ISSN : 2231-0711

Mohammed Ahmed Abdala et al | International Journal of Computer Science Engineering and Technology(IJCSET) | February 2014 | Vol 4, Issue 2,29-35

Real-time and Reliable Video Transport Protocol (RRVTP) for Visual Wireless Sensor Networks (VSNs)

Dr. Mohammed Ahmed Abdala , Mustafa Hussein Jabbar
College of Information Engineering, Al-Nahrain University,
Baghdad, Iraq

***Abstract*- Recent progress of Visual Sensor Networks (VSNs) has been resulted from great development in cameras techniques that has been enabled the development of single chip camera modules that could easily be embedded into inexpensive transceivers. The interconnection of multimedia sources with inexpensive communication devices has enhanced research in the networking of visual sensors. Due to the large size of the multimedia that captures by visual sensor node, video streams require high bandwidth for a multi-hop wireless environment. So the main two factors influencing in design of transport protocol for visual sensor network are reliability and real time.**

This paper proposes Reliable Real Time Video Transport Protocol (RRVTP) that attempts to solve the reliability and real-time problems by sending captured video in real-time and guarantees the delivery of all corrupted/dropped frames for reliable storage (playback).

Keywords— bandwidth, multi-hop, transport protocol, reliability, real-time and playback.

I. INTRODUCTION

The development in CMOS technology has enabled the development of single chip camera modules that could easily be embedded into inexpensive transceivers. Moreover, microphones have for long been used as an integral part of wireless sensor nodes. The interconnection of multimedia sources with inexpensive communication devices has fostered research in the networking of visual sensors [1]. When several number of visual sensor node is connected together will produce what is known as Visual Sensor Networks (VSNs).

A Visual Sensor Network (VSN) is a network of spatially distributed smart camera devices capable of processing and fusing images of a scene from a variety of viewpoints into some form more useful than the individual images. A visual sensor network may be a type of wireless sensor network, and much of the theory and application of the latter applies to the former. The network generally consists of the cameras themselves, which have some local image processing, communication and storage capabilities, and possibly one or more central computers, where image data from multiple cameras is further processed and fused (this processing may, however, simply take place in a distributed fashion across the cameras and their local controllers). Visual sensor networks also provide some high-level services to the user so that the large amount of data can be distilled into information of interest using specific queries [2] [3].

The primary difference between visual sensor networks and other types of sensor networks is the nature and volume of information the individual sensors acquire. Unlike most sensors, cameras are directional in their field of view, and they capture a large amount of visual information which may be partially processed independently of data from other cameras in the network [4].

At the transport layer, research has been focused in congestion control and reliability. Wireless links in VSNs are error prone and thus have a much higher error rate than traditional computer networks. Besides, communication is made across many of those links. Thus, it is important to guarantee reliability in data delivery, implementing loss detection and a retransmission mechanism [5].

Traditional protocols like TCP and UDP can not be used as a transport protocol in VSNs due to the end-to-end delay that caused from retransmission mechanism in TCP and unreliability in UDP protocol. All that led to the search for a new transport protocol for VSNs [4].

II. PROPOSED RRVTP PROTOCOL

Proposed Reliable Real Time Video Transport Protocol (RRVTP) is designed for end-to-end, multiple senders–single receiver, real-time transfer of stream data. The protocol provides facility for jitter compensation and detection of out of sequence arrival in data that are common problems during transmissions on a network. This protocol attempts to solve the Reliability and Real-time problems by sending captured video in Real-time without reliability and guarantees the delivery of all corrupted/dropped frames for reliable storage (playback).

III. PROTOCOL DESIGN

The protocol must be light weighted (designed with less complexity) in order to reduce overhead, multi-platform compatible and ability to be implemented on different architectures and operating systems.

The main goals for RRVTP which have been taken into account when designing the protocol are:

- Real-Time: the main goal of the protocol is to deliver voice, video and other events that are captured from

different kinds of sensors to the monitoring center in real-time for live examination to match the usage of surveillance applications.

- Reliability: the second goal is to guarantee that all information captured from the sensors are delivered without corruption or loss and stored in sequence for future retrieval.

- Efficiency: finally, the protocol must find the best compromisation between real-time and reliability and provide an efficient usage of the transfer channel and lowers the overhead on the processing units.

