International Journal of Engineering Science Invention Research & Development; Vol. IV, Issue III, SEPTEMBER 2017
www.ijesird.com, E-ISSN: 2349-6185

SECURE AND RELIABLE ROUTING PROTOCOLS IN MULTI-HOP WIRELESS NETWORKS USING MANET

Thanikonda Madhavi[1], Dr.ch.Balaswamy[2]
M.Tech (DECS)[1] , Professor & HOD[2], Department of ECE,QIS College of Engineering and Technology
Pondur Road, Vengamukkapalem, Ongole- 523272, AP
tanikondamadhavi2@gmail.com[1] , ch.balaswamy7@gmail.com[2]

***Abstract*— In multi-hop wireless networks, the mobile nodes usually act as routers to relay packets generated from other nodes. However, selfish nodes do not cooperate but make use of the honest ones to relay their packets, which has negative effect on fairness, security and performance of the network. In propose a novel incentive mechanism to stimulate cooperation in multi-hop wireless networks. Fairness is achieved by using credits to reward the cooperative nodes. The overhead is significantly reduced by using a cheating detection system (CDS) to secure the payment. Extensive security analysis demonstrates that the CDS can identify the cheating nodes effectively under different cheating strategies. Simulation results show that the overhead of the proposed incentive mechanism is incomparable with the existing ones. The circulated design makes it difficult to build a exceedingly secure and dependable yet insubstantial data storage scheme. on top of the one dispense, sensor information are subject to not only Byzantine failures, but also lively pollution attacks, as along the time the adversary may modify pollute the stored data by compromise individual sensors. On the erstwhile hand, the resource-constrain environment of WSNs precludes the applicability of overload for security designs. To address the challenge, in this object propose framework based integrated dynamic data storage scheme with dynamic reliability guarantee.**

***Keywords*-WSN, CDS, MANET, selfish node.**

I. INTRODUCTION

A wireless sensor network (WSN) is a self-organized wireless network system consisting of a number of sensors, which gather information from their surrounding environments and transmit it to a data sink or a base station (BS). In WSN applications, the main objective is to monitor and collect sensor data and then transmit the data to the BS. Sensors in different regions of the field can collaborate in data collection, and provide more accurate reports about their local regions. Most deployed WSNs measure physical phenomena like temperature, pressure, humidity, or location of objects to improve the fidelity of reported measurements, and data aggregation reduces the communications overhead in the network, leading to significant energy savings. The characteristics of low cost, low-power, and multifunctional sensors have rendered WSNs very attractive.[2][3] Nowadays, with the development of cloud technology WSNs have been rapidly deployed many practical applications, including home security, battlefield surveillance, monitoring movement of wild animals in the forest, healthcare applications etc. Recently, extensive research efforts have been dedicated to explore new roles for WSNs in remote and inaccessible environments. In a sensor network, each node is both a sensor and a router, and its computing capability, storage capacity and communications ability are limited. Moreover, in many WSN applications, sensor nodes are deployed in harsh environments, which make the replacement of failed nodes either difficult or expensive. Thus, in many scenarios, a wireless node must operate without battery replacement for an extended period of time. HEED ensures that only one CH within a certain range achieves the uniform CH distribution across the network. Therefore, the head nodes consume a great deal of power in the HEED protocol, resulting in their quick depletion of energy. The EECS protocol leads to a fair distribution for cluster heads, in which cluster heads are selected based on the residual energy and location of nodes. In EECS, a competitive algorithm is suggested for the CH selection phase.[10][11][13]

II. EXISTING SYSTEM

A recent advance in sampling theory, known as compressive sensing (CS) provides a promising solution to reducing the required number of measurements to represent the original signal, where high-dimensional sparse signals can be recovered from highly incomplete low-dimensional vectors. Taking advantage of the combination of CS and network coding, a variety of DDS schemes such as CStorage, ICStorage, and compressive network coding based distributed data storage (CNCDS) have been proposed to improve the energy efficiency. Transmitting the linear projections of sensor readings, these DDS schemes can greatly reduce the number of transmissions and receptions. However, they have been designed by only taking into account the spatial correlation among sensor readings from geographically neighboring nodes. Note that sensor readings in a WSN from natural phenomena generally exhibit correlations in both spatial and temporal domains.

