transmit the current fragment immediately.

3.3.2. Modification of SMAC Operation in Overhearing Nodes

A node which overhears a packet from an ongoing communication cannot infer the success of the transmission because of the overhearing avoidance mechanism of the original SMAC protocol. However, the accuracy of the monitored buffer occupancy information of the neighbors in a node depends on the inference that a frame transmission around the node will be successful or not. The aim of the modifications applied to the original SMAC is to make sure that a node will estimate and update the end-of-transmission buffer occupancy conditions of the sender and the receiver nodes using the current BOC information only when the RTS-CTS exchange is realized to be successful. For this purpose, the overhearing avoidance mechanism of SMAC is relaxed not to put the overhearing nodes into SLEEP mode till they infer that the RTS-CTS exchange is successful.

The listen period of the SMAC protocol is modified as illustrated in the flowchart given in Figure 3.7: As the idle period starts, the sensor starts to listen to the medium for possible incoming or ongoing transmissions till the SLEEP interrupt is received. In case of any packet reception, the source, the destination, the type of the packet, the duration of the whole transmission and the buffer occupancy information is gathered from the packet. Either destined to itself or not, upon reception of any type of packet, a node updates the current buffer occupancy information of the source of the packet. In addition, it calculates the number of data packets to be transmitted / received by using the duration field of a packet. If the received packet is destined to itself, the node waits for possible extra packets depending on the type of the received packet and the duration of the transmission.

If the packet is overheard from an ongoing communication, then the following actions are taken depending on the type of the received packet:

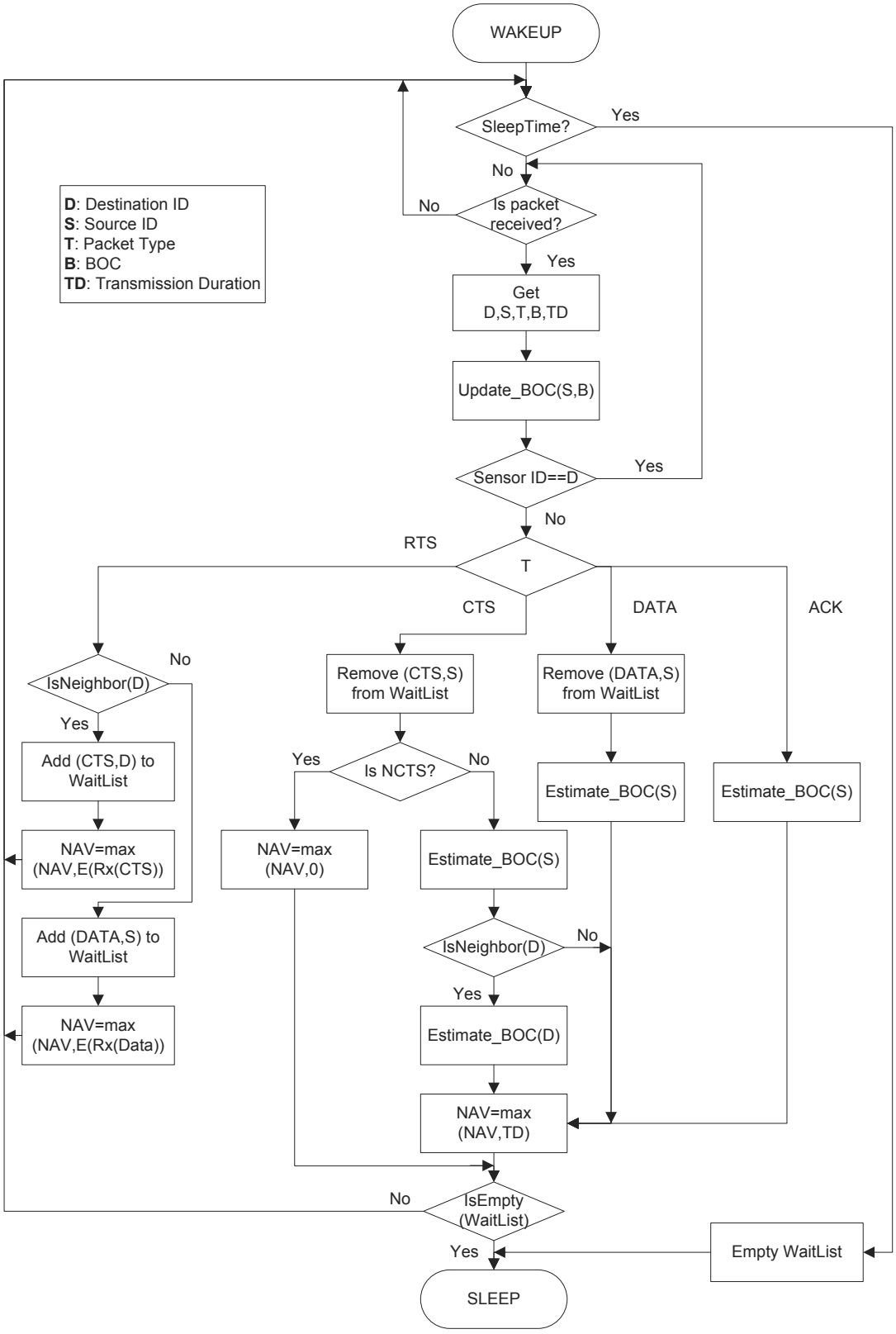


Figure 3.7. Operation of modified SMAC in overhearing nodes.

If a node overhears an RTS packet from a neighbor, it creates an entry in the WaitList containing the type of the next packet expected to be received in this communication and the address of the expected source. This list is created in order to identify multiple communications that does not interfere with each other around that node. WaitList is necessary to update the buffer occupancy information for the receiver of the transmission, once the frame transmission takes place in that communication. For this purpose, it checks the destination of the packet. If the destination is a neighbor of the node, it is possible for the node to overhear the CTS message. An entry in the WaitList is created regarding that CTS will be received from the destination. The NAV value is set as the expected time to receive a CTS packet.

If the destination of the RTS packet is not a neighbor of the overhearing node, then the destination node is hidden from the overhearing node. Hence, an entry regarding that a DATA packet will be received from the source node is created in the WaitList to understand that frame transmission takes place. The NAV value is set as the expected time to receive a DATA packet.

If a node overhears a CTS packet from a neighbor, it checks the WaitList for the relevant entry regarding the received packet, and removes the entry, if exists. Next, it inspects the duration field of it in order to determine whether the node has available buffer space for the data packets requested to be sent by the source of the RTS packet. The node overhearing a CTS packet updates the buffer occupancy information of the source of the CTS as it has received the packets. If the destination of the RTS packet is also a neighbor of the node, the node updates the buffer occupancy information of the source of the RTS as it has transmitted the packets. The node also sets NAV as the time required to finalize the transmission of the packets. If the overheard packet is NCTS, the buffer occupancy information of the sources of the RTS and the CTS packets is not updated and the NAV value is not altered.

If a node overhears a DATA packet from a neighbor, it checks the WaitList for the relevant entry regarding the received packet, and removes the entry, if exists. Next, the node updates the buffer occupancy information of the source of the DATA as it

has transmitted the packets. The node also sets NAV as the time required to finalize the transmission of the packets.

If a node overhears an ACK packet, the node updates the buffer occupancy information of the source of the ACK as it has received the packets. The node also sets NAV as the time required to finalize the transmission of the packets.

Subsequently, in each case, the node goes into SLEEP mode either when the WaitList is empty or when the SLEEP interrupt is received due to its normal sleep schedule regardless of the contents of the WaitList. The contents of the WaitList are removed before the node goes into SLEEP mode.

3.3.3. BOC Estimation Mechanism

When an overhearing node infers that an RTS-CTS exchange of its neighbors is successful, the BOC of the neighbors involved in communication should be estimated and the relevant entries in NIT should be updated prior to setting the NAV value to the end of transmission and putting the node into SLEEP mode. The estimation mechanism uses the information stored in the duration field in the overheard packet that indicates the duration of the remaining transmission. The operation of the estimation mechanism is given in Figure 3.8.

The accuracy of the buffer occupancy information of the neighbors monitored by a sensor is presented in Table 3.1. The simulation scenario and simulation parameters are provided in the performance evaluation section. The presented results are obtained when the sensors use the LBRF scheme. At the time of frame delivery, each sensor compares the BOC information of the source of the CTS message in the NIT with the actual value of the BOC extracted from the received CTS message. The related statistics are collected in the overall network. The accuracy results in Table 3.1 represent the percentage of the number of equalities observed in all BOC comparisons. As the frame rate of the application increases, we observe a decrease in the accuracy of the monitored buffer occupancy information. Since the frame creation rate is high,

1: **procedure** ESTIMATEBOC(NIT,$packet$)
2: $currentboc \leftarrow NIT(packet.nodeid).boc$
3: **if** $(packet.type = RTS)$ **then**
4: $estimatedboc \leftarrow currentboc - \left(\frac{(packet.td)c - \text{SIZE}(RTS)}{\text{SIZE}(DATA)+\text{SIZE}(ACK)}\right)$
5: **else if** $(packet.type = CTS \vee packet.type = ACK)$ **then**
6: $estimatedboc \leftarrow currentboc + \left(\frac{(packet.td)c}{\text{SIZE}(DATA)+\text{SIZE}(ACK)}\right)$
7: **else if** $(packet.type = DATA)$ **then**
8: $estimatedboc \leftarrow currentboc - \left(\frac{(packet.td)c - \text{SIZE}(ACK)}{\text{SIZE}(DATA)+\text{SIZE}(ACK)}\right)$
9: **end if**
10: $NIT(packet.nodeid).boc \leftarrow estimatedboc$
11: **end procedure**

c: Channel capacity in $bits/sec$

td: Remaining transmission duration after reception in sec

SIZE($\cdot$): Size of a packet in $bits$

Figure 3.8. Pseudo-code for the BOC estimation mechanism.

the buffer occupancy condition of neighbors rapidly changes in case of event detection. This change results with an inconsistency between the stored and the actual buffer occupancy information of the destination sensor node, since the packet creation rate is higher than the update rate of the buffer occupancy information performed upon overhearing any packet from the destination sensor. This phenomenon causes the accuracy results to be less than 100% for all frame rates in the simulations. However, the accuracy rates are observed to be at least almost 95%.

Table 3.1. Accuracy of the monitored buffer occupancy information.

Frame Rate	**Accuracy**
3 fps	0.970
4 fps	0.954
5 fps	0.950
6 fps	0.949

3.3.4. Discussion on the Cost of the Modifications

The aim of the modifications in SMAC operation is to relax the overhearing avoidance mechanism not to put overhearing nodes into SLEEP mode till they infer that RTS-CTS exchange is successful. In other words, the awake time of a node in the listen period is extended in the cases when there occur communication in the neighborhood of the node. Hence, actually, the modifications do not put extra burden on a node in terms of computation and memory usage, but it increases the energy expenditure of a node due to longer awake times and overhearing some extra packets.

Different than the original SMAC, the modified version operates on a table called NIT, which stores location information, distance to sink information, buffer occupancy level information, the routing class and region information of the neighbors. In addition, the reception time of the last received packet from a neighbor is also stored in NIT. The space complexity of this table is $O(N^2)$ for small N and $O(N)$ for large N, where N is the number of neighbors of a node. In the modified version, buffer occupancy level, routing class and last reception time entries are updated upon reception of a packet from a neighbor. The computational complexity of the update operation is at most $O(N)$, which is composed of a search in the table and update fields operations. If the routing class of a neighbor extracted from a packet is different than the stored class in NIT, the node recalculates its self routing class. The complexity of this operation is also $O(N)$, which may require one or more search operations on NIT (Figure 3.4).

The modified version perform some operations on a list called WaitList, depending on the type of the packet received from a neighbor. The length of this list is at most N. Hence the complexity of adding an element to the list or removing an element from the list operations is $O(N)$. In some cases, the destination of the packet is checked for inclusion in the neighbor list which requires a search operation in NIT of complexity $O(N)$. Estimation of buffer occupancy conditions of the neighbors is an $O(1)$ time operation. Hence, the complexity of the operations branched depending on the packet type (Figure 3.7) is $O(N)$, concluding the overall complexity of the modifications to be $O(N)$.

We explain energy cost of the modifications with an illustrative scenario presented in Figure 3.9. In that scenario, where r_t is the transmission range, the node A want to transmit a video frame to node B. R_1 is the region where the nodes can hear the packets from only node A, and R_3 is the region where the nodes can hear the packets from only node B. The nodes in R_2 region can hear the packets transmitted by A and B.

In the original SMAC, when the node A transmits the RTS packet to node B, the nodes in region R_1 and R_2 go to SLEEP mode upon overhearing of the RTS packet. The nodes in region R_3 go to SLEEP mode when they overhear the CTS packet transmitted by node B as a reply to the RTS packet. In the modified version of SMAC, when the node A transmits the RTS packet to node B, the nodes in region R_1 do not go to SLEEP mode until they overhear the first DATA packet transmitted by node A. Hence, extra energy consumption of the nodes in R_1 is due to idle listening during the time between the overhearing of RTS and DATA packets, and reception of a DATA packet. The nodes in region R_2, also, do not go to SLEEP mode until they overhear the CTS packet transmitted by node B. Hence, extra energy consumption of the nodes in R_2 is due to the idle listening during the time between the overhearing of RTS and CTS packets, and the reception of a CTS packet. There is no extra energy consumption of the nodes in R_3 since in each version they go to SLEEP mode as soon as they overhear a CTS packet from node B, since it contains adequate information for them to infer the success of the transmission.

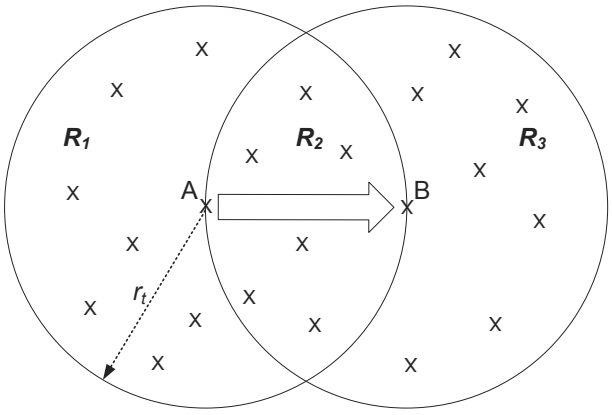


Figure 3.9. A scenario illustrating the overhearing regions during a transmission.

Although, we do not provide a separate analysis for the energy cost of the modifications, the energy consumption comparison of the routing algorithms that use original and modified SMAC is presented in Section 3.5.3. In order to ease the understanding of the energy expenditure results, we will define the energy expenditure properties of SMAC in the following section.

3.3.5. Energy Consumption Properties of SMAC

The basic scheme of SMAC [76] described in Section 3.3.1 is illustared in Figure 3.10. Each node sleeps for a duration of T_{sleep} and then wakes up to see if any other node wants to talk to it. During the SLEEP period, the node turns off its radio, and the energy consumption parameter of the node in that period is the SLEEP power. The complete cycle of awake and sleep is called a frame. The awake interval is composed of synchronization and listen periods. Each period has a contention window with many time slots for senders to perform carrier sense. For example, if a sender wants to send a packet (SYNC or DATA), it starts carrier sense when the receiver begins listening. It randomly selects a time slot to finish its carrier sense. If it has not detected any transmission by the end of that time slot, it wins the contention and starts sending its packet. During carrier sensing the energy consumption parameter of SMAC is IDLE power, whereas the related parameters in data transmission and data reception are TX power and RX power. At the end of the data transmission, the communicating nodes sleep until the beginning of the subsequent SMAC frame.

Energy consumption of a node that uses SMAC as the underlying MAC protocol is highly related with the durations of the synchronization, listen and sleep periods. Typically, the duration of synchronization (T_{sync}) and listen periods (T_{listen}) are determined as the time required for the contention window in that period plus the time required to send a SYNC packet or the time required to send RTS and CTS packets,

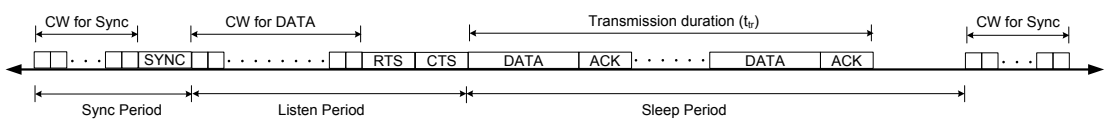


Figure 3.10. Frame structure of SMAC.

respectively. The duty cycle (r_{dc}) is a system parameter for SMAC and is defined as the ratio of the awake interval (T_{active}) to the frame length (T_{frame}). The size of each period is normally fixed according to physical-layer and MAC parameters such as the radio channel bitrate (c) and the contention window sizes for the data period (W) and the SYNC period (W_s). The frame length and the duration of the sleep period can be calculated as:

$$\begin{aligned} T_{frame} &= T_{active}/r_{dc} \\ &= \left(T_{sync} + T_{listen}\right)/r_{dc} \\ &= \left((W_s - 1)t_{slot} + t_{SYNC}\right. \\ &\quad \left. + \left(W - 1\right)t_{slot} + t_{RTS} + t_{CTS}\right)/r_{dc} \\ T_{sleep} &= T_{active}(1 - r_{dc})/r_{dc} \end{aligned} \tag{3.1}$$

where t_{slot}, t_{SYNC}, t_{RTS}, t_{CTS} are the durations for a single contention slot and the packets of SMAC in a channel of bitrate c in terms of bits per second.

3.4. Discussion on the Cost of LBRF

The operation of LBRF is based on the information stored in NIT, the entries of which is updated by the modified SMAC. As we discussed in Section 3.3.4, the space complexity of NIT is $O(N^2)$ for small N and $O(N)$ for large N, where N is the number of neighbors of a node. The pseudo-code of relay selection mechanism given in Figure 3.2 will be used in the determination of the computational complexity of LBRF. According to the pseudo-codes the actual computation is done in the first procedure. In that procedure there is a search operation in NIT to determine the neighbors residing in the delivery region. In addition there are two search operations to find the closest neighbor to the sink in the delivery region with the minimum buffer occupancy level. These three operations is implemented as a one pass traversal over NIT and using a sorted list where the neighbors in the delivery region are inserted into the list according to the buffer occupancy level (primary) and the distance to the sink (secondary). At the end of the traversal, the neighbor node which is at the head of the list is going

to be the relay node. Hence, the computational complexity of determining the relay node is $O(N)$ due to the $O(N)$ time traversal operation over NIT and the sorted list operations of maximum complexity $O(N)$. The space complexity is also $O(N)$ since the length of the sorted list is at most N, and it operates on the elements of structure type that contains node identity, buffer occupancy level and distance information.

The overhead of LBRF on the network is not considerable. LBRF uses no extra control packet for its operation. Instead, it piggybacks buffer occupancy condition and routing class information to each MAC layer packet, which results with an insignificant increase in the length compared to the length of a data packet.

3.5. Performance Evaluation of LBRF

We evaluate the performance of LBRF routing scheme by comparing it with GPSR (which is based on greedy forwarding) and two random forwarding schemes via simulations using OPNET [80] in a target tracking video application scenario. In the first random scheme, called random forwarding (RF), the relay for a frame is determined randomly among the neighbors residing in the forward region of the sensor. In the second random scheme, called weighted random forwarding (WRF), the neighbors in the forward region are weighted in random selection with respect to their provision of advancement towards the sink. In other words, in WRF the closest neighbor to the sink can be determined as the relay with the highest probability whereas the closest neighbor to the node can be determined as the relay with the lowest probability. The comparison of LBRF and GPSR schemes enables us to assess the advantages of applying load balance and the disadvantages of greedy forwarding. The comparison of LBRF and the random forwarding schemes RF and WRF provides an assessment for the necessity of cross layer assistance in load balancing.