IV. RRVTP Connection Scenario

After the node(s) is being attached, it starts transmitting data packets containing the information captured by the sensor. Each reading is represented as a frame and each frame assigned an incremented sequence number. The frame is stored in a temporary database before being sent. At the same time of the transmission of the data a timer working on a periodic manner is started. It helps making decision about the retransmission process starting by sending a query received packets trigger that contain the last sequence number that has been sent by the client when timer ends. After that the sink respond to that request by issuing two types of packets. The first one contain lists of received packet sequence details, the second one contain a list of the lost packets sequence details.

Retransmission process occurs in parallel with the data transmission process in a way that it does not affect the real-time feature of the protocol. Fig. 1 explains in detail how this protocol works

V. Packet Header Format

In the following sections, the RRVTP structure is reviewed especially the packet types that are exchanged between client and server. A brief description on packet headers given below:

A. Packet Types

There are several types of packets that are being exchanged between nodes and sink in RRVTP as described below:

1) Data Packet: is sent from node to sink when an event occurs. This packet carries video data, as shown in Fig. 2.

2) Query Received Packet: is sent from node to sink to query whether the data has been received or not, as shown in Fig. 3.

3) Acknowledgement Packet: is being sent from sink to node which acknowledges packets that have been received, as shown in Fig. 4.

4) Acknowledgement Packet: is being sent from sink to node with negative acknowledges packets that have not been received, as shown in Fig. 5.

5) Retransmission Data Packet: that is sent from node to sink which contain data that are not received in the first time, as shown in Fig. 6.

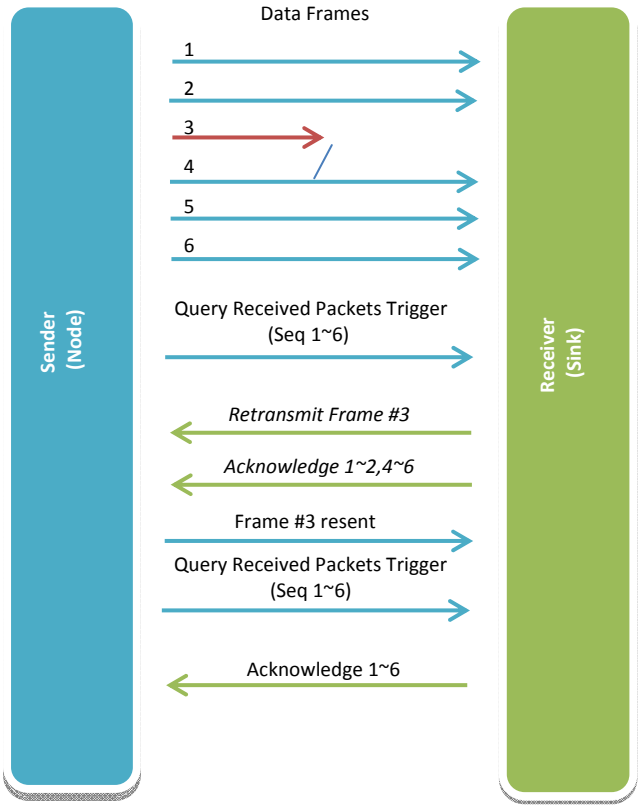


Fig. 1 RRVTP operation steps

Format	Content						
	Version	Packet Type	Session Id	Time Stamp	Sequence Number	Payload type	Data
Length(bit)	4-bits	4-bits	16-bits	32-bits	32-bits	8-bits	

Fig. 2 Data packet

Format	Content				
	Version	Packet Type	Session Id	First Sequence	Last Sequence
Length(bit)	4-bits	4-bits	16-bits	32-bits	32-bits

Fig. 3 Query received packet

Format	Content			
	Version	Packet Type	Session Id	Lists
Length(bits)	4-bits	4-bits	16-bits	

Fig. 4 Acknowledgement packet

Format	Content			
	Version	Packet Type	Session Id	Lists
Length(bits)	4-bits	4-bits	16-bits	

Fig. 5 Negative acknowledgement packet

Format	Content						
	Version	Packet Type	Session Id	Time Stamp	Sequence Number	Payload type	Data
Length(bit)	4-bits	4-bits	16-bits	32-bits	32-bits	8-bits	

Fig. 6 Retransmission data packet

B. Header Field Description

1) Version: This field identifies the version of RRVTP. The version defined by this specification is 1.

2) Packet Type: This specifies the RRVTP Message type. Table 1 shows packet types and its decimal and binary code.

