

BETTER ROUTING WITH OVERCROWDING VARIETY IN WAHNET

RAJULAPATI NAGARJUNA ^{#1}, BUJJIBABU LINGAMPALLI ^{#2}

^{#1} M.Tech Scholar , Department of Computer Science and Engineering,
D.N.R College of Engineering and Technology,
Balusumudi, Bhimavaram, India.

^{#2} Assistant Professor , Department of Computer Science and Engineering,
D.N.R College of Engineering and Technology,
Balusumudi, Bhimavaram, India.

ABSTRACT

As we all know that wireless sensor networks are achieving a lot of user's attention towards its usage in terms of efficiency, accuracy and quick response time. Recently wireless adhoc networks(WAHNET's) is also becoming more familiar by combining the advantages of both mobile ad-hoc networks (MANETs) and infrastructure wireless networks because of their ultra-high performance .For this wireless adhoc networks we must try to adopt an efficient routing protocol for data transfer with high network capacity and scalability. However, till now almost various routing protocols try to combine the ad-hoc transmission mode with the advanced or modern cellular transmission mode, by inheriting the limitations that are available in the ad-hoc data transmission. The main problem what we try to observe in this current application is how to route the packets across a multi-hop network consisting of multiple sources of traffic and wireless links while ensuring bounded expected delay. The data is initially divided into packets and where each and every packet transmission can be overheard by a sequence of intermediate nodes that are available in the router among which the next relay is selected opportunistically. The main challenge in the design of minimum-delay routing policies is balancing the trade-off between routing the packets along the shortest paths to the destination and distributing the traffic according to the maximum backpressure. Combining important aspects of shortest path and backpressure routing, this paper provides a systematic development of a new distributed opportunistic routing policy with congestion diversity (D-ORCD) protocol. This proposed protocol is proved to be best with single destination, to calculate the over delay under admissible traffic. By conducting various experiments on our proposed protocol, we finally came to a conclusion that our proposed approach is best in sending dedicated packets to valid destination nodes under no congestion delay within the network.

Key Words:

, Mobile Ad hoc Network, Data Scalability ,Data Packets, Distributed Network, Congestion Control, Opportunistic Routing.

I. INTRODUCTION

A Wireless Sensor Network (WSN) is a collection of several nodes ranges from a few to several hundreds and even thousands of nodes, where each and every group of nodes is connected either to single sensor or group of sensors. Sensor network typically has several parts which are clearly shown in figure 1.

1. A radio transceiver device with an inbuilt internal antenna or device connected to an external antenna.
2. A microcontroller
3. An electronic circuit board for interfacing mainly with the sensors and an energy source, usually a battery or an embedded form of energy harvesting.

Sensor may vary in size when compared with different type of sensors just like of a shoebox down to the size of a grain of dust. The amount for purchasing of single sensor nodes is similarly variable in its price, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. While deploying any sensor some valuable resources like energy, memory, computational speed and communication bandwidth mainly depends on size and cost of the sensor what we use. The topology (I.e. arrangement of nodes) of the WSNs can vary from a basic star network to an advanced mesh network. The propagation technique between the nodes of the wireless network can be routing or flooding [1], [2].

