Impact of Watchdog Mechanism in VANET – A Review

S. Raagavi
Department of Computer Science
Periyar University, Salem-636 011
raagavisargunam12@gmail.com

S. Sathish
Department of Computer Science
Periyar University, Salem – 636 011
sathishkgm@yahoo.com

Abstract- **Vehicular Ad-Hoc Network is a type of the ad-hoc network. It is communicated from the road side units. Vehicular Ad-Hoc Network is the supper class of Mobile Ad-hoc networking in which vehicles is moving high speed on road side to exchanging information in efficient manner. Vehicular Ad-Hoc Network also provides value added services like email, audio and video sharing etc. Watchdog is a security function in journalism informs the public about society especially in circumstances where a significant portion of the public would demand changes in response. Watchdog might involve for fact checking statements of public officials. In order to provide the reliable information to the environment, the mechanism like watch dog is required to monitor the misbehaving node in the network. Lot of watch dog methods is available to monitor the environment and provide the secured transformation. This paper mainly focus the survey of watch dog methods like watchdog timer, watchdog monitoring and co-operative mechanism in Vehicular Ad-Hoc Network environment.**

Keywords- Watchdog, Timer, Monitoring, Cooperation, VANET

1. INTRODUCTION

Vehicular Ad-Hoc Networks, (VANET), are a particular kind of Mobile Ad Hoc Network, (MANET), in which vehicles act as nodes and each vehicle is equipped with transmission capabilities which are interconnected to form a network. The topology created by vehicles is usually very dynamic and significantly non-uniformly distributed (Bhois, 2014)(C. Oliveira, 2013). The availability of navigation systems on each vehicle makes it aware of its geographic location as well as its neighbours. However, a particular kind of routing approach, called Geographic Routing, becomes possible where packets are forwarded to a destination simply by choosing a neighbour who is geographically closer to that destination. With the rapid growth of vehicles and roadside traffic monitors, the advancement of navigation systems, and the low cost of wireless network devices, promising peer-to-peer (P2P) applications and externally-driven services to vehicles became available. For this purpose, the Intelligent Transportation Systems (ITS) have proposed the Wireless Access in Vehicular Environments (WAVE) standards that define an architecture that collectively enables vehicle-to-vehicle (V2V) and (V2I) vehicle-to-infrastructure wireless communications.

VANET can be divided into three categories: (i) The Wireless Wide Area Network (WWAN), in which the access points of the cellular gateways are fixed in order to allow direct communication between the vehicles and the access points. However, these access points require costly installation, which is not feasible. (ii) The Hybrid Wireless Architecture, in which WWAN access points are used at certain points while an ad hoc communication provides access and communication in between those access points. (iii) The Ad Hoc V2V Communication, which does not require any fixed access points in order for the vehicles to communicate. Vehicles are equipped with wireless network cards, and a spontaneous setting up of an ad hoc network can be done for each vehicle. This study will focus on studying ad hoc V2V communication networks, which are also known as VANETs.

The purpose of VANET is to allow wireless communication between vehicles on the road including the roadside wireless sensors, enabling the transfer of information to ensure driving safety and planning for dynamic routing, allowing mobile sensing as well as providing in-car entertainment. As VANETs have unique characteristics which include dynamic topology, frequent disconnection of the networks, and varying environments for communication.

The routing protocols for traditional MANET such as Ad hoc On-demand Distance Vector (AODV) are not directly usable for VANETs. Vehicular Delay-Tolerant Networks (VDTNs) were proposed as new kind of vehicular networks, whose design supports communications in environments where an end-to-end path between

the source and destination may not be available. Compared to MANETs, VANET present many other constraints such as the high mobility of nodes, the network topology changing, and the short times of connection. These constraints require that different types of conventional attacks to which they are exposed ad hoc mobile networks are valid for VANET, but the behaviour of VANETs against these attacks is not the same.

As mobile communication has drastically changed the lifestyles of human being, vehicular communication is expected to play a very important role as a future development of the society. Industrial sectors, telecommunication sectors, government research agencies, academic researchers are focusing in developing more secure transportation on the roads through Vehicular Ad-hoc Networks (Douglass, 2011).

1.1. Vehicle to Vehicle Communication

It refers to inter vehicle communication. Vehicles or a group of vehicles connect with one another and communicate like point to point architecture. It proves to be very helpful for cooperative driving.

