# GEOGRAPHIC ROUTING

## 1. Introduction

A Wireless Sensor Network (WSN) is a network composed of large number of small, low power, low-cost devices called sensors. The main task of WSNs is to monitor and report environmental conditions. One of the main characteristics of WSNs is that all the sensors belong to a group and work toward a common goal. An individual senor node has little value of its own unless it works in cooperation with other sensors in a distributed fashion. They exchange messages frequently according to the application demands and report the information to a single or to multiple sinks.

In computer networks, the process of forwarding a message from source to destination through a series of intermediate relays nodes is called routing. Compared to wired networks, the topology of WSNs changes more frequently. Therefore, more update messages are sent and cause network congestion and enormous bandwidth consumptions. Consequently, conventional routing methods such as Distance Vector algorithms are not acceptable for WSNs. In addition, memory constrained sensor nodes are incapable of saving the global information of network topology as the link state algorithms do. To overcome these problems, Geographic Routing algorithms are introduced. In Geographic Routing, each node only maintains the state of its one-hop neighbors and propagates topology information of its one-hop neighbors. Therefore, in order to make forwarding decisions Geographic routing only requires the position of the packet's destination and the position of its one hop neighbors. These two properties have made Geographic Routing very assuring. All Geographic Routing protocols have two modes of operations. In first mode, they greedily forward a packet to a neighbor, which is the best choice toward the destination. "Best" is defined according to the routing algorithm; it could mean the closest node to destination or a node with highest energy level, etc. If the node fails to find the best neighbor, the algorithm switches to the second recovery mode. Different Geographic Routing protocols have different backup recovery processes.

### The Problem

Commonly, large numbers of sensor nodes are required to cover an area. Therefore, nodes must be cheap to make use of the network economic. For this purpose, wireless sensors are made in small sizes, battery powered and memory constrained. They can communicate over a restricted area, as their radio range is small. Like any other broadcast oriented wireless technology, WSNs are vulnerable to numerous security threats. In addition, they are often deployed in unattended, unreliable environments where physical security is unavailable. These restrictions have made secure routing challenging in WSNs. Different attacks are applicable on different layers of WSNs. Hardware oriented attacks such as eavesdropping, interference, and jamming attack can disrupt the physical Layer operations. MAC layer attacks cause selfish misbehavior of nodes and gaining unfair share of bandwidth. Attacks such as selective forwarding, Sinkhole attack, Sybil attack, etc, target the network layer and routing protocols. Attacks such as SYN flooding, session hijacking, etc. lead to malfunctioning at the transport layer. Finally, some of the possible application layer attacks are viruses, worms, spywares, and Trojan Horses. They can attack both operating systems and user applications.

## Contribution

Geographic Routing protocols have been extensively studied in ad hoc and wireless sensor networks. However, many of these protocols have not paid much attention to security at the design phase. All concerns are focused on overcoming the inherent constraints of WSNs. In this study, we attempt to analyze attacks on two unsecure Geographic Routing protocols namely GEAR and GPSR. The reason for choosing these two from a variety of unsecure Geographic Routing protocols is comparing an energy aware protocol to a distance centric protocol in an attack-wise manner. Some attacks target the routing algorithm on path selection based on the distance metric, some target the energy consumption level of the sensor nodes and some take advantage of both metrics. In this research, the impact of different types of attacks are studied, attacks such as routing misdirection, selective forwarding, Sybil attack, sinkhole attack, Byzantine attack, beaconing attack, etc. The goal is to improve the robustness and security degree of the two aforementioned protocols. To achieve this, various approaches have been tried. One approach is disjoint multipath routing in which every compromised path has a probable uncompromised alternative path. Another approach is to make the protocols trust-aware. This means forwarding decisions are made not only based on the routing algorithm but also based on the history of system behavior. Nodes that have been treated loyally in the recent past are rewarded while the suspicious malicious ones are punished. Finally, we present that if we want to secure the protocols only by relying on the capabilities of regular nodes and using the software methods without additional help of auxiliary facilities such as hardware support and cryptographic algorithms, we need to keep a partial history in every node and in every sent message.