In this letter, we exploit both spatial and temporal (spatiotemporal) correlations among sensor readings to further improve DDS energy efficiency. The new approach is referred to as spatiotemporal compressive network coding (ST-CNC). Specifically, the projections of sensor readings within several consecutive time slots are first generated, and then linearly combined along multiple random paths in a network coding based manner. Prove that the equivalent sensing matrix can be constructed by the product of temporal and spatial sensing matrices. Consequently, CS can be employed to jointly recover the original sensor reading block generated from the entire sensor node across multiple time slots by visiting only an arbitrary small subset of sensor nodes. In other proposed scheme can jointly compress and recover a spatiotemporal sensor reading block, whereas existing DDS-CS schemes only repeat the previous operation for data handling in each time slot. However, the increased data dimension of the CS framework could result in degraded recovery performance. To solve this problem, we derive a separable sensing operator that allows the spatial matrix and temporal sensing matrix to be optimized separately. Simulation results demonstrate that compared with the conventional DDS schemes, the proposed scheme can significantly reduce the number of transmissions and receptions with almost the same recovery performance, resulting in much higher energy efficiency.[5][7]

A. SPATIOTEMPORAL DDS PROTOCOLS

The proposed ST-CNC scheme extends the work in and considers a more realistic scenario, where sensor readings across all the nodes exhibit both spatial and temporal correlations. It starts with the initialization of each node, and then encodes and disseminates the sensor readings in a two dimensional CS based manner. Finally, a Kronecker structure based mathematical model is formulated for data recovery. Next we describe the operation of each stage in detail.

(S-I) Initialization.

Given a compressible sensor reading block **X** ∈ RN×L and a temporal sensing matrix Φt ∈ RG×L(G_L), each node forms its initial transmission packet. Specifically, the packet of the n-th node, denoted by **r**n, has three independent components given by

$$\mathbf{r}_n = \begin{cases} \mathbf{r}_n\{1\} = [\phi_{n,n}] \\ \mathbf{r}_n\{2\} = [n] \\ \mathbf{r}_n\{3\} = \phi_{n,n}\mathbf{\Phi}_t\mathbf{X}^T_{n,:}. \end{cases}$$

The first component in **r**n is a random coefficient being +1 or −1 with equal probability, the second one contains the node index n, and the third one is linear projection of **X**Tn ,: on Φt .

(S-II) Broadcasting.

Each sensor node randomly selects itself as a source node with a probability P0, and it is assumed that there are Ns(Ns < N) source nodes and their packets are broadcast to neighboring nodes. If a node p(p ∈ [1,N]) receives a packet from a neighboring node q(q ∈ [1,N]), it will check whether the received packet index (**r**q{2}) has a common element as the stored packet index (**r**p{2}).

$$\mathbf{r}_p\{2\} \cap \mathbf{r}_q\{2\} = \emptyset$$

node *p* will update its packet as follows

$$\mathbf{r}_p = \begin{cases} \mathbf{r}_p\{1\} = [\mathbf{r}_p\{1\}, \beta\mathbf{r}_q\{1\}] \\ \mathbf{r}_p\{2\} = [\mathbf{r}_p\{2\}, \mathbf{r}_q\{2\}] \\ \mathbf{r}_p\{3\} = \mathbf{r}_p\{3\} + \beta\mathbf{r}_q\{3\}. \end{cases}$$

Introduce a weighted factor β to help provide additional degree of freedom aiming for further performance optimization, whose value in detail will be determined in Section the third component in **r**p is updated according to a network coding based manner.[9]

(S-III) Intermediate Nodes Forwarding.

In the following, only the reception nodes in Stage II which have implemented the update operation in continue to broadcast their updated packets to neighboring nodes with a probability Pf. Again, if a node receives a packet from its neighboring node and at the same time the condition is satisfied, this node will update its packet according to the broadcasting operation will continue until there are no updated reception nodes in the last iteration.

B. RECOVERY ACCURACY IMPROVEMENT

The proposed DDS scheme increases the amount of data under the Kronecker product framework and could result in degraded recovery performance; therefore, we further improve this framework to provide a good recovery performance. A sensing matrix is crucial to CS, because it determines the efficiency in recovering the original compressible signal. In many CS applications, a random measurement matrix such as a Gaussian matrix is used. However it is shown in that a well-designed measurement matrix helps further improve the recovery accuracy compared with a random matrix. According to a smaller value of μ(A) will lead to a more accurate recovery of θ and x. Thanks to the Kronecker structure in as and At can be optimized separately according to the following theorem.

