Scalable Computing: Practice and Experience
Volume 11, Number 4, pp. 359–371. http://www.scpe.org
ISSN 1895-1767

ERCTP: END-TO-END RELIABLE AND CONGESTION AWARE TRANSPORT LAYER PROTOCOL FOR HETEROGENEOUS WSN*

ATIF SHARIF, VIDYASAGAR M. POTDAR AND A. J. D RATHNAYAKA†

Abstract. Other than hardware optimization, the communication protocols for heterogeneous Wireless Sensor Network (WSN) are currently playing an important role for achieving the longevity of the network. Among other layers of heterogeneous WSN communication protocol stack, researchers are putting efforts in developing transport layer protocols in order to avoid congestion in WSN and provide data or application level reliability support thereby ensuring the QoS objectives of the heterogeneous WSN application. In this paper we have envisaged a light weight transport protocol design, End-to-End Reliable and Congestion Aware Transport Layer Protocol (ERCTP), which achieves high data reliability by the introduction of distributed memory concept within network and minimum packet drop due to congestion by the effective implementation of congestion detection and rate adjustment scheme that uses stochastic control framework. The proposed scheme is evaluated extensively against the TCP-Westwood+ (TCP-WW+), TCP-Westwood (TCP-WW), TCPNewReno(TCP-NR), and TCPReno(TCP-R). The ERCTP has been tested for 24 mote topology and results reveal that the ERCTP effectively controls congestion and exhibits highest good throughput of 0.2941 Mbps, $\leq$ 100 msec average End-to-End (E-2-E) data packet latency for heterogeneous packet information, $>$ 99% data packet reliability and overall energy efficient behavior (lowest per packet communication cost) in comparison to TCP-WW+, TCP-WW, TCP-NR and TCP-R.

Key words: transport layer, IEEE 802.11, MAC layer, congestion, NACK, cross-layer design, WSN, reliability

1. Introduction.

Heterogeneous WSN is comprised of tiny embedded devices termed as "motes" that have inbuilt features for multitude sensing, processing and communicating information over wireless channel [7, 25, 18]. These devices sense information from the targeted area and communicate it wirelessly to remote base station or sink. There are a range of application scenarios for such ad-hoc heterogeneous networks[1] that range from modern health care to military applications by involving multiple disciplines of control, signals processing and embedded computing [7, 9, 21].

Transport layer of heterogeneous WSN, in general provide E-2-E network connection and the key objective of a transport protocol is to attain reliable data transport, while avoiding congestion and achieving high energy efficiency. Energy efficiency is considered to be the biggest concern for such networks and achieving high energy efficiency is of paramount importance for the longevity of such networks [26, 9]. To combat against these challenges energy efficient hardware design along with efficient communication protocol design for such networks is heavily investigated in the research community [7, 20].

In WSN, the transport layer protocol is responsible for reliably communicating the sensed information form source mote to the sink. In WSN the term "data reliability" is defined in terms of successful data delivery probability from the source mote to the sink mote. The main reason for packet drop includes congestion, collisions due to hidden motes, poor Signal-to-Noise Ratio (SNR) caused by bad channel conditions and link breakage due to mote failure. Undelivered data packets dropped as a result of any of the above listed conditions may lead to unwanted data retransmission that consume considerable amount of mote's energy budget. Although link reliability mechanisms like MAC layer protocol and Automatic Repeat Request (ARQ)[21] keeps the E-2-E packet loss ratio within acceptable limits, but mission critical applications like military, surveillance and industrial automation demand high E-2-E data reliability, which requires the use of highly reliable transport functionality for heterogeneous WSN. The reliability level is enhanced in WSN by the introduction of distributed intermediate storage or buffer motes. The buffer mote is not only meant for buffering the sensed packet information for some defined time interval (depending upon the link conditions, congestion severity etc) but also responsible for prioritized forwarding of information and possible data packet retransmissions as a result of packet drops in forward hops.

Congestion control is an important design parameter for heterogeneous WSN transport protocol. In WSN the term "congestion control" is defined as a way of achieving uncongested network that results in drop-less transmission of data packets from source to sink within the defined E-2-E packet latency threshold. Congestion occurs when:

*This paper is an enhanced version of the paper accepted in 24^{th} IEEE International Conference on Advanced Information Networking and Applications, AINA 2010, Australia, 20–13 April 2010.

†DEBII, Curtin University of Technology, Perth, WA. {a.sharif, v.potdar, a.rathnayaka}@curtin.edu.au wsn.debii.curtin.edu.au

[1]throughout the paper this term means network having scalar and multimedia sensing capability

1. Mote transmit more combined upstream traffic resulting in packet-arrival rate to exceed the packet processing rate at the receiving mote,
2. Mote's data throughput exceeds the link's available data threshold limit, and
3. Due to wireless link issues such as contention, interference, and blind mote problem.

Congestion causes packet drops and unnecessary packet retransmissions followed by significant network's energy depletion. The congested network regions also termed as "hot-spots" may lead to packet drops and retransmitting the lost packet information may result in considerable energy loss.

The congestion control feature of WSN transport protocol [15] is comprised of three functional modules i. e. Congestion Detection, Congestion Notification and Rate Adjustment. Congestion Detection module detects the level of congestion within the WSN network by either taking the E-2-E or H-b-H reliability information. The measured level of congestion then dictates the new rate adjustment for data communication. Upon detecting congestion, the Congestion Notification module informs the neighboring motes about the severity of the congestion in order to avoid extreme congested scenarios, which may lead to enormous data packet drop. Based on the notified congestion level the Rate Adjustment module then defines the new rate adjustment for the source motes in order to mitigate congestion. In most scenarios it is the sink mote that broadcasts this piece of information to the entire network. Given the importance of reliability and congestion control in WSN communication the contribution of this paper is four fold:

1. Firstly, we develop a stochastic control framework for congestion control.
2. Secondly, we design a new transport protocol that is capable of handling both reliability and congestion simultaneously for heterogeneous WSN.
3. Thirdly, we develop data prioritization module for assigning priorities to diverse sensors to allocate more bandwidth for critical applications, while achieving weighted fairness [2].
4. Finally, we provide simulation results that show the performance of the ERCTP in terms of system's good throughput, average E-2-E data packet latency in mili seconds (msec), average data packet drop and average per packet energy consumption for the entire data transmission monitored at source, intermediate/relay and sink motes.

