A Novel, Unified Dynamic Data Dissemination [D-Cube] and Traffic Classification Technique for Congestion Avoidance in High Speed Wireless Multimedia Sensor Networks

Uma S Visweswaraiya, Kargar S Gurumurthy
Department of ECE, UVCE, Bangalore University, Bangalore, India
umakeshav2000@gmail.com, shkgurunurthy@gmail.com

Uma S Visweswaraiya¹, Kargar S Gurumurthy²
Department of ECE, RNSIT, RITM, VTU
Bangalore, India
Uma.S.V@rnsit.ac.in

Abstract — Rapid technological advances and innovations in the area of autonomous systems push the researchers towards autonomous networked systems with emphasis on Wireless Sensor Networks (WSNs). In WSN event-driven applications, reporting the detected events in the area is critical, resulting in sudden bursts of traffic due to occurrence of spatially-correlated or multiple events, causing loss of data. Also, nodes have very limited power due to hardware constraints. Packet losses and retransmissions resulting from congestion, cost precious energy and shorten the lifetime of sensor nodes. Till now, in WSNs, Congestion control techniques are based on detection of congestion and recovery, but they cannot eliminate or prevent the occurrence of congestion. Collision and hence congestion also results in a time-variant channel capacity. Therefore, this research uses a unified approach to address this issue dynamically. In the first phase an efficient medium access control (MAC) technique [Dynamic Data Dissemination] is used to coordinate the access of nodes to the shared medium without interference. It uses the queue buffer length of the sensor nodes to estimate the congestion. Later, it dynamically disseminates the traffic, passing it through a DiffServ Server in the second phase, where the packets from Phase 1 are further classified into different priority classes to provide a congestion-free routing path to the destination with improved Quality of Service (QoS).

Keywords—congestion; routing; protocol; Wireless Sensor Networks (WSNs); throughput; delay; data-rate; Quality of Service (QoS); Differentiated Services (DiffServ)

I. INTRODUCTION

The needs of man today have pushed the vision of ambient intelligence from concept to reality. Towards this direction, there has been an unprecedented research interest for autonomous networked systems with emphasis on WSNs till this day [1]. WSNs are deployed for several mission-critical tasks (e.g. as platforms for health monitoring, process control, environmental observation, battlefield surveillance) and are expected to operate unattended (without human intervention) for extended periods of time.

Typically, WSNs comprise of small (and often cheap), cooperative devices (nodes), which may be (severely) constrained in terms of computation capability, memory space, communication bandwidth and energy supply. Increasingly, with the rapid development of low-cost hardware CMOS cameras and microphones, autonomous sensor devices are becoming capable of ubiquitously retrieving multimedia content such as audio and low-rate video streams from the environment [2]. In the context of WSN, autonomous nodes may interact (a) with the environment so as to sense or control physical parameters, and (b) with each other in order to exchange information or forward data towards one or more dedicated sink nodes. Typically, WSNs operate under light load, but large, sudden, and correlated synchronized impulses of data may suddenly arise in response to a detected or constantly monitored event, causing congestion. Alternatively, some nodes may be constantly generating streaming data. All the data must be directed in a multi-hop manner to the sink node(s). A large number of generated packets in conjunction with variable wireless network conditions may result in unpredictable behavior in terms of traffic load variations and link capacity fluctuations. The problem is worsened due to topology changes driven by node failures, mobility, or intentional misbehavior. These stressful situations are likely to provoke congestion in WSN environments. The problem of congestion in wireless sensor networks is unveiled, thoroughly investigated and effectively tackled in the remainder of this study.