## Wireless Sensor Networks

A WSN is a network of hundreds and thousands of small, low power, low-cost devices called sensors. A sensor is an object that performs the sensing task and converts all forms of energy into electrical energy. The core application of WSNs is to gather information about physical objects or areas and report events. Physical property that should be monitored identifies the sensor type. For example if we want to monitor temperature: Thermistors or thermocouples sensors are required. Pressure gauges sensors are for monitoring pressure, Accelerometers are used for monitoring motion vibration, etc. Some examples of WSN usage are military and civilian domain, robotic landmine detection, battlefield surveillance, environmental monitoring, wildfire detection, and traffic regulation, life-saving operations, vehicle tracking, structural health monitoring, economic forecasting etc. Since the transmission ranges of sensor nodes are not large, they cannot transmit their data directly to the base station. Therefore, they cannot form a star topology. Thus, multi-hop communication is more common for sensor networks in which, sensors form a mesh topology and every sensor node serves as a relay for other sensor nodes. This reduces the power consumption and allows for larger coverage, however, it introduces the problem of routing: the task of finding a multi-hop path from a sensor node to a base station, which is one of the most important challenges in WSNs.

# Challenges in Wireless Sensor Networks

As described above, there are many challenges and constraints in WSN technology, which leads to the design of protocols and algorithms differ from other type of distributed systems. Some of the challenges are as follows:

**Energy constraints**

Most of the time it is not possible to replace the batteries of sensors and when the battery is depleted, the sensor is discarded. As a result, sensor nodes should be able to operate during their entire mission time with the initial installed battery

**Self-* properties**

Since there is no possibility of maintenance and repair in remote areas and harsh environments, sensor nodes must be self-managing which means they must be able to autonomously configure themselves, cooperate with other nodes and accommodate to failures and environmental changes without human intervention. For example, sensors, which monitor the catastrophe areas or battlefields, are thrown out of the airplanes over the target areas. Not all of these sensor nodes survive intact after such a drop and may never start their sensing activities. However, those who are survived must start a consecutive setup and configuration process autonomously including determining their positions, communicating with neighboring sensor nodes and the initiation of their sensing responsibilities. A sensor node should be self-organized, which means that it should be able to adapt configuration parameters based on system and environmental situation. For example, a sensor device can change its transmission

power to communicate with more or less number of neighboring nodes. Self-optimization is the ability to monitor and optimize the use of the limited system resources such as battery by automatic sleep and wake up. Self-protection is the ability to detect and protect from attacks and intrusions. Finally, self-heal property allows sensor nodes to autonomously recover from network disruptions . One form of self-healing is Self-stabilization. "A system is called self-stabilizing if and only if, regardless of the initial state and regardless of the privilege selected each time for the next move, at least one privilege will always be present and the system is guaranteed to find itself in a legitimate state after a finite number of moves ." A system turns out to be in an inconsistent state for many reasons. The temporal violation of algorithm assumptions, the variation of memory content due to harsh environments affects message loss at a higher accepted rate, topology changes due to node depletion, node destruction, mobility or new joining nodes, all can lead to temporal inconsistencies. The manual reconfiguration of large scale WSNs is not possible in order to recover from all these inconsistencies. Therefore, self-stabilization is a much-demanded property for algorithms in WSNs. The system assumptions could be violated not only due to the benign faults but also by a powerful adversary disrupting the network functionality. It is difficult to anticipate all possible states of a large-scale network after such an attack. Self-stabilization makes sure that the network can recover from any state as long as the assumptions hold once more.

Typically, algorithms are either secure or self-stabilized and not often secure and self-stabilized at the same time. Researches have been aimed at achieving high level networking protocols for WSNs that are both self-stabilizing and secure and could stand up to both faults and attacks. For example a secure and self-stabilizing algorithm can work based on this assumption that the part of the lost messages, including messages that are lost due to benign collisions or messages that are lost due to attacks do not pass a certain threshold in order to keep the adversary undetected. Therefore, the adversary does not attack all messages of a node. However, if the adversary passes this threshold, even a self-stabilized algorithm does not guarantee to provide the expected level of service. If the adversary is detected and eliminated and the message delivery assumption holds again, the self-stabilizing algorithm recovers quickly and delivers the promised level of service.

**Decentralized Management**

In large scale WSNs, sensor nodes cannot work in a centralized manner in which a Base station supervises the routing process and topology changes. Instead, sensor nodes must collaborate with each other locally, without a global knowledge. This brings all the challenges associated with decentralized systems such as self-learning via flooding and overhead of prompt reaction against topology changes .