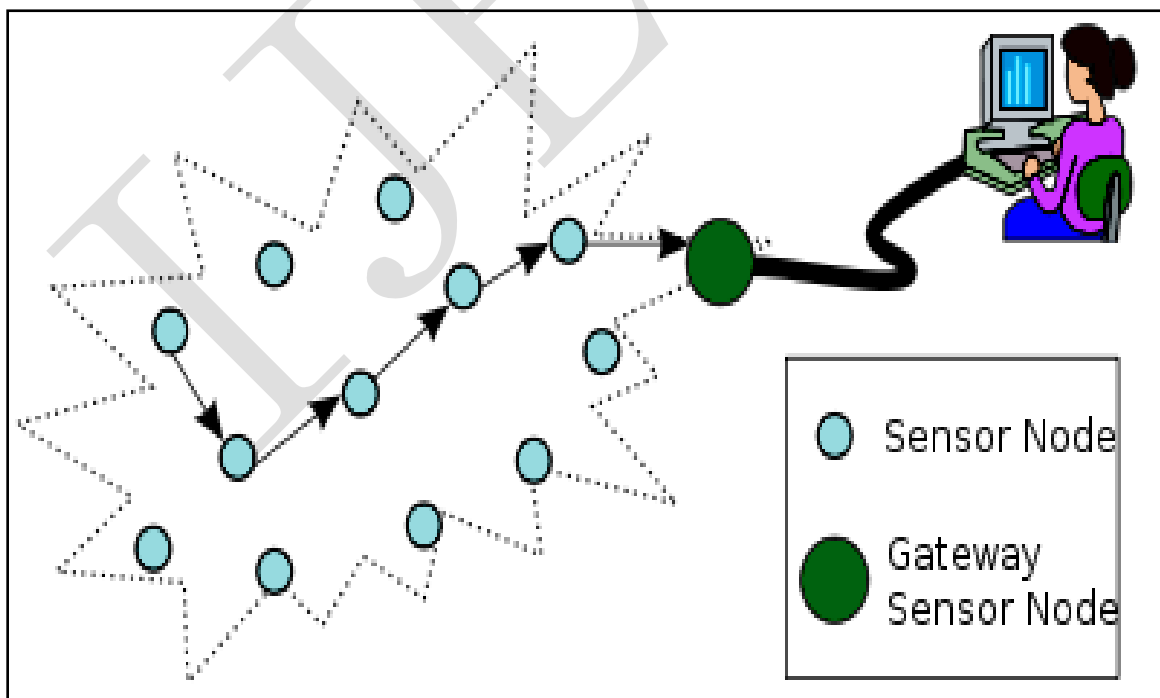


FIGURE.1. REPRESENTS THE TYPICAL MULTI-HOP WIRELESS SENSOR

NETWORK ARCHITECTURE

Recently with the huge usage of wireless sensor networks in a variety of applications, a lot of research efforts have been made to develop sensor hardware and network architectures in order to effectively deploy WSNs. For the general purpose network deployment, normal WSN cannot able to fulfill the needs like sensing range, transmission range, and bandwidth range for sensing the data remotely. To achieve this, it is very crucial to identify the impact parameters of network on its performance with respect to the application specifications. In CSE and telecommunications, wireless sensor networks are an active research area with numerous workshops and conferences arranged each year for the improvement of its performance [3],[4].

Owing to these unique characteristics, MANET is becoming more and more widely implemented in the industry [5]. However, considering the fact that MANET is popular among critical mission applications, network security is of vital importance. Unfortunately, the open medium and remote distribution of MANET make it vulnerable to various types of attacks. For example, due to the nodes' lack of physical protection, malicious attackers can easily capture and compromise nodes to achieve attacks. In particular, considering the fact that most routing protocols in MANETs assume that every node in the network behaves cooperatively with other nodes and presumably not malicious [5], attackers can easily compromise MANETs by inserting malicious or non-cooperative nodes into the network. Furthermore, because of MANET's distributed architecture and changing topology, a traditional centralized monitoring technique is no longer feasible in MANETs. In such case, it is crucial to develop an intrusion-detection system (IDS) specially designed for MANETs. Many research efforts have been devoted to such research topic [6].

Although there was a lot of conventional routing algorithms that are available in the real world environment, no routing algorithm fulfilled in achieving high level of data accuracy and integrity in sending the dedicated packets to the valid destination with no loss. So a new routing technique like opportunistic routing for a multi-hop wireless ad hoc networks has been proposed to overcome the deficiencies that are available in primitive routing algorithms [7]–[10]. Opportunistic routing reduces the impact of very weak wireless links by exploiting the broadcast nature of wireless transmissions and the path diversity. Two well known authors in [11] provided a modified Markov formula for opportunistic routing and a novel framework for many versions of opportunistic routing [12],[13], with the variations due to the authors' choices of costs.

In particular, it is shown that for any packet, the optimal routing decision, in the sense of minimum cost or hop-count, is to select the next relay node based on an index. This index is equal to the expected cost or hop-count of relaying the packet along the least costly or the shortest feasible path to the destination. When multiple streams of packets are to traverse the network, however, it might be desirable to route some packets along longer or more costly paths, if these paths eventually lead to links that are less congested. More precisely, as noted in [6], [7], the opportunistic routing schemes in [7]–[10] can potentially cause severe congestion and unbounded delay (see the examples given in [6]). In contrast, it is known that an opportunistic variant of backpressure [8], diversity backpressure routing (DIVBAR) [7] ensures bounded expected total backlog for all stabilizable arrival rates. To ensure throughput optimality (bounded expected total backlog for all stabilizable arrival rates), backpressure-based algorithms [12], [13]

do something very different from [7]–[11]: rather than using any metric of closeness (or cost) to the destination, they choose the receiver with the largest positive differential backlog (routing responsibility is retained by the transmitter if no such receiver exists). This very property of ignoring the cost to the destination, however, becomes the bane of this approach, leading to poor delay performance in low to moderate traffic (see [9]). Other existing provably throughput optimal routing policies [9]–[12] distribute the traffic locally in a manner similar to DIVBAR and hence, result in large delay.

