

# Hop Count Filtering Algorithm to Analyze Node Failure for Intrusion Prevention and Packet Drop Avoidance

AbhijitMore ,AmolBaviskar, JaypalBaviskar , RohitWaghmare

PadmabhushanVasantdadaPatilPratishthan's College of Engineering, Mumbai 400022

Department of Information Technology, Vishwatmak Om Gurudev College of Engineering, Mohili-Aghai, Maharashtra IEEE  
Member

*abhijitmore23@yahoo.in, amolbaviskar222@gmail.com, jaypal.j.baviskar@ieee.org, wagh4all@gmail.com*

**Abstract**—This paper presents a technique which aims to detect cuts (disconnection) between nodes in order to avoid packet drop and prevent intrusion by Hop Count Filtering algorithm. It is advantageous as it not only avoids provides packet drop but also provides alternative path which would be indeed the shortest available path for packet transfer. It avoids unwanted access to data and also blocks the intruder's services so that he/she could not have access to data. As it runs on Java it can be made to run on any platform A Cut detection system is used to detect the disconnection between nodes. This avoids unwanted loss of data by detecting the failed node and routing the data package via alternative path. The Intrusion detection system for network is generally used with the networking applications where the hacking attempts are made by the hackers. Here the HCF algorithm is used to identify whether the hacking attempt is made or not, and the Security principals are used for identifying the normal IP Address and Hacked IP address. The Database helps to prevent the hacked IP packets and alarm is generated by the system and information is shown at the receiver side. DOS attack is also taken into consideration which contains Disruption of configuration information, such as routing information. DOS is also gets denied. This paper not only detects intrusion but also prevents it by tracing the intruders IP address and then it blocks the intruder's services so that intruder is unable to hack any further.

**Keywords** - Hop Count Filtering algorithm, Java, Cut Detection system, packets, Intruder, DOS attack.

\*\*\*\*\*

## I. INTRODUCTION

Wireless sensor networks (WSNs) are a promising technology for monitoring large regions at high spatial and temporal resolution. In fact, node failure is expected to be quite common due to the typically limited energy budget of the nodes that are powered by small batteries. Failure of a set of nodes will reduce the number of multi-hop paths in the network. Such failures can cause a subset of nodes that have not failed to become disconnected from the rest, resulting in a cut. Two nodes are said to be disconnected if there is no path between them. so we used Cut detection system to detect the cuts and also to alter the flow of data from currently selected path to the next shortest path. IDS are often used as another wall to protect computer systems. Here project secures the system from attackers.

Attackers mainly use Intrusion detection technique, Intrusion detection (ID) is defined as the problem of identifying individuals who are using a computer system without authorization (i.e., crackers) and those who have legitimate access to the system but are abusing their privileges (i.e., the insider threat).

## II. LITERATURE SURVEY

Wireless sensor networks (WSNs) are a promising technology for monitoring large regions at high spatial and temporal resolution. Failure of a set of nodes will reduce the number of multi-hop paths in the network. Such failures can cause a subset of nodes that have not failed to become disconnected from the rest, resulting in a cut. Two nodes are said to be disconnected if there is no path between them. We consider the problem of detecting cuts by the nodes of a wireless network. We assume that there is a specially

designated node in the network, which we call the source node. Since a cut may or may not separate a node from the source node, we distinguish between two distinct outcomes of a cut for a particular node.

When a node  $u$  is disconnected from the source, we say that a DOS (Disconnected from Source) event has occurred for  $u$ . When a cut occurs in the network that does not separate a node  $u$  from the source node, we say that CCOS (Connected, but a Cut Occurred Somewhere) event has occurred for  $u$ . By cut detection we mean

- (i) detection by each node of a DOS event when it occurs, and
- (ii) detection of CCOS events by the nodes close to a cut, and the approximate location of the cut. In this article we propose a distributed algorithm to detect cuts, named the Distributed Cut Detection (DCD) algorithm.

The algorithm allows each node to detect DOS events and a subset of nodes to detect CCOS events. The algorithm we propose is distributed and asynchronous: it involves only local communication between neighboring nodes, and is robust to temporary communication failure between node pairs. The convergence rate of the computation is independent of the size and structure of the network. All cut nodes information is been updated and created into a log file of system and filtered log record of cut nodes are maintained.