TABLE I
PACKET TYPES OF RRVTP

Number of packet in decimal	Packet Type	Code
0	DataPacket	0000
1	QueryReceivedPacket	0001
2	AckPacket	0010
3	NackPacket	0011
4	RetransDataPacket	0100

3) Time Stamp: The sampling instant must be derived from a clock that increment monotonically and linearly in time to allow synchronization and jitter calculations.

4) Sequence Number: The sequence number increments by one for each RRVTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. The initial value of the sequence number should be random (unpredictable) to make known-plaintext attacks on encryption more difficult, even if the source itself does not encrypt according to the method because the packets may flow through a translator that does.

5) Session Id: This field is used to distinguish between the nodes and to overcome the NAT problem if used within the network.

6) Payload Type: This field identifies the format of the RRVTP payload and determines its interpretation by the application. A profile may specify a default static mapping of payload type codes to payload formats.

7) First Sequence and Last Sequence: They indicate the first and the last sequence number for the range that has been sent from node to sink and it did not acknowledge. It will be described in details in reliability technique (section VI).

8) Lists: Each single list, consist of two fields, the first one indicates first sequence number in list and the second one indicates the number of packets in the list. Further description with more details about reliability technique will be given in (section VI).

VI. RELIABILITY TECHNIQUE

Packets may be lost during transport due to network congestion and errors. By means of an error detection code, such as a checksum, the transport protocol may check that the data is not corrupted, and verify correct receipt by sending an ACKPacket or NACKPacket to the sender. Automatic repeat request schemes may be used to retransmit lost or corrupted data.

Reliability mechanism starts when timer timeout expires in the sender node, where it will send QueryReceivedPacket -to the Sink- to query whether packets sent have been received or not.

QueryRecievedPacket contains the first and the last sequences to packets data that are sent to sink and are not acknowledged.

In sink, when QueryReceivedPacket is received where sink begins checking packets range among the first and the last sequences that are sent by client. Then sink will send two types of packets: AckPackt which contains sequence numbers for received packets and NackPackt which contains sequence numbers for lost or corrupted packets.

The node will delete packets from temporary database that have been sent by node and received earlier by sink, then node get acknowledged through AckPacket. While the node will resend the packets that have been lost or corrupted and which have not been acknowledged by sink.

To explain the process shown in Fig. 7, it has been assumed that a node has transmitted 25 video packets that did not get acknowledged yet, as shown in Fig. 7 (a). When sink receives QueryReceivedPacket - which contain first sequence number (11) and last sequence number (35) - from node, it will check packets that has been received, as shown in Fig. 7 (b).

Packets that has been received during sending process will be arranged as lists as shown in Fig. 7 (c), as well as packets that has been lost will be arranged in lists as shown in fig. 7 (d).

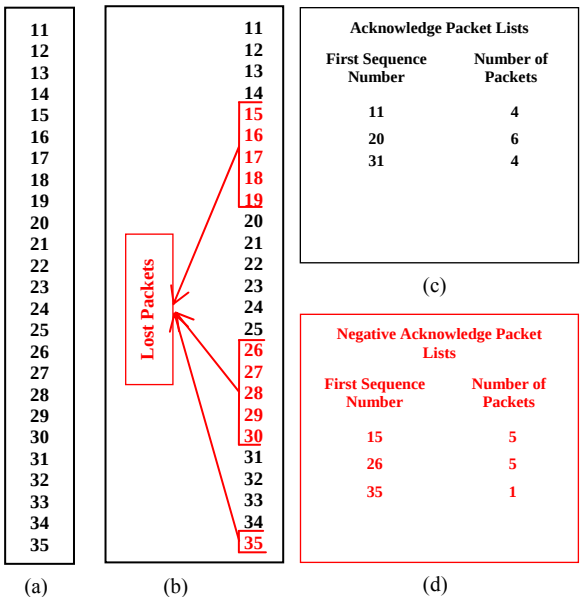


Fig. 7 Reliability mechanism example

Then, sink will send ACKPacket that contain lists for packet received, and send NACKPacket that contain lists for packet lost or corrupted.

When a node receives ACKPacket and NACKPacket, it will delete the packets that are listed in ACKPacket and will resend packets that are listed in NACKPacket.

VII. Flow and Congestion Control Techniques

Flow control is the primary reason to reduce the flow of excess traffic. While congestion control occurs when resources are scarce and highly in demand, and processing and transmission speeds lag behind the speed of incoming traffic. The two cases are not separable both can be controlled by limiting the amount of traffic entering a network: transmission rate of a sender [6].