The rest of the paper is organized as follows. After introduction the related work is presented in Section 2 followed by Section 3 and Section 4 where we describe the overview and details of the proposed transport protocol. Section 5 describes the simulation setup used for observing the behavior of the proposed protocol. In Section 6 we outline the simulation results and then we conclude the paper in Section 7.

2. Related Work. In this section we discuss the existing transport protocol schemes for WSN that targets both the congestion control and reliability aspects. The Table 2.1 includes the fundamental differences of these protocols based on the congestion control and reliability support.

In [7], the STCP (Sensor Transmission Control Protocol) solely depends on the buffer occupancy for congestion detection. It notifies congestion implicitly and the congestion avoidance is performed using rate adjustment as well as traffic redirection. STCP achieves controlled variable packet reliability for different flows in the network. STCP offers E-2 E reliability and the loss recovery mechanisms for continuous and event-driven dataflow utilizing NACK and ACK respectively. However, STCP does not incorporate multimedia flow of data, which is an imperative demand for heterogeneous applications.

DST (Delay Sensitive Transport Protocol) [13] detects the congestion considering mote delay estimation in addition to the buffer occupancy and uses implicit (Imp) congestion notification messages for informing the severity of the congestion. It avoids congestion by source rate adjustment policy. DST supports E-2-E event reliability but does not offer an explicit (Exp) loss recovery mechanism for lost packets.

Flush [16] detects congestion based on buffer size, route length and link quality and notifies the congestion implicitly to initiate rate adjustments. Flush ensures the Event reliability in upstream direction and enable E-2-E loss recovery mechanism by the Exp use of NACK signaling.

PORT (Price-Oriented Reliable Transport Protocol) [27] on the other hand detects congestion based on link loss rates and "Node price". The term "Node price" is defined as the number of transmission attempts made before a packet is successfully transferred to the sink. PORT uses Imp congestion notification messages for informing sink about the severity of the congestion and avoids congestion by using traffic redirection and source readjustment policy. It supports reliability in the reverse direction (from sink to source, E-2-E) only.

[2] is a data flow technique allowing different scheduling priorities to statistically multiplexed data flows

It does not support Exp mechanism for packet retrieval (for reliability assurance), as it relies heavily on its congestion control and optimal routing functionality to prevent congestion and packet drop.

CTCP (Collaborative Transport Control Protocol) [11] attains active congestion detection via transmission error losses and buffer overflow. It informs the congestion severity by the Exp use of congestion notification packets and avoids congestion using rate adjustment policy. CTCP achieves controlled variable packet reliability for different applications in network and accomplishes Hop by Hop (H-b-H) reliability by sending ACK and double ACK to each mote. CTCP uses considerable control signaling overhead for ensuring reliability.

ART (Asymmetric and Reliable Transport Protocol) [24] and RCRT (Rate Control Reliable Transport Protocol) [17] detect congestion based on the successful delivery of packets and use Imp congestion notification. ART identifies congestion if ACK is not received from essential dominating sensor motes within specific time interval. However, RCRT on the other hand identifies congestion based on time estimation to recover the loss. Both support E-2-E loss recovery mechanism by the Exp use of ACK and NACK control signaling. ART ensures bidirectional reliability by the use of ACK and NACK for event and query reliability, whereas RCRT uses NACK for ensuring packet reliability in the upstream direction and cumulative ACK to safely delete the memory. RT^2 (Real-Time and Reliable Transport) [14] achieves reliable H-b-H loss recovery between actors using Exp selective-acknowledgments (SACK) and also addresses cross-layer feature and intermediate mote feedback information to obtain the route failures, congestion notifications and transmission rate feedbacks. It uses information in the form of mote delay and buffer size for congestion detection and uses Imp means to notify congestion.

3. Proposed Protocol Overview:ERCTP. As discussed above majority of communication from source to sink in WSN is of H-b-H nature. Congestion, poor link quality, mote failure, collisions due to hidden motes etc. are some of the major factors of packet drop in WSN. As WSN is an energy scarce network, there is a price to be paid in terms of energy for every packet drop, which is given by:

$$E_{price} = N_h \times (E_{NACK} + E_{send})\ joules \tag{1}$$

where,

E_{price} = Total energy required for dropped packet retransmission (joules),

N_h = Number of hops between the sink and the source motes,

E_{NACK} = Energy consumed by a mote to transmit the NACK packet (joules), and

E_{Send} = Energy consumed by the mote to transmit the actual missing data packet (joules).

Hence in order to conserve the mote's energy budget (or network's energy budget) packet drop should be minimized to an extent such that the applications specific QoS is satisfied. It is therefore transport protocol's responsibility to control congestion, and if at all it occurs then notify the source motes and readjust their sending rate in order to mitigate congestion effectively. The heterogeneous WSN transport protocol also has an additional feature of distributed network storage which is invoked when packet recovery is desired. As WSN is comprised of energy scarce motes so loss of information due to packet drop caused by congested network condition is inevitable not only for certain application scenarios but also from the energy consumption point of view. In the next section we will describe our proposed transport protocol scheme, named ERCTP, that ensures Congestion detection, Congestion avoidance and Data loss recovery. The block diagram of the ERCTP is shown in Figure 3.1. The proposed scheme is comprised of the following functional modules: Congestion Control Module and Reliability Module.

3.1. Congestion Control Module. Congestion control module is composed of three separate sub-modules i. e. congestion detection module, congestion notification module and congestion avaoidance using source rate adjustment module. The basic functionality of these modules was discussed in Section 1 earlier and hence we have not repeated it again. For congestion control we envisage a scheme based on the stochastic control framework theory since we are considering the entire network communication in probabilistic terms. Stochastic control framework provides an effective mean to control congestion by observing the average E-2-E packet latency at the sink mote. Here the packet latency is given by:

$$T_{delay} = T_{pr} + T_Q + T_{PP} \tag{2}$$

where,

T_{delay} =average E-2-E packet latency (msec),

$T_{PP} = \tau$ =average packet processing time at a given mote(msec),

T_Q =average mote interface queue latency(msec), and

T_{pr} =average 1-hop propagation time (msec).