II. RELATED WORK

In this section we will mainly discuss about the related work that was carried out in order to prove the performance of our proposed D-ORCD protocol for data transfer in dynamic routing path and in order to avoid the congestion delay during data transfer. Here in this section we mainly discuss about the working nature of intrusion detection system for identifying the attackers during data transfer inside the network. Also we will look after the primitive technique like hash based message authentication codes in order to provide security for the data during communication. Now let us look about that in detail as follows:

MAIN MOTIVATION

An Intrusion detection system (IDS) is internet software which is mainly deployed on the hardware designed to detect any unwanted attempts to access, manipulating, and/or disabling of computer mainly through a network [14],[15]. An intrusion detection system is mainly used to identify several types of malicious behaviors that can easily compromise the security and trust of a computer system. Some of the attacks include network attacks against vulnerable services, host based attacks such as privilege escalation attack, unauthorized logins attack and attempting to access some invalid files like viruses and worms[13].

An IDS is mainly composed of several components:

- 1. SENSORS:** This is used for generating security events.
- 2. CONSOLE:** Which is used to monitor events and alerts, while controlling the sensors?
- 3. CENTRAL ENGINE:** Which is mainly used for recording the events logged by the sensors in a database and use a system of rules to generate alerts from security events received.

Intrusion detection system software is basically executed and deployed in wireless sensor networks. The development of such a variety of wireless sensor networks was originally motivated by military applications such as battlefield surveillance and attacker identification through sensor which is clearly shown in figure 2. However, now a day's these wireless sensor networks is also used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control. From the below figure 2, we can clearly find out that wireless sensor networks are used by a military networks, where

the information is passed through the satellite and this information can be sent or received through the surface sink, surface station, onshore sink. All the individual sensors are formed as clusters with the help of a gateway that was located within a set of sensors and they all will communicate within their sensing region.