IDS are often used as another wall to protect computer systems. Here project secures the system from attackers.

Attackers mainly use Intrusion detection technique, Intrusion detection (ID) is defined as the problem of identifying individuals who are using a computer system without authorization (i.e., crackers) and those who have legitimate access to the system but are abusing their privileges (i.e., the insider threat). Existing Intrusion Detection Security Softwares: In order for us to determine how database can help advance attack detection it is important to understand how current Project work to identify an Attack.

There are two different approaches to Security detection:

- Misuse detection
- Anomaly detection

Misuse detection is the ability to identify intrusions attacks based on a known pattern for the malicious activity. These known patterns are referred to as signatures. The second approach viz. anomaly detection, is the attempt to identify malicious traffic based on deviations from established normal network traffic patterns. Most of the software's which can be purchased today are based on misuse detection. Current software products come with a large set of signatures which have been identified as unique to a particular vulnerability or exploit. Most software's vendors also provide regular signature updates in an attempt to keep pace with the rapid appearance of new vulnerabilities and exploits.

### III. HOP COUNT FILTERING METHODOLOGY

Hop Count Filtering is popular technique. It can be used as a defensive algorithm [1] for spoofed traffic. It can be also used for detecting the sinkhole attacks in Wireless Sensor Network [2].

The five modules included in Hop Count Filtering algorithm are listed below:

- Distributed cut detection
- Cut
- Source Node
- CCOS and DOS
- Network Separation

#### A. Distributed cut detection

The calculation permits every hub to distinguish DOS occasions and a subset of hubs to identify CCOS occasions. The calculation we propose is appropriated and nonconcurrent: it includes just nearby correspondence between neighboring hubs, and is powerful to brief correspondence disappointment between hub sets. A key part of the DCD calculation is a dispersed iterative computational stride through which the hubs figure their (invented) electrical possibilities. The union rate of the calculation is autonomous of the size and structure of the system.

#### B. Cut

Wireless sensor systems (WSNs) are a promising innovation for observing extensive districts at high spatial and worldly

resolution. In reality, hub disappointment is relied upon to be very normal because of the regularly constrained vitality spending plan of the hubs that are fueled by little batteries. Disappointment of an arrangement of hubs will lessen the quantity of multi-bounce ways in the system. Such disappointments can bring about a subset of hubs that have not neglected to wind up detached from the rest, bringing about a cut. Two hubs are said to be detached if there is no way between them.

#### C. Source Node

We consider the issue of distinguishing cuts by the hubs of a remote system. We expect that there is an exceptionally assigned hub in the system, which we call the source hub. The source hub might be a base station that serves as an interface between the system and its users. Since a cut could possibly isolate a hub from the source hub, we recognize two unmistakable results of a cut for a specific hub..

#### D. CCOS and DOS

At the point when a hub  $u$  is separated from the source, we say that a DOS (Disconnected from Source) occasion has happened for  $u$ . At the point when a cut happens in the system that does not separate a hub  $u$  from the source hub, we say that CCOS (Connected, however a Cut Occurred Somewhere) occasion has happened for you.

By cut detection we mean

- (i) detection by each node of a DOS event when it occurs, and
- (ii) detection of CCOS events by the nodes close to a cut, and the approximate location of the cut.

#### E. Network Separation

Failure of a set of nodes will reduce the number of multihop paths in the network. Such failures can cause a subset of nodes that have not failed to become disconnected from the rest, resulting in a cut. Because of cut, some nodes may separate from the network, that results the separated nodes cant receive the data from the source node. The flowchart for Cut Detection Algorithm is demonstrated below in Fig. 1

### IV. IMPLEMENTATION OF PROPOSED ALGORITHM

A. For Intrusion Prevention the algorithm is implemented by demonstrating complete simulation of Hop-count filtering. GUI modules are to be designed for the Source Node, Router 1, Destination Node and the Hacker. The simulation can be shown in two ways;

- A process with hacker enabled
- A process with hacker disabled Overall algorithm is divided into two main modules;
- Reporting or GUI modules
- Intrusion detection modules The logical implementation of the Hop Count algorithm involves the idea to count two factors before accepting the packet;