TABLE 2.1
Comparison for Transport Layer Protocols

Protocols	RCRT	FLUSH	STCP	PORT	ART	CTCP	RT^2	DST
Congestion Control	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Congestion Detection	Time to recover loss	Route length and link quality	Buffer size	Mote price and link-loss rates	ACK received to "Essential motes"	Transmission error packet loss and Buffer size	Mote delay/ Buffer size	Mote delay/ Buffer size
Congestion Notification	Implicit	Implicit	Implicit	Implicit	Implicit	Explicit	Implicit	Implicit
Congestion Avoidance	Rate Adjustment	Rate Adjustment	Traffic redirection / Rate adjustment	Traffic redirection / Rate adjustment	Reduce traffic of "Non-essential motes"	Rate Adjustment	Rate Adjustment	Rate Adjustment
Reliability Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Reliability Direction	Up	Up	Up	Up	Both	Up	Up	Up
Reliability Measure	Packet	Packet	Packet	Event information	Event	Packet	Packet	Event
Loss Recovery	E-2-E	E-2-E	E-2-E	-	E-2-E	H-b-H	H-b-H	E-2-E
Loss Notification	NACK & cumulative ACK	NACK	NACK/ ACK	-	ACK (events)/ NACK (queries)	ACK/ Double ACK	SACK	-

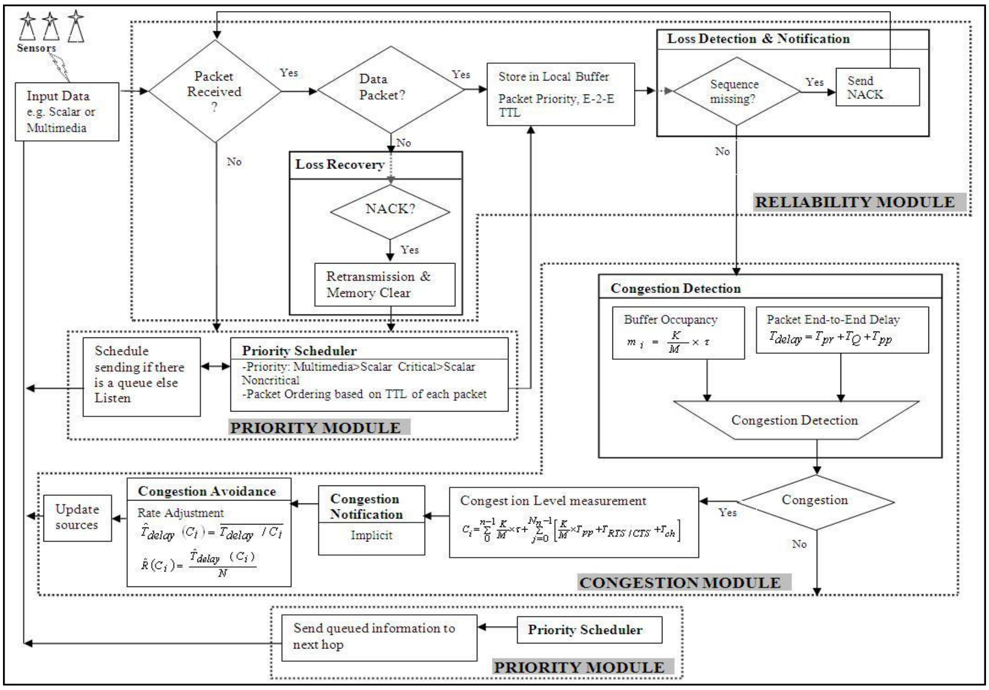


Fig. 3.1. *Proposed Transport protocol Scheme*

To detect congestion occurrence we define an index termed as the Congestion Index (C_i) in msec, which represents the congestion state of the network and is governed by the intermediate mote's buffer occupancy level and E-2-E propagation delay. Buffer Occupancy Index (m_i) for any intermediate mote is defined as 'ratio of occupied storage locations to the maximum available storage locations multiplied by the time required to process one data packet information'. The buffer mote uses the local queue for storing the packet and will read the information based on the prioritized nature of the data and the TTL information for that particular packet ID.

$$m_i = \frac{Total\,Free\,Space\,in\,the\,local\,storage}{Total\,Memmory\,space\,for\,local\,storage} \times \tau \tag{3}$$

or

$$m_{i-local} = \frac{K_{local}}{M_{local}} \times \tau_{local} \tag{4}$$

where,

τ = average local processing time for one data packet at any given intermediate mote.

For n number of intermediate storage motes each having congestion state m_i (for i^{th} mote), then C_i is mathematically given by:

$$C_i = \sum_{i=0}^{n-1} m_{i-local} + T_{E-2-E\,Pr\,delay} \tag{5}$$

If 'N_h' is the number of hops between source and sink mote, T_Q as the interface queue delay and per mote per link propagation delay (including the local queue latency) of T_{pr} then E-2-E propagation latency is given by:

$$T_{E-2-E\,Pr\,delay} = N_h \times T_{pr} \tag{6}$$

where,

$$T_{pr} = T_Q + T_{MAC} \tag{7}$$

$$T_Q = m_{i-interface} \times \tau_{inetrface} \tag{8}$$

and T_{MAC} is given by:

$$T_{MAC} = T_{RTS/CTS} + T_{ch} \tag{9}$$

where,

T_{MAC}= MAC access delay,

$T_{RTS/CTS}$ = Latency due to ongoing transmission as indicated by RTS/CTS, and

T_{ch} = Channel access delay.

So Eq.(5) i. e. congestion index, can now be expressed as:

$$C_i = \sum_{i=0}^{n-1} \frac{K_{local}}{M_{local}} \times \tau_{local} + \sum_{j=0}^{N_h-1} [\frac{K_{interface}}{M_{interface}} \times \tau + T_{RTS/CTS} + T_{ch}] \tag{10}$$

Eq. (10) represents the congestion index, which is helpful in deciding the future rate adjustments for the source motes. Now that we have calculated the congestion index, we move on to the next stage i. e. congestion detection.