Congestion control involves measures taken for controlling the traffic injected into the network in order to avoid or mitigate network collapse "in press" [3]. Robust, scalable and self-adaptable congestion control approaches aim to keep the network operational under congestion conditions, whilst keeping packet loss and end-to-end delay within tolerable levels as well as maximizing network lifetime. Lately, with the emergence of mission-critical applications (e.g. health monitoring, plant automation), there has been increased interest [4] in providing performance assurances, especially for metrics such as packet delivery ratio, delay and energy consumption. Interestingly, in the recent times, various applications based on multimedia wireless networks are...
beginning to arise, where many types of sensors such as cameras, audio sensors, vibration sensors, and light sensors have to be integrated in the same sensor node. In addition, it is expected that the number of such highly capable sensor nodes in multimedia applications will scale to tens, hundreds or even thousands. Due to the constrained nature of WSNs, the new approaches should be simple to implement at individual node level with minimal exchange of information. The focal point of this study is to design a robust, scalable, self-adaptive and energy-efficient congestion control (CC) mechanism for delivering enhanced application fidelity at the sink in terms of packet delivery ratio and delay, under varying network conditions and simultaneously giving differential treatment to different types of traffic so as to improve their quality. The organization of the rest of the paper is as follows: Section II describes the related work, Section III gives insight into the proposed Unified technique, Section IV describes the Implementation methodology, and finally Section V concludes the paper.

II. RELATED WORK

Congestion in a wireless sensor network is either controlled or avoided for improving the data transmission in both continuous monitoring and event-reporting applications. Some of the protocols that have been proposed to reduce congestion in WSNs can be broadly classified as i) Congestion Detection and Avoidance protocols (CODA and ESRT) [5][6], ii) Rate Control protocols (WRCP, LACAS, WCP and CONSEQ) [7][8][9][10], iii) Routing protocols (DFS, HMAC, PFSQ RMST) [11][12][13][14] and ADHOC routing protocols like DSDV, AODV) [15][16] iv) Medium Access Control protocols [TDMA, CSMA/CA, DCF etc.][17][18].

Among MAC protocols, in the recent years, various MAC protocols with different objectives were proposed for wireless sensor networks. Two major types of media access control protocols are prevalent in WSNs: Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

TDMA is a schedule-based, contention-free and energy efficient protocol. TDMA divides the channel into time slots and only one node is allowed to transmit or receive data within each time slot, hence avoiding collisions and having bounds on the delay. But, TDMA needs the sensor nodes to be time-synchronized, which is not possible for large scale sensor networks due to the significant amount of signaling traffic needed. Also, TDMA protocols have limited scalability and adaptivity to the changes in number of transmitting nodes. TDMA can be used in scenarios where a small set of nodes is continuously transmitting packets with infrequent activation/deactivation of transmitting nodes.

CSMA/CA is a contention-based, random access MAC protocol that requires no coordination among the nodes accessing the channel. In CSMA/CA, when a sensor node has a packet to transmit, it senses the channel before transmitting and is allowed to transmit after the channel is sensed idle. On successful reception, the destination node issues an acknowledgement packet. The asynchronous mechanism of accessing the shared medium is not able to prevent collisions. When a collision is detected, the lost packet is retransmitted by the MAC layer after a random amount of time (back-off procedure) and is discarded after a finite number of unsuccessful retransmission attempts. CSMA/CA protocol employs a retransmission scheme in order to increase the reliability of the lossy wireless channel. The main advantage of CSMA/CA is that it adapts quite well with the variable condition of traffic and is quite robust against interferences. The main drawback of CSMA/CA is that it consumes more energy than contention-free protocols because of energy wastage in collisions and idle listening and does not provide bandwidth and delay guarantees.

The majority of MAC protocols designed for WSNs rely on the CSMA/CA protocol [12][24]. CSMA/CA has been preferred to TDMA in the scenarios where the traffic is bursty and multiple consecutive and contiguous packets generated from the same collision neighborhood need to be sent. Simulation studies conducted by [24] using NS2 [26] revealed that, in WSNs where the wireless medium is shared using CSMA/CA-like protocols, wireless channel contention losses can dominate buffer drops and increase quickly with offered load. The problem of channel losses is worsened around hot spot areas, as for example, in the proximity of an event, or around the sink. These phenomena result in the starvation of channel capacity in the vicinity of senders, while the wireless medium capacity can reach its upper limit faster than queue occupancy [10].