Theorem 1: Consider two matrices As and At , one has

$$\{\mu(As \otimes At) = \max\{\mu(As), \mu(At)\}.$$

Theorem 1 indicates that minimizing μ(As ⊗At) is equivalent to minimize μ(As) and μ(At) separately. To minimize μ(As), we know that Φs is formulated by the transmission protocol, and the weighted factor β in Φs can be adjusted aiming for reducing μ(As). We adopt the method of numerical search of β in the region [0,1]. Meanwhile, this method also considers the numerical stability for signal recovery. We adopt the algorithm in to minimize μ(At), where the temporal sensing matrix which commonly uses a Gaussian matrix is redesigned.

C. COMPOSITE BIT REPRESENTATION

An important backhaul rate reduction results if a suitable composite representation of the quantized signals is used. In fact, BS *j* may quantize the signal on subcarrier *n* for two BSs *i*1 and *i*2 using *b*(*i*1,*j*) *n* and *b*(*i*2,*j*) *n* bits. However, considering that both quantized signals are sent to the RNC and are obtained from the same signal *Y* (*j*) *n* , we can provide both quantization representations with fewer bits than *b*(*i*1,*j*) *n* + *b*(*i*2,*j*) *n* .

For instance, if *b*(*i*1,*j*) *n* = *b*(*i*2,*j*) *n* , only one of the two signals is sufficient to reconstruct both and thus only one can be sent to the RNC. Note that if we were to represent a given signal with the maximum number of bits required by the other BSs, I would increase the backhaul rate in some links during phase two. In this section we illustrate this composite representation. With reference to subcarrier *n*, instead of simply sending *i*∈*J*\{*j*} *b*(*i*,*j*) *n* bits to the RNC, BS *j* defines a new composite quantizer that still allows the RNC to a) reconstruct each quantized signal *Y* (*i*,*j*) *n* , *i* ∈ *J* \ {*j*}, and b) send the corresponding bits to each cooperative BS in the second phase.[10]

D. NETWORK THROUGHPUT MAXIMIZATION

First formulate the network throughput maximization problem that is solved at the RNC to schedule the backhaul transmissions. Then, we propose an iterative greedy solution to the resulting optimization problem.

E. PROBLEM FORMULATION

The RNC has got CSI from the BSs. However, to reduce the scheduler complexity and by assuming that the coherence bandwidth of the channel is larger than the FSB bandwidth, the RNC makes decisions by assuming

$$G(j,k)\, s = H(j,k)\, n\ ,\ n \in Ns$$

$$\mathrm{SINR}^{(k)}(b)=\frac{\frac{1}{\left|\mathcal{N}^{(k)}\right|}\sum_{n\in\mathcal{N}^{(k)}}H_n^{(j,k)H}\left(H_n^{(j,k)}H_n^{(j,k)H}+\Psi_n^{(j)}\right)^{-1}H_n^{(j,k)}}{1-\frac{1}{\left|\mathcal{N}^{(k)}\right|}\sum_{n\in\mathcal{N}^{(k)}}H_n^{(j,k)H}\left(H_n^{(j,k)}H_n^{(j,k)H}+\Psi_n^{(j)}\right)^{-1}H_n^{(j,k)}}.$$

Scheduler allocates the same number of quantization bits to all the subcarriers $n \in Ns$ of the same FSB, i.e., $b(i,j)\ n$ constant for $n \in Ns$.

III. PROPOSED SYSTEM

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. A WSN typically has little or no infrastructure. It consists of a number of sensor nodes working together to monitor a region to obtain data about the environment. Today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring and soon. In such applications one of the major challenge is where and how to store the sensed data. Data storage in WSNs mainly falls into two categories, namely centralized data storage and distributed data storage Data availability, security, query processing and data retrieval, network lifetime, energy efficiency are the major challenges faced by data storage in wireless sensor networks. Then about various distributed data storage where information is stored on more than one node, often in a replicated fashion in wireless sensor network. There are two main approaches: data-centric storage and fully distributed data storage. In a fully distributed data storage approach, all nodes contribute equally to sensing and storing. In data- centric storage approach some distinguished storage nodes are responsible for collecting data.[1][2]