3.1.1. Congestion Detection. In WSN, Congestion Detection is defined as "the means of detecting the congestion state of the network based on the link capacity and intermediate motes buffer occupancy". Eq. (10) is used for measuring the congestion state of the network which can then be used for the Mean-Square Estimation (MSE) of a new T_{delay} or new rate value for source motes (as we are taking the entire E-2-E scenario). Here we limit ourselves to C_i computation related to congestion only, avoiding any wait state for carrier sensing at the MAC level.

The joint density function[3] that relates the E-2-E data packet latency (T_{delay}) and Congestion index C_i (both are dependent variables) is given by $f_{T_{delay}C_i}(t_{delay},c_i)$, so the posteriori estimate for T_{delay}, $\hat{T}_{delay}(C_i)$ changes as C_i changes and is fixed for some constant z i. e. $C_i = z$. At the start we assume no congestion i. e. $C_i = C_0 = z_0$. Hence, the estimated error (J) [10] is given by:

$$J = \int_{-\infty}^{\infty} \int \left[T_{delay} - \hat{T}_{delay}(C_i)\right]^T \left[T_{delay} - \hat{T}_{delay}(C_i)\right] \, dT_{delay}.dC_i \tag{11}$$

or using the Baye's rule [10] we can write Eq.(11) as follows:

$$J' = \int_{-\infty}^{\infty} f_{C_i}(C_i) \int_{-\infty}^{\infty} \left[T_{delay} - \hat{T}_{delay}(C_i)\right]^T \left[T_{delay} - \hat{T}_{delay}(C_i)\right] f_{T_{delay|C_i}}(T_{delay}|C_i).dT_{delay}.dC_i \tag{12}$$

Here $f_{C_i}(C_i)$and the inner integral is non-negative so the value of conditional mean-square error J can be minimized by treating C_i as a constant and minimizing J for every such C_i. The value of J becomes:

$$J' = \int_{-\infty}^{\infty} \left[T_{delay} - \hat{T}_{delay}(C_i)\right]^T \left[T_{delay} - \hat{T}_{delay}(C_i)\right].f_{T_{delay|C_i}}(T_{delay}|C_i) \, dT_{delay}.dC_i \tag{13}$$

Now considering the conditional case and using the fact that $\hat{T}_{delay}(C_i)$changes as C_i changes and is fixed for $C_i = z$, where z is a constant it comes out to be:

$$\frac{\partial J'}{\partial T_{delay}(C_i)} = -\overline{2\hat{T}_{delay}/C_i} + 2\hat{T}_{delay}(C_i) = 0 \tag{14}$$

or finally MSE of the new $T_{delay}(C_i)$ is given by:

$$\hat{T}_{delay}(C_i) = \overline{T_{dealy}/C_i} \tag{15}$$

where for channel load threshold limit of λ_{Thresh} ,

$$\hat{T}_{delay}(C_i) \leq \lambda_{Thresh} \tag{16}$$

Hence the new estimated rate value can be expressed as:

$$\hat{R}(Ci) = \frac{\hat{T}_{delay}(Ci)}{N} \tag{17}$$

3.1.2. Congestion Notification. The purpose of the congestion notification module is to inform the neighboring motes about the severity of the congestion in order to avoid extreme congestion scenarios, which may lead to enormous data loss. Based on the computed $T_{delay}(C_i)$ value from Eq. (15), the sink periodically informs the source about the new rate value using Eq. (17) and the level of congestion C_i using Eq. (10).

3.1.3. Source Rate Adjustment. The purpose of the Source Rate Adjustment Module is to define the new transmission rate adjustment for child motes in order to mitigate congestion. The protocol monitors the instantaneous network statistics which helps sink to explicitly and periodically send the estimated value of rate adjustment to source motes, which is obtained based on congestion index calculation. Here this packet is generated separately for each source for every $'x'$ number of packets received from each source at the sink. This rate adjustment value indicates how much transmission rate should be increased or decreased for prospective data transmissions. For example, the rate adjustment figure would be either a negative value in case of foreseen congestion or a positive value if the channel is underutilized. Source motes then start sending the data packets with this new rate value $R(C_i)$, adjusted based on this feedback, in order to reduce the network's congested state and to avoid channel under utilization.

[3] function of two or more randomly distributed variables from which a single probability can be obtained indicating the range within which all the variables in the function falls.

3.2. Reliability Module. The purpose of the reliability module is to:

1. Retain the sensed information at designated intermediate motes for a defined interval of time,
2. Resend the sensed information upon receipt of NACK packet i. e. when packet loss occurs, and
3. Free up the motes memory upon receipt of ACK packet or after interval expiry.

To ensure reliability the ERCTP has introduced the concept of Distributed Memory Storage (DMS). In DMS the designated intermediate motes[4] are used to temporarily store packet information (thereby ensuring the transport level reliability), which can then be used for retransmission in case the sink fails to receive the designated packet information. In case the sink breaks its communication link with the immediate storage mote due to poor channel condition or due to storage motes energy depletion, then the NACK for the missing packet is propagated back to the next storage mote for corresponding information retrieval thus ensuring reliability of the proposed light weight transport protocol. The probability that a data packet is dropped is basically the sum of various probabilistic failures and is given by:

$$P_f = P_{lf} + P_{nf} + P_{hn} \tag{18}$$

where,

P_f = Total probability of data packet being dropped,

P_{lf} = Probability of packet drop due to link failure,

P_{nf} = Probability of packet drop due to mote failure, and

P_{hm} = Probability that a packet is dropped due to a hidden mote.

The idea of P_{lf}, P_{nf}, P_{hm} comes from the MAC/Wireless-Physical layer, by looking at motes energy, Bit Error Rate (BER) and packet collisions. So the total probability (success probability,$P_{success}$) with which data packet arrives at sink station is given by:

$$P_{success} = 1 - P_f \tag{19}$$

Hence there always exists a probability of packet failure and basically the index of P_f i.e α depicts the time to retain the data packets at the intermediate storage motes memory which is given by:

$$\frac{10 \times RTT}{\alpha}(msec) \tag{20}$$

where

α = storing time factor in this case, and

RTT = Round Trip Time in msec.