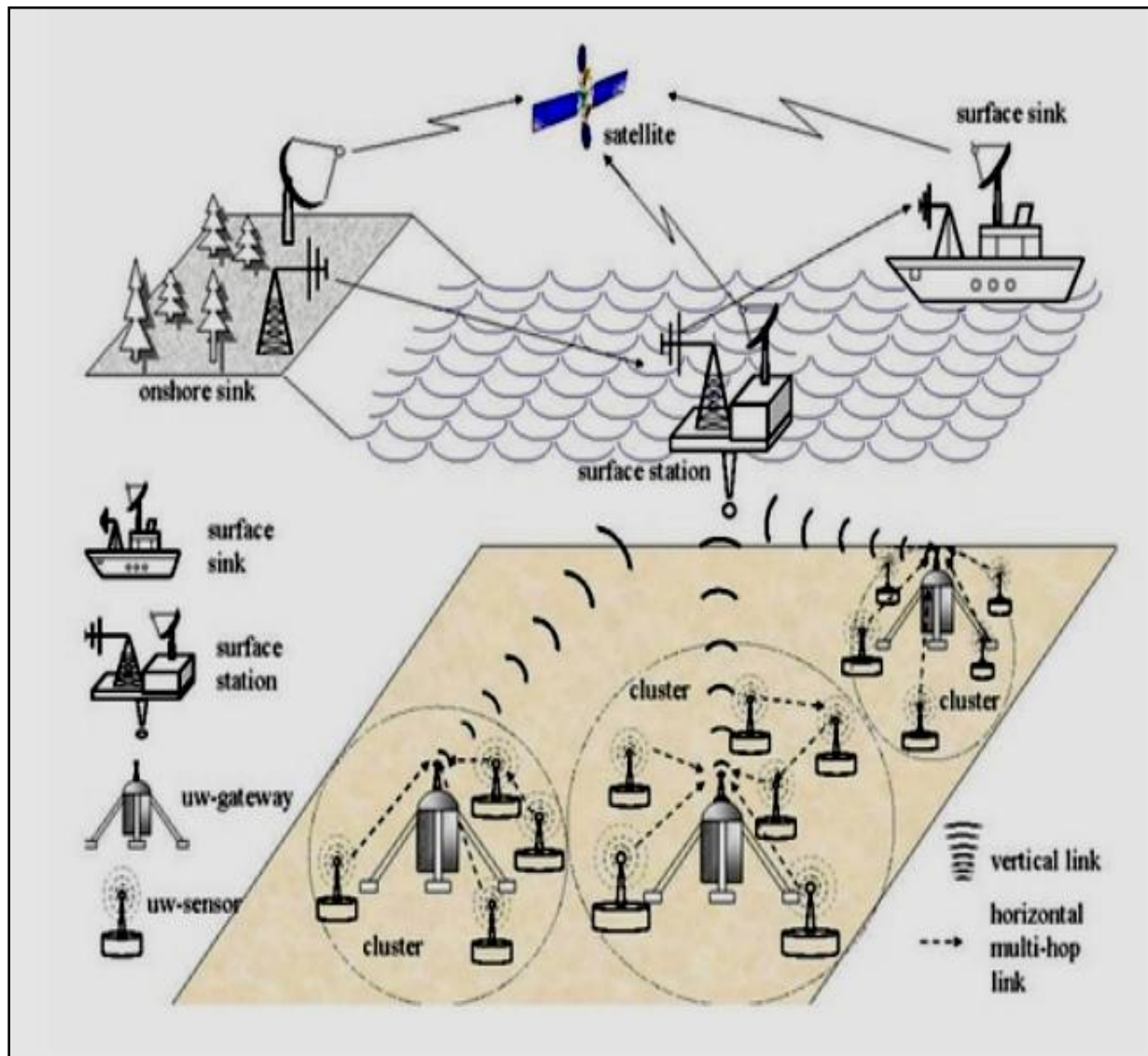


FIGURE.2. REPRESENTS THE MILITARY APPLICATION WHICH USES WIRELESS SENSOR NETWORK

HASH BASED MESSAGE AUTHENTICATION CODES

In cryptography, one among the best methods is a keyed-hash message authentication code (HMAC). This is a specific construction for calculating a message authentication code (MAC) which involves a cryptographic hash function in combination with a secret cryptographic key. This HMAC is used to verify the message integrity and also the

authentication of the message. Generally for calculating the HMAC function we can use any of the two cryptographic hash functions like MD5 or SHA-1. If MD5 algorithm is used for generating the MAC we termed it as HMAC-MD5, and if the hash function is generated by SHA1 then it is termed as HMAC-SHA1 accordingly. Generally the strength of any cryptographic algorithm like HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output, and on the size and quality of the key[16]. An iterative hash function breaks up a message into blocks of a fixed size and iterates over them with a compression function. Generally if we consider an example to differentiate the block size of HMAC, they generally operate on 512-bit blocks. The size of the output of HMAC is the same as that of the underlying hash function (128 or 160 bits in the case of MD5 or SHA-1, respectively), although it can be truncated if desired.