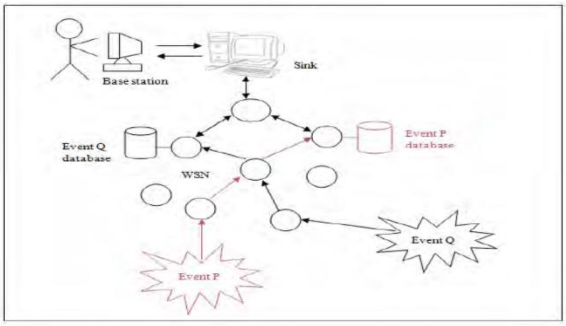


Figure 1. Distributed Storage

Both the schemes make use of various techniques for distributed data storage and each technique is characterized by different properties like topology, security, load-balancing and reliability. Data availability, security, query processing and data retrieval, network lifetime, energy efficiency are the major challenges faced by data storage in wireless sensor networks.

A. FULLY DISTRIBUTED DATA STORAGE [FDDS]

In this approach, all nodes contribute equally to sensing and storing. All nodes try to store the sensor readings locally and, then, delegate other nodes in the WSN to store newly collected data as soon as their local memories are full. Fully distributed data storage can be categorized into mainly four classes as such as 1) Topology based FDDS, 2) Security based FDDS 3) Load- balancing based FDDS, and 4) Reliability based FDDS.

B. TOPOLOGY BASED FDDS

In this approach data storage in wireless sensor networks are based on the topology of the network. Most commonly tree topologies are adopted. Mesh topology are also introduced in some special cases. Some examples are given as follows.

C. Pro Flex

The main objective of ProFlex is to introduce distributed data storage for heterogeneous wireless sensor networks with mobile sink. It is a probabilistic and flexible data storage schemes. ProFlex constructs multiple data replication structures. When compare with related protocols, ProFlex has an acceptable performance under message loss scenarios, decreases the overhead of transmitted messages, and decreases the occurrence of the energy hole problem. The protocol is composed of three phases: tree construction, importance factor distribution and data distribution. Tree topology is responsible for making multiple replication structures. Advantages of ProFlex include 1.Reduced message loss, 2.Decrease the overhead of transmitted messages, 3.Decrease occurrence of energy whole problem and 4.Applicable to large scale WSN. Guarantee to security of data is the main disadvantage of ProFLex.

D. SECURITY BASED FDDS

Security based fully distributed data storage perform distributed data storage by considering security and privacy of data as the main constraint. Several research papers are there which focus on security while data storing. Some examples are given as follows.

E. C&R-DS (CONFIDENTIAL AND RELIABLE DATA STORAGE)

The objective of C&R-DS is to introduce a technique that prevents attackers from gain information from sensor collected data. To preserve confidentiality introduce some encryption mechanism, so that data at the storage node is not available to attacker. A Two-tiered sensor network consists of three types of nodes: sensors, storage nodes, and a sink. Sensors are inexpensive sensing devices with limited storage and computing power. They are often massively distributed in a field for collecting data.

Storage nodes are powerful wireless devices that are equipped with much more storage capacity and computing power than sensors. Each sensor periodically sends collected data to its nearby storage node. Attackers are more motivated to compromise storage nodes thus algorithm to provide confidentiality and Algorithm to provide reliability are introduced for secure data storage. The advantages include 1.Confidentiality, 2.Reliability, 3.Splitting of data gives a) network band width b) network overhead c) increase efficiency and 4.Autherization[4][5]

F. LOAD-BALANCING BASED FDDS

These schemes perform fully distributed data storage based on load-balancing using different approaches. Such schemes address the problem of low-memory capacity of sensor nodes in WSN. Some examples which provide load-balancing is given as follows.

(i) Steady State Phase

After the set up phase, the compressed values of the data packets are sent by all the C-Ms. Here, they do not send the sensed value CMi they rather send the difference between the sensed data value and the data value of the corresponding C-H. Let the compressed value denoted as Δi. The i-th C-Ms data value denoted as value CMi and the corresponding C-H data value denoted as value CH. Therefore, $\Delta I = |value\_CMi - value\_CH|$.

Note that, at the binging of each occurrence, the C-H sends the complete set of the sample data values of all the C-Ms and based on the information the compression becomes achievable.