We evaluated the performance of routing schemes using 10 different network topologies with corresponding 10 different target trajectories where 200 video sensors are uniformly deployed in a square shaped environment of each side 400 m. Two targets are moving in the environment according to the random waypoint mobility model where

the target speeds are 2 m/s and the pause times are 0 seconds. In each scenario, the sink node is placed in the middle of the right side of the environment. The presented values are the mean values of the results obtained in these 10 scenarios.

In the simulation, we assumed that the video sensors are capable of taking images and compressing them with the cameras integrated on their hardware [81, 82]. Since the size of the data transmitted is directly related to the size of the image, SQCIF (128 x 96) format is assumed. The image module employs intra-frame encoding which results in compressed images of size 10 Kbits. Predictive encoding alternatives such as ISO MPEG and H.26x cannot practically be used in VSNs due to the high complexity involved [2]. Distributed source coding techniques are promising alternatives for encoding video in VSNs as they exploit the inter-frame redundancy with affordable complexity in the sensor nodes [83]. However, due to the lack of practical implementations yet available, we resort to the JPEG compression available on the image module.

Table 3.2. Simulation parameters for the evaluation of LBRF.

Parameter	**Value**
#Sensors/Area/Sink Loc.	200/400 m x 400 m/(400,200)
Data Packet Size	1000 bits
Video Frame Size	10 packets
Frame Rate	3,4,5,6 fps
Buffer Size	100 Kbits
Channel Rate	250 Kbps
MAC Layer	SMAC (Message Passing)
Duty Cycle	15%
Contention Window Size	64 slots
Sensing Radius	30 m
Transmission Radius	60 m
Initial Energy	200 J
TX Power	81 mW
RX Power	30 mW
IDLE Power	30 mW
SLEEP Power	0.03 mW

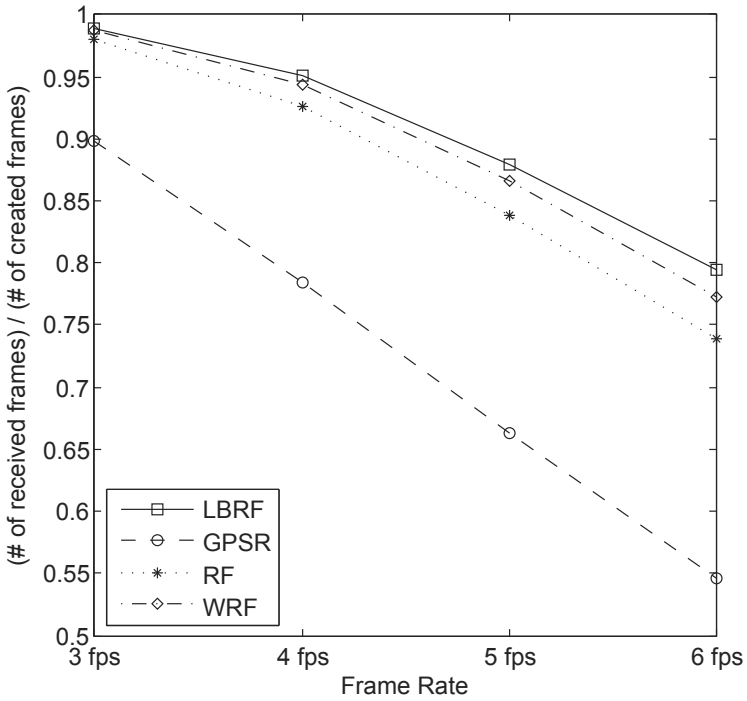
Generally, higher video quality is required for better VSN application performance. Video quality can be adjusted in the system by varying the image resolution and the camera frame rate. In our case, we fix the image resolution since a lower resolution may not be tolerated by the identification application, whereas a higher resolution results in frame sizes that cannot effectively be carried in the network. Therefore, in the simulations the frame rate of the cameras on the sensors is varied to alter the video quality throughout the network. Event triggered data generation is simulated where the triggering event is the visual detection of a target. Since the cameras support background subtraction feature, an image is produced only when the scenery changes significantly. Hence triggering occurs when the target is within the camera detection range of 30 m and is within the Field of View (FOV) of 52 degrees [3]. The communication ranges of the video sensors are 60 m. Other simulation parameters are presented in Table 3.2. The radio power consumption parameters are set as in [84], which are typical values for Mica2 Mote sensors.

3.5.1. Frame Delivery Performance

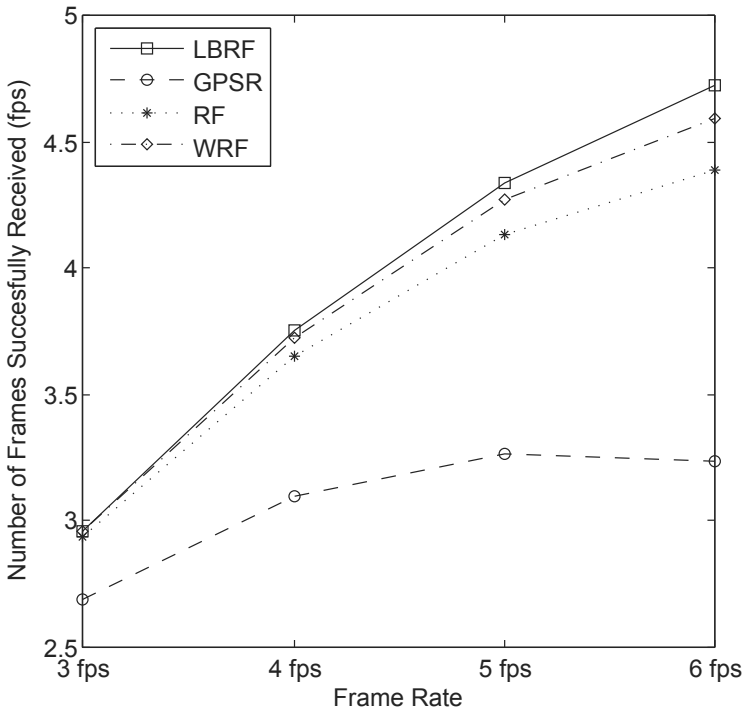
Figure 3.11 presents the frame delivery performance and the efficiency of the routing schemes in terms of successful frame delivery ratio and average frame rate received by the sink respectively as compared to the video frame rate of the application.

When we compare the performance of LBRF and GPSR routing schemes in terms of successful frame delivery to the sink in Figure 3.11a, we observe that LBRF outperforms GPSR. The two schemes achieve their maximum performance when the video quality is at its lowest, where the delivery ratio of GPSR is 89.7% and the delivery ratio of LBRF is 98.9%. The performances of the schemes for the maximum tested video quality are 54.5% and 79.5% for GPSR and LBRF, respectively. As the video quality increases, the performance gap between GPSR and LBRF widens from 11.2% to 25%.

The main reason for the performance gap between GPSR and LBRF in Figure 3.11 is the difference between the loss rates of both schemes in the relay nodes.

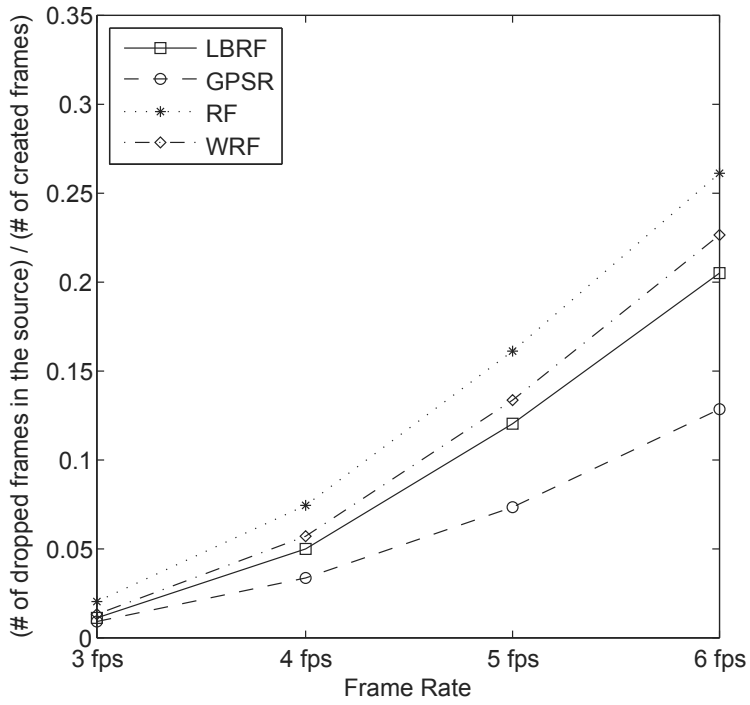


(a)

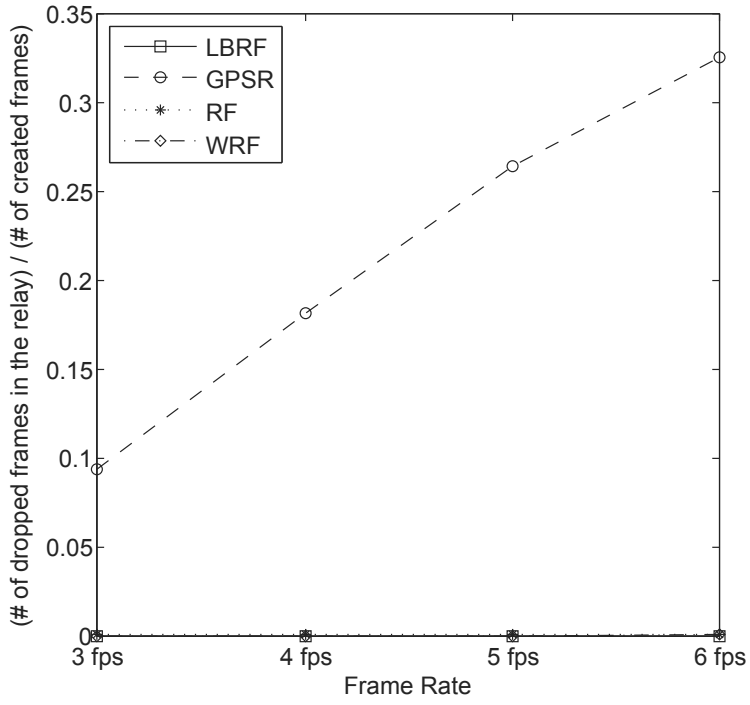


(b)

Figure 3.11. Frame delivery performance comparison: (a) Frame delivery ratio (b) Perceived frame rate.



(a)



(b)

Figure 3.12. Frame drop rate comparison: (a) Drop rate in the source (b) Drop rate in the relay.

In order to investigate the frame loss characteristics of these schemes, we analyze the frame drop rates in the source and in the relay nodes (Figure 3.12). It is remarkable that almost no frames are dropped in the relay nodes using LBRF, whereas the majority of the frame loss occurs in the relay nodes when GPSR is used (Figure 3.12b). GPSR determines its relay node as the one which provides the maximum advancement towards the sink in a greedy manner. The greedy approach results with the selection of the same sensor as the relay node by many sensors (Figure 1.1a). As a result, these relay nodes are more likely to have higher buffer occupancy levels. The buffer occupancy levels of these relays get higher when a target is detected by them. The presented loss rates of GPSR are the result of the buffer overflows triggered by the high buffer occupancy levels of the relay sensors. In contrast to GPSR, LBRF determines its relay node as the one which provides the maximum advancement towards the sink only among the sensors having the smallest buffer occupancy level which is available to receive a video frame. In that way, each sensor tries to achieve homogeneity in the buffer occupancy levels of its neighbors providing positive advancement, which supplies considerably lower levels of buffer occupancy with respect to GPSR. Preserving the homogeneity increases the probability for a node to find a relay with available buffer space. Combined with the cancellation of delivery due to insufficient buffer space in the destination, the loss rate in the relay nodes for LBRF is almost 0 for all quality levels, which results with the survival of more frames relayed in the network. However, providing homogeneity increases the number of nodes containing frames in their buffers, which as a result increases the number of nodes contending for the medium access in the neighborhood. The increase in the number of survived frames increases the average buffer occupancy levels of the sensors leaving less space for the frames to be created and the increase in the number of contenders decreases the output rate of a sensor. The joint effect of these two factors results with higher loss rates in the source nodes for LBRF as compared to GPSR (Figure 3.12a). As the video quality increases, LBRF utilizes more sensors as a relay, which increases the number of contending nodes. Since the increase in the number of frames does not affect the relay selection of GPSR, the performance gap between the loss rates in the source nodes of LBRF and GPSR gets larger. On the other hand, since GPSR always utilizes the same relay node, increase in the video quality results with an increase in the buffer occupancy level of these relay

nodes which in turn results with an increase in the performance gap between the loss rates in the relay nodes of LBRF and GPSR.

When we compare the performance of LBRF and the random forwarding schemes in terms of successful frame delivery, we observe that LBRF outperforms both RF and WRF schemes (Figure 3.11a). The loss rates in the relay nodes are almost 0 for all three routing schemes (Figure 3.12b). Hence, the difference between the frame delivery ratios are due to the drop rates in the source nodes (Figure 3.12a). Loss results show that LBRF provides more homogeneous and lower buffer occupancy levels in the neighborhood of a sensor. As a result more space is left in LBRF for the frames to be created as compared to RF and WRF schemes, which consequently decreases the loss rate in the source. The difference between RF and WRF schemes is due to the higher residence time of the frames in the network caused by shorter average advancement towards the sink in relay determination in RF.

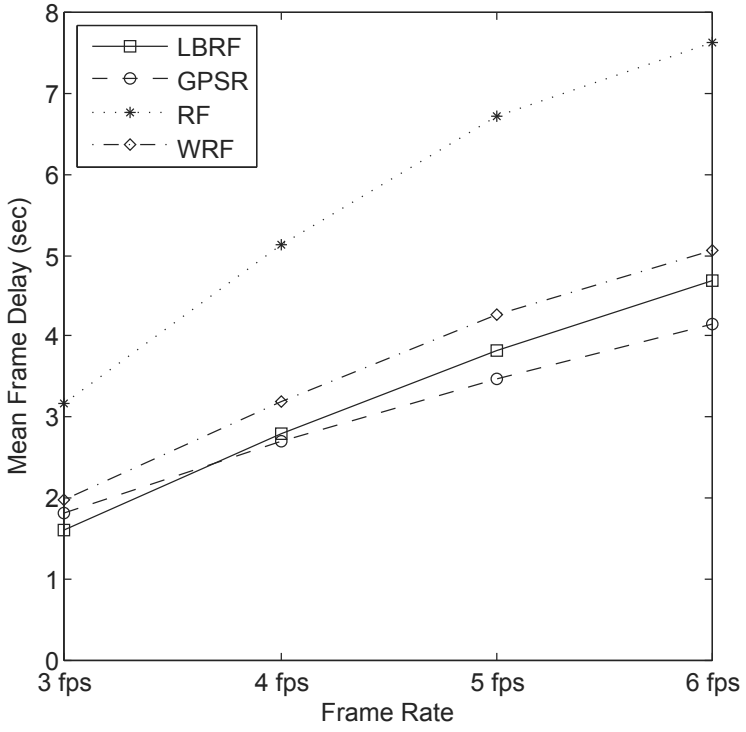
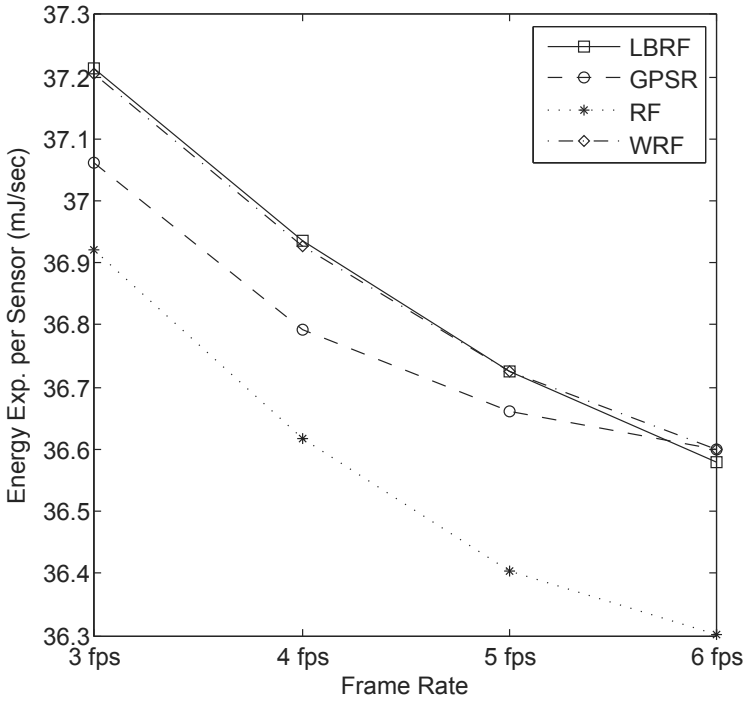


Figure 3.13. Frame latency performance comparison.

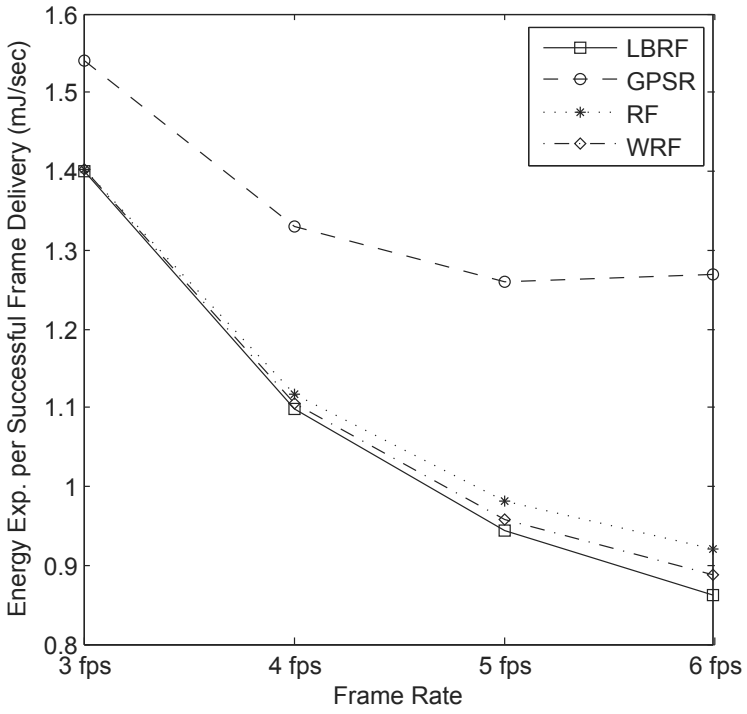
3.5.2. Frame Latencies

We compared the performance of the routing schemes in terms of mean delay of frames successfully delivered to the sink in Figure 3.13. When we compare LBRF and GPSR schemes for the minimum video quality, we observe the effects of the number of contenders and the buffer occupancy levels to the frame delivery latency. When the load is light, LBRF distributes the load among smaller number of neighbors in the forward region, which decreases the number of contenders in the neighborhood with respect to higher load conditions. In addition, we observe less accumulation in the buffers of the contending sensors. In contrast to LBRF, multiple sensors determine the same node as their only relay in GPSR resulting with less number of sensors selected as relay. As a consequence, the number of contenders remains less than that of LBRF, which is a positive factor for decreasing the delay in the network. However, higher accumulation of frames is observed in the buffers of these relays, which negatively affects the network delay performance. The effect of the number of contenders in LBRF is observed to have less effect than the buffer accumulation in GPSR in light load conditions. As the video quality increases, LBRF utilizes more neighbors in the forward region of a sensor, which increases the number of contenders, whereas the number of contenders is constant for all quality levels in GPSR. Since LBRF keeps more frames alive in the network as shown in Figure 3.11a, more frames accumulate in the buffers of the utilized sensors. Hence, as the video quality is increased in the network, we can observe that the delay performance of LBRF is more affected by the increase in the number of contenders and the increase in the buffer occupancy levels.