Some congestion control approaches for WSNs are solely based on buffer occupancy to detect congestion. In these approaches, congestion is detected when the instantaneous queue length of a node exceeds its limited buffer capacity, leading to packet drops, or when the queue length exceeds a certain percentage of the buffer capacity, leading to long delays. These approaches assume that the underlying MAC protocol can efficiently provide a stable radio link capable of resolving collisions. A large number of congestion control approaches in WSNs, as for example CODA [5], Fusion [19], BGR [20] and Siphon [21] use both buffer occupancy and local wireless channel load to infer congestion. In these approaches, while a packet waits to be sent, the sensor node samples the state of the channel at a fixed interval. Based on the number of times the channel is busy, it calculates a utilization factor. If utilization rises above a certain level (e.g. the theoretical upper bound of the
channel throughput), the congestion bit is set. Otherwise, the congestion bit is cleared.

Recent studies by [22] and [23][24], argue that congestion can be also inferred by inspecting packet inter-arrival time and packet service time (or alternatively incoming and outgoing traffic rates). In these approaches, congestion is inferred when the inter-arrival time is smaller than the service time that is the incoming packet rate is higher than the outgoing traffic rate leading to accumulation of packet in queues. Congestion avoidance is the core concept for this research model to proactively identify and alleviate congestion in the network and adjust the network to handle the future traffic. In this study, buffer occupancy and wireless channel load are cooperatively used to infer congestion.

III. PROPOSED TECHNIQUE

Congestion control (CC) is concerned with measures taken in order for a network (a) to avoid congestion (congestion avoidance approaches) and (b) to mitigate congestion (congestion mitigation approaches) and operate within an acceptable performance level, even when the demand is near or it exceeds the capacity of the network resources.

In Wireless sensor Networks (WSN), Congestion control techniques detect congestion and attempt to recover from packet losses due to congestion, but they cannot eliminate or prevent the occurrence of congestion. Congestion avoidance techniques, instead of reducing the traffic rate to control the congestion, distribute the data traffic through the other nodes to achieve good throughput. Hence Congestion avoidance technique that uses the queue buffer length of the sensor nodes to estimate the congestion and disseminate traffic to provide a congestion-free routing path is preferred in WSN.

Congestion avoidance is the core concept of our unified technique. This work brings out a novel technique to avoid Congestion in High speed multimedia WSNs. It comprises of three phases:

1. Congestion estimation using queue buffer length at the sensor nodes
2. Congestion avoidance by dynamic data dissemination (D^3)
3. Added Congestion Control by traffic classification for better QoS

A. Phase I: Congestion estimation

The queue buffer length is used as an important parameter to identify the congested nodes in the network. This is a MAC based approach where Congestion is detected in the network when the sensor nodes' queue overflows and packets start to drop. The node is said to be congested and cannot handle any further packets until its buffer starts to clear. Every time there is a new packet to transmit, the buffer length values are checked and that node with the smallest filled queue buffer value gets the chance to forward the packets. This can help to dynamically choose an alternate, better path for data transmission, and also by using only the queue buffer length as a measure, the overhead of computations at a node can be eliminated.

B. Phase 2: Dynamic Data Dissemination

This protocol provides event reporting and packet delivery ratio, by dynamically diffusing the traffic in the network using multiple forwarders in addition to backup forwarding. Unlike other protocols where the sender node chooses its receiver node based on the current congestion level, the D^3 technique provides the option for each node to decide if it can participate in the data transmission. Using anycasting, the response time for a node to reply could be completely avoided and the data transmission decisions could be handled at the MAC layer. This will make the D^3 protocol very light weight and more efficient. Further, instead of having one forwarder for data transmission, there are multiple probable forwarders for each node. This could provide the opportunity for the data packets to get transmitted in a congestion-free path much faster. Since the forwarders get the opportunity to make the participation decision, the proposed technique can help to alleviate congestion in the network efficiently by managing the data transmission at each hop level dynamically. In order to improve event reporting, a backup mechanism could support data transmission when all the probable forwarders are not available.