1.2. Vehicle to Infrastructure Communication

Number of base stations positioned in close proximity with a fixed infrastructure to the highways is necessary to provide the facility of uploading/downloading of data from/to the vehicles. Each infrastructure access point covers a cluster.

1.3. Cluster to Cluster Communication

In VANETs network is splited into clusters that are self-managed group of vehicles. Base Station Manager Agent (BSMA) enables communications between the clusters. BSMA of one cluster communicates with that of other cluster.

1.4. Security of VANET

In VANET greedy drivers or the other adversaries can be condensed to a greater extent by authentication mechanism that ensures that the messages are sent by the actual nodes. Authentication, however, increases privacy concerns, as a basic authentication scheme of connecting the identity of the sender with the message. It, therefore, is absolutely essential to validate that a sender has a certain property which gives certification as per the application. For example, in location based services this property could be that a vehicle is in a particular location from where it claims to be

- Message integrity
- Message non reputation
- Access control
- Privacy
- Real time guarantees

Organization of the paper as follows Section I describe the introductory part, Section II describes the overview of Watch dog mechanism, Section III provides the Cooperation mechanism of VANET, Section IV describe the Monitoring method, Section V describe the timer method and Section VI conclude the paper.

2. OVERVIEW OF WATCHDOG

A watchdog is a device used to protect a system from specific software or hardware failures that may cause the system to stop responding. The application is first registered with the watchdog device. Once the watchdog is running on your system the application must periodically send information to the watchdog device. It also including for watchdog timer defined for sometimes called a computer operating properly or COP timer, or simply a watchdog is an electronic timer that is used to detect and recover from computer malfunctions. Example concluding for ATM process. Watchdog technique is a fundamental building block to many trust systems that are designed for securing wireless sensor networks (WSNs). Unfortunately, this kind of technique consumes much energy and hence largely limits the lifespan of WSN. Efficiency in WSNs and proposed several preliminary solutions, they have overlooked to optimize the watchdog technique which is perhaps among the top energy consuming units. Methods to minimize the energy cost of watchdog usage while keeping the system's security in a sufficient level. The results have successfully confirmed that our watchdog optimization techniques (F. Li, 2009)

A watchdog timer (WDT) is a hardware timer that automatically generates a system reset if the main program neglects to periodically service. It is often used to automatically reset an embedded device that hangs because of a software or hardware fault as Shown in figure 1.

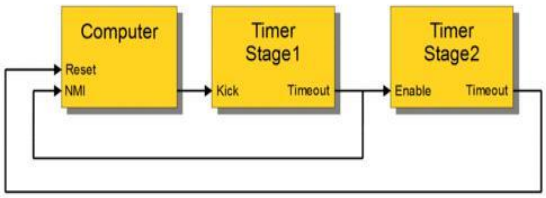

Figure 1. Watchdog Timer

Vehicular Delay Tolerant Networks (VDTNs) an end-to-end relay path between bundles source and destination nodes may not be available. To accomplish such goal, VDTNs rely on nodes cooperation. Thus, in order to maintain the network efficiency, it is very important to ensure that all network nodes follow the protocol. This is not an easy task since nodes may diverge from the protocol due to a selfish behaviour or to maintain their data or resources integrity. This paper proposes a cooperative watchdog system to detect and act against misbehaviour nodes in order to reduce their impact in the overall network performance (G. Li, 2011).

Watchdog is an application that can 'watch' other applications to ensure they are actively running. If an application should shutdown abnormally or become hung, Watchdog can restart this application. Watchdog is an optional purchase item that can be associated with drop box and Gateway. Watchdog utilizes a heartbeat between the target application and itself (Surana K.A., 2012). If the target application cannot 'respond' to the heartbeat, Watchdog will restart the program.

Watchdog will not be able to monitor any generic process for drop box and gateway can be monitored. Watchdog only monitors target application on the machine for which it is being run. Watchdog will not monitor a process on another workstation. The Watchdog screen displays the current status of all Target applications. Basic information such as Process Name along with Process Pathname and working directory are show. The current Status of the Process is shown as either 'Running' or 'Not Running'. If the Target Process is not supported on the Security key, 'No Support for Process' will be displayed.