In the comparison of LBRF and the random forwarding schemes, we observe that LBRF outperforms RF and WRF schemes for each video quality in terms of frame delivery latency. The justification for the lower latency is the longer average advancement towards the sink and the provision of more homogeneous and lower buffer occupancy levels in the neighborhood in the determination of relays for the frames.



(a)



(b)

Figure 3.14. Energy consumption comparison: (a) Energy expenditure per sensor node (b) Energy expenditure per received frame.

3.5.3. Energy Consumptions

We compared the performances of the routing schemes in terms of average energy expenditure per sensor node and average energy expenditure in the network per frame successfully delivered to the sink in Figure 3.14a and Figure 3.14b respectively. The general behavior of the energy expenditure is observed to be decreasing as the video quality increases. This interesting result, which is also mentioned in [85], is related with the reduced idle listening duration due to the overhearing avoidance and virtual carrier sensing mechanisms of SMAC. The overhearing avoidance mechanisms enables the overhearing sensors to sleep earlier than the actual sleep time, while the virtual carrier sensing mechanism prevents a node to wake up prior to the end of an ongoing transmission. SMAC requires some nodes to follow multiple sleep schedules causing a significant proportion of the nodes to stay awake much longer than the expected awake period determined by the duty cycle [86]. Hence, such overhearing nodes which follow more than one schedule continue to sleep in other schedules, where they would be awake and wasting energy for idle listening in case of no transmissions in their vicinity. The total energy conservation due to decreased idle listening in these overhearing sensors is generally higher than the energy expenditure of the communicating nodes. Hence, as the number of transmissions increases, the energy conservation in the network increases because of the decreased idle listening [87]. The same argument holds for the modified SMAC algorithm since it also applies an overhearing avoidance mechanism and a virtual carrier sensing mechanism.

LBRF scheme operates with the modified version of SMAC in which the overhearing avoidance mechanism of SMAC is relaxed to put the nodes into sleep mode later than the original SMAC in order to overhear more packets from the neighbors containing fresh buffer occupancy information. Hence, the modified version of SMAC increases the energy expenditure of the sensors. In addition, since more frames are successfully delivered by LBRF, the energy expenditure for transmissions is higher as compared to GPSR for any video quality. For lower video quality, LBRF causes the sensors to spend more energy than GPSR. However as the video quality increases, LBRF utilizes more sensors as a relay and the number of sensors in the vicinity of

a transmission increases accordingly. Since the number of sensors utilized as a relay is constant in GPSR for all video qualities, the gap between the energy expenditure of these two protocols decreases as the video quality increases. Furthermore, LBRF spends less energy than GPSR for the highest tested video quality, which implies that the mentioned higher energy expenditures of LBRF are compensated by the increased amount of sleep time all over the network. Note that, the lowest energy expenditure is observed for the RF scheme. The highest residence time of the frames in the network due to usage of RF results with the lowest duration spent for idle listening in the network. As a consequence, the energy expenditure in the network is in the lowest level for each video quality when the RF is used as the routing scheme.

Figure 3.14b presents the energy spent in the network per successful frame delivery which is a metric to assess the efficiency of energy consumption. We observed previously in Figure 3.11b that LBRF outperforms GPSR in terms of average frame rate delivered to the sink and the performance difference between the two schemes becomes larger as the video quality gets better. Combining these results with those in Figure 3.14a, we observe that LBRF utilizes the energy budget more efficiently than GPSR for all video quality levels. Moreover, as the video quality increases, the performance gap between the two schemes widens.

In the cross comparison of LBRF, RF and WRF schemes in terms of the energy spent in the network per successful frame delivery, we observe that the three routing schemes achieves similar performance where LBRF causes the network to spend slightly less energy per successful frame delivery.

By assessing the performance comparison results in terms of frame delivery ratio, frame latency and energy efficiency, we can conclude that load balancing is quite beneficial compared to greedy forwarding. In addition, we can confirm that the cross layer assistance in the provision of load balancing is advantageous, where the higher frame delivery ratio is achieved with lower frame latency and lower energy expenditure per successfully delivered frame.

4. DIRECTIONAL LOAD BALANCED SPREADING

We propose a new cross-layer routing scheme called DLBS (Directional Load Balanced Spreading) by combining the load-balancing approaches applied in LBRF and DGR [17] routing schemes. The load-balancing approach in the LBRF scheme aims to solve the problem introduced in Figure 1.1a. LBRF tries to establish an efficient utilization of the available buffer space in the sensors, which provides positive advancement towards the sink. By considering the buffer states of these sensors, it tries to distribute the load equally among them. This approach locally improves the survival of the routed packets as well as provides more available buffer space for the traffic created by the sensor node itself. However, load balancing in a restricted zone leads to an increase in the number of contenders which increases the average medium access time. Hence, the accumulation of more frames in a higher number of contending nodes results with increased cumulative waiting time of the frames, leading to an increased average frame delay.

In contrast to LBRF, DGR scheme applies a direction-based load balancing approach by distributing the load on a sensor into predetermined multiple directions to traverse over multiple possibly non-interfering paths converging to the sink. These directions are determined by deviation angles defined as the deviation from the direction of the sink (Figure 4.1). DGR divides a single video stream into multiple sub-streams and each sub-stream is delivered in a round robin fashion over these paths. The partitioning of a video stream and the delivery of these sub-streams over disjoint paths in a round-robin fashion provides a fair load distribution over these paths traversing different regions of the network. This implicitly provides a spatial load balance in the network. However, each of the paths in DGR are determined as in single-path static routing schemes, and may suffer from the congestion problem presented in Figure 1.1a in case of the existence of multiple source nodes in the network.

DLBS is a hybrid routing technique which combines the ideas in LBRF and DGR to benefit from the advantages and to reduce the effects of the disadvantages

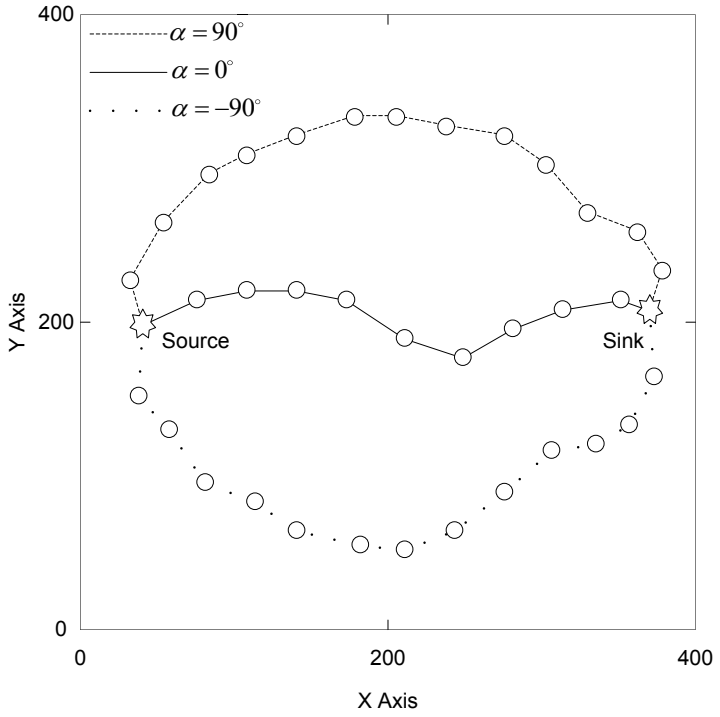


Figure 4.1. Illustration of the paths determined by deviation angle α.

of both approaches. DLBS utilizes the round-robin angle selection strategy during delivery of frames in each video source node to provide spatial load balancing. In each direction, DLBS applies LBRF like routing mechanism to avoid the congestion problem and to further improve the load-balance. In that way, DLBS aims to improve the perceived quality of the video from several video sources by increasing the successful frame delivery performance and decreasing the mean frame delay.

DLBS utilizes the buffer occupancy level information of candidate next hop sensors in relay selection which is acquired by the simple piggybacking mechanism, as in LBRF. Hence, DLBS works with the modified version of SMAC as the underlying MAC layer protocol. The problem of routing around dead-ends is solved by adopting the same mechanism in LBRF. The only difference is that the general DLBS scheme is applied by only the nodes in the routing-class C_1 and the nodes belonging to other routing classes applies the general LBRF scheme. The directional routing approach in DLBS is similar to the one in DGR. However, DGR requires node-disjoint path construction from the source to the sink prior to the transmission of the video stream. In

contrast to DGR, DLBS uses a hop by hop strategy in route establishment where the next node in a path is determined in an ad-hoc manner considering the buffer occupancy conditions of the candidate relay sensors at the time of delivery which provides a more dynamic relay selection strategy.

4.1. DLBS Routing Mechanism

In DLBS, there are two main strategies that are utilized in the decision of a candidate relay for a frame. The first strategy is called as the angle based spreading (ABS) and the second strategy is the LBRF scheme. The aim of ABS strategy is to determine the next hop node for a given *DeviationAngle*. In a source node, *DeviationAngle* is selected from a set of predetermined deviation angles in a round-robin fashion per successful frame delivery, which is inferred from a successful RTS-CTS exchange. The angle of transmission is embedded in each packet of the frame for further reference by the relay nodes in their routing decisions. In the relay nodes, ABS strategy adjusts the deviation angle embedded in the packets of the frame in a hop-by-hop manner in order to direct the frames towards the sink quickly after providing a spatial spreading. The relay selection mechanism uses the adjusted deviation angle in the relay nodes. In angle adjustment the relay nodes uses a variable called *SrcToSinkHopCount*. The value of this variable is determined by the source node as the expected number of total hops to the sink and recorded in the packets of the frame to be transmitted. *SrcToSinkHopCount* is calculated assuming the case where the network is dense and the route to the sink can be established with the minimum number of hops, which is the ratio of the distance to the sink to the communication range of the sensor.

ABS determines a candidate next hop node for a given deviation angle by considering the buffer occupancy levels and the distances of the nodes to the strategic mapping location, which will be introduced in the following section. ABS utilizes an LBRF-like strategy to determine the relay node for a frame. If ABS cannot determine a relay in the direction of the current deviation angle either because of inexistence of a sensor in that direction, or unavailability of the buffers of existent sensors, DLBS applies the LBRF routing mechanism in order to find a suitable relay for the frame.

If the *SrcToSinkHopCount* of a source or a relay node is smaller than or equal to two, then nodes only use the LBRF, to avoid unnecessary spreading of the frames near the sink.

4.2. Relay Selection in the ABS Strategy

ABS strategy tries to find a relay for a deviation angle (α) by rotating the XY coordinate system to the X^vY^v virtual coordinate system and then rotating the X^vY^v virtual coordinate system to the X^mY^m mapping coordinate system. In this section, the deviation angle refers to the actual angle used in a source node or the adjusted one used in a relay node. Initially, each node calculates the virtual coordinates [17] of its neighbors, which are defined as the coordinates in the virtual two-dimensional coordinate system where the node is the origin, and the X-axis is the line between the node and the sink. If (x_s^r, y_s^r) is the position of a node, (x_n^r, y_n^r) is the position of its neighbor and (x_t, y_t) is the position of the sink, the virtual coordinates (x_n^v, y_n^v) of the neighbor is calculated with the following rotation mapping:

$$\begin{cases} x_n^v = \cos(\theta)(x_n^r - x_s^r) + \sin(\theta)(y_n^r - y_s^r), \\ y_n^v = \cos(\theta)(y_n^r - y_s^r) - \sin(\theta)(x_n^r - x_s^r), \end{cases} \tag{4.1}$$

$$\theta = \arctan\left(\frac{y_t - y_s^r}{x_t - x_s^r}\right) \tag{4.2}$$

ReferenceLine is defined as the straight line between the node and the sink, and *DeviationAngle* (α) is defined as the angle that specifies how much a path is expected to deviate from the *ReferenceLine* at the origin point. The *DeviationLine* is acquired by rotating the *ReferenceLine* by α. The rotation of the position of the sink around the sensor by α, which means the rotation of the virtual coordinate of the sink around the origin by α, yields the position of the virtual sink, which is located at the endpoint of the *DeviationLine*. The position of the virtual sink is referred as the *StrategicMappingLocation* (Figure 4.2). If the virtual coordinates are rotated around

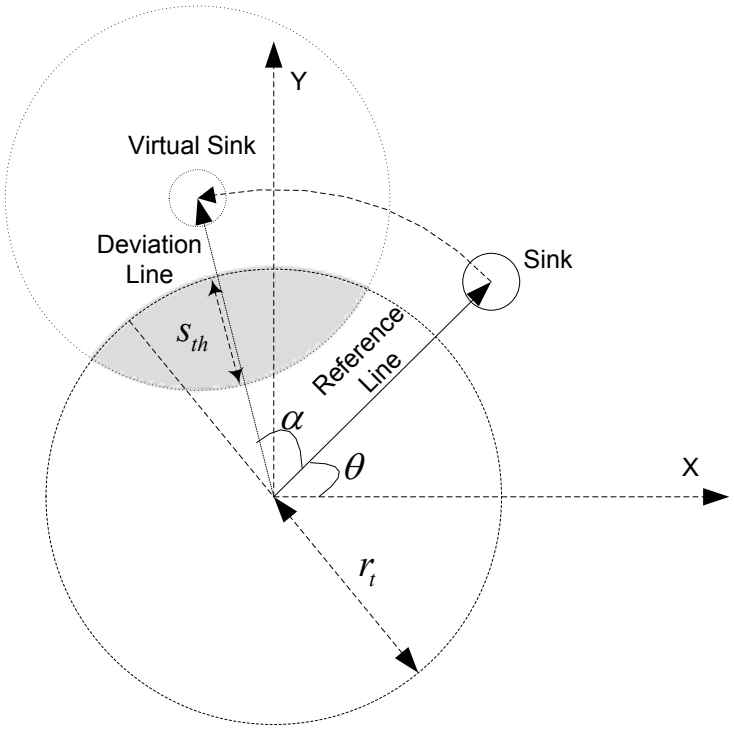


Figure 4.2. Relay selection in the ABS strategy.

the origin by α, the rotated coordinates are defined as mapping coordinates [17]. If $\alpha < 0$, the rotation is clockwise, and if $\alpha > 0$, the rotation is counterclockwise. The mapping coordinates are used in the determination of the candidate next hop sensor for a deviation angle. The mapping coordinates (x_n^m, y_n^m) of a neighbor node is calculated as

$$\begin{cases} x_n^m = \cos(\alpha)x_n^v + \sin(\alpha)y_n^v, \\ y_n^m = \cos(\alpha)y_n^v - \sin(\alpha)x_n^v, \end{cases} \tag{4.3}$$

The set of candidate relays towards the virtual sink in the direction of *DeviationAngle* (α) are determined in a similar way as in the LBRF routing mechanism. However, we limit the number of candidate relays, in order to limit the propagation away from the sink. For this purpose, we introduce the *SpreadingThreshold* (s_{th}), where $0 \leq s_{th} \leq r_t$. If the distance of a node to the *StrategicMappingLocation* (i.e. virtual sink) is d^α, the distance of a neighbor node to the *StrategicMappingLocation* is

d_n^α and the transmission radius of a node is r_t; then the set of candidate relays are the neighbors satisfying $(r_t - s_{th}) < (d^\alpha - d_n^\alpha) \leq r_t$. The region where the candidate relays are scanned is illustrated by the shaded area in Figure 4.2. As in LBRF, a sensor is excluded from the set of candidate relays if its buffer occupancy level is not enough to store the frame to be delivered. The remaining candidate sensors are sorted according to their buffer occupancy levels from smallest to largest. The relay node is determined as the nearest node to the *StrategicMappingLocation* among the candidate relay nodes which have the smallest buffer occupancy level at the time of delivery.

4.3. Angle Adjustment in ABS Strategy

Angle adjustment is performed in the relay nodes to determine the direction of the next hop in the route discovery process and depends on the variables *DeviationAngle*, *SrcToSinkHopCount* and *HopCount*. These parameters are extracted from the packets of the received frame where the *DeviationAngle* (α) and *SrcToSinkHopCount* variables are embedded by DLBS in the source node and *HopCount* variable, which is the number of hops traversed from the source up to the relay node, is provided by the MAC layer.

Instead of using the actual value of the deviation angle extracted from the packets of the received frame, a relay node adjusts the deviation angle, α, as a function of the values of *SrcToSinkHopCount* and *HopCount*. There are several alternatives for this function [17], among which the following function is utilized in our routing scheme:

$$\alpha_H = \frac{\max\left(SrcToSinkHopCount - HopCount, 0\right)}{SrcToSinkHopCount}\alpha \tag{4.4}$$

The rationale behind the selection of this function is that the α_H value decreases slowly resulting with longer paths towards the sink. The usage of longer path lengths aims the exploitation of the network regions in proximity to the perimeter of the network, where the network traffic is expected to be relatively lower than the inner regions of the network. However, the total energy expenditure per packet increases with the number of hops in the routing path. Hence, we prefer a linear adjustment

function for the deviation angle.

In Figure 4.3, a sample relay selection scenario is presented, where the deviation angle α is 90°. In this scenario, s_{th} is taken as $\frac{r_t}{2}$. There are three nodes in the selection region which are the nodes A, B and C with buffer occupancy levels 3,3, and 5 respectively. Since, the node A is the closest one among the candidates to the *StrategicMappingLocation* (the position of the virtual sink) with the smallest buffer occupancy level, it is determined as the relay node for α=90°.

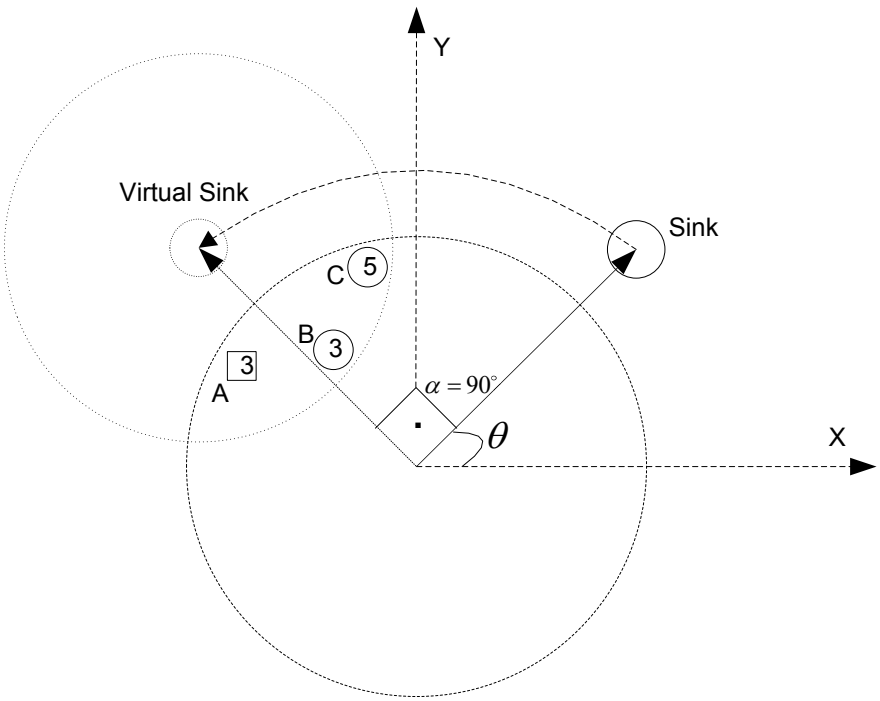


Figure 4.3. A sample relay selection scenario in DLBS where α=90°.