- Actual TTL TTL computed after sending packet from source to destination
- Final TTL TTL supposed to come as stored in database
- Actual TTL = Initial TTL- (packet received-1)
- Packet received will be in the form of (message + number of routers)
- Final TTL = Initial TTL- (stored routers)

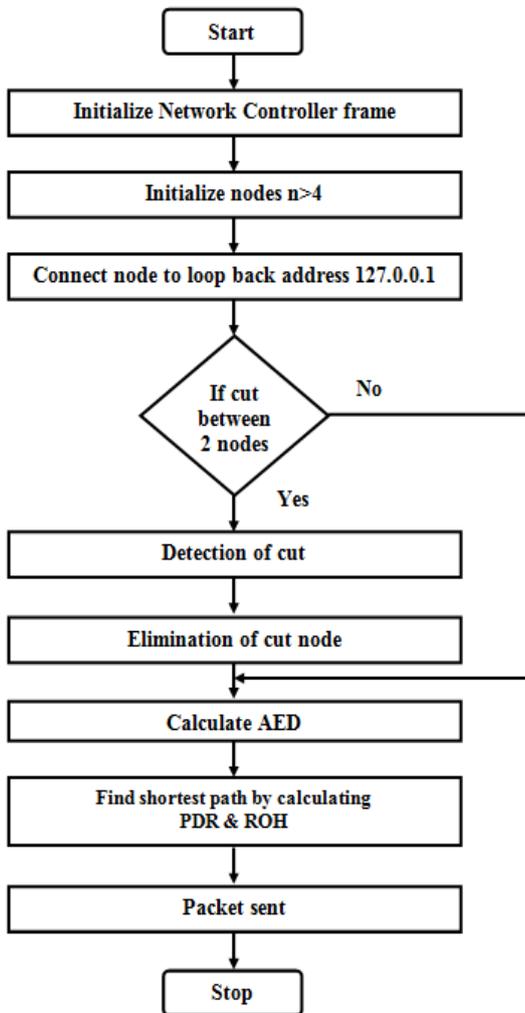


Fig. 1. Flowchart for Cut Detection Algorithm

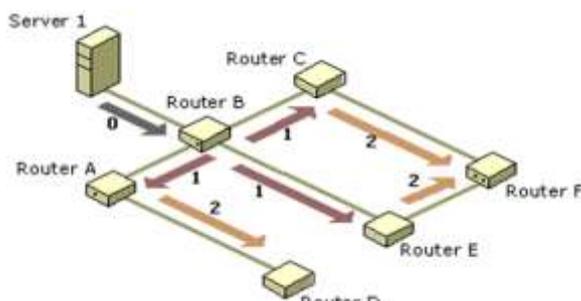


Fig. 2. Implementation of Hop Count Algorithm

For example, as seen in Fig. 2 suppose number of routers stored =2 and initial TTL=34.

1. Then, without hacking a packet transmits from Sourcerouter1-router2-destination.  
 Actual TTL =  $(34 - (3-1)) = 32$   
 Packet received will be in the form of (message + number of routers)  
 Final TTL =  $(34 - (2)) = 32$   
 Since both actual TTL and Final TTL are equal we conclude packet is not spoofed.

2. Then, with hacking a packet transmits from Sourcerouter1-hacker-router2-destination.  
 Actual TTL =  $(34 - (4-1)) = 31$   
 Packet received will be in the form of (message + number of routers)(here hacker is enabled between routers) Final TTL =  $(34 - (2)) = 32$   
 Since both actual TTL and Final TTL are not equal we conclude packet is spoofed.

B. For Intrusion Detection and Prevention The Fig. 3 shown below gives detailed steps involved in intrusion prevention as well as Intrusion Detection