When a node 'i' has a data packet to transmit, it computes the virtual queue length for all its forwarder set nodes, based on the number of packets in node i's queue, the number of packets in the neighbor's queue, and the number of packets dropped by node i due to an excessive number of retransmissions after the most successful transmission. Though this protocol does not use any additional control messages, it has to maintain multiple queues and compute the virtual queue length for all forwarders for each data packet [25]. The main objective of this research is to avoid control messages and unwanted computation that will degrade the network performance. The key aspects of the technique are as shown below:

- This protocol provides event reporting, packet delivery ratio,..., by dynamically diffusing the traffic in the network using multiple forwarders. This could provide the opportunity for the data packets to get transmitted in a congestion-free path much faster.
- The queue buffer length is used as an important parameter to identify congested nodes in the network.
- By using anycasting approach, where forwarders get the opportunity to make the participation decision,
the response time for a node to reply could be completely avoided.

In order to improve event reporting, a backup mechanism could support data transmission when all the Most Probable forwarders (MPFs) are not available.

This will make the proposed D\(^3\) technique light weight and more efficient. The forwarders are set up as shown below:

**Most Probable forwarders [MPFs]** - The D\(^3\) technique addresses the difficulty faced with a single forwarder, by adopting multiple probable forwarders for each node.

**Backup forwarder [BF]** - A backup node is used only when none of the probable forwarders are available for data transmission.

C. Forwarder Configuration for Traffic Diffusion

1) Potential Forwarders Setup: The multiple forwarder setup has many advantages compared to the single forwarder setup, like increased network reliability, reduced congestion, and increased Quality of Service (QoS). The most probable forwarders configuration procedure involves the following steps:

1. Identifying one-hop neighbors.
2. Choosing the forwarder set from the neighbors.

The base station is located at the top-left corner of the deployment area and all the sensor nodes know their exact location in the area. The sensor nodes also have the knowledge of their hop distance from base station. With this information, the one-hop neighbors are identified.

**Algorithm for Choosing MPFs**

1. Each sensor node initially identifies all its one-hop neighbors based on transmission radius.
2. The neighbors can be active forwarders, if they are closer to the base station by one hop level than the source node.
3. Now, from the list of one-hop neighbors, the neighbors that are closer to the base station are identified using the hop distance to base station and are chosen as MPFs.
4. From the reduced list of one-hop neighbors, four neighbors are chosen based on their hop distance to the base station.

The sensor nodes closer to the base station or within the transmission radius of the base station, which are basically in one-hop distance, have only base station as its forwarder/receiver.

Some of the sensor nodes, due to their location, might not be able to find four forwarders and might even have a single forwarder. With at most four potential forwarders, the network performance is improved by providing multiple data paths for transmission. The data packets, instead of being dropped at a node due to the unavailability of a forwarder, can be transmitted to the base station through any potential forwarder.

An example forwarder setup is as shown in Figure 1. The hop levels are identified for the sensor nodes with level 'i'-2 being the closest to base station and level 'i' being the farthest from base station. There are nodes labeled from A to O that can sense the events and forward the data packets. For the sensor node H, there are three MPFs: A, B, and C. For node I, there is only a single MPF. And the node O contains the full set, K, L, M, and N, as MPFs. Based on the location of the sensor nodes and its transmission radius, the number of MPFs will vary. Since the area for event monitoring can be anywhere in the network, the sources are randomly chosen and hence a source node is also an MPF.

The D\(^3\) technique could reduce contention also by avoiding all the MPFs involved in the data transmission and using the Backup forwarder when required. With fewer nodes contending for the channel, the contention in the network could reduce as compared to all the MPFs.

![Figure 1. Potential Forwarders Setup](image1.png)
When a sensor node is able to find only four potential forwarders and no possible backup forwarder in the hop level above, the backup forwarder can also be chosen in the same hop level. This can be seen in Fig. 3, the backup forwarder J is in the same level as node O. Though this backup forwarder setup in the same hop level is very rare, the nodes still try to have a backup. This ensures data transmission without packets being held at the intermediate nodes due to the unavailability of potential forwarders and thereby relieving congestion in the network.