Based on the formulation, we are now going to simulate ERCTP by taking into consideration the possible real world effects of hidden mote transmissions and individual mote energy consumption for packet communication.

4. Simulation Setup. In this section we will describe the network topology and the parameters used for extensive testing of ERCTP for heterogeneous WSN. The aim of the simulation setup is to monitor the average system good throughput, average data packet drop, average E-2-E data packet latency and average per packet energy consumed by the source, relay and sink motes.

4.1. Performance Metrics. The ERCTP is evaluated against the following performance metrics:

1. Average Good throughput,
2. Average Data packet drop,
3. Average E-2-E Data packet latency, and
4. Average per packet energy consumed.

4.1.1. Average Good throughput . Average Good Throughput (AGT) is defined as the percentage ratio of sum of data sent by all sources to the data received by the sink. Mathematically it can be written as:

$$AGT = \frac{Total\, sent\, data}{Total\, received\, data} \times 100 \tag{21}$$

4.1.2. Average Data packet drop. Average packet drop (PD) is defined as the % of the data loss (difference between sent and received data) to the sent data. The main contributors of data loss are collisions at the receiving end in the presence of blind motes, congestion and link failure due to mote energy depletion. Mathematically it can be written as:

$$PD = \frac{Total\, sent\, data\, packets - Total\, received\, data\, packets}{Total\, sent\, data} \times 100 \tag{22}$$

[4]also called as “storage motes or buffer motes

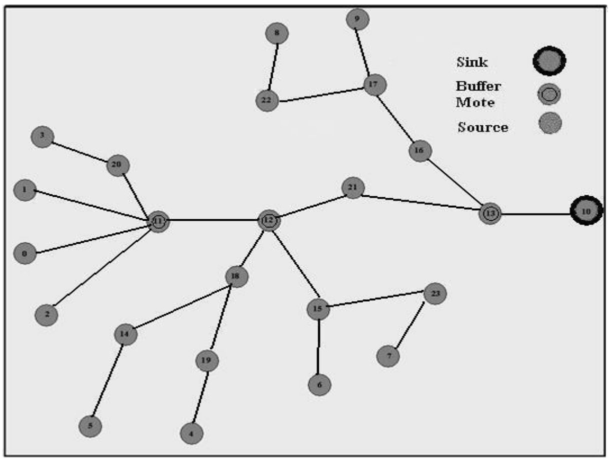


FIG. 4.1. *Network Topology*

4.1.3. Average E-2-E Data packet latency. E-2-E data packet latency $T_{delay}(E-2-E)$ is defined as the total time a packet would take from the source to sink (E-2-E). This includes all the possible delays as a result of queuing, retransmissions at the MAC layer, propagation delays and transfer time. Mathematically it can be written as:

$$T_{delay}(E-2-E) = (T_{rec} - T_{send}) \times 1000\, msec \tag{23}$$

4.1.4. Average per packet energy consumed. Average per packet energy E_{avg} consumed by source, relay or sink motes is defined as the % of total energy consumed in packet processing (that includes transmit, receive, relaying etc) to the total number of packets that it handles. Mathematically it can written as:

$$E_{avg} = \frac{Total\, Energy\, consumed}{Total\, packets} \times 100\, mJ \tag{24}$$

4.2. Network Topology. The network topology used for evaluating the ERCTP is shown in the Figure 4.1. The motes $0-9$ are considered as source motes while motes 11, 12 and 13 are considered as intermediate storage motes and mote 10 is considered to be a sink mote. The remaining network parameters, source nature and their priorities are listed in the Table 4.1 and Table 4.2 respectively.

Let us consider that the network is comprised of N motes placed in open/free space environment. The separation between the sending mote and receiving mote given by d_{tx-rx} is maintained to avoid the issue of hidden motes[5]. As outlined in Eq. 25 & Eq. 26 this distance is calculated as follows:

$$d_{int} \geq 1.778 * d_{tx-rx} \tag{25}$$

$$d_{tx-rx} \leq 0.5624 * d_{int} \tag{26}$$

where,

d_{int} = receiver interference region, and is given by:

$$d_{int} \geq 4\sqrt{SNR_{Thresh}} * d_{tx-rx} \tag{27}$$

$$SNR_{rec} = \frac{P_{rx}}{P_{int}} = \left(\frac{d_{int}}{d_{tx-rx}}\right)^4 \geq SNR_{Thresh} \tag{28}$$

For present simulations we have used a value of 10 for SNR_{Thresh}.

4.3. Comparative Transport Layer Protocol Standards . In this section we will briefly describe the transport layer protocols which we have used for the detailed evaluation against ERCTP and they are:

1. TCP-R,
2. TCP-NR,
3. TCP-WW, and
4. TCP-WW+.

ERCTP is evaluated against TCP-WW+, TCP-WW, TCP-NR and TCP-R. The source codes of these protocols were freely available but we were unable to find the codes for many recent protocols that we have mentioned in the related work section. TCP-NR is an extended version of TCP-R. TCP-NR improves retransmission during the

[5]for more detail refer to [22]

Table 4.1
Network Parameters

Parameter	Values
Frequency (Hz)	$914e^{+6}$
Transport Protocols	ERCTP, TCP-WW+ [5], TCP-WW [8], TCP-NR[3], TCP-R[4]
MAC	IEEE802.11 [1]
RX and CS Threshold (W)	$3.6252e^{-10}$ & $1.559e^{-11}$
Routing agent	Ad hoc On-Demand Distance Vector (AODV) [19]
CP Threshold	10
Ifqlen (packets)	50
Mote Initial power (W)	100
Mote Idle power (W)	$712e^{-6}$
Mote Rx power (W)	$35.28e^{-3}$
Mote Tx power (W)	$31.32e^{-3}$
Mote Sleep power (W)	0.001
Data Packet Size (Bytes)	512
x	10
α	$0 < \alpha \leq 1$
SNR_{Thresh}	10

Table 4.2
Source Priority

Source	Priority	Nature of Source
$1, 4, 7$	1	Multimedia
$3, 5, 6$	2	Scalar-critical
$0, 2, 8, 9$	3	Scalar-less critical

fast recovery phase of TCP-R. TCP-WW is a sender-side-only modification to TCP-NR and TCP-WW+ is an evolution of TCP-WW. The main difference in congestion avoidance feature of TCP-WW variants with respect to TCP-R variants is that Reno variants halves the congestion window after three duplicate acknowledgments or after a timeout where as Westwood variants attempt to select a slow start threshold and a congestion window considering the bandwidth estimation. In TCP-WW+, the original bandwidth estimation algorithm in TCP-WW has been modified to cope with ACK compression effect [12], as the TCP-WW fails to function properly in the presence of ACK compression.