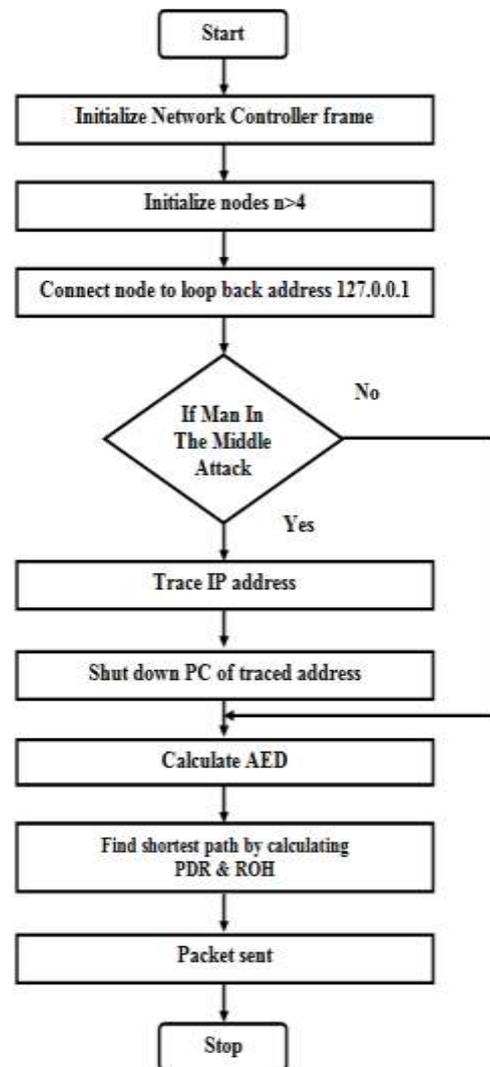


Fig. 3. Flowchart for Intrusion Detection and Prevention

V. ANALYSIS OF IMPLEMENTED ALGORITHM USING VARIOUS DIAGRAMS

A. USE CASE Diagram

The USE CASE Diagram delineated in Fig. 4 in the Unified Modeling Language (UML) is a type of behavioral diagram

defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

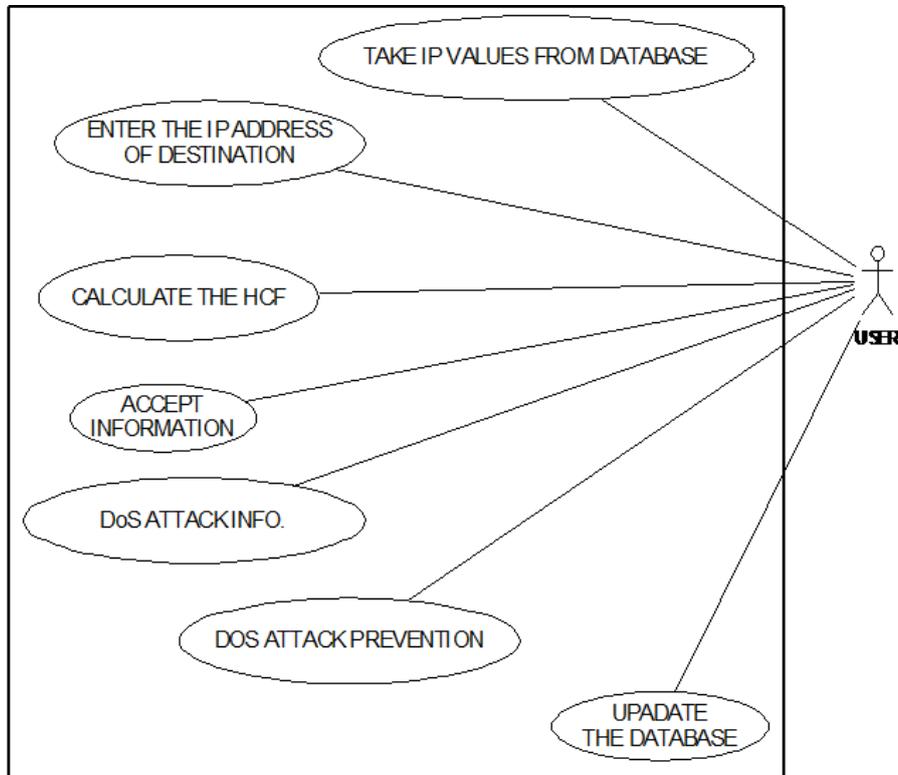


Fig: Use Case Diagram

C. DATA FLOW Diagram (DFD)

A DATA FLOW Diagram (DFD) demonstrated in Fig. 5 is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design).