**Design Constraints**

All the mentioned constraints affect the design of network protocols, operating systems and middleware in WSNs. For example, TelosB devices only have10 Kbytes of RAM and 48 Kbytes of flash memory, therefore, the installed operating systems must have small footprints and must be efficient in its resource management tasks. Alternatively, a list of neighbors is stored in a sensor node. Respectively, any algorithms that may require more computational power and storage capacities than can be provided by low-cost sensor nodes are not desirable

**Security**

Like all other secure computer systems, WSNs need to address three well-known services in the CIA security model: Confidentiality, Integrity, and Availability. Here, we explain what does this mean in the scope of WSNs.

**Confidentiality:** requires the protection of the exchanged data from insider, passive adversaries. The most common methods to achieve confidentiality are cryptographic schemes such as encryption which requires efficient key management schemes

**Integrity:** The receivers in WSN should be able to recognize whether the exchanged data between the two parties has been changed. In addition, the integrity service should also make sure that the exchanged content is not deleted, replicated, outdated or maliciously injected

**Availability:** Due to many threats to the WSN, some portion of the network or some of the functionalities or services could be temporarily damaged or unavailable. E.g., some sensors could die prematurely. Hence, availability services make sure that the necessary functionalities and services are always up and running, even in presence of malicious nodes. Availability properties of WSNs is studied in form of Denial-of-Service type attacks

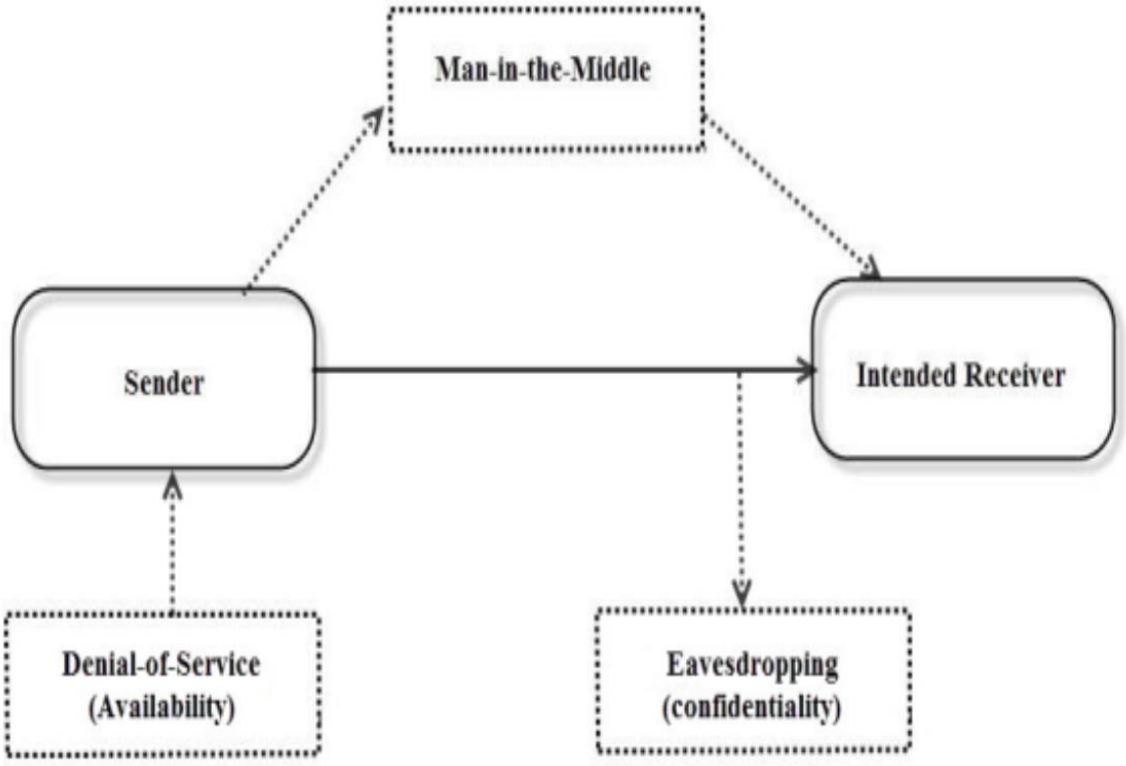


**Figure.1. The CIA model and examples of attacks.**

Resource constrained sensor nodes cannot support conventional security algorithms that are heavy in terms of computation or those, which require negotiation and authentication with remote devices . Meanwhile, despite the fact that cryptography provides integrity, confidentiality and authentication, it is defenseless against an internal adversary. Therefore, a security mechanism inside a WSN is required to deal with internal adversaries. Wireless sensor nodes typically operate in remote and hard to reach locations, deployed in public access environments and often scale large. Thus, it is not possible to continuously monitor and protect sensor nodes from attacks. The wireless medium of communication is error prone. Errors such as channel errors, routing failures or collisions hence, packets may be lost or corrupted in the middle of the way. By the way, it is difficult to distinguish benign communication errors such as node and link failures from malicious behaviors . Wireless