III) Backup Response: A backup forwarder is used to support data transmission when all the potential forwarders fail to respond. The D3 protocol will avoid packet drops during congestion, when all queue buffers of MPIs are full, by diffusing the data packets using a backup forwarder. When a node sends a RTS [Request to send] packet and does not get a CTS [Clear to send] reply after waiting for SIFS [Short Interframe Space] interval of time, it retries with another RTS packet. In the standard IEEE 802.11MAC, the maximum number of RTS retries is seven. In a situation in which all of the potential forwarders have their queue buffers full, it might take more RTS attempts to receive a CTS reply, or no CTS reply is received. The packets will get held at the sender’s queue and start to create congestion. In order to ease such a situation and also reduce the delay incurred in waiting for a CTS reply from the potential forwarder, the backup forwarder is used in the D3 protocol.

The condition for selecting the backup forwarder to be far from the other potential forwarders helps to avoid the congested zone and try to find a congestion-free forwarder to send the packets. The backup technique provides more robustness to avoid congestion in some backlogged situations.

D. Differentiated Services based Congestion Control Algorithm

The proposed differentiated services based algorithm helps in further alleviating the congestion in WMSNs by ensuring that the data leaving each and every node in the network gets prioritized differential treatment. The following are the components of the proposed differentiated services algorithm: 1) Traffic classes 2) Congestion detection 3) Procedure for calculating the modified bandwidth share for upstream nodes. Each node estimates the available bandwidth by monitoring the wireless channel or through other Traffic Classes. We assume that data originating from the WMSN can be mapped to the following traffic classes.

I) Real-time Loss Tolerant Data: Time and mission critical data is mapped to this traffic class. Such data demands low delay and high reliability. Invariably, critical scalar data is short lived, therefore the mean bandwidth consumed by such data is small.

II) Real-time Loss Tolerant Data: Critical scalar data that requires low delay and low reliability is mapped to this traffic class. Dense sensor nodes deployment can be one reason that the data mapped to this traffic class is loss tolerant. If one reading is corrupted or dropped during the transmission, it is highly likely that the reading sent by a nearby sensor node will compensate for the lost reading.

III) Real-time Loss Tolerant Multimedia Streams: Real-time multimedia streams are mapped to this traffic class. Real-time multimedia data is loss tolerant but requires bounded delay and jitter.

IV) Delay Tolerant and Loss Tolerant Multimedia Streams: There are multimedia streams that can be processed off-line, such streams arise from environmental monitoring systems. These multimedia streams can tolerate delay as well as packet loss. Data belonging to this traffic class can be stored at a sensor node and it is transmitted when the network is experiencing low traffic load. There is a limit on the data storage capabilities of sensor nodes and with current state of the art nodes, storing a large amount of data may not be feasible.

V) Delay Tolerant and Loss In tolerant Data: Data that emerges in response to a query that does not involve mission critical aspects of the system is mapped to this traffic class.
VI) Delay Tolerant and Loss Tolerant Data: Data that is neither mission critical nor time critical e.g., non-critical regular reporting data.

Based on this allocation, Congestion is detected and new calculated BW is allocated to the different streams of data.

We made the following assumptions for designing the congestion control algorithm.

1) Any intermediate node \( n \) knows the mean data arrival rate (i.e., \( \lambda_i \)) pertaining to traffic class \( i \) from the upstream node \( j \). Applications local to node \( n \) may generate traffic pertaining to class, with mean arrival rate of \( \lambda_i^n \). Therefore, the mean arrival rate in a particular traffic class \( i \) is given by the following equation.

\[
\lambda_i = \left( \sum_{j=1}^{\omega} \lambda_i^j \right) + \lambda_i^n
\]  
(1)

In equation 1, \( \omega \) represents the one hop away upstream nodes (for which a node is acting as a relaying node).

2) Each node estimates the available bandwidth (\( \beta_{arb} \)) by monitoring the wireless channel or through other available bandwidth estimation algorithms for wireless networks.