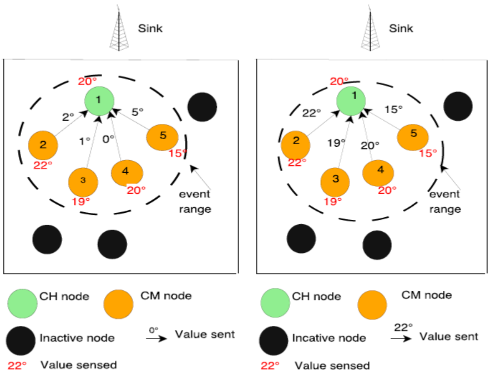


Figure 2. (a) The proposed protocol (b) The classical protocol

Region range is Revent. In the case of the proposed protocol, nodes transmit their values to the BS. Upon receiving the values from the nodes the BS then calculates Si for each ni. Therefore BS may select either node n1 or node n4 as a C-H as both nodes minimize the total difference value measured. Other nodes in turn become C-Ms. And during the steady state phase the all the C-Ms transit only the Δi value to the C-H rather than the complete temperature values. As a result a compressed value with less coded bits compared to the complete data in the classical clustering scheme.

With is reduction on transmitting bits energy saving is achieved in a great extent demonstrated in simulation results.[6][8] The LQE algorithm is shown in Algorithm 1

(ii)_Algorithm 1. Link Quality Estimation Algorithm

```
if receiving an ACK then
    update ETl1] at the rates used for to last packet;
    update average RSSls of recent ACKs measured on RX antennas RSSI;
    timer + +; error = 0; succ[ratecur] + +;
    if receiving a "complete ACK" then
        success + +;failure = 0;
        failure V = O;failureH = 0;
        successV + =Fsiperfect tx(rate), last tx(rate));
        successH+ = Fsh(minRSSI, maxRSSI);
    else if receiving a "partial ACK" then
        success = O;failure + +;
        successV = 0; successH = 0;
        failure V + = Fjv(perfect tx(rate), last tx(rate));
        failureH+ = FjJlminRSSI, maxRSSI);
    end if
else ifmissing an ACK but tries < retry limit then
    fail[ratecur] + +;
    ratecur = Lookup (multirate retry);
else if missing an ACK and tries >= retry limit then
    timer + +; failure = 0; success = 0;
    error + +; err [rate] + +;
end if
```

G. PROTOTYPES AND TEST BEDS

The feasibility of cooperative techniques has been demonstrated in “over-the-air" networks of limited size. The proposed system showed significant gains in mean sum-rate capacity (as a function of measured SINR) compared to a conventional time-multiplexed baseline. Two outdoor testbeds for implementing network coordination have been developed under the EASY-C project (Enablers for Ambient Services and Systems Part C- Cellular networks), a collaboration between academia and industry for the research and development of LTE-Advanced technologies. One testbed in Berlin, Germany, consists of four base station sites (seven sectors) connected through a high-speed optical fiber network. An even larger testbed consists of ten base station sites (28 sectors) distributed in downtown. Network coordination has been recently demonstrated over limited portions of each testbed. Using two distributed base antennas and two users, the Berlin testbed demonstrated downlink network coordination for an FDD LTE trial system. It accounted for many practical implementation aspects including synchronization, CSI uplink feedback, limited modulation and coding schemes, and a finite-bandwidth backhaul connection between the bases. Zero-forcing precoding based on limited CSI feedback was implemented jointly across the two bases. The Dresden testbed demonstrated a similarly detailed field trial for an LTE uplink system, also consisting of two bases and two users.[5][8][9]

IV. NETWORK SIMULATOR (NS2)

- NS is an object oriented discrete event simulator
  - Simulator maintains list of events and executes one event after another
  - Single thread of control: no locking or race conditions
- Back end is C++ event scheduler
  - Protocols mostly
  - Fast to run, more control
- Front end is OTCL
  - Creating scenarios, extensions to C++ protocols
  - fast to write and change

A. EVENT SCHEDULAR

In this Event scheduler while we processing many data’s at a time it will process one by one (i.e) FIFO concept , so there is no congestion while transferring the packets.

B. PACKETS

It is the collection of data, whether header is called or not all header files where present in the stack registers.