If a Heartbeat is missed, it is displayed with the number of attempts tried. When the number of misses equals the number of maximum attempts defined by the WatchDog.INI file, Watchdog will restart the program. The number of times the Target program has been restarted is also logged. The Enabled / Disabled button allows on to Enable or Disable the heartbeat communications between Watchdog and the Target application.

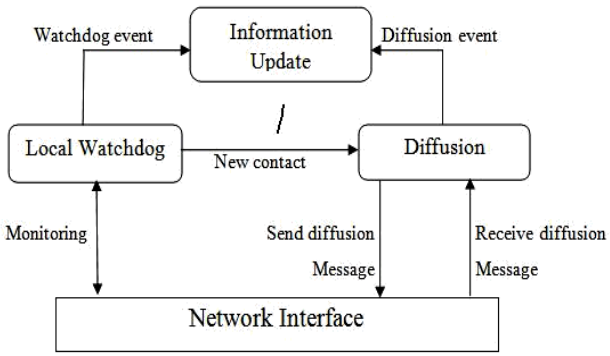

Figure 2. Overview of watchdog

When Enabled, Watchdog will monitor the program, likewise, when Disabled Watchdog will cease to monitor the program. To the right of the screen are the target applications being monitor. Selecting one of the Targets will display its status information (Hongmei Deng, 2002).

Watchdog mechanism in MANET, a node can listen the traversing packets of its neighbourhood and monitor their activity by overhearing the forwarding transmissions. The watchdog can deduce if nodes are acting as selfish,

block or grey whole routers. Independent of routing protocols used and can be detect attacks in ad-hoc networking. Mobile ad hoc network is a wireless technology that contains high mobility of nodes and does not depend on the background administrator for central authority, because they do not contain any infrastructure. Nodes of the MANET use radio wave for communication and having limited resources and limited computational power. The Topology of this network is changing very frequently because they are distributed in nature and self-configurable. The below figure 2 describe overview of watchdog.

3. WATCHDOG A COOPERATION SYSTEM IN VANET

Watchdog process to detect misbehaviour nodes and selfish node in vehicular. to Vehicular Delay-Tolerant Networks (VDTNs) were proposed as new kind of vehicular networks, whose design supports communications in environments where an end-to end path between the source and destination may not be available. Like other ad hoc networks, VDTNs rely their operation on cooperation between mobile nodes (e.g., vehicles), which are exploited to store-carry-and-forward bundles. VDTNs consider three kinds of nodes: mobile, terminal, and relay nodes. Mobile nodes move along path and may interact with the other two types of VDTN nodes. Mobile nodes move along paths and may interact with the other two types of VDTN nodes. Terminal nodes are usually placed at the edge of the VDTN network and are responsible for the heavy data processing and interaction with other networks (such as, the Internet), while relay nodes are placed at road intersections increasing the number of network contacts and storing a higher number of bundles that can be picked by any passing-by vehicle. Contrary to other vehicular networks, in VDTNs, each contact opportunity is processed in two phases: control plane and data (M. Raya and J.-P. Hubaux, 2007).

A Cooperative Watchdog System (CWS) is proposed to support network nodes to detect selfish nodes. To perform such task, CWS assigns a reputation score to each network node. Thus, each time nodes participate in a contact opportunity, the CWS updates their reputation score based on the considerations of three modules (classification, neighbour's evaluation, and decision). The classification module aims to categorize nodes into different types according to their reputation score.

The Cooperative value is then transmitted to the decision module in order to punish or reward nodes in function of their cooperative behaviour. The neighbour's evaluation module determines how neighbours evaluate a node's reputation on the network. This is accomplished by asking them their opinion about it. At the end of a contact opportunity, the decision module updates nodes reputation score based on the information transmitted by the other modules. With this approach, the CWS manages to classify, monitor, and act against such kind of nodes.

Address the problem of detecting misbehaving vehicles in Vehicular Ad Hoc Network (VANET) using Quality of Service Optimized Link State Routing (QoS-OLSR) protocol. According to this protocol, vehicles might misbehave either during the clusters' formation by claiming bogus information or after clusters are formed. A vehicle is considered as selfish or misbehaving once it over-speeds the maximum speed limit or under-speeds the minimum speed limit where such a behaviour will lead to disconnected network. As a solution, we propose a two-phase model that is able to motivate nodes to behave cooperatively during clusters' formation and detect misbehaving nodes after clusters are formed. Incentives are given in the form of reputation and linked to network's services to motivate vehicles to behave cooperatively during the first phase.