IJETED

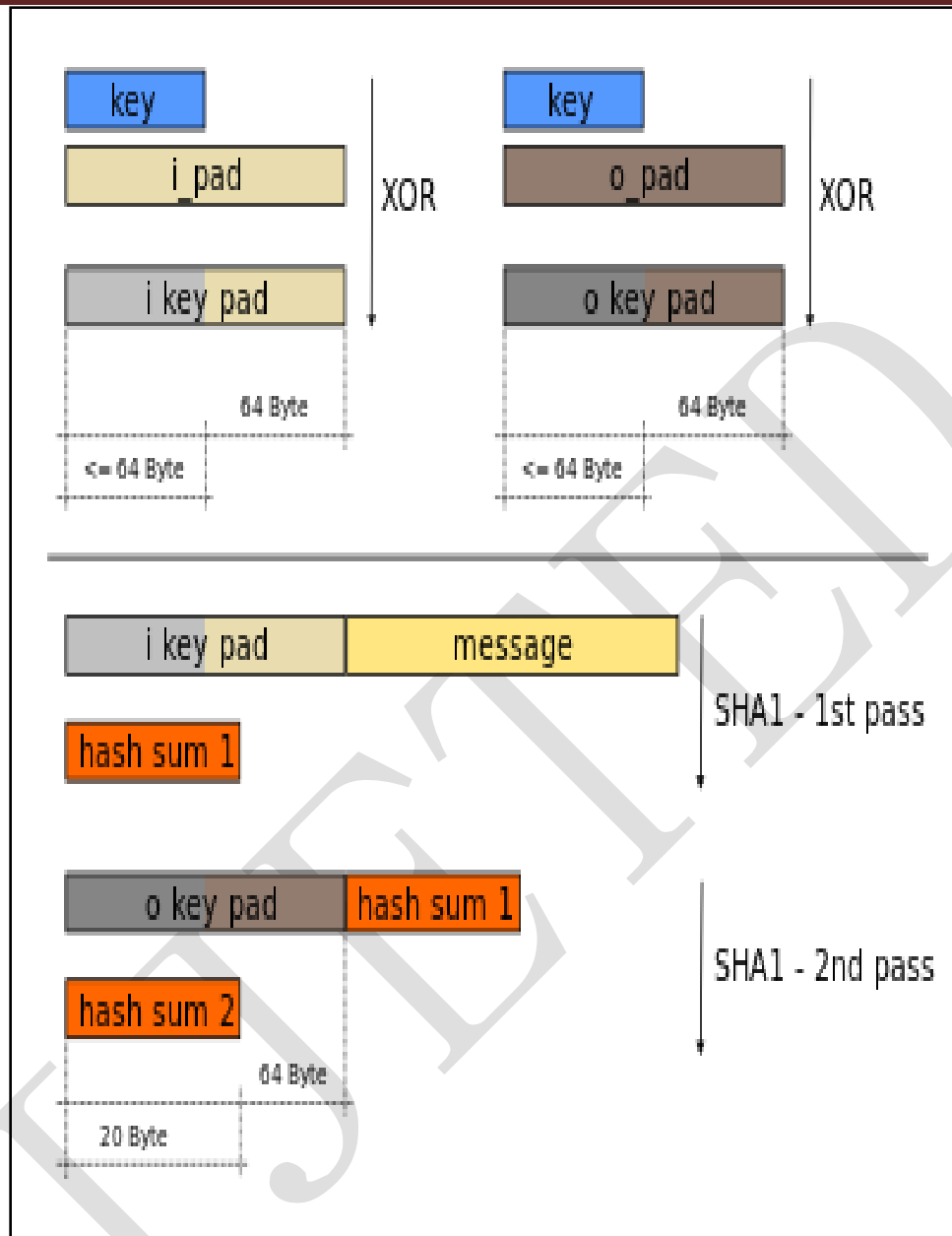


FIGURE 3. REPRESENTS THE HMAC GENERATION BY USING SHA1 HASH ALGORITHM

From the above figure 3, we can clearly get an idea of how the hash function is generated for the message which is transmitted between sender and receiver. Generally SHA1 takes input parameter as data and it will generate a hexa decimal code at the sender side. Once the message has been transmitted through the network to the receiver side, the message will be in an encrypted manner and it is authentic. So if user wants to decrypt the data he should substitute his identity so that then the data will be decrypted.

III. PROPOSED NOVEL DISTRIBUTED OPPORTUNISTIC ROUTING POLICY WITH CONGESTION DIVERSITY (D-ORCD) PROTOCOL

In this section we will find out the proposed Novel D-ORCD protocol that was used in current application in order to send dedicated packets from a valid source to destination with no packet loss under network congestion during data transmission.

MAIN MOTIVATION

We describe the guiding principle behind the design of Distributed Opportunistic Routing with Congestion Diversity (D-ORCD). In this section we mainly propose a time-varying distance vector, which enables the network to route packets through a neighbor with the least estimated delivery time. D-ORCD opportunistically routes a packet using three stages of: (a) transmission, (b) acknowledgment, and (c) relaying. During the transmission stage, a node transmits a packet. During the acknowledgment stage, each node that has successfully received the transmitted packet, sends an acknowledgment (ACK) to the transmitter node. D-ORCD then takes routing decisions based on a congestion-aware distance vector metric, referred to as the *congestion measure*. More specifically, during the relaying stage, the relaying responsibility of the packet is shifted to a node with the least congestion measure among the ones that have received the packet. The congestion measure of a node associated with a given destination provides an estimate of the best possible draining time of a packet arriving at that node until it reaches destination. Each node is responsible to update its congestion measure and transmit this information to its neighbors which is shown clearly in figure 4. Next, we detail D-ORCD design and the computations performed at each node to update the congestion measure.