The system now starts an algorithm to calculate the modified share of the bandwidth for each one hop away upstream node, given that the node \( n \) is acting as a relaying node for that particular one hop away upstream node. If the congested node is generating some flows locally then the algorithm also calculates the modified share of bandwidth for the congested node. The following linear program assigns the modified bandwidth to each one hop away upstream node. While optimizing the bandwidth assignment procedure, the linear program considers the priority of the node to which bandwidth will be assigned. The modified bandwidth information is sent to all concerned one hop away upstream nodes and to the node itself. Nodes can then cut down transmission rates for each traffic class depending on the available bandwidth, priority of the traffic class, and amount of data generated in each traffic class. If the receiving node also experiences congestion, then the process continues until the source of the congestion gets informed.

\[
\max \sum_{j=1}^{\omega} n_j \beta_j
\]  
(2)

\[
\varphi_i^{\min} \leq \beta_i \leq \varphi_i^{\max} \quad (\forall j \in \omega)
\]  
(3)

\[
\sum_{j=1}^{\omega} \beta_i \leq \beta_{arb} \quad (\forall j \in \omega)
\]  
(4)

In the above linear program, \( \beta_j \) is the decision variable i.e., bandwidth that will be assigned to the \( j^{th} \) node, \( n_j \) represents the priority of the \( j^{th} \) node and \( 0 \leq n_j \leq 1 \), and \( \omega \) represents the upstream nodes for which the node is acting as the relaying node. \( \varphi_i^{\min} \) and \( \varphi_i^{\max} \) give the minimum and maximum bandwidth that can be assigned to the \( j^{th} \) node, and Equation 5 ensures that the total assigned bandwidth does not exceed the available bandwidth. Please note, \( j = 0 \) represents the congested node and its rate is adjusted if the congested node is generating flows locally. The methods for calculating different parameters used in the above linear program are discussed below. The value of \( \eta_j \) for each node is derived from the priorities of the flows originating from the node \( j \). If the mean bandwidth utilized by the node \( j \) in traffic class \( i \) is \( \bar{\beta}_i^j \), then the mean total bandwidth utilized by node \( j \) is

\[
\bar{\beta}_i^{\text{tot}} = \sum_{i=1}^{\omega} \bar{\beta}_i^j
\]  
(5)

The value of \( n_j \) is calculated as follows:

\[
n_j = \frac{\sum_{i=1}^{\omega} (\alpha_i \times \varphi_i^{j})}{\beta_{arb}^{n_j}}
\]  
(6)

In Equation (6), \( \alpha_i \) represents the priority of the traffic class \( i \), \( \beta_{arb}^{n_j} \) represents the bandwidth available at the congested node.

Algorithm: Congestion Detection and Control

Data: \( \lambda_i, \alpha_i, \mu_i, \beta_i^j, \beta_i^{\text{tot}}, \beta_{arb}^{n_j}, \omega \)

Result: \( \beta_i \)

1) \( i \leftarrow \text{Congestion} \leftarrow 0 \)

2) if \( \forall j, \exists (\lambda_j > \mu_j) \) then

3) \( i \leftarrow \text{Congestion} \leftarrow \text{monitorQueue}(i) \)

4) end

5) if \( i \leftarrow \text{Congestion} \) then

6) \( \eta_i \leftarrow \text{calculateEta}(\alpha_i, \beta_i^j, \beta_i^{\text{tot}}, \beta_{arb}^{n_j}) \)

7) \( \varphi_{\min} \leftarrow \text{calculatePhiMin}(\beta_{arb}^{n_j}, \omega, \beta_i^{\text{tot}}) \)

8) \( i \leftarrow 0 \)

9) for \( i \leq 6 \) do

10) \( \varphi_{\max} \leftarrow \beta_i^j \)

11) end

12) \( \beta_i \leftarrow \text{reassignBW}(\varphi_{\min}, \varphi_{\max}, \eta_i) \)

13) \( i \leftarrow 0 \)

14) for \( i \leq \omega \) do

15) \( \text{relaycongestion}(\beta_i) \)

16) \( i \leftarrow i + 1 \)

17) end

18) end
IV. EXPERIMENTAL RESULTS

The $D^3$ technique is tested on a sample wireless Multimedia Sensor Network (WMSN) with three levels as shown in Figure 1 using Network Simulator 3 (NS-3)[26][27].