RRVTP is designed so that both sender and receiver (node and sink) are aware of the channel capability and bandwidth variance, hence can identify if a congestion has occurred.

Discussions of two mechanisms, one at each side (sink side and node side) that fulfill this awareness is given below:

A. *Sink Retransmission Rate Control*

The sink can sense if congestion has occurred much simpler than the node. It performs a comparison on the packets sequence numbers, and it can find the number of delivery packets over timer interval (Δt) in the network. Packet Delivery Ratio (PDR) can be calculated according to:

$$PDR\% = \frac{\text{Packet received successfully with } \Delta t}{\Delta n} * 100\% \quad \ldots \quad (1)$$

Where Δn is total packet number that sent through time interval (Δt). If PDR is less than 100%, then sink will never send NACKPacket. As a result for that, node will not send lost packets because node will not receive request to retransmission of the lost packets from sink.

When congestion is found by the sink, it holds the retransmission requests of NACKPacket to another time when the channel is not congested and it will send ACKPacket only.

The sink must maintain a list of the on hold retransmission requests which will be sent when the channel is clear; this will heavy affect the real-time performance of RRVTP in a good way.

B. *Node Transmission Rate Control*

The sink replies to the node's inquiry by sending an acknowledgment (ACKPacket) of the frames that have been received successfully and negative acknowledged (NACKPacket) to the frames that are not received by the sink.

When congestion control happen in the network, sink will never send NACKPacket to the node. In this case, the node can sense congestion by performing calculations on this ACKPacket lists to find delivery packets ratio by using equation (1) and then decrease transmission packets to delivery packet ratio.

If the sender node did not receive ACKPacket after timer timeout, then the node will reduce traffic gradually until it reaches the appropriate value.

The packets that have not been sent by the node when it had reduced traffic will be considered by sink as lost packets. Sink will request retransmission all packets that have not been received. This means that the node receives lists of the received and not received packets that it queried.

VIII. Video Stream Reliability When System Failure

When a node captures a particular event it starts sending video stream to sink and when this traffic has not been delivered to sink (for the following reasons), the node will continue to capture video and store video packets to a temporary database as lost packets.:

a. *Connection cut off between sink and node.*
b. *Sink hardware or software crash.*

When the connection re-establishes between the node and the sink in the case of cut off connection, or sink restart in case sink crash, the server will request to retransmit video packets that have been captured through system failure.

IX. Performance Metrics

The protocol performance is evaluated through a number of comprehensive simulations for RRVTP. Network Simulator version 2 (NS-2) is chosen to be the platform to conduct all simulations in our work. Considering that the most widely used transport layer protocols are TCP and UDP, both TCP and UDP are compared with the RRVTP. Comparison is aimed to show whether RRVTP is capable of keeping its delay performance at an acceptable level and, at the same time, provides reliable transmissions.

These two aims lead to two main criteria of the simulations: average end-to-end delay and proportion of data packets delivered in real time. The end-to-end delay measures the average delay for a data packet when travelling from a source node to a destination node. It is the ratio of the data packets received successfully by the destination to the data packets sent by the sources.

This evaluation is made by building three scenarios each of them depends on three important parameters throughput, number of hops and number of sources. In every scenario, the end-to-end delay and packets delivery ratio with and without applying congestion control techniques is monitored.

A. Evaluation result

Simulations are conducted for three different scenarios. Scenario one measured reliability and real-time by using 5 nodes. The network simulation size is 80x80 meters, while the distance between every two nodes is about 22 meters so that range of nodes transmission is 25 meters. There is one sender to sink and there are 3 nodes routing traffic from sender node to sink as shown in Fig. 8.

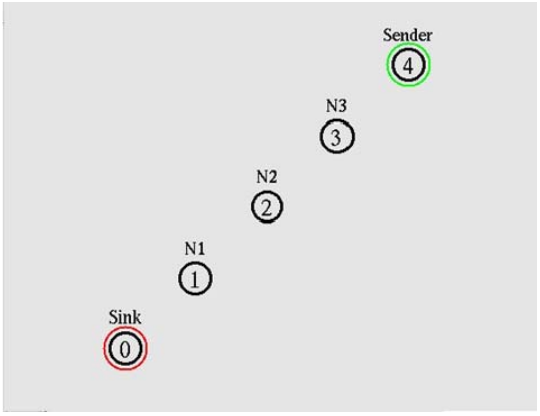


Fig. 8 Arrangement of nodes

In scenario two, reliability and real-time parameters are measured by using 9 nodes. The network simulation size is 160x160 meters, while the distance between every two nodes is about 50 meters so that range of nodes transmission is 50 meters. There are 4 senders to sink – two senders direct with sink and the other senders connect with sink through one node as shown in Fig. 9.