Misbehaving vehicles can still benefit from network's services by behaving normally during the clusters' formation and misbehave after clusters are formed. To detect misbehaving vehicles, cooperative watchdog model based on Dempster–Shafer is modeled where evidences are aggregated and cooperative decision is made. Simulation results show that the proposed detection model is able to increase the probability of detection, decrease the false negatives, and reduce the percentage of selfish nodes in the vehicular network, while maintaining the Quality of Service and stability.

4. WATCHDOG MONITORING IN VANET

Watchdog will generate a log [WatchDog.LOG] in the directory it is run from. This log contains information about when a process has stopped responding to a heartbeat, when it was restarted along with information about when a process has been manually shut down. The following are examples of Watchdog log entries in Table 1.

The monitoring of watchdog executing on-chip process. Two types are include in the monitoring of watch dog execution in on-chip process are On-Chip Watchdog to Monitor RTOS (Real-Time Operating System) Activity and second one is Validation of an On-Chip Watchdog for Embedded Systems. First one is the use of Real-Time Operating System (RTOS) became a mandatory condition to design safety-critical real-time embedded systems based on multicourse processors. At the same time, these systems are becoming more and more sensitive to transient faults originated from a large spectrum of noisy sources such as conducted and radiated Electromagnetic Interference (EMI) (Johnson. A. K. 2004). Therefore, the system's reliability degrades. Namely RTOS-Watchdog (RTOS-WD), was described in VHDL and is connected to the address busses between the cores and their local Cache memories. Present a hardware-based Infrastructure Intellectual Property (I-IP) core able to monitor the RTOS' activity in a multicore processor system-on-chip (MPSoC).

To detect RTOS task scheduling faults in a single-core processor, for RTOSs based on the Round-Robin and Preemptive scheduling algorithms. It is important to highlight that the RTOS-WD represents a generic passive solution and consequently does not interfere within the execution flow of the RTOS running by the system. A case-study based on the dual-core Plasma processor IP core running different test programs under the control of a typical preemptive RTOS was implemented. The case-study was prototyped in a Xilinx Virtex4 FPGA mounted on a dedicated platform (board plus control software) (S. Ben Dia, 2006). For validation, the MPSoC (Multicore Processor System-on-Chip**)** was exposed to conduct EMI. The obtained results demonstrate that the proposed approach provides higher fault coverage and reduced fault latency when compared to the native fault detection mechanisms embedded in the kernel of the RTOS.

Second one is Due to stringent constraints such as battery powered, high-speed, low-voltage power supply and noise exposed operation, safety-critical real-time embedded systems are often subject to transient faults originated from a large spectrum of noisy sources; among them, conducted Electromagnetic Interference (EMI) and radiation. Present the most recent results involving the validation analysis of hardware based intellectual property (IP) core, namely Real-Time Operating System - Guardian (RTOS-G). This is an on-chip watchdog that monitors the RTOS' activity in order to detect faults that corrupt tasks' execution flow in embedded systems running preemptive RTOS. The results demonstrate the proposed approach provides higher fault coverage and reduced fault latency when compared to the native (software) fault detection mechanisms embedded in the kernel of the RTOS (B. Nicolescu, 2006).

Table 1. Example of Watchdog Logs Entries

Watchdog LOG entry	***Description of entry***
01/27/0515:32:16 Started Watchdog	Watchdog has been Started and Stopped
01/27/05 15:36:38 Shutdown Watchdog	
01/27/05 15:33:56 w[1] WARNING: Process not running (DAD.exe)	Watchdog has detected that DAD is no longer running. When it is unable to start <check> the process, it will restart the application.
01/27/05 15:33:56 w[1] ERROR: Unable to start process (DAD.exe)	
01/27/05 15:35:06 w[1] WARNING: Process not running (DAD.exe)	Watchdog has detected that DAD is no longer running. Here it detected that the program was manually exited and shut down normally.
01/27/05 15:35:06 w[1] NORMAL : Detected normal exit (DAD.exe)	This message indicates that the heartbeat for the application cannot be found. Watchdog is starting an application that was not running when Watchdog was started.
01/27/05 15:38:32 w[1] WARNING: Unable to detect heartbeat (DAD.exe)	
01/27/05 15:38:51 w[2] NORMAL : Process started (Dist-WO.exe)	This entry will be entered if Watchdog terminates a program because the process is not responding to heartbeats.
01/27/05 15:39:19 w[2] WARNING: Killing process (dropbox.exe)	