communications make it easy for an adversary to eavesdrop on sensor transmissions leads to one of the most challenging security threats called Denial of Service attack with aim of disrupting the availability of WSN operation. DoS attacks can be applied in different ways. As an example, powerful wireless signals jam the communication channels and prevent legitimate nodes from successful communication .

# Geographic Routing Protocols

In wired networks, routing is performed by routers, high performance devices that specifically designed for the purpose of forwarding messages. The underlying network is stable and network topology does not change frequently. Therefore, prompt reactions and rapid propagation of update messages in short intervals are not required. In contrast, in WSNs, there are no specific devices for routing process. Every node acts as a router, cooperates on forwarding a message from source to destination. Wireless sensor networks are inherently more dynamic than the wired networks as network topology can be changeable. Thus, conventional routing protocols, which are designed for wired networks, generally fail to satisfy the requirements of wireless networks. These facts lead to invention of new routing protocols specifically for operation in ad hoc networks. Typically, these protocols are classified into three categories: proactive, reactive and Geographic Routing protocols. The first two are called topology-based protocols, while the third one is a location-based protocol. Proactive routing protocols need to maintain the information about the entire network topology and propagate frequent updates due to topology changes. On the other hand, reactive routing protocols need to discover the routes on demand via excessive flooding. Since the topology-based protocols are beyond the scope of this research, we suffice it to say that they are not very efficient in lightweight sensor nodes for the aforementioned reasons as they exhaust the network bandwidth and need intense memory storage.

Alternatively, in geographic routing protocols, forwarding decisions are made using geographic position of the nodes. All nodes in the network know their own positions as well as the position of all their neighbors. In addition, every sent packet carries the location of the final destination in its header. The node uses these three inputs to choose the next hop. All routing decisions are made locally based on internal node state and therefore, very little routing information is kept in each node. Traffic overhead and computation time are considerably reduced because no energy is spent on frequent route discovery, route request and reply messages. Node memory requirements are decreased, as there is no need for keeping information about the entire network topology. These three advantages, no need for keeping routing tables, independence of remote topology changes and flooding free route discovery process, are three main reasons for the appropriateness of Geographic Routing for WSNs. This makes it so practical, as once the position of the destination is known, all operations are local .

## Geographic Routing Algorithm

The formulation of above definition is as follows:
"Let G = (V, E) be a Euclidean graph. The task of a geographic ad hoc routing algorithm A is to transmit a message from a source S $\in$V to a destination D $\in$ V by sending packets over the edges of G while complying with the following conditions:

- All nodes v $\in$V know their geographic positions as well as the geographic positions of all their neighbors in G.
- The source S is informed about the position of the destination D.
- The control information which can be stored in a packet is limited by O(log n) bits, that is, only information about a constant number of nodes is allowed.

- Except for the temporary storage of packets before forwarding, a node is not allowed to maintain any information . "

There are various types of Geographic Routing protocols. Some examples are GPSR, GEAR, GAF, GHT, GOAFR, GFG, etc.
Before going further, we need to explain about the two main approaches of packet forwarding in Geographic routing; Greedy forwarding and Face routing.

## Greedy Routing

The first introduced approach to Geographic Routing was greedy forwarding . This approach is conceptually simple and the implementation is easy. Initially, with respect to the routing algorithm, the concept of being "best located neighbor" needs to be defined. This can be interpreted as "closest to the destination" or "having the highest energy level", or the combination of both, etc. Then every node forwards the message to its best-located neighbor for example to a node, which has the minimum distance to the destination. Greedy forwarding advances until it reaches a node without any "better" neighbor. It then stops. This dead-end which is one of the conventional problems of all Geographic Routing protocols is called "local minima", "local maxima", "hole" or "void". Consequently, a backup forwarding approach such as Face routing is required for escaping from it. We will explain about it in the following section. As a result, greedy forwarding cannot be used solely as a solution for geographic routing. However if not stuck, it reaches the destination so efficiently.