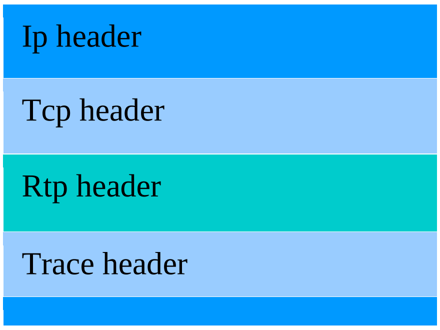
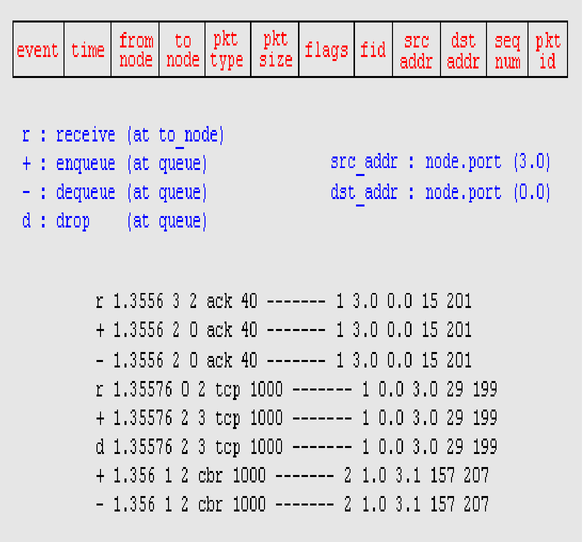


Figure 3. Packets Size

(i) *Turn on Tracing*

Trace packets on individual link Trace file format



event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id

```
r : receive (at to_node)
+ : enqueue (at queue)              src_addr : node.port (3.0)
- : dequeue (at queue)              dst_addr : node.port (0.0)
d : drop    (at queue)

          r 1.3556 3 2 ack 40 ------- 1 3.0 0.0 15 201
          + 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
          - 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
          r 1.35576 0 2 tcp 1000 ------- 1 0.0 3.0 29 199
          + 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
          d 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
          + 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
          - 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
```

Figure 4. Turn on Tracing

(ii) *Create Network Topology (Physical Layer)*

The Physical Layer is the first and lowest layer in the seven-layer OSI model of computer networking. The implementation of this layer is often termed PHY. The Physical Layer consists of the basic hardware transmission technologies of a network. It is a fundamental layer underlying the logical data structures of the higher level functions in a network. Due to the plethora of available hardware technologies with widely varying characteristics, this is perhaps the most complex layer in the OSI architecture

(iii) *Transport Connection (Transport Layer)*

Transport layers are contained in both the TCP/IP. This is the foundation of the INTERNET. The OSI model of general networking. The definitions of the Transport Layer are slightly different in these two models. This article primarily refers to the TCP/IP model, in which TCP is largely for a convenient application programming interface to internet hosts, as opposed to the OSI model of definition interface. The most well-known transport protocol is the (TCP). It lent its name to the title of the entire internet protocol suite TCP/IP. It is used qwwwwwwwwwwwwwwwwwwwwwwwwwwa4344.for connection-oriented transmissions, whereas the connectionless user datagram suite (UDP) is used for simpler messaging transmissions. TCP is the more complex protocol, due to its stateful design incorporating reliable transmission and data stream services.

C. GENERATE TRAFFIC (APPLICATION LAYER)

In TCP/IP, the Application Layer contains all protocols and methods that fall into the realm of process-to-process communications via an Internet Protocol (IP) network using the Transport layer protocols to establish underlying host-to-host connections.

In the OSI model, the definition of its Application Layer is narrower in scope, explicitly distinguishing additional functionality above the Transport Layer at two additional levels: session layer and presentation layer OSI specifies strict modular separation of functionality at these layers and provides protocol for each layer.