5. WATCHDOG TIMER IN VANET

A method with Observer Pattern and Finite State Machine for watchdog implementation is proposed. By using Observer Pattern and Finite State Machine, the watchdog will be notified automatically when the program's state changes. Complex protection strategies can be applied based on the transition of different states. The simulation

indicates great improvement on reliability and maintainability of the program, especially for those complex programs with multi-task or multi-interrupt.

The Definition of observer pattern for one-to-many dependency between objects so that when one object changes state, all of its dependents are notified and updated automatically (E. O. Ochola, 2013). There are two kinds of class in the Observer Pattern which is "Subject" and "Observer" (Silva & L. Bolzani, 2011). The subject and observers define the one-to-many relationship. The observers are dependent on the subject such that when the subject's state changes, the observers get notified. Each subject has to implement three kinds of method, Register Observer, Remove Observer and Notify Observers.

The Framework of observer pattern in order to use Observer Pattern in the design of watchdog, Subject can be thought as a detector. The detector can be placed anywhere in the program, so that it can automatically monitor the state of the program. Furthermore, the original program logic can be replaced by the detector so that the state variable can be saved inside the detector. Because most of the time the watchdog program is independent of main program, a watchdog subject is needed in order to monitor the state of watchdog. Generally speaking there are at least two Observers, one is watchdog manager which is used to control the watchdog, and the other is watchdog simulation form which is used to refresh the state of main program (C. Oliveira, 2013). A finite state machine is a device, or a model of a device, which has a finite number of states it can be in at any given time and can operate on input to either make transitions from one state to another or to cause an output or action to take place. A finite state machine can only be in one state at any moment in time. The main purpose of the finite state machine is to decompose an object's behavior into easily manageable way which is called "state". However while the feed dog strategy becomes more sophisticated, much more than two states are needed for the watchdog. The watchdog must monitor on only the main loop, but also other critical locations such as working threads and interrupt functions so that the control logic of watchdog will become very complicated and hard to maintain (Sergio Marti, 2000) (T.HO, 2004). In order to solve the problem above, finite state machine can be used because the behavior of watchdog is only determined by current state. The program consists of two timer interrupts (Timer1, Timer2), two working threads (Thread1, Thread2), and two external interrupts (Input1, Input2). The timer interrupt and working thread all doing their own tasks and will pass the computation result as parameters to the main program. Two external interrupts will both receive an integer variable to deal with accordingly. The feed dog strategy is: (i) Stop feeding dog if one of the timers interrupts stop working. But it must resume feeding the watchdog if the timer interrupts returns to normal. (ii) The watchdog must not be fed if both the timer interrupts stop working, even if they return to normal afterwards. The program must be reset by watchdog to ensure the stability of the program. (iii) Once Thread1 stops working, main program must try to restore Thread1 first, and continue to feed the watchdog if it restores the Thread1 successfully, or stops feeding dog if not. Also it must resume feeding the watchdog if Thread1 returns to normal afterwards. (iv) Stop feeding dog if Thread2 stops working. But resume feeding dog if the Thread2 returns to normal. (v) The watchdog must not be fed if both the working threads stop working, even if they return to normal afterwards. (vi) Stop feeding dog if the variable received from Input1 is larger than the variable received from Input2, and will resume feeding the watchdog if input1 becomes equal or less than input2 afterwards. (vii) The priority of the external interrupts is higher than the timer interrupts and working threads as shown in fig 3. The states of finite state machine are Normal and Abnormal.

Table 2. Finite State Machine's State Transition

Current State	Condition	State Transition
Normal State	Timer1 has errors	Timer1ErrorState
Normal State	Timer2 has errors	Timer2ErrorState
Normal State	Timer1 has errors	Timer1ErrorState
Timer1ErrorState	Timer1 has errors	Timer1_2ErrorState
Timer2ErrorState	Timer2 has errors	Timer2_21ErrorState
Timer1ErrorState	Timer1 restore normal	Normal State
Input Normal State	If input1 bigger than input2	InputErrorState

Normal State, which means that the program is in abnormal state, and continue feeding dog. Timer1ErrorState, which means that Timer1 has stopped working, and stop feeding dog. Timer2ErrorState, which means that Timer2 has stopped working, and stop feeding dog. Timer1_2ErrorState, which means that both Timer1 and Timer2 have

stopped working, and stop feeding dog. Thread1RestartState, which means that Thread1 has stopped working for the first time, and will try to restart Thread1.