Scenario three differs from the two previous scenarios. This scenario will measure real-time, reliability and end-to-end delay between sender node and sink when number of hopes are increased. The network simulation size is 50x50 meters, while the distance between every two nodes is about 7.5 meters so that range of nodes transmission is 8 meters. The number of nodes used in this scenario is 41. Fig. 10 (a, b, c and d) shows the network environment.

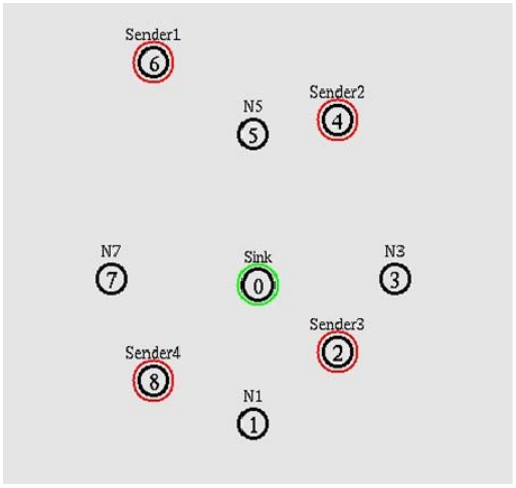
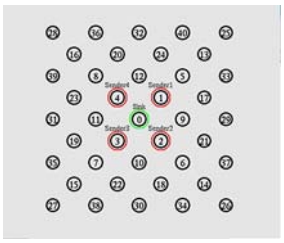
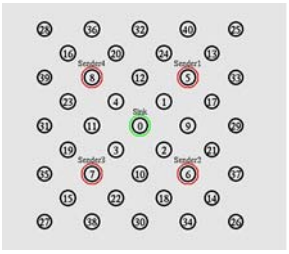


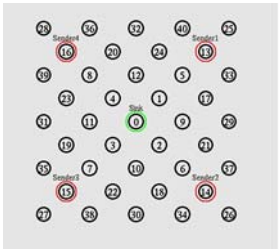
Fig. 9 Snapshot of NS2 simulation environment in scenario two



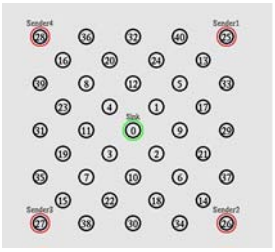
(a) Zero Hop



(b) One Hop



(c) Two Hops



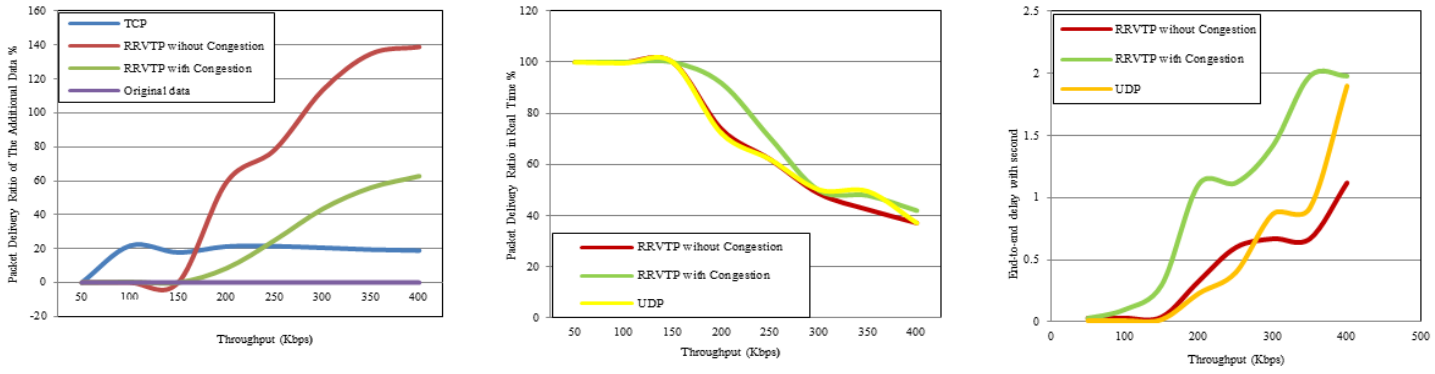
(d) Three Hops

Fig. 10 Scenario three network environment

Table II lists some basic network specifications across all simulation scenarios.