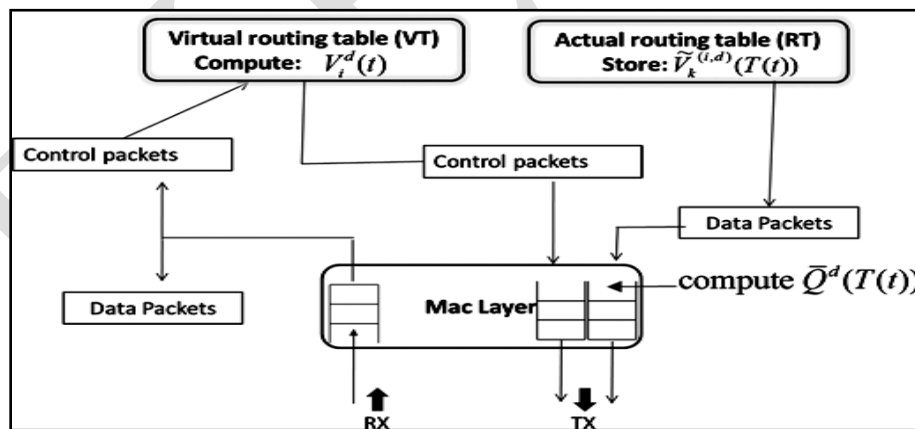


FIGURE 4. REPRESENTS THE OPERATION OF D-ORCD PROTOCOL

D-ORCD relies mainly on a routing table at each node to determine the next best hop. The routing table at node consists of a list of neighbors $N(i)$ and a structure consisting of estimated congestion measure for all neighbors in $N(i)$ associated with different destinations. The routing table acts as a storage and decision component at the routing layer. The routing table is updated using a “virtual routing table” at the end of every “computation cycle”: an interval T_c of units of time. To update virtual routing table, during the progression of the computation cycle

the nodes exchange and compute the temporary congestion measures. The temporary congestion measures are computed in a fashion similar to a distributed stochastic routing computation of [4] using the backlog information at the beginning of the computation cycle (generalizing the computations of distributed Bellman-Ford). We conceptualize this in terms of the virtual routing table updating and maintaining these temporary congestion measures. We assume that each node has access to a common global time to ensure that the nodes update the routing table roughly at the same time.