Thread1ErrorState, which means that Thread1 has stopped working, and stop feeding dog. Thread2ErrorState, which means that Thread2 has stopped working, and stop feeding dog. Thread1_2ErrorState, which means that both Thread1 and Thread2 have stopped working, and stop feeding dog. Input normal state, which means that input1, is equal or less than input2, and continue feeding dog.

Input error state, which means that Input1 is larger than Input2, and stop feeding dog. Because the priority of input error state is higher than the other states, two interfaces will be needed, which are: Instate, which is inherited by all the state other than input error state. I Global State, which has higher priority than State, and is inherited by input error state only. The finite state machine's state transition table is show in table.

Figure 3. Watchdog simulation

6. CONCLUSION

The watchdog techniques are a diagnosis mechanism useful to detect a cooperative system for misbehaviour node and the efficiently evaluated in the watchdog timer method observer pattern and finite state machine greatly improve the reliability and maintainability. This paper mainly focus the three different watch dog mechanisms namely Timer, Monitoring and Cooperative method used in the VANET environment. These methods were analysed and studied in order to know the impact in the VANET. It is more useful to detect and remove the malicious node in the VANET to provide the efficient and reliable data transmission between them.

REFERENCES

M. Raya and J.P. Hubaux (2007). Securing vehicular ad hoc networks. Journal of Computer Security, 15(1), 39–68.

Bhois, K. (2014). Vehicular communication a survey. IET Networks, 3(3):204–217.

B. Nicolescu, N. Ignat, Y. Savaria, G. Nicolescu (2006). Analysis of Real-Time Systems Sensitivity to Transient Faults Using Micro Kernel. IEEE Transactions on Nuclear Science, 53(4).

C. Oliveira, J. Benfica, L. Bolzani Poehls, F. Vargas, J. Lipovetzky, A. Lutenberg, E. Gatti, F. Hermandez, A. Boyer (2013). Reliability analysis of an on-chip watchdog for embedded systems exposed to radiation and EMI. 9th Int. Workshop on Electromagnetic Compatibility of Integrated Circuits (EMC Compo).

Douglass, B. P. (2011). Design Patterns for Embedded Systems in C. An Embedded Software Engineering Toolkit, U.K.: Elsevier

Ochola, E. O., Eloff, M. M. van der Poll, & J. A. (2013). The Failure of Watchdog Schemes in MANET Security: A Case of an Intelligent Black-Hole. Proceedings of the SAICSIT 2013 Conference.

Li, F., Wu, J. (2009). Frame: an innovative incentive scheme in vehicular networks. IEEE International Conference on Communications.

Li, G., Semerci, M., Yener, B. & Zaki M. J. (2011). Graphs Classification via topological and label attributes,. Workshop on Mining and Learning with Graphs,.

Hongmei Deng, W. L. (2002). Routing Security in Wireless Ad Hoc network. IEEE Communication Magazine, 40.

Johnson, A. K. (2004). Modelling mobility for vehicular ad-hoc networks. Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, ACM.

S. Ben Dia, R. R. (2006). Electromagnetic Compatibility of Integrated Circuits – Techniques for Low Emission and Susceptibility. Springer.

S. Lee, G. P. (2007). Secure incentives for commercial ad dissemination in vehicular networks. Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing.

Sergio Marti, T. J. (2000). Mitigating routing misbehaviour in mobile ad hoc networks. Proc. Annual International Conference on Mobile Computing and Networking.

Silva, D., & L. Bolzani, F. V. (2011). An intellectual property core to detect task scheduling-related faults in RTOS-based embedded systems. IEEE 17th Int. On-Line Testing Symposium (IOLTS), 19-24.

Surana, K.A., Rathi, S.B., Thosar, T.P. & Snehal Mehatre, (2012). Securing Black Hole Attack in Routing Protocol AODV in MANET with Watchdog Mechanisms. World Research Journal of Computer Architecture, 1(1)