5. Simulation Results and Discussions. Network Simulator (NS-2) [2] is used as a simulation platform for testing and comparing the functionality of the ERCTP with TCP-WW+, TCP-WW, TCP-NR and TCP-R. We have considered heterogeneous traffic flow including multimedia with a packet size of 512 bytes. The simulation has been performed five times and the average values of the results are plotted to ensure reliability in the given results.

5.1. Average Good Throughput Comparison. Good throughput comparison of the ERCTP with TCP variants like TCP-WW+, TCP-WW, TCP-NR and TCP-R is shown in Figure 5.1. From the Figure 5.1 we can see that the TCP-R and the ERCTP offer high system good throughput i. e. 0.2941 and 0.2927 Mbps in comparison to TCP-WW+, TCP-WW and TCP-NR whose good throughputs are 0.2874, 0.2902 and 0.2668 Mbps respectively. The ERCTP, which is basically a sink enabled E-2-E congestion control, utilizes the stochastically estimated values of T_{delay} (function of C_i) for defining the new transmission rate values for every

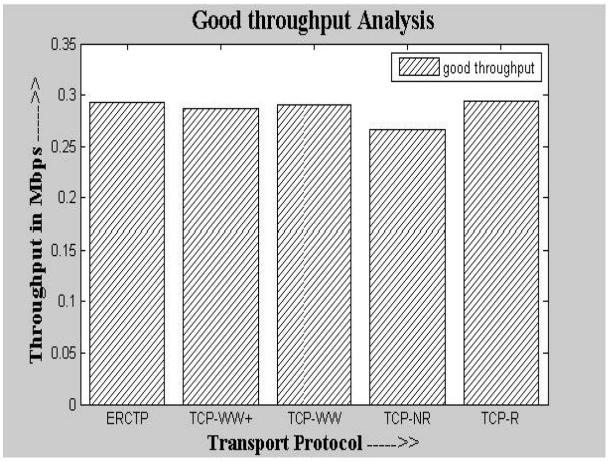
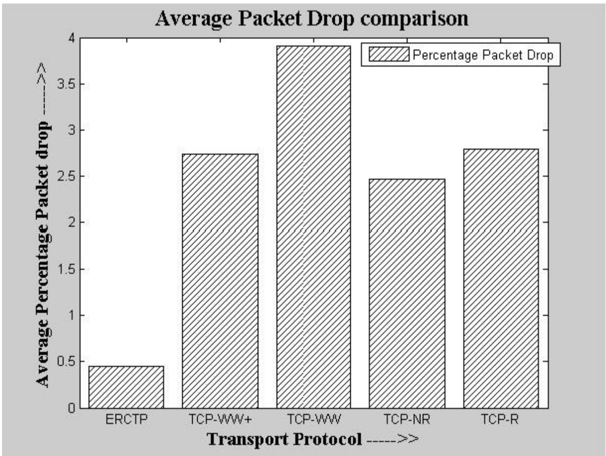


FIG. 5.1. *Average E-2-E Good Throughput Comparison*

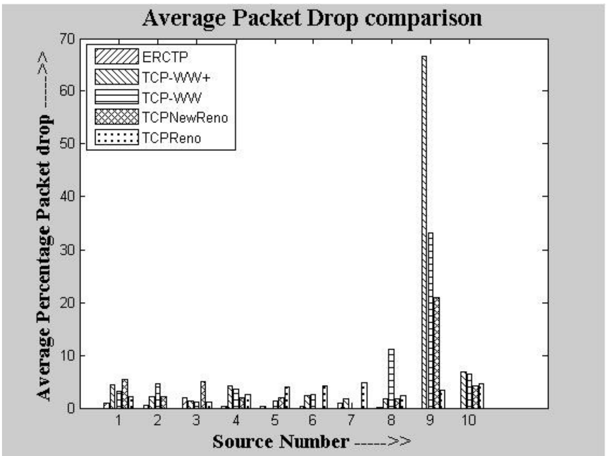
source and updates itself by the feedback parameter also called estimation error 'J'. Since the rate estimation involves the real time monitored system statistics in form of 'C_i' that is directly coupled to T_{delay} (the time gap between successive packet transmissions in other words the transmission frequency) which is why ERCTP shows high good throughput. By intelligently employing the system statistics in a stochastic control framework, not only prevents congestion but also the associated packet drops thus resulting in energy efficiency. In comparison, TCP-WW+ and TCP-WW (also called the sender-side variants of TCP-NR) relies on mining of the ACK control signaling for setting the congestion control parameters like slow-start threshold limit and congestion window which are then used in estimating the source transmission rate values. As mining of ACK stream involves the close monitoring of per packet ACK besides the actual data flow, it includes the E-2-E T_{delay} for data packet with an addition of the ACK reception latency (which is variable) at the source side, which would increase the time gap between the successive transmissions and hence results in a drop of system good throughput.

5.2. Average Data Packet Drop Comparison. Figure 5.2.a and 5.2.b shows the average E-2-E data packet drop comparison of the ERCTP with TCP variants like TCP-WW+, TCP-WW, TCP-NR and TCP-R. From these figures it is clear that among all protocols only the ERCTP offers high packet reliability for both the high and low priority packets. Of the total communication, the ERCTP exhibits approximately 0.46% packet drop in comparison to 2.74%, 3.91%, 2.47% and 2.79% packet drops for TCP-WW+, TCP-WW, TCP-NR and TCP-R respectively. Other than the use of storage motes for storing packets at intermediate buffer motes another major reason the ERCTP offers high packet reliability is its congestion control dependency over the real time/ instantaneous monitored network statistics that actually gives true idea of network congestion. In comparison the TCP variant like TCP-WW+ and TCP-WW avoids packet drops by the use of an additive increase/adaptive decrease (AIMD) paradigm to enhance the classic AIMD algorithm. When it experiences packet loss, it performs E-2-E bandwidth estimation (at the sender side) for future packet transmissions. As this sender based dynamic phenomenon of rate adjustment is entirely based on received ACK monitored statistics it could result in enormous packet drops before the new bandwidth estimation has taken place, which is why their average percentage data packet drop ranges between $2.74 - 3.91\%$ of the entire communication.