Pseudo-codes for the DLBS scheme in the source and the relay nodes are provided in Figure 4.4 and Figure 4.5 respectively.

4.4. Demonstrative Analysis of Tradeoff between Load Partitioning over Multiple Paths and Single Path Approaches

In order to compare the effect of sending the overall load over the shortest path of length H hops and sending distinct k partitions of load through k parallel routes of

```
1: procedure DetermineRelayInSource(N,NIT,STSHC,α_list,α,Frame)
2:     relay ← 0
3:     DetermineDeviationAngle(α_list, α)                ▷ Round robin fashion
4:     if (STSHC ≤ 2) then                               ▷ STSHC: Source to sink hop count
5:         α ← 0
6:     end if
7:     Calculate mapping coordinates of the neighbors and the sink w.r.t α
8:     N_s ← {n : n ∈ N and of C_1 residing in the shaded region in Figure 4.2}
9:     boc_min ← argmin_{n∈N_s}(NIT(n).boc)
10:    relay ← argmin_{n∈ boc_min} (DistToVirtualSink(n))
11:    if (relay = 0) then
12:        SearchForCandidateRelay(N,NIT,1,FORWARD,relay)      ▷ LBRF
13:    end if
14:    if (relay ≠ 0) then
15:        SetDeviationAngle(Frame,α)
16:        SetSourceToSinkHopCount(Frame,STSHC)
17:        SetRelay(Frame,relay)
18:    end if
19:    return Frame
20: end procedure
```

Figure 4.4. Pseudo-code for the relay selection mechanism of DLBS in the source.

length $H+h$ hops, we provide a demonstrative analysis of the delay performance of the routes illustrated in Figure 4.6. The detailed analysis requires (i) analysis of the contention phase, (ii) the analysis of the duty cycle, (iii) analysis of delay and throughput in multihop sensor networks. Moreover, the dynamic nature of the route determination process and the target trajectories further complicates a detailed analysis. We provide a simplified analysis of the delay performance in order to the demonstrate the tradeoff between the load partitioning over multiple paths and the single path approach. Figure 4.6a and Figure 4.6b are the simplified representations of the paths traversed by the frames in LBRF and DLBS/DGR schemes respectively. For simplicity, we assume

```
1: procedure DetermineRelayInRelay(N,NIT,STSHC,Frame)
2:     relay ← 0
3:     GetDeviationAngle(Frame,α)
4:     GetSourceToSinkHopCount(Frame,STSHC_F)
5:     GetHopCount(Frame,hopcount)
6:     if (STSHC_F > hopcount ∨ STSHC ≤ 2) then
7:         α ← 0
8:     else
9:         α ← α ( (STSHC_F − hopcount) / STSHC_F )
10:    end if
11:    Calculate mapping coordinates of the neighbors and the sink w.r.t α
12:    N_s ← {n : n ∈ N and of C_1 residing in the shaded region in Figure 4.2}
13:    boc_min ← argmin_{n∈N_s}(NIT(n).boc)
14:    relay ← argmin_{n∈ boc_min} (DistToVirtualSink(n))
15:    if (relay = 0) then
16:        SearchForCandidateRelay(N,NIT,1,FORWARD,relay)        ▷ LBRF
17:    end if
18:    if (relay ≠ 0) then
19:        SetRelay(Frame,relay)
20:    end if
21:    return Frame
22: end procedure
```

Figure 4.5. Pseudo-code for the relay selection mechanism of DLBS in a relay.

that the nodes are represented as M/M/1 queues. Assuming a perfect MAC layer, no collisions and no interference between disjoint paths, the average delays in each node in Figure 4.6a and Figure 4.6b are $\frac{1}{\mu-\lambda}$ and $\frac{1}{\mu-\lambda/k}$ respectively, where μ is the service rate of the sensor and λ is the data input rate of the sensor. Hence, the total delay in each path can be found as $T_1 = \frac{1}{\mu-\lambda}H$ for single path route and $T_2 = \frac{1}{\mu-\lambda/k}(H+h)$ for each route of multiple parallel paths.

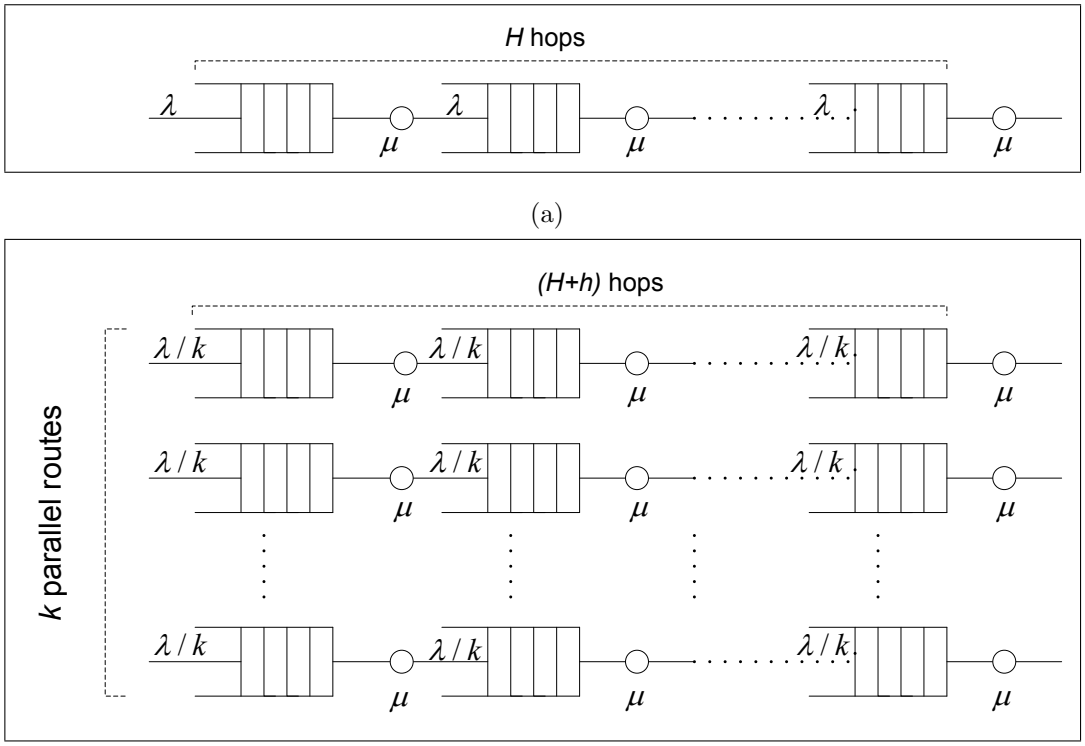


Figure 4.6. The illustration of (a) a single path sensor network route with H hops (b) multiple parallel paths with $H + h$ hops.

We aim to find the upper bound of h (extra hops traversed by frames) so that the delay for transmitting a partition of load over a longer route is smaller than the delay of transmitting the whole load over a single and shorter route. Hence,

$$\begin{aligned} T_2 &\leq T_1 \\ \frac{1}{\mu - \lambda/k}(H + h) &\leq \frac{1}{\mu - \lambda} H \\ k(\mu - \lambda)(H + h) &\leq (k\mu - \lambda)H \\ k(\mu - \lambda)h &\leq \lambda(k - 1)H \\ h &\leq \frac{k - 1}{k} \frac{\lambda}{\mu - \lambda} H \end{aligned} \tag{4.5}$$

From (4.5), we can see that in the heavy traffic conditions where λ approaches μ, the number of extra tolerable hops (h) approaches and even may get greater than the number of hops (H) in the shortest path. Hence, the frame latency decreases as

we choose to split the load into parts and transmit over longer routes. The obvious solution of transmitting the load through a single path is advantageous only if λ is much smaller than μ (in the light traffic conditions). However, due to the dynamic nature of the target trajectories and bursty nature of packet generation, the load generated in the network usually approaches to the output rate. This behavior can be observed in Figure 4.8a. Even at the rate of 3 fps, we can observe a higher number of frame drops in the source node when the LBRF scheme is used, i.e. the input rate increases in the relay nodes resulting with a higher input rate than the output rate. This situation results with an increased packet accumulation and finally more packet losses in the sensor buffers. The increase in the delay is caused by the increase in the packet accumulation in the LBRF case, which have paths usually shorter than DLBS.

4.5. Discussion on the Cost of DLBS

As in the case of LBRF, the operation of DLBS is based on the information stored in NIT with space complexity $O(N^2)$ for small N and $O(N)$ for large N, where N is the number of neighbors of a node. The pseudo-codes of relay selection mechanisms in the source and relay nodes given in Figure 4.4 and Figure 4.5 will be used in the determination of the computational complexity of DLBS. We can observe that the operation of the pseudo-codes are similar to the operation of LBRF (Figure 3.2). The difference is in the determination of the candidate neighbors with respect to α direction instead of with respect to the sink direction which requires extra computations of complexity $O(N^2)$ for small N and $O(N)$ for large N. After determining the candidates, there are two search operations to find the closest neighbor to the sink in the delivery region with the minimum buffer occupancy level. These operations is implemented using a sorted list where the neighbors in the delivery region are inserted into the list according to the buffer occupancy level (primary) and the distance to the sink (secondary). At the end of the traversal, the neighbor node which is at the head of the list is going to be the relay node. Hence, the computational complexity of determining the relay node is $O(N)$ due to the sorted list operations of maximum complexity $O(N)$. Hence the overall time complexity of relay selection mechanisms in DLBS is $O(N^2)$ for small N and $O(N)$ for large N. As compared to time complexity of LBRF which is

$O(N)$, the computational overhead of DLBS on the sensors is higher especially when the the number of neighbors of a node is small. The space complexity is also $O(N)$ since the length of the sorted list is at most N, and it operates on the elements of structure type that contains node identity, buffer occupancy level and distance information.

The overhead of DLBS on the network is not considerable. DLBS uses no extra control packet for its operation. It piggybacks deviation angle and source to sink hop count information to each MAC layer packet in addition to the information piggybacked by LBRF which are buffer occupancy condition and routing class information. As compared to LBRF the length of the packets is longer in DLBS but this is an insignificant increase in the length as compared to the length of a data packet.

4.6. Performance Evaluation of DLBS

We evaluate the performance of DLBS routing scheme by comparing it with LBRF and DGR schemes via simulations using OPNET [80] in a target tracking video application scenario. In DLBS, the *SpreadingThreshold* (s_{th}) is set as 30 m, which is the half of the communication range. Other simulation parameters and the assumptions are same with the ones presented in the performance evaluation section of Chapter 3.

Table 4.1. Performance of DLBS for different α sets.

α set	**Delivery ratio**	**Latency (sec)**	**Energy efficiency (mJ/sec)**
$\{60°, -60°\}$	0.840	4.153	0.813
$\{60°, 0°, -60°\}$	0.841	4.278	0.814
$\{90°, -90°\}$	0.844	4.107	0.805
$\{90°, 0°, -90°\}$	0.856	4.034	0.797

In order to determine the α set to be used in the simulations, we evaluated the performance of DLBS for different α sets in terms of mean frame delivery ratio, mean frame latency and mean energy expenditure in the network per successful frame delivery, when the frame rate is set as 6 fps. From the results in Table 4.1, we deduce that DLBS obtains the best performance when the α set is $\{90°, 0°, -90°\}$. Hence, in the simulations of DLBS and DGR we used the α set as $\{90°, 0°, -90°\}$.

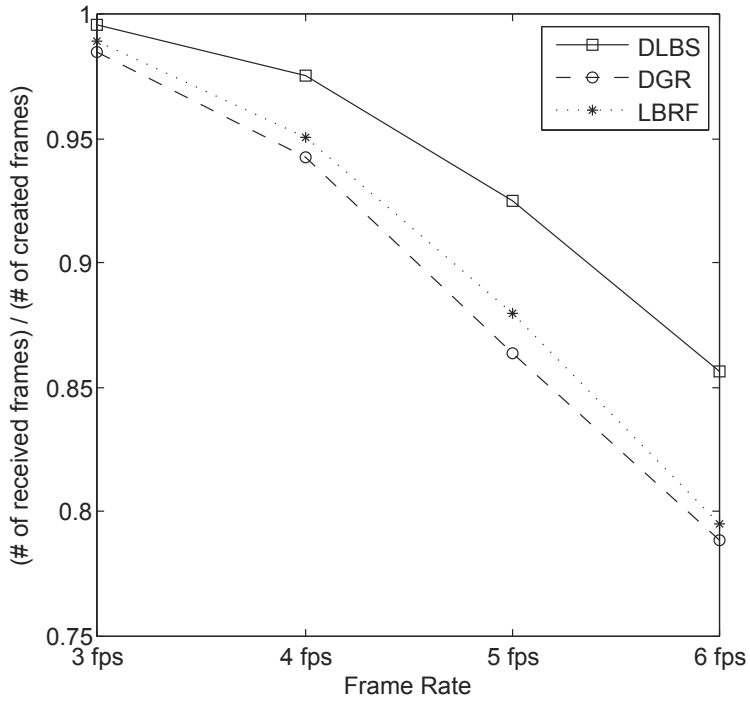


Figure 4.7. Frame delivery performance comparison.

4.6.1. Frame Delivery Performance

Figure 4.7 presents the frame delivery performance and the efficiency of the routing schemes in terms of successful frame delivery ratio and average frame rate received by the sink respectively as compared to the video frame rate of the application.

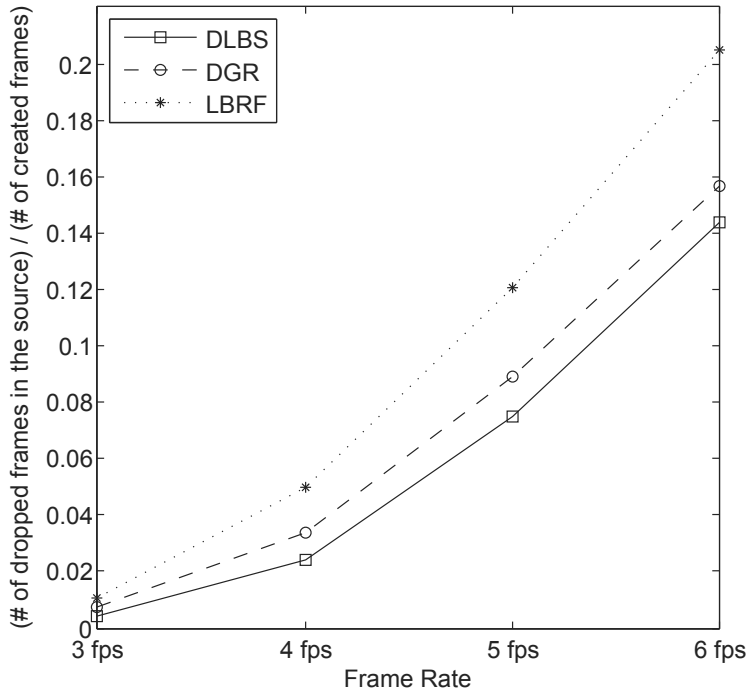
In order to provide a comparison between the two load-balancing approaches (local, spatial), we compare LBRF and DGR schemes. We observe that the successful frame delivery performance of LBRF is slightly higher (approximately 1%) than the performance of DGR. LBRF tries to increase the frame delivery performance by dynamically utilizing the storage space of alternative sensors which provides positive advancement towards the sink in a load balanced manner considering the buffer occupancy conditions of them. However, it is not successful to effectively alleviate an existent congestion in the direction of the sink, since all traffic is restricted to be routed along a narrow area in the network. On the other hand, DGR scheme tries to increase the frame delivery performance by dividing and spreading the load in multiple directions, which provides the distribution of the load to a wider area in the network

with respect to LBRF. However, since the paths in each direction are constructed deterministically in DGR, it does not utilize alternative sensors in these directions. It also does not consider the buffer occupancy conditions of the sensors along the routes. Therefore, it cannot prevent an occurrence of congestion in these sensors.

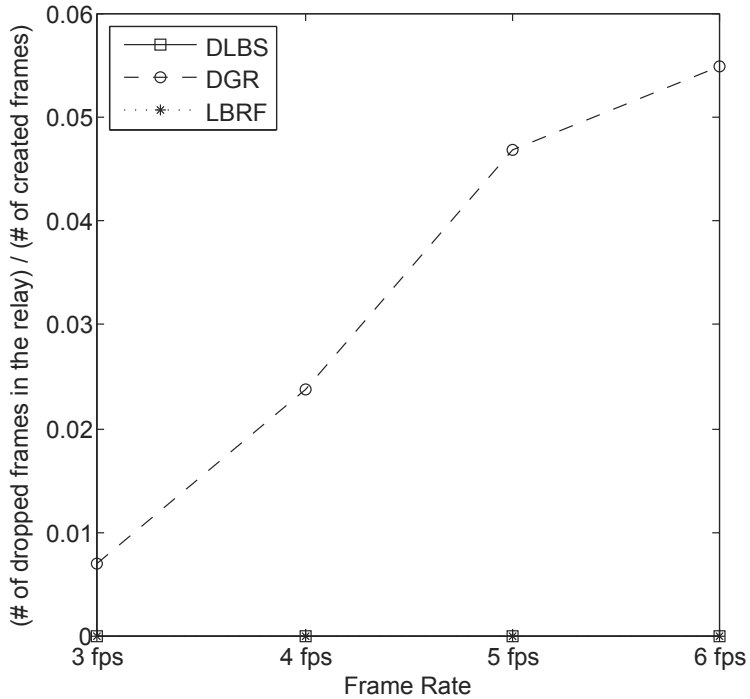
DLBS combines the spatial and the local load balancing approaches and it employs dynamic buffer occupancy aware load balancing in multiple directions. This combination provides DLBS to benefit from the advantages and to reduce the effects of the disadvantages of the both approaches. In comparison of DLBS and DGR, we observe that DLBS achieves a substantially higher frame delivery ratio. The performance difference between the two schemes is approximately 1-7% as the video quality is varied between 3 fps-6 fps. DLBS also outperforms LBRF and the maximum performance difference between the two schemes is achieved as 6.1% when the video quality is 6 fps.

As we compare LBRF and DGR schemes, we can observe the effects of the advantages and disadvantages of the local and spatial load balancing approaches. LBRF has a better performance in terms of drop rate in the relay (Figure 4.8b), since it aims to keep the relayed frames survived in the system by utilizing more sensors for storage using a buffer occupancy aware routing mechanism. The increase in the number of contenders results with a decrease in the output rate per sensor. Hence more created frames are dropped in the sensors, since less buffer space is reserved for their storage, due to accumulating relayed frames. DGR does not consider the buffer occupancy condition of the relays, and uses a static route for each direction, which leads to more number of relayed frame drops in existence of congestion. However, it has a better performance in terms of drop rate in the source (Figure 4.8a), due to the spreading of the load to a wider area in the network and inexistence of a buffer reservation for relayed frames, jointly increasing available space for newly created frames.