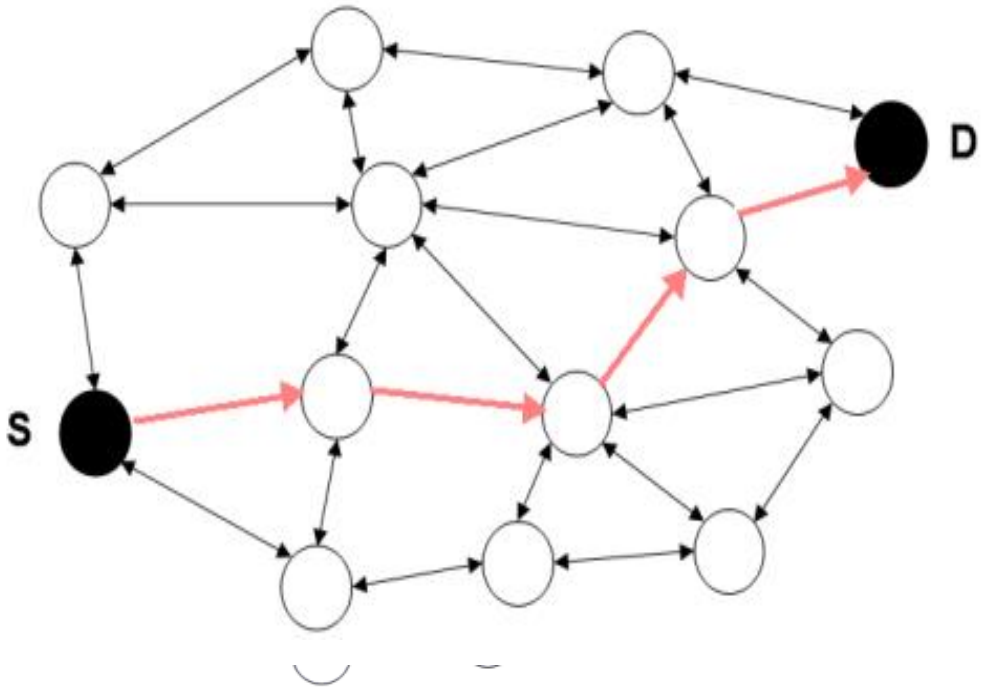


**Figure.2. Greedy Routing.**

**If d is the distance between source and destination, it reaches the destination with cost: O ($d^2$).**

## Face Routing

Face routing sometimes called perimeter forwarding was the first Geographic Routing algorithm that guaranteed successful message delivery without getting stuck in the middle of the way . It is defined based on the concept of "faces", contiguous polygonal regions separated by the edges of a planar graph. In a planar graph, no two edges cross each other. Face routing uses two principles of "the right hand rule" and "face change" to proceed. The right-hand rule goes around a face in a cycle on clockwise direction. There are some minor differences between protocols in how to explore the face and where to switch faces. The original algorithm keeps track of the points where it crosses the line SD (the hypothetical line that connects the source S and the destination D). After routing the face completely, the algorithm returns the intersection point that is closest to the destination. It then proceeds by routing the next face closer to D. The same steps are repeated until the message either reaches a face containing the destination or reaches a node that is no longer considered as the local minima. It then falls back to greedy mode and moves forward. As mentioned it always guarantees delivery, however if n is the total number of nodes in the network, it takes at most O (n) steps, which is similar to flooding. Thus, it cannot be used as the main routing approach and always should be used as a backup method of greedy forwarding

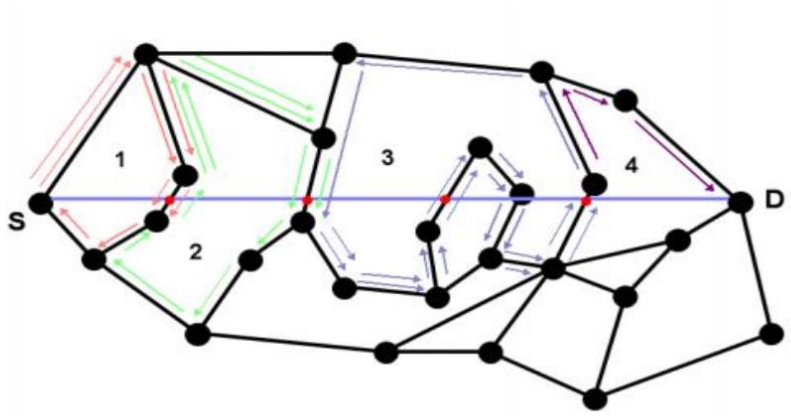


**Figure.3. Face Routing.**

**Original face routing: if n is the total number of nodes in the network, it takes at most O (n) steps, which is similar to flooding**