5.3. Average E-2-E Data Packet Latency Comparison. Figure 5.3 shows the E-2-E data packet latency comparison of the ERCTP with TCP-WW+, TCP-WW, TCP-NR and TCP-R. From this comparison it is obvious that the ERCTP, which uses real time instantaneous monitored statistics of the packet E-2-E latency coupled to the network congestion index, outperforms the rest. For the packet information coming from sources that uses TCP-WW+ and TCP-WW as transport agents, which probes the incoming ACKs for every sent data packet information for defining their rate plan based on the estimation of bandwidth, actually suffers from a large variable delay. For all TCP variants including TCP-NR and TCP-R, for large number of hops between sources and sink motes, this variable delay is in between $370 - 700$ msec E-2-E whereas for the ERCTP with similar number of hops the average E-2-E data packet latency is in between $70 - 85$ msec. For all data packet priorities the E-2-E data packet latency offered by ERCTP is < 100 msec while it is $\gg 100$ msec for TCP



(a) E-2-E percentage overall packet drop comparison



(b) Percentage packet drop comparison for various sources

FIG. 5.2. *Average Packet Drop Comparison*

variants and is highly variable depending upon the number of hops between the source and sink mote. Thus, these results clearly shows the use of intermediate storage mote's buffer occupancy and instantaneous network channel statistics, that defines the E-2-E congestion index of the entire network coupled to the rate adjustment scheme, for achieving acceptable E-2-E data packet latency confined to the QoS objectives of the application [23, 6].

5.4. Average Per Packet Energy Consumed Comparison. Figure 5.4 shows the per packet energy consumed (in mili Joules, mJ) by the source, relay and sink motes that uses various transport layer protocols. From the Figure 5.1 we notice that only TCP-R and ERCTP offer high throughput i. e. 0.2941 and 0.2927 Mbps while lowest of 0.2668 Mbps offered by TCP-NR. So if we start relating this system good throughput statistics with per packet energy consumption statistics as shown in Figure 6 then it is obvious that among all transport layer protocols only the ERCTP shows energy efficient behavior. For ERCTP, per packet energy consumed by the source, relay and sink motes is 0.3364, 0.4793 and 0.6942 mJ. Since TCP-NR has lowest throughput among all, its per packet energy consumption by the source motes is small i. e. 0.3126 mJ while highest being consumed by TCP-WW+, TCP-WW and TCP-R i. e. 0.4475, 0.4410 and 0.4394 mJ respectively. The reason why TCP-WW+ and TCP-WW offers high per packet energy cost is being justified by its control channel probing (mining of ACK control signaling for bandwidth estimation for future transmissions). Finally, in the case of TCP-R, which offers approximately similar system good throughput as offered by the ERCTP, has shown high per packet energy consumption behavior for source, relay and sink motes (i. e. on average per

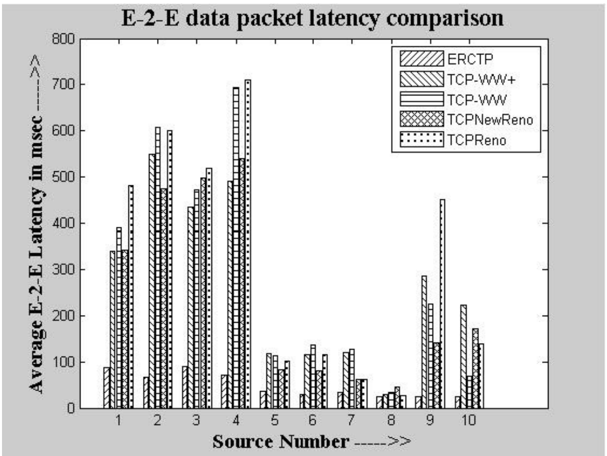


FIG. 5.3. *Average E-2-E Data Packet Latency Comparison*

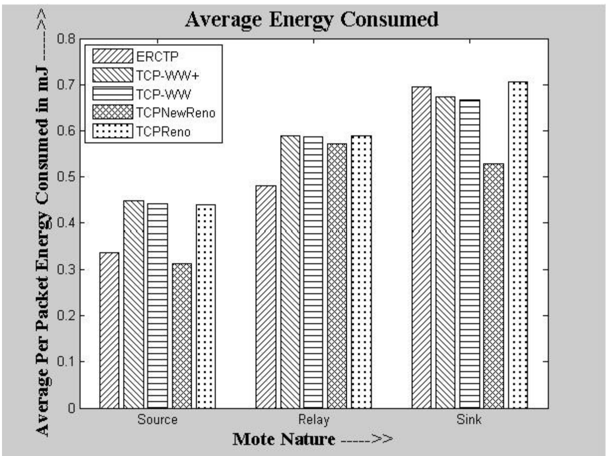


FIG. 5.4. *Average Per Packet Energy Consumed Comparison*

packet energy consumed by source, relay and sink motes are 0.1030, 0.1101 and 0.0114 mJ) and it is caused by frequent retransmissions and associated control signaling.

6. Discussions and Conclusions. Longevity of WSN is an important design challenge and was addressed in multiple disciplines of control, signal processing, and communication protocol design for WSN. It can be achieved by ensuring data reliability and congestion control, which are the two vital aspects of any transport protocol design for WSN. In this paper, we have surveyed various transport protocol schemes for WSN and proposed a light weight transport protocol, ERCTP, which is capable of detecting congestion, notifying it, source rate readjustment and data retrieval for any possible data loss that may happen either due to congestion or due to poor channel conditions i. e. packet is considered to be dropped for high BER, collisions at the receiver side due to transmissions from a hidden mote (hidden mote problem).