A summary of the simulation parameters is given in Table I. All the nodes in the area are distributed uniformly and randomly. For the data packets, each source generates Constant Bit Rate (CBR) traffic and the number of sources is varied to evaluate the performance at different loads. All the experiments are conducted multiple times (10) and each data value is taken as an average of 10 independent runs with randomly chosen sources. The data packet generation from each source is 256 Kbps and the packet size is varied from 64 bytes to 512 and 1000 bytes.

We use the same data packet generation, packet size, queue size (5/10), and simulation duration of 60 seconds with NS-3's default power setting to compare $D^3$ performance with CONSEQ [10] which also exploits multiple forwarders to reduce congestion in the network.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>100m x 100m</td>
</tr>
<tr>
<td>Deployment Strategy</td>
<td>Uniform Random</td>
</tr>
<tr>
<td>Transmission Radius</td>
<td>25 m</td>
</tr>
<tr>
<td>Total Number of Nodes</td>
<td>25</td>
</tr>
<tr>
<td>Number of Sources</td>
<td>1-5</td>
</tr>
<tr>
<td>Data Packet Size</td>
<td>64, 512, and 1000 bytes</td>
</tr>
<tr>
<td>Number of Packets Sent (data rate)</td>
<td>1kb/sec to 30kb/sec</td>
</tr>
<tr>
<td>Queue Size</td>
<td>5, 10, and 15</td>
</tr>
</tbody>
</table>

Table I. Test Network Setup

<table>
<thead>
<tr>
<th>Number of Sources</th>
<th>Number of Packets Sent</th>
<th>Number of Packets Received</th>
<th>Number of Packets Sent</th>
<th>Number of Packets Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6301</td>
<td>6301</td>
<td>6301</td>
<td>6301</td>
</tr>
<tr>
<td>2</td>
<td>10602</td>
<td>10602</td>
<td>10602</td>
<td>10602</td>
</tr>
<tr>
<td>3</td>
<td>11143</td>
<td>11143</td>
<td>11143</td>
<td>11143</td>
</tr>
<tr>
<td>4</td>
<td>11214</td>
<td>11214</td>
<td>11214</td>
<td>11214</td>
</tr>
<tr>
<td>5</td>
<td>13457</td>
<td>13457</td>
<td>13457</td>
<td>13457</td>
</tr>
<tr>
<td>6</td>
<td>13542</td>
<td>13542</td>
<td>13542</td>
<td>13542</td>
</tr>
<tr>
<td>7</td>
<td>14631</td>
<td>14631</td>
<td>14631</td>
<td>14631</td>
</tr>
</tbody>
</table>

Table II. Number of Packets Sent and Received

We used the standard network performance metrics such as packet delivery ratio, throughput, and average end-to-end delay for evaluation of our technique. Table II shows the number of packets sent and received by the CONSEQ and $D^3$ protocol. The approximate number of packets sent from different numbers of sources is calculated for CONSEQ from the protocol's packet delivery ratio and the number of packets delivered graphs. From the Table II, it is clear that $D^3$ delivers more packets than CONSEQ.

The traffic dissemination approach to proactively avoiding congestion at the nodes makes our technique deliver more packets even with high traffic loads. As observed from Figure 4(a), the $D^3$ outperforms CONSEQ even under high traffic load. Figure 4(b) shows the delay comparison of $D^3$ and CONSEQ. The average end-to-end delay for the first 20%, 50%, and 100% of the packets are measured for $D^3$ protocol because the number of packets delivered in both of the protocols are different. The average end-to-end delay of the first 20% of packets delivered by $D^3$ is less than CONSEQ's delay under all the tested conditions. The average delay of the first 50% of packets is comparable with CONSEQ's delay for more sources. The delay for 50% of packets in $D^3$ is less than CONSEQ's delay, even though 50% of the total number of packets delivered here is more than CONSEQ's total number of packets.

Figure 4. Comparison of $D^3$ with CONSEQ with respect to a) Number of Packets Delivered and b) Delay

In $D^3$, a node does not need the queue status of its neighbors to involve in data transmission, whereas in CONSEQ, load is balanced after computing the virtual queue length of all the forwarders. This mechanism of $D^3$ reduces the transmission delay at each hop level even at high traffic. The delay for all the packets (100%) is higher than CONSEQ delay because the total number of packets delivered by $D^3$ is significantly more than CONSEQ. On
an average, the difference between $D^3$ and CONSEQ in end-to-end delay of all the packets delivered (100%) for different number of sources is 0.18 seconds. While incurring only 0.18 seconds more, $D^3$ delivers 80% more packets than CONSEQ.