DLBS outperforms both DGR and LBRF schemes by utilizing a higher number of sensors as relay in a wider area in the network providing more storage space for relayed frames. DLBS benefits from buffer occupancy aware routing mechanism of LBRF to reliably relay the frames. In addition, it benefits from the idea of dividing and spreading



(a)



(b)

Figure 4.8. Frame drop rate comparison: (a) Drop rate in the source (b) Drop rate in the relay.

the load into different directions offered by DGR to provide more alternative sensors to be utilized as a relay. Hence, we observe that the drop rate of relayed frames in DLBS is as low as the drop rate in LBRF whereas the drop rate in the source is lower than the drop rate in DGR.

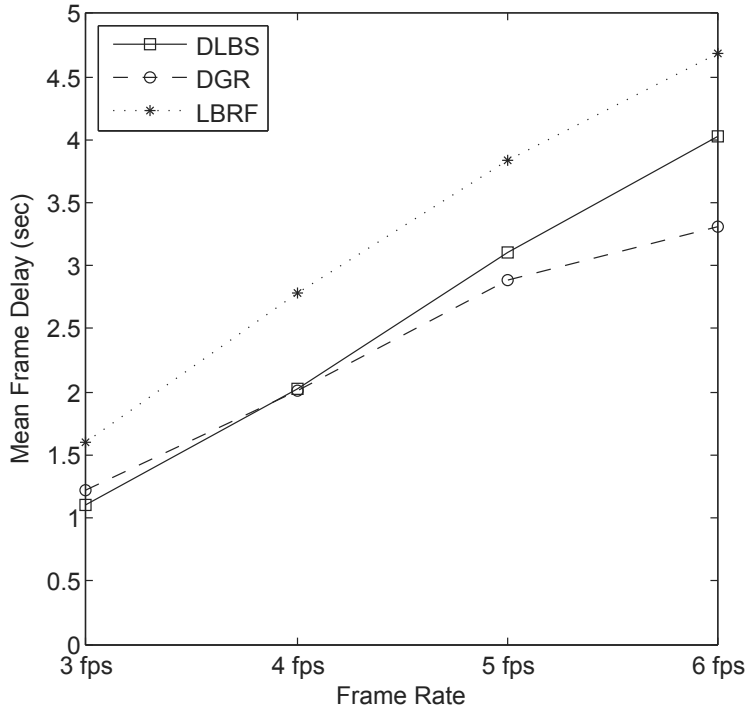


Figure 4.9. Frame latency performance comparison.

4.6.2. Frame Latencies

DGR partitions the load, spreads each partition in different directions in the network, and is a multidirectional static routing scheme. The behavior of DGR is analogous to multi-path GPSR. Since the load in each direction in DGR is less than the load in GPSR, its latency performance is better than the performance of GPSR for all video quality levels. Since the performance evaluation results in Chapter 3 show that GPSR outperforms LBRF in terms of latency performance, there is a significant performance gap between LBRF and DGR routing schemes.

DLBS benefits from the load partitioning and spreading approach of DGR but suffers from the factors adversely affecting the latency performance explained for LBRF.

Hence the performance improvement of DLBS with respect to LBRF remains almost same as the performance improvement of DGR with respect to GPSR for all video quality levels.

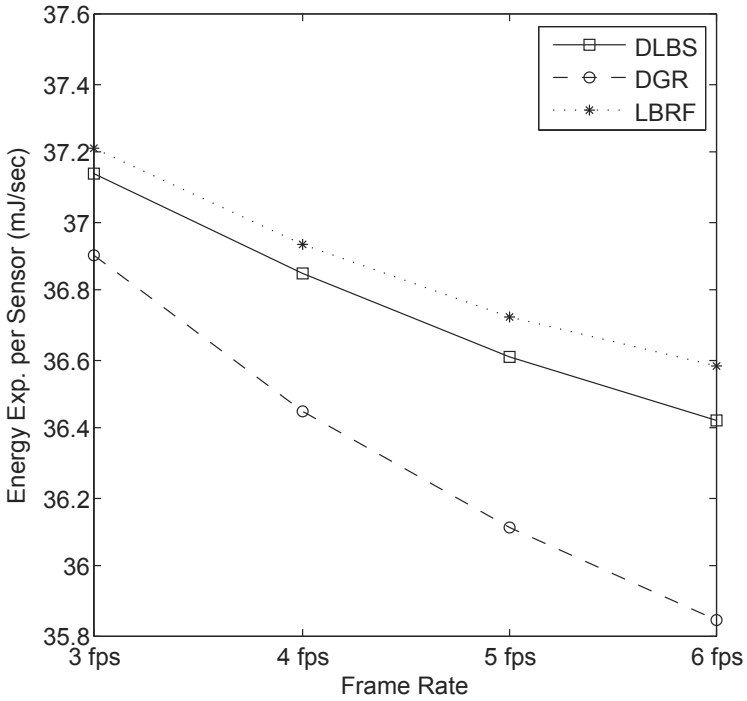
4.6.3. Energy Consumptions

We compared the performances of routing schemes in terms of average energy expenditure per sensor node and average energy expenditure in the network per frame successfully delivered to the sink in Figure 4.10a and Figure 4.10b respectively. The general behavior of the energy expenditure is similar to the behavior observed in the performance evaluation section of Chapter 3 which has a decreasing trend with the increased video quality. This result was justified with the reduced idle listening duration due to the overhearing avoidance and virtual carrier sensing mechanisms of SMAC [87].

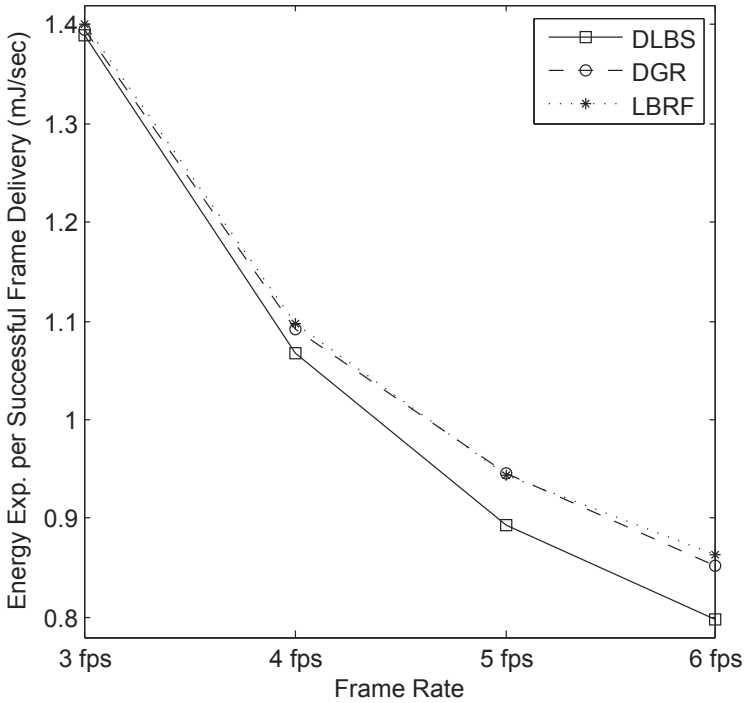
LBRF and DLBS schemes operate with the modified version of SMAC in which the overhearing avoidance mechanism of SMAC is relaxed to put the nodes into sleep mode later than the original SMAC in order to overhear more packets from the neighbors containing fresh buffer occupancy information. Hence, the modified version of SMAC increases the energy expenditure of the sensors.

The energy consumption of DGR per sensor node is better than LBRF, since it operates with the original version of SMAC and it accesses to a wider area in the network enabling more sensors not involved in the frame delivery to switch into SLEEP mode. In addition, the higher number of contenders in LBRF leads to more collisions as compared to DGR which increases the energy consumption per sensor node in the network.

Since DLBS also accesses a wider area in the network as DGR, the energy consumption is less than LBRF. However, since DLBS delivers 1-7% more frames (Figure 4.7a) and increases the number of contenders as compared to DGR, and uses the modified version of SMAC, the energy consumption per sensor node is more than DGR.



(a)



(b)

Figure 4.10. Energy consumption comparison: (a) Energy expenditure per sensor node (b) Energy expenditure per received frame.

Figure 4.10b presents the energy spent in the network per successful frame delivery which is a metric to assess the efficiency of energy consumption. When we compare LBRF and DGR, we observe that their performances are similar in terms of the energy spent in the network per successful frame delivery. Although energy expenditure of LBRF per sensor node is more than that of DGR, LBRF delivers a higher number of frames balancing the energy expenditure per frame of both schemes. On the other hand, DLBS provides more efficient video delivery as compared to LBRF and DGR, by delivering significantly more frames within the same simulation duration with a better utilization of the energy budget.

5. MULTI-SINK LOAD BALANCED RELIABLE FORWARDING

In a multi-sinked target tracking video sensor network, the important decision is the selection of the destination sink for a frame to be delivered by a sensor node. As in classical multi-sink routing algorithms, each sensor node can be statically assigned to a sink, generally based on distance, and each frame created by a sensor can be delivered to the same sink. However, since the size of a video frame is large, it is highly probable for the network to experience a congestion, the degree of which depends on the minimum frame rate requirement of the target tracking application. In addition, the inherent sensor network characteristics of high node density and multi-hop communication increase the severity of a possible congestion. If multiple targets roam in the sensor field and the majority or all of the targets are closer to one of the sinks, the congestion exacerbates. Hence, instead of static assignment of sensors to the sinks, the sensors should dynamically determine the destination sink in the delivery of a frame by using dynamic criteria along with the static criterion of sensor distance to a sink.

In this chapter, we propose a new cross layer geographic routing scheme called MLBRF (Multi-Sink Load Balanced Reliable Forwarding) which integrates network and MAC layers to provide reliable and energy efficient data delivery in multi-sinked WSNs. MLBRF proposes a dynamic multi-criteria sink selection mechanism which applies fuzzy inference on evaluation of the multi-criteria and the decision of the sink node that a frame to be directed. MLBRF uses LBRF scheme both in the inference system and in the routing of the frames to the decided sinks. In the next section, we will describe the multi-criteria fuzzy sink selection mechanism. Since MLBRF is a geographic routing scheme, it should deal with possible dead-ends caused by connectivity holes. In Section 5.2, we will discuss the modifications required on the routing around dead-ends solution of LBRF for adaptation to multi-sinked network structures.

5.1. Fuzzy Sink Selection Mechanism for MLBRF

In a multi-sinked sensor network, an evaluation and comparison technique is required to decide on the sink to direct the frame. Moreover, since the usage of distance as the only criterion has some drawbacks that causes congestion in the network, the decision mechanism requires additional dynamic criteria that provide inference on the traffic density in the direction of a sink. The number of the nodes that have frames in their buffers and residing in the forward region of a node towards a sink is an indicator for the traffic density. In addition, the buffer occupancy levels of these nodes extend a node's knowledge about the traffic density. Although the neighborhood buffer occupancy knowledge is local, it provides a dynamic and discriminating information to be used in the selection of the sink node.

In order to map different metrics into a single value to be used in the sink decision, we use fuzzy inferencing which has the expertise of modeling the complex real world using rule sets defined by a model similar to the human reasoning [88]. The components of our fuzzy sink selection system are singleton fuzzifier, product inference engine and centroid defuzzifier (Figure 5.1).

Assuming the number of sinks in the network is n, we define a three-dimensional pattern vector for each sink s_i as:

$$\omega_{s_i} = (\alpha_{s_i}, \beta_{s_i}, \gamma_{s_i}), 1 \leq i \leq n \tag{5.1}$$

where α_{s_i} is the number of the nodes that have frames in their buffers in the forward region of a node in the direction of s_i (the number of contenders), β_{s_i} is the buffer occupancy of the candidate node c_{s_i} determined by LBRF considering the frame will be delivered to s_i and γ_{s_i} is the distance of c_{s_i} to s_i.

A sensor extracts α_{s_i} and β_{s_i} information from the BOC entries stored in NIT and kept up-to-date by LBRF reporting mechanism. A sensor examines the forward region in the direction of each s_i and determines a candidate relay c_{s_i} by using LBRF

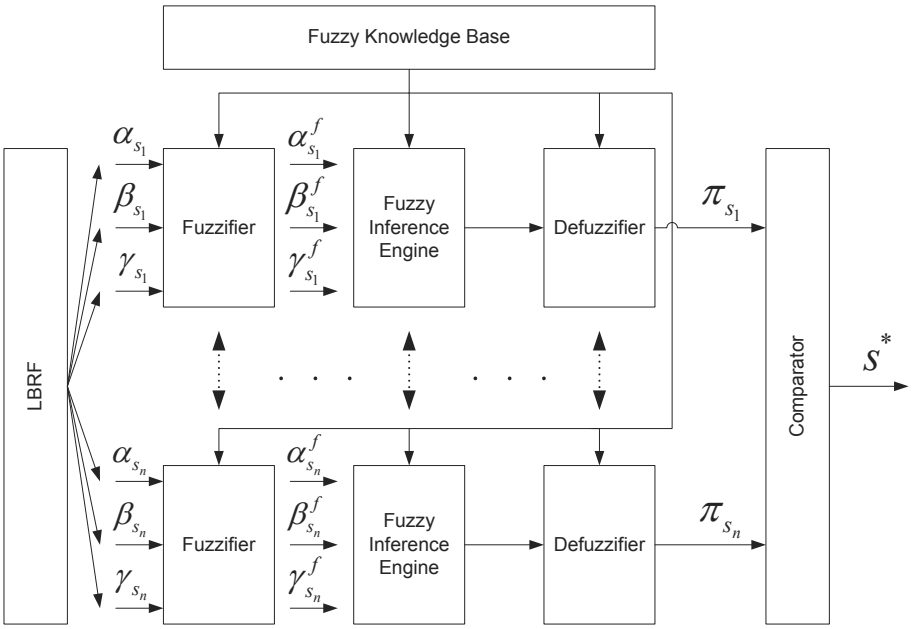


Figure 5.1. Block diagram of the fuzzy sink selection mechanism.

mechanism. Since, MLBRF assumes that the sensors have the location information of their neighbors and the sinks, γ_{s_i} can be calculated.

Our singleton fuzzifier maps each element of this pattern vector to a fuzzy variable using membership functions. The fuzzifier defines three fuzzy classes named LOW (L), MEDIUM (M), and HIGH (H) for each fuzzy variable. Each fuzzy variable is assigned a fuzzy value vector where there is a membership function mapping an element of the pattern vector to an element of the fuzzy value vector for each of the fuzzy classes as:

$$F : (\alpha, \beta, \gamma) \rightarrow (\alpha^f, \beta^f, \gamma^f) \tag{5.2}$$

$$\chi^f = \left[\mu_\chi^L(\chi), \mu_\chi^M(\chi), \mu_\chi^H(\chi)\right] \tag{5.3}$$

$$\mu_\chi^L, \mu_\chi^M, \mu_\chi^H : D(\chi) \rightarrow [0, 1] \tag{5.4}$$

where χ represents the variables α, β and γ and $D(\chi)$ is the domain of the represented variable. The domains of the membership functions are defined as:

$$D(\alpha) = [0, n_{max}],\ D(\beta) = [0, b_{max}],\ D(\gamma) = [0, d_{max}] \tag{5.5}$$

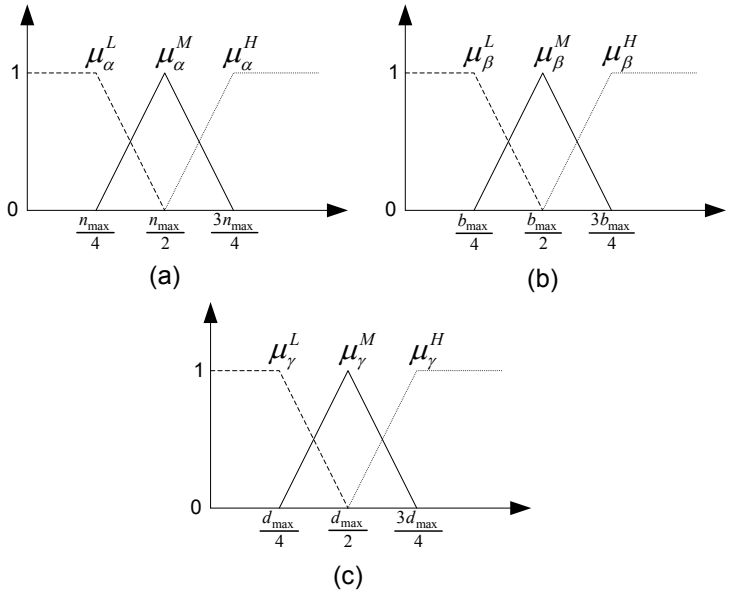


Figure 5.2. Membership functions for the fuzzy variables α, β and γ.

where n_{max}, b_{max} and d_{max} are the maximum of α_{s_i}, β_{s_i} and γ_{s_i} values respectively.

Hence, by using the fuzzifier F, we map the pattern vectors ω_{s_i} to the fuzzy pattern vectors $\omega_{s_i}^f = (\alpha_{s_i}^f, \beta_{s_i}^f, \gamma_{s_i}^f)$, for each sink s_i as

$$F(\alpha_{s_i}, \beta_{s_i}, \gamma_{s_i}) = (\alpha_{s_i}^f, \beta_{s_i}^f, \gamma_{s_i}^f) \tag{5.6}$$

Membership functions of the fuzzifier for the three fuzzy variables are presented in Figure 5.2. The computational complexities of the function calculations are considered in the choice of these fuzzy functions among other alternatives [88]. As an example, if the distance, γ, of the candidate relay in the direction of a sink is 150 m where d_{max}=400 m, then the corresponding γ^f is [0.5,0.5,0] since $\mu_\gamma^L(150)$=0.5, $\mu_\gamma^M(150)$=0.5 and $\mu_\gamma^H(150)$=0 as can be seen in Figure 5.2c.

One of the important components of our fuzzy inference system is the rule base which is designed to mimic the real world behavior. Since there are three fuzzy variables in a fuzzy pattern vector and three fuzzy classes for each fuzzy variable, the rule base contains 3^3=27 different rules. Consequence class f_c^r of an implicative rule r is

determined by a rule evaluation method (REM) where the structure of r in our rule base is as

$$r : \text{If } \alpha_r \text{ is } f_1^r \text{ and } \beta_r \text{ is } f_2^r \text{ and } \gamma_r \text{ is } f_3^r \text{ then output is } f_c^r$$

where each of f_1^r,f_2^r and f_3^r is one of the fuzzy classes L,M or H. The aim of the REM is to determine the output of a rule considering a prioritization between the elements of the pattern vector and the fuzzy classes. This prioritization is performed by employing a weighted voting mechanism. The voting mechanism considers that the effect of the number of contenders is higher than the effects of the buffer occupancy and the distance of the candidate relay in the selection of the destination sink and the weight of α is set as twice as much of β and γ. Hence, the weights of the fuzzy classes for β and γ are set as 3 for L, 2 for M and 1 for H where the corresponding weights for α are set as 6, 4 and 2. This arrangement is beneficial since for each element of the pattern vector lower values of the number of contenders, the buffer occupancy levels and the distances are preferred in the comparison of the routing directions. The output of a rule r, θ_r, is calculated by summing up the weights of f_1^r, f_2^r and f_3^r and represents the current membership degree of the sensor to a sink inferred from that rule. The consequence fuzzy class f_c^r of a rule r is determined by the θ_r value of r.

Table 5.1 shows the rule base of our fuzzy inference system, where the θ column corresponds to the output values of the rules which is calculated using the REM and the c column corresponds to the consequence fuzzy classes determined by the unique values of θ.