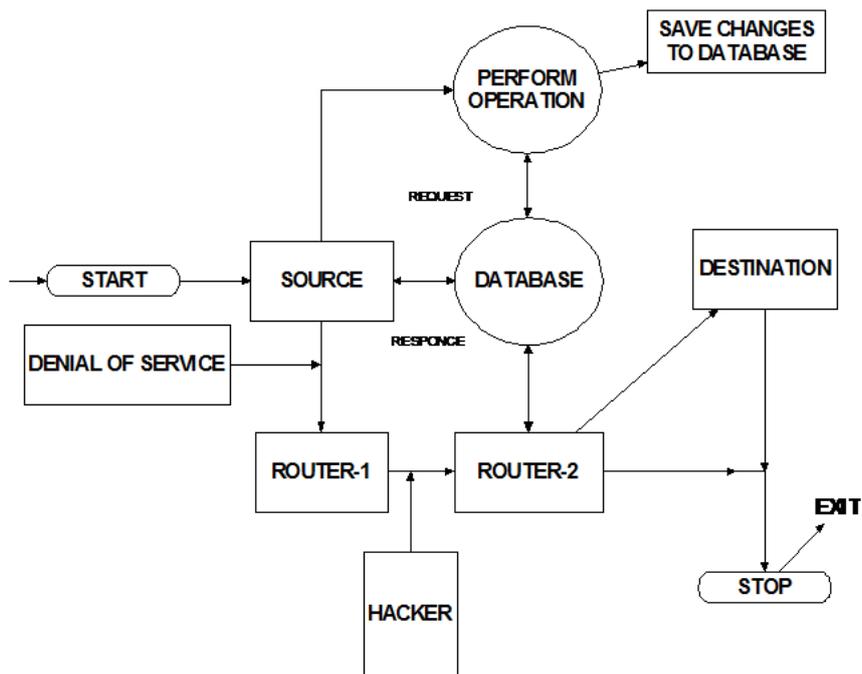


Fig: Data Flow Diagram

D. E-R Diagram An Entity-Relationship Model (ERM) shown in Fig. 6 is an abstract and conceptual representation of data. Entityrelationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams.

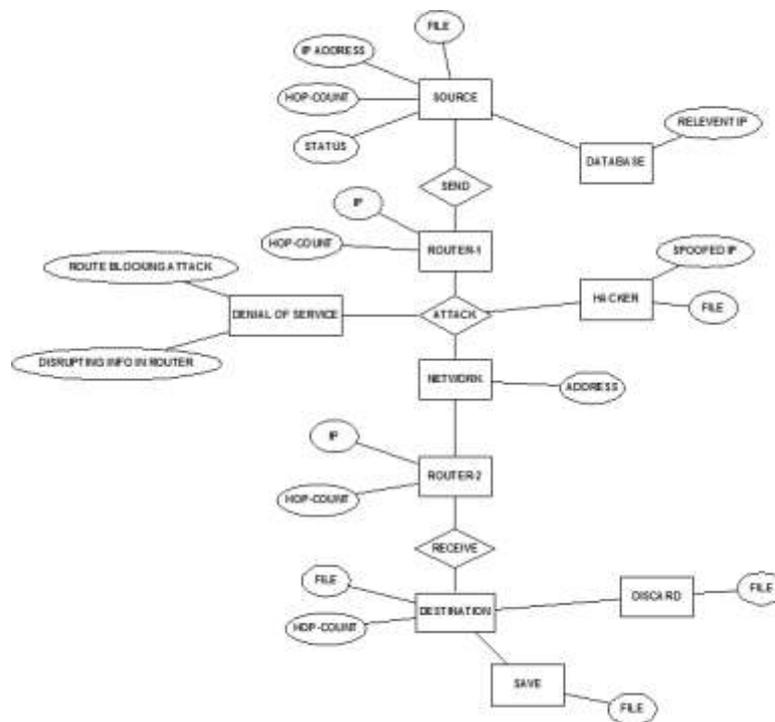
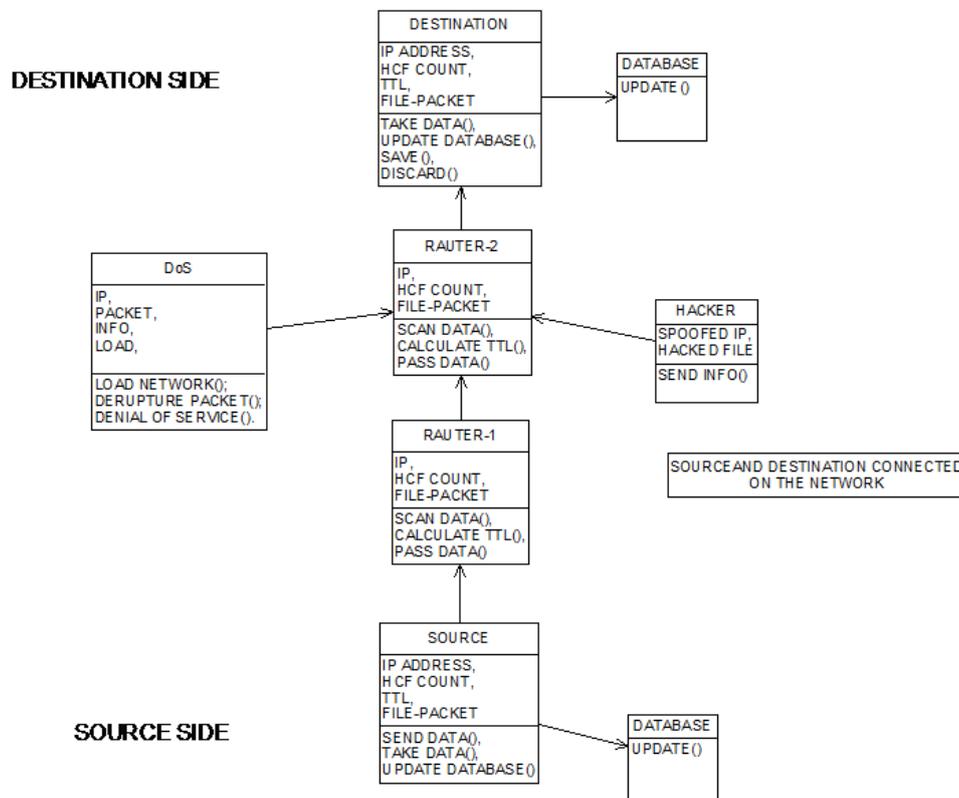