We extensively simulated the ERCTP against TCP-WW+, TCP-WW, TCPNewReno and TCPReno and the results reveal that considerable reduction in E-2-E data packet latency has been observed for the ERCTP around $<$ 100 msec for heterogeneous packet information. Also for the ERCTP and TCP-R highest average good throughput is achieved i. e. 0.2927 and 0.2941 Mbps while effectively maintaining the $>$ 99% achieved source priority (at sink) for various sources with only 3 buffer motes. For the entire communication ERCTP exhibits the energy efficient behavior in comparison to rest all and exhibits only $<$ 0.45% packet drop which is minimum of all. The results reveal that the rate adaptation mechanism prevents major data packet drop if the stochastically distributed network conditions are known. Under uncongested/congested network conditions the ERCTP performs well in comparison to TCP variants.

In the next phase we will try to melt the prevalent or reciprocal functionalities of Transport layer and underlying MAC and Wireless-Physical layers, as we have seen that Transport layer functionality is mostly dependent on MAC and Wireless-Physical layers [22]. Cross-layering the transport with MAC and Wireless-Physical layers enables an idea of low level signal and channel details like received signal strength, packet ACK at MAC level, channel access time, channel loading, packet retransmission, intermediate queue occupancy, mote energy updates etc at the transport level, thereby helping in the design of more robust transport layer protocol for achieving high network throughput while simultaneously addressing the issues of congestion detection and data reliability.

REFERENCES

[1] *Ieee p802.11, main general info page.*
[2] *The network simulator:ns-2, http://www.isi.edu/nsnam/ns/index.html, 2008.*
[3] *Tcp new reno, http://www.faqs.org/rfcs/rfc3782.html, 2008.*
[4] *Tcp reno, http://tools.ietf.org/html/rfc1644, 2008.*
[5] *Tcp westwood+, http://c3lab.poliba.it/index.php/westwood, 2008.*
[6] *How delay and packet loss impact voice quality in voip,*, 2008.
[7] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, *A survey on wireless multimedia sensor networks*, Comput. Netw., 51 (2007), pp. 921–960.
[8] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, *Tcp westwood: end-to-end congestion control for wired/wireless networks*, Wirel. Netw., 8 (2002), pp. 467–479.
[9] S. Ci, H. Wang, and D. Wu, *A theoretical framework for quality-aware cross-layer optimized wireless multimedia communications*, Adv. MultiMedia, 2008 (2008), pp. 1–10.
[10] L. L. Frank, L. Xie, and D. Popa, *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory, Second Edition*, CRC Press, 2007, ch. 1-7, pp. 1–552.
[11] E. Giancoli, F. Jabour, and A. Pedroza, *Ctcp: Reliable transport control protocol for sensor networks*, in International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP, dec. 2008, pp. 493 –498.
[12] L. A. Grieco and S. Mascolo, *Tcp westwood and easy red to improve fairness in high-speed networks*, in PIHSN '02: Proceedings of the 7th IFIP/IEEE International Workshop on Protocols for High Speed Networks, London, UK, 2002, Springer-Verlag, pp. 130–146.
[13] V. Gungor and O. Akan, *Dst: delay sensitive transport in wireless sensor networks*, in International Symposium on Computer Networks, 2006, pp. 116 –122.
[14] V. C. Gungor, O. B. Akan, and I. F. Akyildiz, *A real-time and reliable transport (rt) 2 protocol for wireless sensor and actor networks*, IEEE/ACM Trans. Netw., 16 (2008), pp. 359–370.
[15] Y. Iyer, S. Gandham, and S. Venkatesan, *Stcp: a generic transport layer protocol for wireless sensor networks*, oct. 2005, pp. 449 – 454.
[16] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica, *Flush: a reliable bulk transport protocol for multihop wireless networks*, in SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems, New York, NY, USA, 2007, ACM, pp. 351–365.
[17] J. Paek and R. Govindan, *Rcrt: rate-controlled reliable transport for wireless sensor networks*, in SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems, New York, NY, USA, 2007, ACM, pp. 305–319.
[18] Q. Pang, V. W. S. Wong, and V. C. M. Leung, *Reliable data transport and congestion control in wireless sensor networks*, Int. J. Sen. Netw., 3 (2007), pp. 16–24.
[19] C. Perkins, E. Royer, and S. Das, *Ad hoc on-demand distance vector (aodv) routing rfc 3561, 2003.*, 2003.
[20] V. Potdar, A. Sharif, and E. Chang, *Wireless sensor networks: A survey*, in WAINA '09: Proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops, Washington, DC, USA, 2009, IEEE Computer Society, pp. 636–641.
[21] Z. Rosberg, R. Liu, A. Dong, L. Tuan, and S. Jha, *Arq with implicit and explicit acks in wireless sensor networks*, nov. 2008, pp. 1 –6.
[22] A. Sharif, V. Potdar, and A. Rathnayaka, *Dependency of transport layer protocols on the ieee 802.11 and ieee 802.15.4 mac/phy layer protocols for wsn: A step towards cross-layer design*, Int J. of Busn. Data Comm. and Netw., IJDBCN, 3 (2010), pp. 1–30.
[23] E. D. Siegel, *Applications and infrastructure issues*, Wiley Computer Publishing John Wiley & sons.
[24] N. Tezcan and W. Wang, *Art; an asymmetric and reliable transport mechanism for wireless sensor networks*, Int. J. Sen. Netw., 2 (2007), pp. 188–200.
[25] C. Wang, K. Sohraby, B. Li, M. Daneshmand, and Y. Hu, *A survey of transport protocols for wireless sensor networks*, Network, IEEE, 20 (2006), pp. 34 – 40.
[26] N. Zhao and L. Sun, *Research on cross-layer frameworks design in wireless sensor networks*, mar. 2007, pp. 50a –50a.
[27] Y. Zhou, M. Lyu, J. Liu, and H. Wang, *Port: a price-oriented reliable transport protocol for wireless sensor networks*, nov. 2005, pp. 117 –126.

Edited by: Jemal H. Abawajy, Mukaddim Pathan, Al-Sakib Khan Pathan, and Mustafizur Rahman
Received: September 24th, 2010
Accepted: November 10th, 2010