D. CODE OVERVIEW

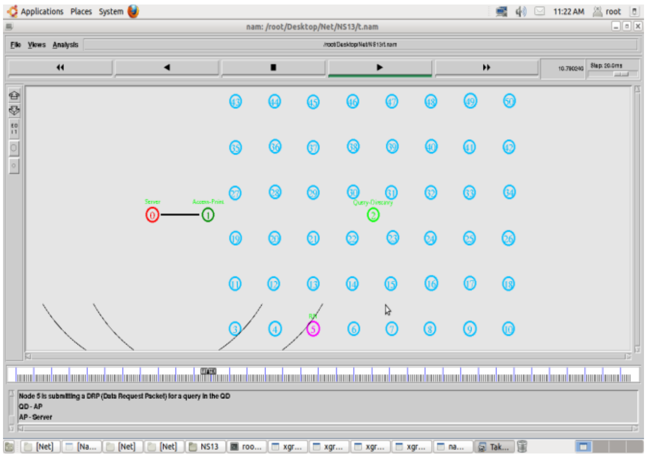
In this document, we use the term “interpreter” to be synonymous with the OTcl interpreter. The code to interface with the interpreter resides in a separate directory, tclcl. The rest of the simulator code resides in the directory, ns-2. We will use the notation ~tclcl/hfilei to refer to a particular hfilei in the Tcl directory. Similarly, we will use the notation, ~ns/hfilei to refer to a particular hfilei in the ns-2 directory. There are a number of classes defined in ~tclcl/. We only focus on the six that are used in ns: The Class Tcl contains the methods that C++ code will use to access the interpreter. The class Tcl Object is the base class for all simulator objects that are also mirrored in the compiled hierarchy. The class TclClass defines the interpreted class hierarchy, and the methods to permit the user to instantiate Tcl Objects. The class Tcl Command is used to define simple global interpreter commands. The class Embedded Tcl contains the methods to load higher level built in commands that make configuring simulations easier. Finally, the class Instance contains methods to access C++ member variables as OTcl instance variables.

E. Class Tcl

The class Tcl encapsulates the actual instance of the OTcl interpreter, and provides the methods to access and communicate with that interpreter. The methods described in this section are relevant to the ns programmer who is writing C++ code. The class provides obtain a reference to the Tcl instance;

- invoke OTcl procedures through the interpreter;
- retrieve, or pass back results to the interpreter;
- report error situations and exit in an uniform manner,
- Store and lookup “TclObjects”.
- Acquire direct access to the interpreter. We describe each of the methods in the following subsections.

V. SIMULATION AND RESULTS



Node 5 is submitting a DRP (Data Request Packet) for a query in the QD
QD - AP
AP - Server

Figure 5. Source to destination node communication

Essentially, we are trying to make this random bipartite graph as sparse as possible, while keeping the flow high enough and also allowing each data node to act independently. All the good codes described in previous sections have the property that they have very few edges ((ns2)) connecting the data nodes and the storage nodes but can still guarantee very good connectivity between the any two subsets. Such bipartite graphs are called expanders and are fundamental combinatorial objects for coding theory.