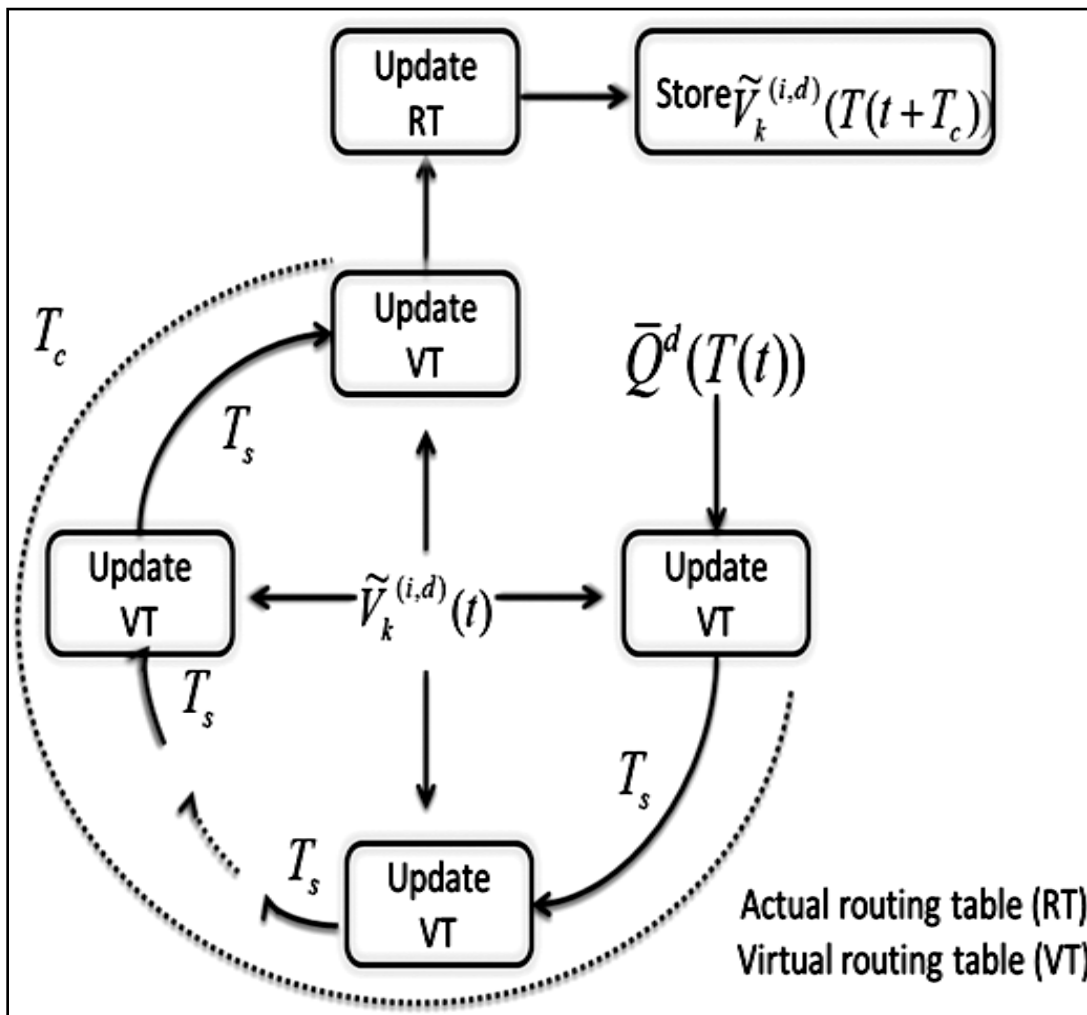


FIGURE 5. ACTUAL ROUTING TABLE IS UPDATED EVERY T_c UNITS OF TIME WHILE VIRTUAL ROUTING TABLE IS UPDATED AFTER RECEIVING ANY CONTROL PACKET.

We denote the temporary congestion measure associated with node $i \in \Omega$ at time t and destination $d \in \Omega$ as $V_i^d(t)$. Each node i computes $V_i^d(t)$ based on congestion measures $\tilde{V}_k^{(i,d)}(t)$ obtained via periodic communication with its neighbours $k \in \mathcal{N}(i)$ and the queue backlog at the start of the computation cycle. D-ORCD stores these temporary congestion measures $\{V_i^d(t)\}_{d \in \Omega}$ and $\{\tilde{V}_k^{(i,d)}(t)\}_{d \in \Omega, k \in \mathcal{N}(i)}$ in the virtual routing table. More precisely, node i periodically computes its own congestion measure and subsequently advertises it to its neighbors using control packets at intervals of $T_s \leq T_c$ seconds. Finally the actual routing table is updated using the entries in the virtual routing table after every T_c seconds. The sequence of operations performed by D-ORCD are shown in Figs. 4 and 5.

Meanwhile, for routing decisions, node i uses the entries in the actual routing table (updated at the end of the last computation cycle): Let $T(t) = \max_n \{nT_c : nT_c \leq t, n \in \mathbb{Z}\}$ be the ending time of the latest computation cycle; node i stores $\tilde{V}_k^{(i,d)}(T(t))$ in the actual routing table and selects the next best hop $K_{D-ORCD}^{(i,d)}$ to minimize the packet's draining time, i.e., $\tilde{V}_k^{(i,d)}(T(t))$.

Next, we describe the distributed computations performed during each computation cycle.

IV. IMPLEMENTATION PHASE

Implementation is the stage where the theoretical design is converted into programmatically manner. In this stage we will divide the application into a number of modules and then coded for deployment. We have implemented the proposed concept on Java programming language with JSE as the chosen language in order to show the performance this proposed NDTR protocol. The front end of the application takes Java Swings, AWT and Socket Programming and as a Back-End Data base we took MS-ACCESS data base. The application can be executed either on a single PC or it can be executed on multiple PC's all connected over a LAN. The application is divided mainly into following 5 modules. They are as follows:

1. Service Provider Module
2. Adhoc Router Module
3. Network Construction Module
4. Receiver Module
5. Node Failures Module

1. SERVICE PROVIDER MODULE

In this module, the service provider will browse the data file path and then send to the particular receivers. Service provider will send their data file to Adhoc router and router will

connect to networks, in a network smallest distance node will be activated and send to particular receiver (A, B, C...). And if any jammer node will found, then service provider will reassign the energy for node.

2. ADHOC ROUTER MODULE

The **Adhoc** Router manages a multiple networks (network1, network2, network3, and network4) to provide data storage service. In network n-number of nodes (n1, n2, n3, n4...) are present, in networks every node consists of distance and energy. In a network shortest distance node will communicate first. The service provider can assign energy for node, view energy for all networks and node history details (view routing path, view boundary nodes, view jamming nodes & view total time delay) in router. Router will accept the file from the service provider and then it will connect to different networks; the all networks are communicates and then send to particular receiver. In a router we can view time delay, jammed nodes and also routing path.

3. NETWORK CONSTRUCTION MODULE

In this module the networks (network 1, network 2, network 3 and network 4) consists of n-number nodes. In networks every node consists of distance and energy. In a network shortest distance node will communicate first. The node consists of lesser energy then that node will be jammed by the jammers. And then it will forward to next lesser distance node within the network. In a network last node will be considered as boundary node.