In Rule 1, the number of contenders is high, the buffer occupancy level of the candidate relay node is high and the distance of the candidate relay to the sink is high. So, each element of the input pattern vector votes bad for this sink implying the lowest degree of membership of value 4, which is class 1. Similarly, in Rule 27, the number of contenders is low, the buffer occupancy level of the candidate relay node is low and the distance of the candidate relay to the sink is low implying the highest degree of

membership of value 12, which is class 9.

Input of the fuzzy inference engine is the three dimensional fuzzy pattern vector.

Table 5.1. Fuzzy rules in the knowledge base.

Index	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	$\boldsymbol{\gamma}$	$\boldsymbol{c}$	$\boldsymbol{\theta}$
1	H	H	H	1	4
2	H	H	M	2	5
3	H	H	L	3	6
4	H	M	H	2	5
5	H	M	M	3	6
6	H	M	L	4	7
7	H	L	H	3	6
8	H	L	M	4	7
9	H	L	L	5	8
10	M	H	H	3	6
11	M	H	M	4	7
12	M	H	L	5	8
13	M	M	H	4	7
14	M	M	M	5	8
15	M	M	L	6	9
16	M	L	H	5	8
17	M	L	M	6	9
18	M	L	L	7	10
19	L	H	H	5	8
20	L	H	M	6	9
21	L	H	L	7	10
22	L	M	H	6	9
23	L	M	M	7	10
24	L	M	L	8	11
25	L	L	H	7	10
26	L	L	M	8	11
27	L	L	L	9	12

For a new pattern vector, contribution of each rule in the fuzzy rule base is given as:

$$\nu_r = \mu_\alpha^{f_1^r}(\alpha)\mu_\beta^{f_2^r}(\beta)\mu_\gamma^{f_3^r}(\gamma) \tag{5.7}$$

where $\mu_\alpha^{f_1^r}$, $\mu_\beta^{f_2^r}$ and $\mu_\gamma^{f_3^r}$ are the membership functions in the rule r corresponding to the fuzzy classes of the fuzzy variables for α, β and γ respectively. We use the product-inference rule in our fuzzy inference engine, since it allows mathematical simplification in the defuzzification process and is commonly used in practice. Hence,

$$\mu_r = \nu_r \mu_c^{f_c^r}(\theta_r) \tag{5.8}$$

where the membership functions of the consequence fuzzy classes are as in Figure 5.3. Note that the membership function of each consequence fuzzy class is chosen to be a triangle that is symmetric with respect to the abscissa value of its apex which is equal to the θ value of the corresponding consequence class. The rationale behind the design of these fuzzy functions is the low complexity offered in the defuzzification process. The improper selection of fuzzy functions needs high processing capability in the calculation of the output of the centroid defuzzifier. Since the processing capabilities of the sensors are limited, such a design is beneficial both in terms of the processing requirement and the energy efficiency.

Using the rule base in Table 5.1 and the product inference rule, the output of the

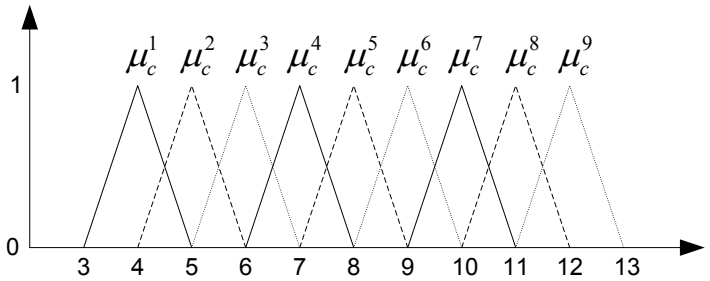


Figure 5.3. Membership functions of the consequence classes.

centroid defuzzifier can be calculated easily as [88]:

$$\pi_{s_i} = \sum_{r=1}^{27} \theta_r \mu_r \Big/ \sum_{r=1}^{27} \mu_r = \sum_{r=1}^{27} \theta_r \nu_r \Big/ \sum_{r=1}^{27} \nu_r \tag{5.9}$$

where θ_r is the is the output of the rule r, which is obtained from the rule base and π_{s_i} is the current membership value of the sensor to the sink s_i.

```
1: procedure MLBRF(N, L_N, l_0, S, L_S,r,s*,relay)
2:     for all s ∈ S do
3:         l_s ← L_S(s)
4:         LBRF(N, L_N, l_0, l_s, r, c_s, α_s, β_s, γ_s)
5:         c_S(s) ← c_s
6:         α(s) ← α_s
7:         β(s) ← β_s
8:         γ(s) ← γ_s
9:     end for
10:    n_max ← max_{s∈S}(α(s))
11:    b_max ← max_{s∈S}(β(s))
12:    c_max ← max_{s∈S}(γ(s))
13:    for all s ∈ S do
14:        Compute π_s using Figure 5.2, Table 5.1, Eqs. (5.7) & (5.9)
15:        π(s) ← π_s
16:    end for
17:    s* ← argmax_{s∈S}(π(s))
18:    relay ← c_S(s*)
19: end procedure
```

N: Neighbor list, L_N: Neighbor locations, l_0: Location of the sensor
S: Sink list, L_S: Sink locations, l_s: Location of the sink s
r: Transmission radius, s^*: Destination sink, $relay$: Relay sensor

Figure 5.4. MLBRF scheme with fuzzy sink selection.

The current membership values π_{s_i} for each sink i are calculated. The destination sink, s^* for the current frame is determined as the one with the highest membership value:

$$s^* = \underset{s_i \in \{s_1,\ldots,s_n\}}{\operatorname{argmax}} (\pi_{s_i}) \tag{5.10}$$

In the cases where multiple highest membership values exist, distances to the sinks are used to break the tie.

The destination sink for a frame is determined only in the source sensor after its creation and is not altered along the way towards the destination. However, for the cases where the frame cannot be routed in the direction of the destination sink due to non-existence of a closer neighbor (dead-end problem), the destination sink for the frame is reevaluated (Section 5.2). In addition, if a source sensor is a neighbor of a sink, fuzzy sink selection mechanism is not processed and the frame is directed to the neighbor sink.

In Figure 5.5, a sample scenario is presented where a sensor has to determine the destination sink for a frame among four candidate sinks. Since in the forward region, F, of the sensor towards Sink1, there are 2 nodes that have frames in their buffers, α_{s_1} is determined as 2. LBRF scheme determines the sensor with 20 packets in its buffer and a distance of 300 m to Sink1 as the relay for that direction. Hence β_{s_1} and γ_{s_1} are determined as 20 and 300 respectively (Figure 5.5a). Note that the domain parameters n_{max}, b_{max} and d_{max} of the membership functions for α, β, and γ are 4, 30 and 300 respectively. With these parameters the output of the defuzzifier for each sink are calculated as π_{s_1}=6.33, π_{s_2}=5.66, π_{s_3}=5.66, π_{s_4}=4.33. Hence the destination sink for the frame is determined as Sink1.

5.2. Routing Around Dead-Ends in MLBRF

In this section, we will discuss the modifications required on the routing around dead-ends solution of LBRF for adaptation to multi-sinked network structures in Sec-

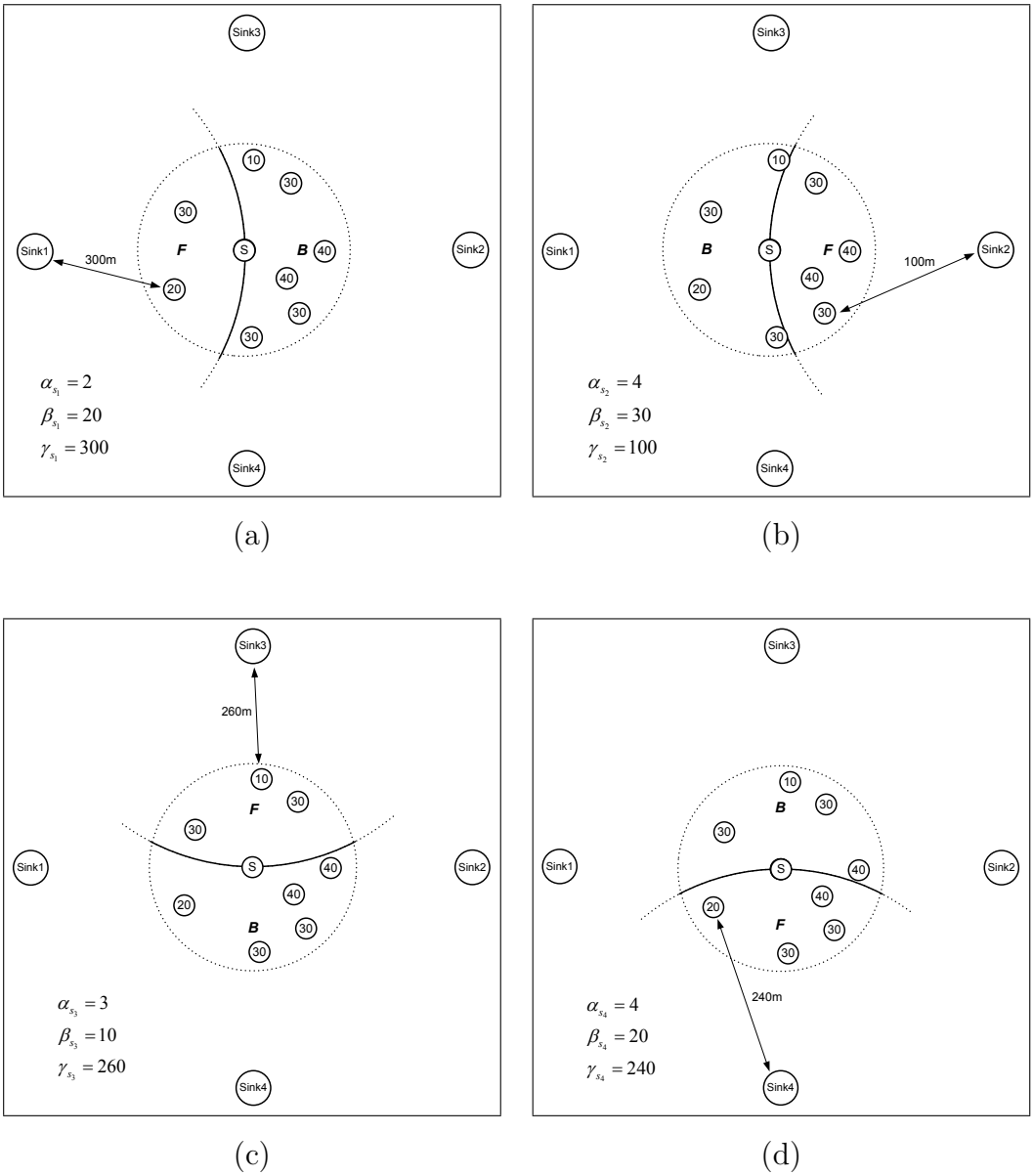


Figure 5.5. Illustration of the determination of α, β, γ in a sample scenario.

tion 3.2. In a multi-sinked network structure, a sensor's routing-class information is formed into an n-tuple as $(\delta_{s_1},\cdots,\delta_{s_n})$ where each δ_{s_i} is the routing-class of the sensor with respect to the sink i. As in the single sink case, each sensor piggy-backs its routing-class tuple to each MAC layer packet and the elements of the own routing-class tuple of a sensor is determined according to the gathered routing-class tuples from the neighbors which are stored in NIT. In the initial phase, all sensors that have neighbors in F in the direction of s_i sets their δ_{s_i} as C_1 and all sensors that do not have neighbors in F in the direction of s_i sets their δ_{s_i} as C_2. The sink s_i sets its δ_{s_i} as C_1, and sets its δ_{s_j} values where $j \neq i$ as C_0. The fuzzy sink selection mechanism determines the

destination sink among the ones which has the corresponding δ_{s_i} set as C_1. In the case where none of the δ_{s_i} is C_1, the fuzzy sink selection mechanism considers the sink directions where the corresponding δ_{s_i} values are C_2. In the same way, the sink selection mechanism may switch to directions with the succeeding higher class, when there is no direction with the current class.

Note that the destination sink for a frame is not altered along the way as stated in Section 5.1. However, in a case where the corresponding δ_{s_i} of a sensor for a frame changes before routing the frame, the destination sink of that frame is re-evaluated and updated accordingly.

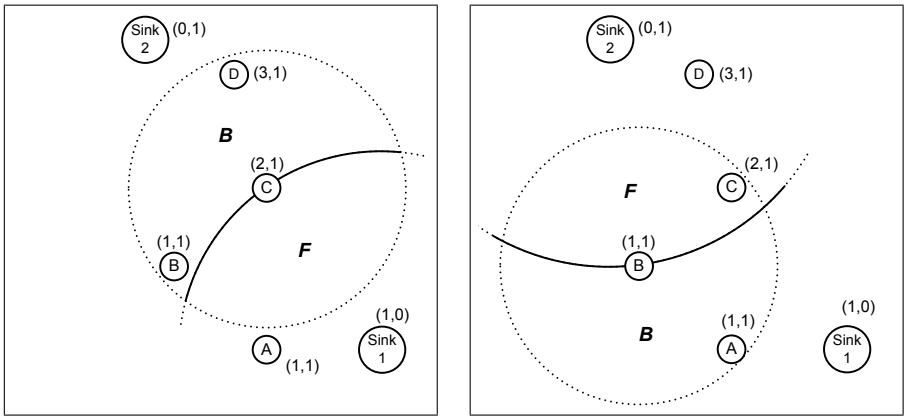


Figure 5.6. Illustration of the routing classes of MLBRF in a sample scenario.

In Figure 5.6, the routing classes in MLBRF are illustrated in a sample scenario. C has no neighbors with δ_{s_1} as C_1 in its forward region towards Sink 1. It determines its δ_{s_1} as C_2, since B with its δ_{s_1} as C_1 resides in its backward region. D has no neighbors with δ_{s_1} as C_1 in its forward region and has no C_1 or C_2 neighbors in its backward region. It determines its δ_{s_1} as C_3, since C with its δ_{s_1} as C_2 resides in its forward region (Figure 5.6a). On the other hand, B determines its δ_{s_2} as C_1, since C with its δ_{s_2} as C_1 resides in its forward region towards Sink 2 (Figure 5.6b). In this scenario, as B considers both of the sinks in fuzzy sink selection, C only considers Sink 2.

5.3. Discussion on the Cost of MLBRF

As in the case of LBRF and DLBS, the operation of MLBRF is based on the information stored in NIT with space complexity $O(N^2)$ for small N and $O(N)$ for large N, where N is the number of neighbors of a node. The pseudo-code of MLBRF scheme with fuzzy sink selection given in Figure 5.4 will be used in the determination of the computational complexity of MLBRF. We can observe that the operation of MLBRF contain two parts. In the first part the input parameters for the fuzzy inference system and the candidate nodes in each sink direction are determined using LBRF relay determination mechanism. Hence, since the number of sinks in an environment is not large, the computational complexity of the first part is $O(N)$. In the second part the fuzzy inference system is executed. The fuzzification and the deffuzzification operations of the inference system requires many computations due to the number of rules in the knowledge base and the number of fuzzy variables. Although, we decrease the number of computations in the deffuzzification process by using symmetric triangle output membership functions, the computational complexity of the fuzzy inference system is $O(N^2)$. Hence the overall complexity of MLBRF scheme with fuzzy sink selection is $O(N^2)$. As compared to the time complexity of LBRF which is $O(N)$, the computational overhead of MLBRF is higher on the sensors. Similarly MLBRF requires higher computation as compared to DLBS in the case of small N. On the other hand, MLBRF and DLBS have similar computational overhead when the number of neighbors of a node is large. The space complexity of MLBRF scheme is $O(N^2)$ due to the memory requirements to store the rules and the membership functions in the knowledge base.

The overhead of MLBRF on the network is not considerable. MLBRF uses no extra control packet for its operation. It piggybacks routing class information with respect to each sink to each MAC layer packet in addition to the information piggybacked by LBRF which are buffer occupancy condition and routing class information with respect to a single sink. As compared to LBRF the length of the packets is longer in MLBRF but this is an insignificant increase in the length as compared to the length of a data packet.

5.4. Performance Evaluation of MLBRF with the Fuzzy Sink Selection Mechanism

We evaluate the performance of the fuzzy sink selection mechanism by comparing it with three other selection mechanisms via simulations using OPNET [80] in a target tracking video application scenario. In the first selection mechanism, the destination sink is determined as the closest one by the source sensor. In the second one, the destination sink is determined randomly. The third mechanism applies a distance-based random sink selection procedure and we call it as DB-Random (Figure 5.7). In this procedure, the reciprocal of the distances to each sink are stored in D_S^-. The membership degree of the sensor in the direction of s, $MD(s)$, is the normalization of $D_S^-(s)$ with respect to $\sum_{s \in S} D_S^-(s)$ where $\sum_{s \in S} MD(s) = 1$. The destination sink is determined randomly using the membership degrees as the weights of the sinks in the random selection, where the closest sink achieves the highest weight. As in the fuzzy case, the destination sink is determined only in the source sensor and LBRF mechanism is used in the delivery to the sink without altering the destination along the way.

In the simulations, 200 video sensors are uniformly deployed in a square shaped environment of each side 400 m. There are four targets in the environment moving according to the random waypoint mobility model where the target speeds are 3 m/s and the pause times are 0 seconds. We varied the number of sinks in the environment between 1 and 4 where the locations of the sinks are (0,200), (400,200), (200,0), and (200,400). We evaluated each scenario for 10 different network topologies with corresponding 10 different target trajectories. The values presented in this section are the mean values of the results obtained in these experiments. Simulation parameters are determined as presented in Table 3.2 in Chapter 3.

Performance of MLBRF with the fuzzy sink selection mechanism in terms of frame delivery, frame latency and energy efficiency is presented in Figure 5.8, Figure 5.9 and Figure 5.10 respectively. The single sink case is given for the purpose of comparison and in this case the operation of MLBRF is equivalent to LBRF. We observe from Figure 5.8 that when we deploy the second sink into the environment, we get a significant

```
1: procedure DB-RANDOM(L_S, destination)
2:     D^-_sum ← 0.0
3:     for all s ∈ S do
4:         D^-_S(s) ← 1 / L_S(s)
5:         D^-_sum ← D^-_sum + D^-_S(s)
6:     end for
7:     RAND[0,1](rand)
8:     MD_sum ← 0.0
9:     for all s ∈ S do
10:        MD(s) ← D^-_S(s) / D^-_sum
11:        MD_sum ← MD_sum + MD(s)
12:        if rand ≤ MD_sum then
13:            destination ← s
14:            break
15:        end if
16:    end for
17: end procedure

L_S: Sink locations
```

Figure 5.7. Distance based random sink selection mechanism.

performance gain where the frame delivery ratio increases from 0.62 to 0.81 in the 6 fps case. The gain is more remarkable in the frame latency performance where the mean frame delay decreases to 3.6 sec from 27.3 sec in the 6 fps case (Figure 5.9). As the number of sinks in the environment is increased, the load directed to each sink is balanced and hence the congestion around the sinks is alleviated. Although, deploying extra sinks into the environment increases the performance in terms of reliability and latency metrics, the breakpoint is the deployment of the second sink. The budget for the sensor network and the QoS requirements of the application are the determining factors for the number of sinks in the environment.