Fig: ER Model

E. Class Diagram

A class diagram delineated in Fig. 7 in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, and the relationships between the classes.



## VI. CONCLUSION

The cut detection proposed algorithm is presented that can also be used for detection of reconnection. If a component that is disconnected due to a cut gets reconnected later, the nodes can detect such reconnection from their states. Simulations are reported in that illustrate the capability of the algorithm to (i) detect cuts in mobile networks and (ii) detect re-connections after cuts. The potentials of nodes far away from the source typically become smaller as the network size increases. To keep the state values become too small which will affect cut detection the parameter  $s$  has to be chosen as a large number for a large network. A guideline for choosing  $s$  depending on the size of the network, and how much a-priori knowledge of the network structure is needed to make the appropriate choice, is being investigated.

The states of the nodes computed by the DSSD algorithm are affected by even those node failures that do not lead to cuts. This is intuitive, since the electrical potential of a node in a resistive electrical network is a function of the network structure which changes due to node failures. This feature raises the possibility of designing algorithms to compute electrical potentials of nodes in a wireless network so as to detect structural changes that are more complex than simply cuts. While a protocol that enables nodes to detect cuts is useful, there is also a need for protocols that allow a base station to detect when and where a cut has occurred. Also the Intrusion detection and prevention, the system fails if firewalls are activated so further extension of this project should include remote controlling the hackers PC if you have to block his services after we trace his IP. Noise can severely limit Security systems effectiveness. Bad packets generated from software bugs, corrupt DNS data, and local packets that escaped can create a significantly high false-alarm rate.

## REFERENCES

- [1] Cheng Jin, Haining Wang, Kang G. Shin, "Hope Count Filtering: A Defense Against Spoofed Traffic" Available at: <http://www.eecs.umich.edu/techreports/cse/2003/CS-E-TR-473-03.pdf>
- [2] D. Dallas, C. Leckie and K. Ramamohanarao, "Hop-Count Monitoring: Detecting Sinkhole Attacks in Wireless Sensor Networks," Networks, 2007. ICON 2007. 15th IEEE International Conference on, Adelaide, SA, 2007, pp. 176-181.