Current Geographic Routing protocols use one of the following rules for changing the face:

**First intersection**: As the algorithm is traversing the face using the right-hand rule, on encountering an edge that crosses the hypothetical line SD at a point closer to D than the point that the current face was entered, the algorithm changes the face at that edge. Thus, this approach does not traverse the entire face (GFG, GPSR) .

**Best intersection**: Works like the first intersection approach, with the difference that it traverses the entire face. (AFR, Compass) .

**Closet-node (other face routing)**: This approach also traverses the entire face and changes the face at the "node" closest to D. (GOAFR+) .

**Closest-point (other face routing)**: The operation is the same as closet node (other face routing), however, the face change occurs at the closest "point" to D.

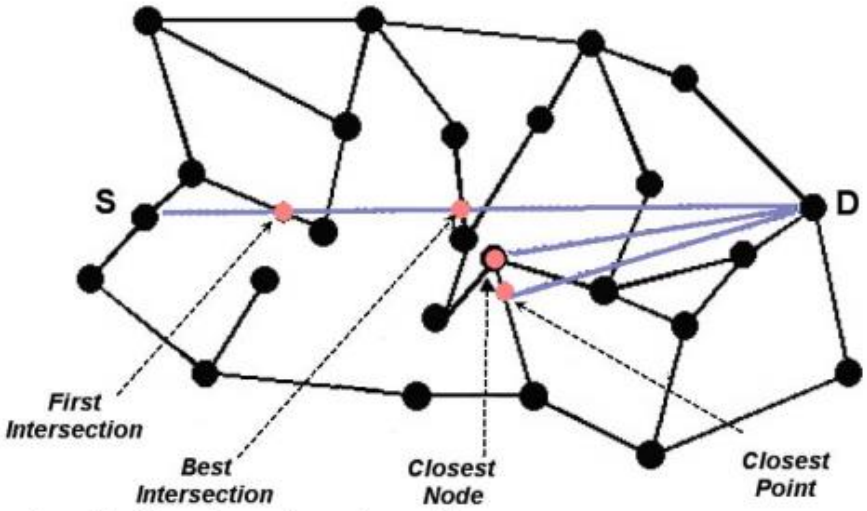


**Figure.4. Points where the face change is performed.**

REFERENCE

1. Wireless Networking in the Developing World: A practical guide to planning and building low-cost telecommunications infrastructure (PDF) (2nd ed.). Hacker Friendly LLC. 2007. p. 425.

2. Pahlavan, Kaveh; Levesque, Allen H (1995). Wireless Information Networks. John Wiley & Sons. ISBN 0-471-10607-0.

3. Geier, Jim (2001). Wireless LANs. Sams. ISBN 0-672-32058-4.

4. Goldsmith, Andrea (2005). Wireless Communications. Cambridge University Press. ISBN 0-521-83716-2.

5. Molisch, Andreas (2005). Wireless Communications. Wiley-IEEE Press. ISBN 0-470-84888-X.

6. Pahlavan, Kaveh; Krishnamurthy, Prashant (2002). Principles of Wireless Networks – a Unified Approach. Prentice Hall. ISBN 0-13-093003-2.

7. Rappaport, Theodore (2002). Wireless Communications: Principles and Practice. Prentice Hall. ISBN 0-13-042232-0.

8. Rhoton, John (2001). The Wireless Internet Explained. Digital Press. ISBN 1-55558-257-5.

9. Tse, David; Viswanath, Pramod (2005). Fundamentals of Wireless Communication. Cambridge University Press. ISBN 0-521-84527-0.

10. Pahlavan, Kaveh; Krishnamurthy, Prashant (2009). Networking Fundamentals – Wide, Local and Personal Area Communications. Wiley. ISBN 978-0-470-99290-6.