The energy efficiency of MLBRF is evaluated in terms of average energy expenditure in the network per successfully delivered frame to the sink (Figure 5.10). The

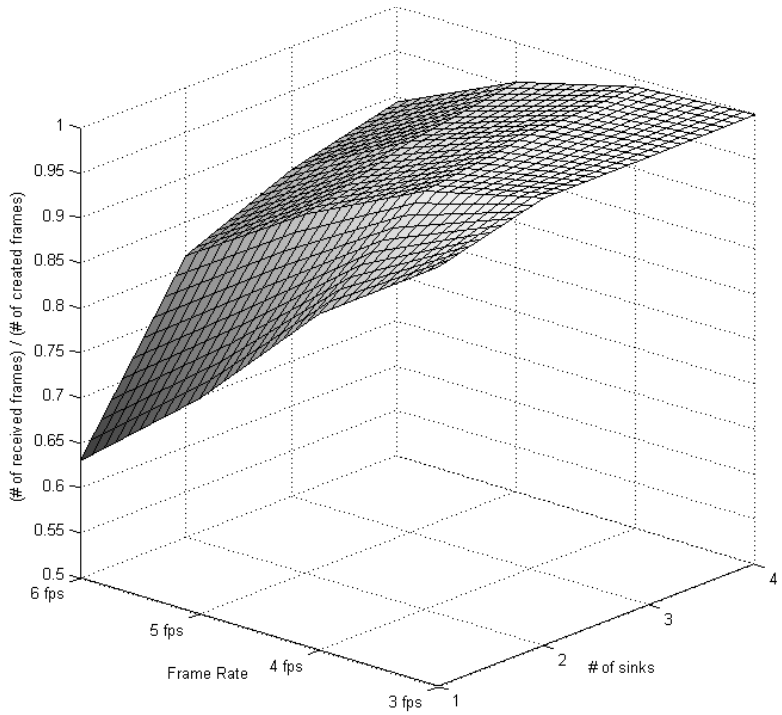


Figure 5.8. Frame delivery performance of fuzzy sink selection.

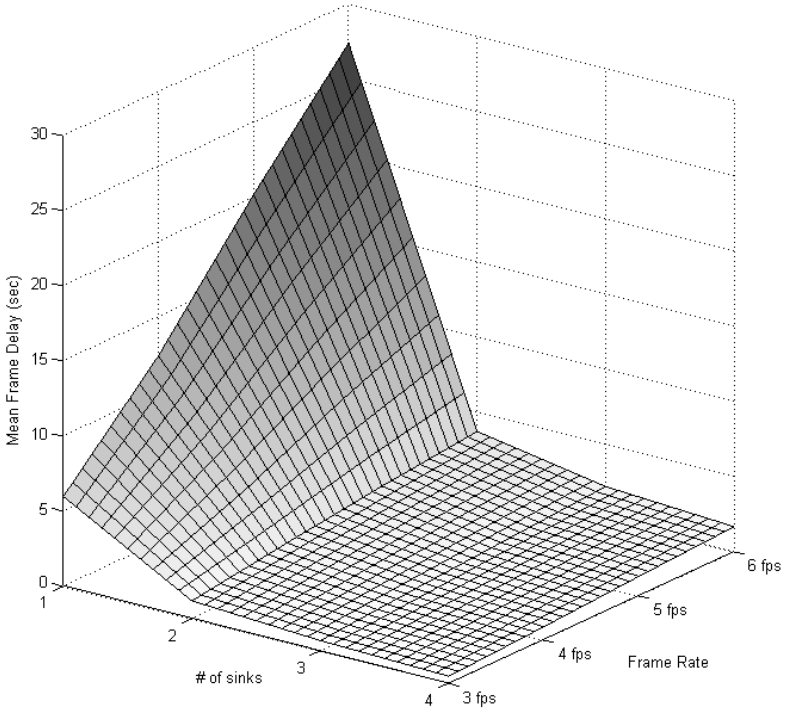


Figure 5.9. Frame latency performance of fuzzy sink selection.

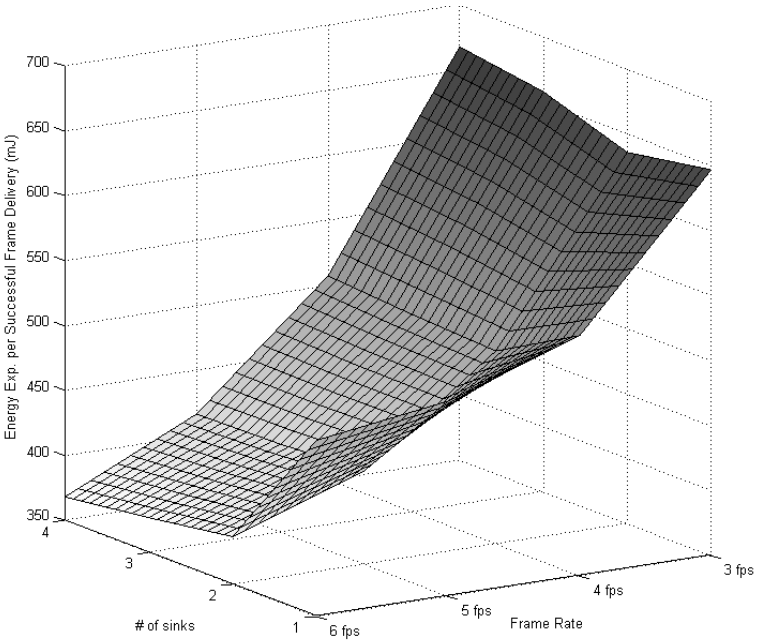


Figure 5.10. Energy efficiency performance of fuzzy sink selection.

energy expenditure characteristics in the single-sink case provides the clues for the behavior in the multi-sink case. The general behavior in the single sink case is that the per frame energy expenditure in the network decreases as the video quality increases. This interesting result, which is also encountered in the performance evaluation of LBRF in Chapter 3, is related with the reduced idle listening duration due to the overhearing avoidance mechanism of SMAC which enables the overhearing sensors to sleep earlier than the actual sleep time [87]. The nodes spend more energy in idle cycles since every node has to sense the channel throughout the predetermined listen duration. However, in a successful transmission cycle or a collision cycle, nodes can go to sleep upon hearing the RTS or CTS packets from other nodes.

In Figure 5.10 we observe that, the energy expenditure per frame decreases when the second sink is deployed into the environment for the 3 fps and 4 fps video qualities. Deployment of extra sinks increase the energy expenditure. On the other hand, for the 5 fps and 6 fps video qualities, the behavior of the energy expenditure is always decreasing with the increased number of sinks in the environment. In the former case, deployment of the second sink decreases the number of idle cycles by dividing the load among the sinks. Further splitting the load by the deployment of extra sinks results in

lighter load conditions which lead to an increase in the number of idle cycles. Hence, the energy expenditure per frame increases due to increased idle listening. In the latter case, splitting the load increases the number of frames survived in the network more than in the previous case (Figure 5.8). Hence, splitting the load causes a sufficiently heavy load distribution throughout the network to decrease the effect of idle listening on energy expenditure per frame.

The performance of the sink selection mechanisms are compared in terms of frame delivery (Figure 5.11), frame latency (Figure 5.12) and energy efficiency (Figure 5.13). The results in these figures are obtained for the case where there are 4 sinks in the environment. We observe in Figure 5.11 and Figure 5.12 that by using the closest sink selection mechanism, MLBRF operates with the worst performance in terms of reliability and latency metrics. In this selection scheme, a sensor is statically assigned to the closest sink and each created frame by this sensor is delivered to that sink without considering the traffic conditions. If majority or all of the targets are closer to one of the sinks, the detecting sensors direct their frames to that sink creating congestion

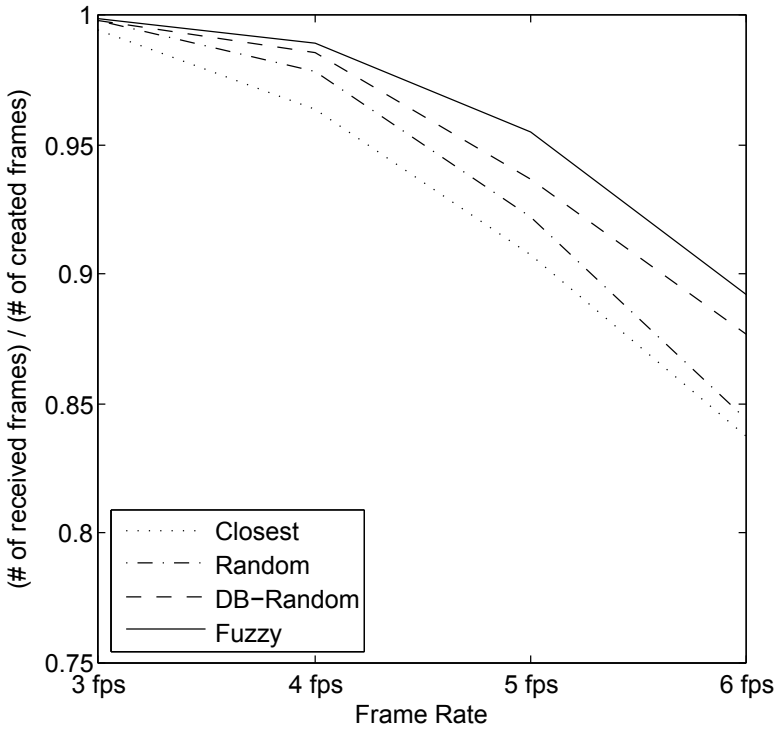


Figure 5.11. Frame delivery performance comparison of the selection schemes.

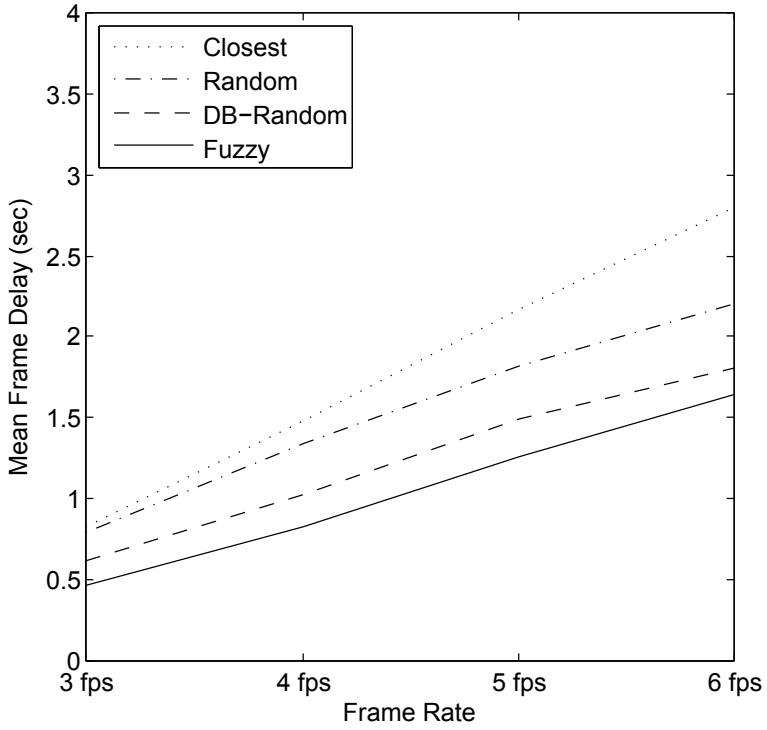


Figure 5.12. Frame latency performance comparison of the selection schemes.

and buffer overflows. The static closest sink selection scheme could not balance the dynamically created load among the multiple sinks.

The random sink selection scheme tries to distribute the created load among the sinks by directing each created frame to a randomly determined sink. The performance of the random sink selection is better than the closest sink selection. However, since it does not consider the distances to the sinks, there may be cases where a sensor closest to a sink directs some of its frames to the farthest sink even in the lightest load conditions towards the closest sink. In such cases, latency and energy expenditure are unnecessarily increased due to usage of longer paths.

The DB-Random sink selection is an improvement on balancing the load among multiple sinks which considers the distances of the detecting sensors to the sinks while performing random selection. We observe from Figure 5.11 and Figure 5.12, that the drawback of the random sink selection is alleviated by DB-Random resulting with better performance. However, although DB-Random provides more balanced load

distribution, it does not consider the load conditions while directing the created frames to the sinks.

We observe that, by using the fuzzy sink selection mechanism, MLBRF achieves the best performance in terms of reliability and latency metrics. The success of the fuzzy sink selection mechanism depends on the provision of an even distribution of the dynamic load among the sinks, by combining the distance criterion with the multiple criteria indicating the congestion level in the neighborhood towards a sink.

In the case of energy efficiency, it is interesting that the closest sink selection scheme has the worst performance (Figure 5.13). Although performance of other schemes are similar, as the video quality increased in the network, the fuzzy sink selection increases its performance and achieves the best performance as compared to the other selection mechanisms. Since the fuzzy selection distributes the load more evenly among the sinks, it provides lighter load conditions in each direction as compared to the other selection schemes for the 3 fps and 4 fps video qualities. Hence,

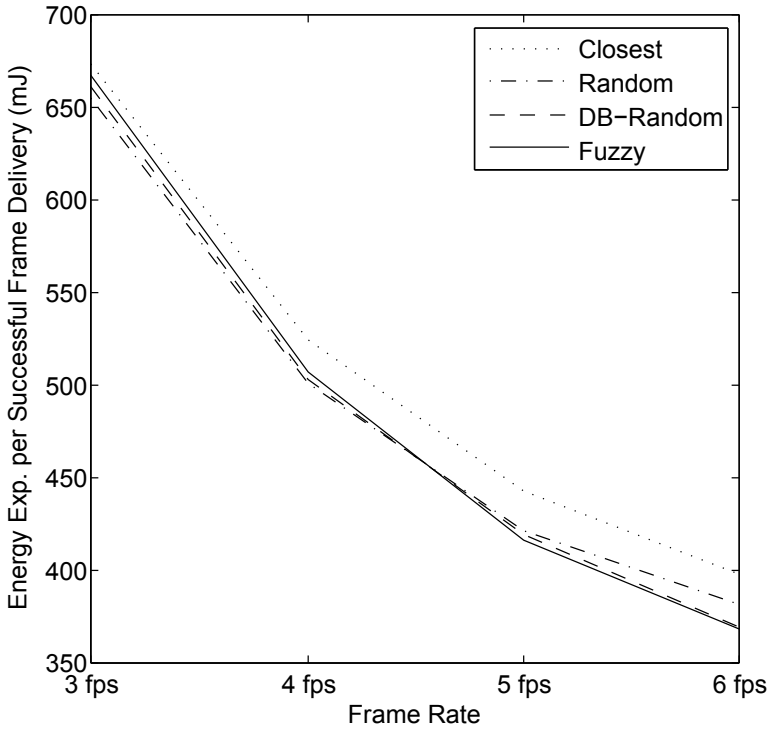


Figure 5.13. Energy efficiency performance comparison of the selection schemes.

due to increased number of idle listening cycles throughout the network, the fuzzy sink selection does not provide the most energy efficient frame delivery for these video qualities. However, for the 5 fps and 6 fps video qualities, the number of frames survived in the network is more than in the previous case by using the fuzzy selection instead of the random selection schemes (Figure 5.11). As a consequence, the increased load in the network decreases the energy expenditure per frame due to decreased effect of idle listening as in Figure 5.10.

6. CONCLUSIONS

Congestion is a major source of data loss in VSNs which adversely affects the reliable data delivery and consequently the success of the video application. In this thesis, we explored distributed and dynamic load balancing techniques to avoid congestion and to provide reliable data delivery in VSNs. We proposed three cross-layer geographic routing schemes LBRF, DLBS and MLBRF which apply three different load balancing techniques.

LBRF scheme introduced the local load balancing technique. In event triggered video applications, regarding the dense deployment of video sensors, many sensors detect the same event and create bursts of video streams destined to the sink which create transient local congestions in the network. Moreover, shortest distance based classical single path routing schemes increase the severity of the congestion by concentrating the traffic on some preferred sensors, resulting with an unbalanced traffic distribution in the network. The main effort in local load balancing is the provision of even load distribution among the candidate neighbors in the direction of delivery. The load balance is achieved in the determination of a relay for a frame using monitored buffer occupancy levels of the neighbors.

LBRF provides load balancing just in the direction of the sink, whereas DLBS routes the frames and provides load balancing in multiple directions which later on converge to the sink. DLBS provides two-folded load balancing, spatial and local, by first partitioning a video stream into multiple parts and delivering the parts over multiple directions in a round robin manner and then by providing a load balance in the direction of the delivery.

Although LBRF and DLBS provide substantial improvements in terms of reliability and energy efficiency, these improvements are limited due to convergecast communication pattern directed to a single sink. Hence, we explored load distribution to multiple sinks as another technique for load balancing in VSNs. Deployment

of multiple sinks is a promising solution in terms of reliability, latency and energy efficiency due to the alleviation of congestion around the sinks. In applications such as target tracking where the load creation is dynamic throughout the network, static load balancing techniques could not effectively distribute the load to multiple sinks. A dynamic load balancing technique is the appropriate solution for the applications where the data creation is triggered upon event detection. MLBRF employs a dynamic multi-criteria sink selection mechanism which applies fuzzy inference in the decision of the destination sink for a frame to combine local indicators of the congestion level in the neighborhood towards a sink with the distance criterion. Deploying extra sinks into the environment increases the performance of the network where the breakpoint is the deployment of the second sink. Hence, the number of sinks to be deployed into the environment should be determined by the budget for the sensor network and the QoS requirements of the application. A surprising outcome of comparative performance evaluation of different sink selection schemes is that even directing the load to randomly determined sinks without considering the distance is superior than directing the load to the closest sink. The comparative results also confirm the success of the fuzzy sink selection mechanism for achieving the best performance in terms of reliability, latency and energy efficiency for high video qualities.

In the future, we want to improve the local load balancing technique to consider both the sender's and the receiver's load conditions and the delivery characteristics which require sender and receiver oriented relay determination approaches. In addition, we want to improve the class based approach for dead-end handling to account for the quality of a class of a sensor which will be based on the number of lower class neighbors in the routing direction for that class. We will accumulate and aggregate this information as moving away from the sink. Using this information, a sensor can discriminate the neighbors of the same class. In that way, we can decrease the volume of the traffic passing through the bottleneck regions in the network, which consequently increases the QoS of data delivery.

Related with the multi-sink study, we want to improve the routing scheme by employing a dynamic sink-switching mechanism along the way considering the deci-

sion history and the current load conditions towards each sink. In addition, we aim to tailor MLBRF routing scheme for forest fire scenarios where sensor and communication destructions are inevitable due to extremely high temperatures. We want to explore the advantages of deploying multiple sinks in the forest fire scenarios and we aim to increase the fire spread prediction quality by proposing fire aware cross layer mechanisms. Moreover, we want to explore the benefits of applying load balanced routing to multiple mobile sinks in terms of reliable and energy efficient data delivery.

REFERENCES

1. Yick, J., B. Mukherjee and D. Ghosal, "Wireless Sensor Network Survey", *Computer Networks*, Vol. 52, No. 12, pp. 2292 – 2330, 2008.

2. Akyildiz, I., T. Melodia and K. R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks", *Computer Networks*, Vol. 51, No. 4, pp. 921–960, 2007.

3. Rahimi, M., R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin and M. Srivastava, "Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks", *Proceedings of the International Conference on Embedded Networked Sensor Systems*, pp. 192–204, 2005.

4. Wang, Y., S. Wenger, W. Jiantao and A. K. Katsaggelos, "Error Resilient Video Coding Techniques", *IEEE Signal Processing Magazine*, Vol. 17, No. 4, pp. 61–82, 2000.

5. Hsiao, P., A. Hwang, H. T. Kung and D. Vlah, "Load-Balancing Routing for Wireless Access Networks", *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 2, pp. 986–995, 2001.