4. RECEIVER /END USER MODULE

In this module, the receiver can receive the data file from the service provider via Adhoc router. The receivers receive the file by without changing the File Contents. Users may receive particular data files within the network only.

5. NODE FAILURES MODULE

In this system, the lesser energy node will be considered as a failure node. Once the failure became active, affected nodes lost their neighbors partially or completely, lost all of their neighbors and became failure nodes.

V. CONCLUSION

In this paper, we for the first time designed and evaluated the performance of Novel distributed opportunistic routing policy with congestion diversity (D-ORCD) by combining the important aspects of shortest path routing with those of backpressure routing. Under this policy packets are routed according to a rank ordering of the nodes based on a congestion measure. Initially the data is divided into packets and where each and every packet transmission can be overheard by a sequence of intermediate nodes that are available in the router among which the next relay is selected opportunistically. The main challenge in the design of minimum-delay routing policies is balancing the trade-off between routing the packets along the shortest paths to

the destination and distributing the traffic according to the maximum backpressure. Combining important aspects of shortest path and backpressure routing, this paper provides a systematic development of a new distributed opportunistic routing policy with congestion diversity (D-ORCD) protocol. This proposed protocol is proved to be best with single destination, to calculate the over delay under admissible traffic. By conducting various experiments on our proposed protocol, we finally came to a conclusion that our proposed approach is best in sending dedicated packets to valid destination nodes under no congestion delay within the network.

VI. REFERENCES

- [1] H. Wu, C. Qiao, S. De, and O. Tonguz. Integrated cell and ad hoc relaying systems: iCAR. J-SAC, 2001.
- [2] Y. H. Tam, H. S. Hassanein, S. G. Akl, and R. Benkoczi. Optimal multi-hop cellular architecture for wireless communications. In Proc. of LCN, 2006.
- [3] Y. D. Lin and Y. C. Hsu. Multi-hop cellular: A new architecture for wireless communications. In Proc. of INFOCOM, 2000.
- [4] P. K. McKinley, H. Xu, A. H. Esfahanian, and L. M. Ni. Unicastbased multicast communication in wormhole-routed direct networks. TPDS, 1992.
- [5] P. T. Oliver, Dousse, and M. Hasler. Connectivity in ad hoc and hybrid networks. In Proc. of INFOCOM, 2002.
- [6] H Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu. Ucan: A unified cell and ad-hoc network architecture. In Proc. of MOBICOM, 2003.
- [7] P. Larsson, "Selection diversity forwarding in a multihop packet radio network with fading channel and capture," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 4, pp. 47–54, Oct. 2001.
- [8] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for Ad Hoc and sensor networks: Multihop performance," *IEEE Trans. Mobile Comput.*, vol. 2, no. 4, 2003.
- [9] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, pp. 33–44, Oct. 2005.
- [10] C. Lott and D. Teneketzis, "Stochastic routing in ad hoc networks," *IEEE Trans. Autom. Contr.*, vol. 51, pp. 52–72, Jan. 2006.
- [11] S. Jain and S. R. Das, "Exploiting path diversity in the link layer in wireless ad hoc networks," in *Proc. WoWMoM*, 2005, pp. 22–30.

[12] P. Gupta and T. Javidi, "Towards throughput and delay optimal routing for wireless ad hoc networks," in *Proc. Asilomar Conf.*, 2007, pp.249–254.

[13] M. J. Neely and R. Uргаonkar, "Optimal backpressure routing for wireless networks with multi-receiver diversity," *Ad Hoc Networks*, vol. 7,no. 5, pp. 862–881, Jul. 2009.

[14] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Contr.*, vol. 37, no. 12,pp. 1936–1949, Aug. 1992.

[15] S. Sarkar and S. Ray, "Arbitrary throughput versus complexity tradeoffs in wireless networks using graph partitioning," *IEEE Trans. Autom. Contr.*, vol. 53, no. 10, pp. 2307–2323, Nov. 2008.

VI. ABOUT THE AUTHORS

RAJULAPATI NAGARJUNA is currently pursuing his 2 years M.Tech in Department of Computer Science and Engineering at D.N.R College of Engineering and Technology, Balusumudi, Bhimavaram, India. His area of interest includes Cloud Computing and Computer Networks.

BUJJIBABU LINGAMPALLI is currently working as an Assistant Professor in Department of Computer Science and Engineering at D.N.R College of Engineering and Technology, Balusumudi, Bhimavaram, India. He has more than 4 years of teaching experience in engineering colleges. His research interest includes Network Security, Computer Organization, Advance Data Structures