The performance of backup forwarding in traffic dissemination is then evaluated. To form a congested state in the network, the base station is located at the center of the deployment area and the traffic converges from all directions onto it. With the base station at the center, 10 sources are randomly chosen. The network is loaded such that traffic converges towards the base station from different directions and at different rates. TABLE III shows the simulation results with backup forwarder and without backup forwarder. Here, more packets are received in a shorter duration than without using backup. Though the packets are sent through backup forwarders in a longer distance path, the delay is less, because the time taken for the packets to reach the base station through backup forwarder is less than the wait time for the potential forwarders when their queues are full.

**TABLE III**

<table>
<thead>
<tr>
<th></th>
<th>Without Backup</th>
<th>With Backup</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of Packets Sent</td>
<td>Number of Packets Received</td>
</tr>
<tr>
<td>1800</td>
<td>1236</td>
<td>0.17</td>
</tr>
<tr>
<td>2750</td>
<td>1995</td>
<td>0.18</td>
</tr>
<tr>
<td>5500</td>
<td>3089</td>
<td>0.20</td>
</tr>
</tbody>
</table>

**TABLE IV**

<table>
<thead>
<tr>
<th>Nodes Specific Parameters</th>
<th>Packets per Second Corresponding to Traffic Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node ID</td>
<td>TC-1</td>
</tr>
<tr>
<td>O</td>
<td>1</td>
</tr>
<tr>
<td>J</td>
<td>0</td>
</tr>
<tr>
<td>N</td>
<td>2</td>
</tr>
<tr>
<td>K</td>
<td>0</td>
</tr>
<tr>
<td>M</td>
<td>0</td>
</tr>
</tbody>
</table>

Sensor nodes labelled O, J, N, K, and M generate traffic corresponding to traffic classes 1, 3, and 5. Sensor nodes start to generate data corresponding to different traffic classes as soon as they wake up. $T^*$ denotes the wake up time of node $n$. Table IV shows the information about the traffic generation rate of each source node corresponding to different traffic classes.

From the network topology and the data generation rates of different source nodes, congestion is made to happen at node M at time 25 seconds. Therefore, the system detects congestion and activates the bandwidth distribution algorithm. Table IV shows different parameter values along with the final outcome as given by the algorithm. The priorities assigned in our simulations are $\alpha_1 = 0.5$, $\alpha_2 = 0.4$, $\alpha_3 = 0.2$.

It can be observed that the proposed congestion control algorithm favours nodes, which produce the bulk of data with higher priorities i.e nodes N, K and M. When congestion control information has reached the source nodes, they can penalize different flows depending upon their priority and data generation rate. The final rates assigned to the different sources by the control algorithm are as shown in **TABLE V**.

**TABLE V**

<table>
<thead>
<tr>
<th>Nodes Specific Parameters</th>
<th>Outcome of the Proposed Algorithm - Bandwidth Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed algorithm assigned rate</td>
<td>6.25</td>
</tr>
<tr>
<td>Node Data Generation rate</td>
<td>7.5</td>
</tr>
</tbody>
</table>
results show that $D^3$ can deliver more packets than CONSEQ in a shorter duration, thus improving the packet delivery ratio and shortening the transmission latency. Also, the algorithm uses a linear differentiating, priority based bandwidth allocation program to optimally assign service rates to congested node and its one hop away upstream nodes. Experimental results have shown that using mean bandwidth consumption in each traffic class along with the class priority achieves better performance. In the future, we want to look at other possibilities like more realistic test scenarios with traffic hotspots, evaluating energy consumption of the nodes using the new technique and develop an energy-efficient $D^3$ protocol.

ACKNOWLEDGMENT

My sincere thanks to Chairman, RNSIT for providing an amiable working environment. Special thanks to the Director and Principal, RNSIT for their encouragement throughout the work.

REFERENCES