6. Dai, H. and R. Han, "A Node-Centric Load Balancing Algorithm for Wireless Sensor Networks", *Proceedings of the IEEE Global Telecommunications Conference*, Vol. 1, pp. 548–552, 2003.

7. Zhang, L., "Load-Balanced Short-Path Routing in Wireless Networks", *IEEE Transactions on Parallel Distributed Systems*, Vol. 17, No. 4, pp. 377–388, 2006.

8. Yan, T., Y. Bi, L. Sun and H. Zhu, "Probability Based Dynamic Load-Balancing Tree Algorithm for Wireless Sensor Networks", *Proceedings of the ICC Networking and Mobile Computing: LNCS 3619*, pp. 682–691, 2005.

9. Fyffe, M., S. Min-Te and M. Xiaoli, "Traffic-Adapted Load Balancing in Sensor Networks Employing Geographic Routing", *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 4389–4394, 2007.

10. Akkaya, K. and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks", *Ad Hoc Networks*, Vol. 3, No. 3, pp. 325–349, 2005.

11. Al-Karaki, J. N. and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey", *IEEE Wireless Communications*, Vol. 11, No. 6, pp. 6–28, 2004.

12. Savvides, A., M. Srivastava, L. Girod and D. Estrin, *Localization in Sensor Networks*, pp. 327–349, Kluwer Academic Publishers, Norwell, MA, USA, 2004.

13. Mao, G., B. Fidan and B. D. Anderson, "Wireless Sensor Network Localization Techniques", *Computer Networks*, Vol. 51, No. 10, pp. 2529 – 2553, 2007.

14. Mauve, M., A. Widmer and H. Hartenstein, "A Survey on Position-Based Routing in Mobile Ad Hoc Networks", *IEEE Network*, Vol. 15, No. 6, pp. 30–39, 2001.

15. Karp, B. and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pp. 243–254, ACM Press, 2000.

16. Giordano, S., I. Stojmenovic and L. Blazevic, "Position-Based Routing Algorithms for AdHoc Networks: A Taxonomy", X. Cheng, X. Huang and D.-Z. Du (Editors), *Ad Hoc Wireless Networking*, Kluwer Academic Publishers, Boston, 2003.

17. Chen, M., V. C. M. Leung, S. Mao and Y. Yuan, "Directional Geographical Routing for Real-Time Video Communications in Wireless Sensor Networks", *Computer Communications*, Vol. 30, No. 17, pp. 3368–3383, 2007.

18. Chen, S. and N. Yang, "Congestion Avoidance Based on Lightweight Buffer Management in Sensor Networks", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 17, No. 9, pp. 934–946, 2006.

19. Akan, O. B. and I. F. Akyildiz, "Event-to-Sink Reliable Transport in Wireless Sensor Networks", *IEEE/ACM Transactions on Networking*, Vol. 13, No. 5, pp. 1003–1016, 2005.

20. Wan, C., S. Eisenman and A. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks", *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 266–279, Los Angeles, California, USA, 2003.

21. Hull, B., K. Jamieson and H. Balakrishnan, "Mitigating Congestion in Wireless Sensor Networks", *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 134–147, Baltimore, MD, USA, 2004.

22. Ee, C. T. and R. Bajcsy, "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks", *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 148–161, Baltimore, MD, USA, 2004.

23. Rangwala, S., R. Gummadi, R. Govindan and K. Psounis, "Interference-Aware Fair Rate Control in Wireless Sensor Networks", *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 63–74, Pisa, Italy, 2006.

24. Wang, C., B. Li, K. Sohraby, M. Daneshmand and Y. Hu, "Upstream Congestion Control in Wireless Sensor Networks Through Cross-Layer Optimization", *IEEE Journal on Selected Areas in Communications*, Vol. 25, No. 4, pp. 786–795, 2007.

25. Mamun-Or-Rashid, M., M. M. Alam, M. A. Razzaque and C. S. Hong, "Congestion Avoidance and Fair Event Detection in Wireless Sensor Network", *IEICE Transactions on Communications*, Vol. E90-B, No. 12, pp. 3362–3372, 2007.

26. Ahmad, M. Z. and D. Turgut, "Congestion Avoidance and Fairness in Wireless Sensor Networks", *Proceedings of the IEEE Global Telecommunications Conference*, pp. 1–6, 2008.

27. Zhang, Z. and G. Cui, "An Effective Congestion Avoidance Altering Routing Protocol in Sensor Networks", *Proceedings of the International Conference on Computer Science and Software Engineering*, Vol. 4, pp. 980–983, 2008.

28. Kang, J., Y. Zhang and B. Nath, "Adaptive Resource Control Scheme to Alleviate Congestion in Sensor Networks", *Proceedings of the First Workshop on Broadband Advanced Sensor Networks*, San Jose, CA, 2004.

29. Sergiou, C., V. Vassiliou and A. Pitsillides, "Reliable Data Transmission in Event-Based Sensor Networks During Overload Situation", *Proceedings of the 3rd International Conference on Wireless Internet*, 2007.

30. Khan, M. I., W. N. Gansterer and G. Haring, "Congestion Avoidance and Energy Efficient Routing Protocol for Wireless Sensor Networks with a Mobile Sink", *Journal of Networks*, Vol. 2, No. 6, p. 42 49, 2007.

31. Karenos, K. and V. Kalogeraki, "Facilitating Congestion Avoidance in Sensor Networks with a Mobile Sink", *Proceedings of the 28th IEEE International Real-Time Systems Symposium*, pp. 321–332, 2007.

32. Toh, C. K., A.-N. Le and Y.-Z. Cho, "Load Balanced Routing Protocols for Ad Hoc Mobile Wireless Networks", *IEEE Communications Magazine*, Vol. 47, No. 8, pp. 78–84, 2009.

33. Puccinelli, D. and M. Haenggi, "Lifetime Benefits Through Load Balancing in Homogeneous Sensor Networks", *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 1–6, 2009.

34. Toumpis, S. and S. Gitzenis, "Load Balancing in Wireless Sensor Networks using Kirchhoffs Voltage Law", *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies*, Rio de Janeiro, Brazil, 2009.

35. Casari, P., M. Nati, C. Petrioli and M. Zorzi, "ALBA: An Adaptive Load - Balanced

Algorithm for Geographic Forwarding in Wireless Sensor Networks", *Proceedings of the IEEE Military Communications Conference*, pp. 1–9, 2006.

36. Chen, M., V. C. M. Leung, S. Mao and T. Kwon, "Receiver-Oriented Load-Balancing and Reliable Routing in Wireless Sensor Networks", *Wireless Communications and Mobile Computing*, Vol. 9, No. 3, pp. 405–416, 2009.

37. Maimour, M., C. Pham and J. Amelot, "Load Repartition for Congestion Control in Multimedia Wireless Sensor Networks with Multipath Routing", *Proceedings of the 3rd International Symposium on Wireless Pervasive Computing*, pp. 11–15, 2008.

38. Oyman, E. and C. Ersoy, "Multiple Sink Network Design Problem in Large Scale Wireless Sensor Networks", *Proceedings of the IEEE International Conference on Communications*, pp. 3663–3667, 2004.

39. Kim, H., Y. Seok, N. Choi, Y. Choi and T. Kwon, "Optimal Multi-sink Positioning and Energy-Efficient Routing in Wireless SensorNetworks", C. Kim (Editor), *Information Networking*, Vol. 3391 of *Lecture Notes in Computer Science*, pp. 264–274, 2005.

40. Vincze, Z., R. Vida and A. Vidacs, "Deploying Multiple Sinks in Multi-Hop Wireless Sensor Networks", *Proceedings of the IEEE International Conference on Pervasive Services*, pp. 55–63, 2007.

41. Dai, S., C. Tang, S. Qiao, K. Xu, H. Li and J. Zhu, "Optimal Multiple Sink Nodes Deployment in Wireless Sensor Networks Based on Gene Expression Programming", *Proceedings of the International Conference on Communication Software and Networks*, pp. 355–359, 2010.

42. Erman, A., T. Mutter, L. van Hoesel and P. Havinga, "A Cross-Layered Communication Protocol for Load Balancing in Large Scale Multi-Sink Wireless Sensor Networks", *Proceedings of the International Symposium on Autonomous Decen-*

tralized Systems, pp. 1–8, 2009.

43. Wang, C. and W. Wu, "A Load-Balance Routing Algorithm for Multi-Sink Wireless Sensor Networks", *Proceedings of the International Conference on Communication Software and Networks*, pp. 380–384, 2009.

44. Eghbali, A., N. Javan, A. Dareshoorzadeh and M. Dehghan, "An Energy Efficient Load-Balanced Multi-Sink Routing Protocol for Wireless Sensor Networks", *Proceedings of the International Conference on Telecommunications*, pp. 229–234, 2009.

45. Intanagonwiwat, C., R. Govindan, D. Estrin, J. Heidemann and F. Silva, "Directed Diffusion for Wireless Sensor Networking", *IEEE/ACM Transactions on Networking*, Vol. 11, pp. 2–16, 2003.

46. Cheng, R.-H., S.-Y. Peng and C. Huang, "A Gradient-Based Dynamic Load Balance Data Forwarding Method for Multi-sink Wireless Sensor Networks", *Proceedings of the IEEE Asia-Pacific Services Computing Conference*, pp. 1132–1137, 2008.

47. Yoo, H., M. Shim, D. Kim and K. H. Kim, "GLOBAL: A Gradient-Based Routing Protocol for Load-Balancing in Large-Scale Wireless Sensor Networks with Multiple Sinks", *Proceedings of the IEEE Symposium on Computers and Communications*, pp. 556–562, 2010.

48. Paone, M., L. Paladina, M. Scarpa and A. Puliafito, "A Multi-Sink Swarm-Based Routing Protocol for Wireless Sensor Networks", *Proceedings of the IEEE Symposium on Computers and Communications*, pp. 28–33, 2009.

49. Wang, W., W. Li, D. Chen and Y. Han, "Ant Colony Based Routing Algorithm for Multi-sink Networks", *Proceedings of the WRI World Congress on Computer Science and Information Engineering*, pp. 423–429, 2009.

50. Chen, Y., E. Chan and S. Han, "Energy Efficient Multipath Routing in Large Scale Sensor Networks with Multiple Sink Nodes", J. Cao, W. Nejdl and M. Xu (Editors), *Advanced Parallel Processing Technologies*, Vol. 3756 of *Lecture Notes in Computer Science*, pp. 390–399, Springer Berlin / Heidelberg, 2005.

51. Egorova-Forster, A. and A. Murphy, "Exploiting Reinforcement Learning for Multiple Sink Routing in WSNs", *Proceedings of the IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, pp. 1–3, 2007.

52. Khalid., M. Y., Javed and M. Y., "NSN Based Multi-Sink Minimum Delay Energy Efficient Routing in Wireless Sensor Networks", *European Journal of Scientific Research*, Vol. 41, No. 2, pp. 399–411, 2010.

53. Kalantari, M. and M. Shayman, "Design Optimization of Multi-Sink Sensor Networks by Analogy to Electrostatic Theory", *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 431–438, 2006.

54. Kalantari, M. and M. Shayman, "Routing in Multi-Commodity Sensor Networks Based on Partial Differential Equations", *Proceedings of the Conference on Information Sciences and Systems*, pp. 402–406, 2006.

55. Li, J., S. Ji, H. Jin and Q. Ren, "Routing in Multi-Sink Sensor Networks Based on Gravitational Field", *Proceedings of the International Conference on Embedded Software and Systems*, pp. 368–375, 2008.

56. Liu, H., Z.-L. Zhang, J. Srivastava and V. Firoiu, "PWave: A Multi-Source Multi-Sink Anycast Routing Framework for Wireless Sensor Networks", I. Akyildiz, R. Sivakumar, E. Ekici, J. Oliveira and J. McNair (Editors), *Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet Networking*, Vol. 4479 of *Lecture Notes in Computer Science*, pp. 179–190, Springer Berlin / Heidelberg, 2007.

57. Ciciriello, P., L. Mottola and G. P. Picco, "Proceedings of the Efficient Routing

from Multiple Sources to Multiple Sinks in Wireless Sensor Networks", *European Conference on Wireless Sensor Networks*, pp. 34–50, 2007.

58. Shah-Mansouri, V., A.-H. Mohsenian-Rad and V. Wong, "Lexicographically Optimal Routing for Wireless Sensor Networks With Multiple Sinks", *IEEE Transactions on Vehicular Technology*, Vol. 58, No. 3, pp. 1490–1500, 2009.

59. Mottola, L. and G. P. Picco, "MUSTER: Adaptive Energy-Aware Multi-Sink Routing in Wireless Sensor Networks", *IEEE Transactions on Mobile Computing*, Vol. 10, No. 12, pp. 1694–1709, 2011.

60. Odorizzi, A. and G. Mazzini, "M-GeRaf: A Reliable Random Forwarding Geographic Routing Protocol in Multisink Ad Hoc and Sensor Networks", *Proceedings of the IEEE International Symposium on Intelligent Signal Processing and Communications Systems*, pp. 553–556, 2007.

61. Odorizzi, A. and G. Mazzini, "M-GeRaf Analysis: Performance Improvement of a Multisink Ad Hoc and Sensor Network Geographical Random Routing Protocol", *Proceedings of the International Conference on Software, Telecommunications and Computer Networks*, pp. 193–197, 2008.

62. Zorzi, M. and R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance", *IEEE Transactions on Mobile Computing*, Vol. 2, pp. 337–348, 2003.

63. Mitton, N., D. Simplot-Ryl and I. Stojmenovic, "Guaranteed Delivery for Geographical Anycasting in Wireless Multi-Sink Sensor and Sensor-Actor Networks", *Proceedings of the IEEE International Conference on Computer Communications*, pp. 2691–2695, 2009.

64. Xu, C., L. Cao and W. Shao, "Multiple Sink Dynamic Destination Geographic Routing in Wireless Sensor Networks", *Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 121–124,

2010.

65. Shirazi, G., P. Wang, X. Dong, Z. A. Eu and C.-K. Tham, "A QoS Network Architecture for Multi-Hop, Multi-Sink Target Tracking WSNs", *Proceedings of the IEEE Singapore International Conference on Communication Systems*, pp. 17–21, 2008.

66. Balakrishnan, M. and E. Johnson, "Fuzzy Diffusion for Distributed Sensor Networks", *Proceedings of the IEEE Military Communications Conference*, pp. 3023–3028, 2005.

67. Pirmez, L., F. Delicato, P. Pires, A. Mostardinha and N. de Rezende, "Applying Fuzzy Logic for Decision-Making on Wireless Sensor Networks", *Proceedings of the IEEE International Fuzzy Systems Conference*, pp. 1–6, 2007.

68. Minhas, M., S. Gopalakrishnan and V. Leung, "Fuzzy Algorithms for Maximum Lifetime Routing in Wireless Sensor Networks", *Proceedings of the IEEE Global Telecommunications Conference*, pp. 1–6, 2008.

69. Minhas, M. R., S. Gopalakrishnan and V. C. M. Leung, "Multiobjective Routing for Simultaneously Optimizing System Lifetime and Source-to-Sink Delay in Wireless Sensor Networks", *Proceedings of the IEEE International Conference on Distributed Computing Systems Workshops*, pp. 123–129, 2009.

70. Haider, T. and M. Yusuf, "A Fuzzy Approach to Energy Optimized Routing for Wireless Sensor Networks", *International Arab Journal of Information Technologies*, Vol. 6, No. 2, pp. 179–185, 2009.

71. Azim, M. A., M. R. Kibria and A. Jamalipour, "An Optimized Forwarding Protocol for Lifetime Extension of Wireless Sensor Networks", *Wireless Communications and Mobile Computing*, Vol. 9, pp. 103–115, 2009.

72. Li, G., Y. Cao, X. Gao and J. Tang, "Energy Balance Routing Protocol for Wireless

Sensor Networks Based on Fuzzy Next-Hop Selection", *Wuhan University Journal of Natural Sciences*, Vol. 14, pp. 148–152, 2009.

73. Manjunatha, P., A. K. Verma and A. Srividya, "Fuzzy Based Optimized Routing Protocol for Wireless Sensor Networks", S. C. Mukhopadhyay and H. Leung (Editors), *Advances in Wireless Sensors and Sensor Networks*, Vol. 64 of *Lecture Notes in Electrical Engineering*, pp. 273–282, Springer Berlin Heidelberg, 2010.

74. Basaran, C., K. Kang and M. Suzer, "Hop-by-Hop Congestion Control and Load Balancing in Wireless Sensor Networks", *Proceedings of the IEEE Conference on Local Computer Networks*, pp. 448 –455, 2010.

75. Leal, L. B., M. V. Sousa Lemos and R. H. Filho, "An Algorithm for Route Selection on Multi-Sink Wireless Sensor Network Using Fuzzy Logic", *Proceedings of the Technological Developments in Networking, Education and Automation*, pp. 591–596, 2010.

76. Ye, W., J. Heidemann and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks", *IEEE/ACM Transactions on Networking*, Vol. 12, No. 3, pp. 493–506, 2004.

77. Stojmenovic, I. and X. Lin, "Loop-Free Hybrid Single-Path Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks", *IEEE Transactions on Parallel Distributed Systems*, Vol. 12, No. 10, pp. 1023–1032, 2001.

78. Ferrara, D., L. Galluccio, A. Leonardi, G. Morabito and S. Palazzo, "MACRO: An Integrated MAC/Routing Protocol for Geographic Forwarding in Wireless Sensor Networks", *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 3, pp. 1770–1781, 2005.

79. Casari, P., M. Nati, C. Petrioli and M. Zorzi, "Efficient Non-Planar Routing Around Dead Ends in Sparse Topologies Using Random Forwarding", *Proceedings of the IEEE International Conference on Communications*, pp. 3122–3129,

2007.

80. OPNET Modeler, http://www.opnet.com/products/modeler/home.html, 2011.

81. Downes, I., L. B. Rad and H. Aghajan, "Development of a Mote for Wireless Image Sensor Networks", *Proceedings of the Cognitive Systems with Interactive Sensors*, Paris, France, 2006.

82. Kulkarni, P., D. Ganesan, P. Shenoy and Q. Lu, "SensEye: A Multi-Tier Camera Sensor Network", *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pp. 229 – 238, Hilton, Singapore, 2005.

83. Zixiang, X., A. D. Liveris and S. Cheng, "Distributed Source Coding for Sensor Networks", *IEEE Signal Processing Magazine*, Vol. 21, No. 5, pp. 80–94, 2004.

84. Miller, M. and N. Vaidya, "A MAC Protocol to Reduce Sensor Network Energy Consumption using a Wakeup Radio", *IEEE Transactions on Mobile Computing*, Vol. 4, No. 3, pp. 228–242, 2005.

85. Langendoen, K., "Medium Access Control in Wireless Sensor Networks", H. Wu and Y. Pan (Editors), *Medium Access Control in Wireless Networks*, pp. 535–560, Nova Science Publishers, Inc., 2008.

86. Ghosh, S., P. Veeraraghavan, S. Singh and L. Zhang, "Performance of a Wireless Sensor Network MAC Protocol with a Global Sleep Schedule", *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 4, No. 2, pp. 99–114, 2009.

87. Zhang, Y., C. He and L. Jiang, "Performance Analysis of S-MAC Protocol under Unsaturated Conditions", *IEEE Communication Letters*, Vol. 12, No. 3, pp. 210–212, 2008.

88. Mendel, J., "Fuzzy Logic Systems for Engineering: A Tutorial", *Proceedings of the IEEE*, Vol. 83, No. 3, pp. 345–377, 1995.