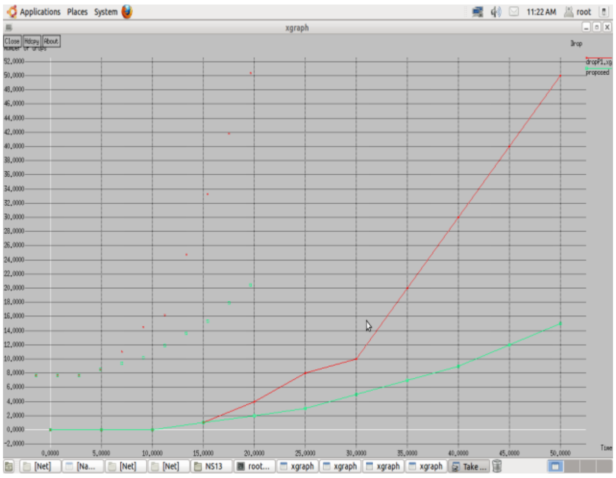


Figure 6. Comparison for Exist and proposed method packet drop

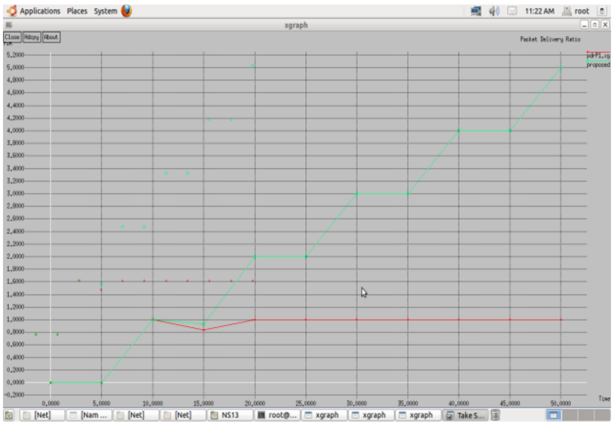


Figure 7. Comparison for Exist and proposed method packet delivery ratio

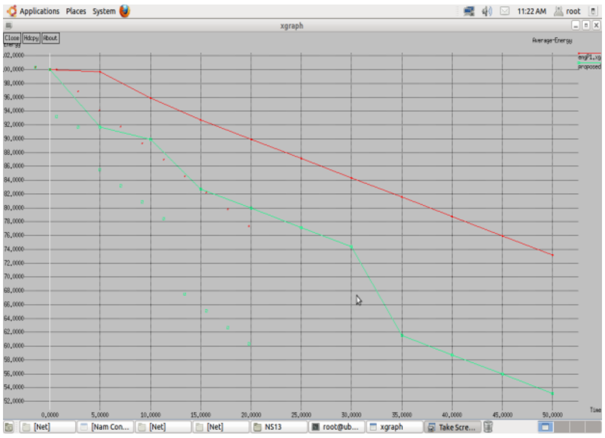


Figure 8. Comparison for Exist and proposed method Energy consumptions

VI. CONCLUSION AND FUTURE WORK

In different distributed data storage schemes in wireless sensor networks and classified these techniques into mainly two types namely fully distributed data storage (FDDS) and data centric storage (DCS). In FDDS all nodes contribute equally to sensing and storing while in DCS some distinguished storage nodes are responsible for collecting a certain data. In every of these classifications the technique can be all over again confidential into topology base, safety based, load-balancing based and dependability based distributed data storage space schemes.

The topology based data storage performs distributed data storage based on the topology of the network and the security based data storage adopts some data storage schemes that support security features. The load-balanced based distributed data storage uses grid like architecture to achieve load balancing and can achieve robustness in distributed storage through reliability based distributed data storage. At last we complete a comparison among different distributed data storage scheme under a variety of constraints like data accessibility, protection, energy efficiency and network lifetime.

REFERENCES

[1] J. Yick, B. Mukherjee, and D. Ghosal, "*Wireless sensor network survey*," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[2] R. Zeng, Y. Jiang, C. Lin, Y. Fan, and X. Shen, "*A distributed fault/ intrusion-tolerant sensor data storage scheme based on network coding and homomorphic fingerprinting*," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1819–1830, Oct. 2012.

[3] Z. Kong, S. Aly, and E. Soljanin, "*Decentralized coding algorithms for distributed storage in wireless sensor networks*," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 261–267, Feb. 2010.

[4] M. Duarte and Y. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4053–4085, Sep. 2011.

[5] A. Talari and N. Rahnavard, "CStorage: Distributed data storage in wireless sensor networks employing compressive sensing," in *Proc. IEEE GLOBECOM*, Houston, TX, USA, Dec. 2011, pp. 1–5.

[6] X. Yang *et al.*, "Energy-efficient distributed data storage for wireless sensor networks based on compressed sensing and network coding," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 5087–5099, Oct. 2013.

[7] B. Ali, N. Pissinou, and K.Makki, "Identification and validation of spatiotemporal associations in wireless sensor networks," in *Proc. SENSORCOMM*, Athens, Greece, Jun. 2009, pp. 496–501.

[8] M. Leinonen, M. Codreanu, and M. Juntti, "Distributed correlated data gathering in wireless sensor networks via compressed sensing," in *Proc. ACSSC*, Pacific Grove, CA, USA, Nov. 2013, pp. 418–422.

[9] Y. Rivenson and A. Stern, "Compressed imaging with a separable sensing operator," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 449–452, Jun. 2009.

[10] V. Abolghasemi, S. Ferdowsi, and S. Sanei, "A gradient-based alternating minimization approach for optimization of the measurement matrix in compressive sensing," *Signal Process.*, vol. 92, no. 4, pp. 999–1009, Apr. 2012.

[11] S. Jokar and V. Mehrmann, "Sparse solutions to underdetermined Kronecker product systems," *Linear Algebra Appl.*, vol. 431, no. 12, pp. 2437–2447, Dec. 2009.

[12] J. Duarte-Carvajalino and G. Sapiro, "*Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization*," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1395–1408, Jul. 2009.