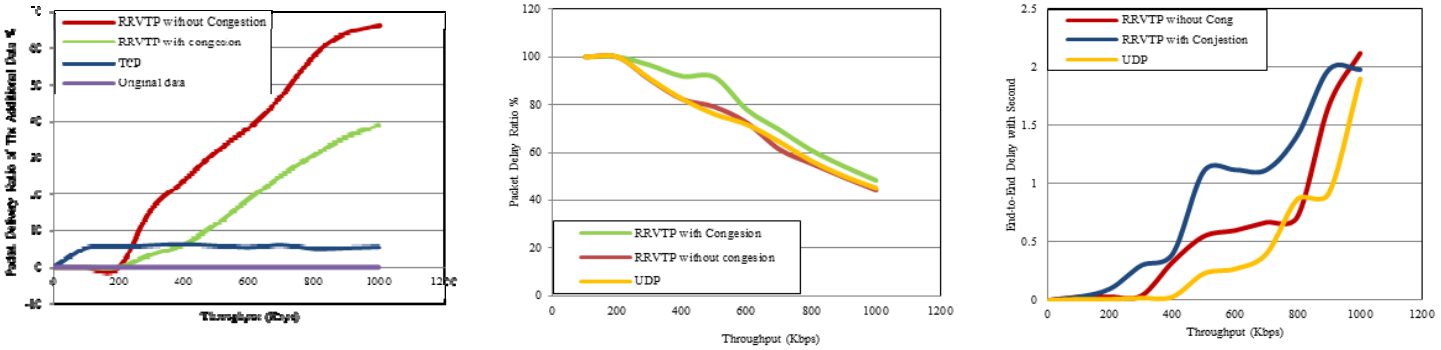
TABLE II
SIMULATION PARAMETERS

Parameters	**Values**
Routing Protocol	AODV
MAC Layer	802.11b
Traffic Type	Constant Bit Rate(CBR)
Bandwidth	1Mbps
Antenna type	Omni Directional Antenna



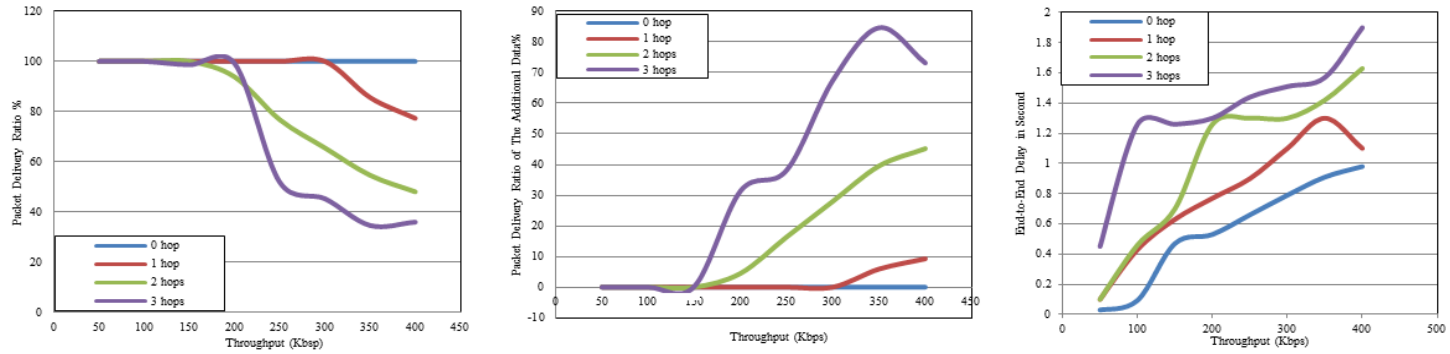
(a) (b) (c)

Fig. 11 Scenario one: one sender single path (a) (Effect of applying congestion control mechanism), (b) (Real time ratio of packets delivery), and (c) (Average end-to-end delay)



(a) (b) (c)

Fig. 12 Scenario Two: Multiple Senders Single Path (a) (Congestion control mechanisms), (b) (Real time data delivery), and (c) (Average end-to-end delay)



(a) (b) (c)

Fig. 13 Scenario Three: Multiple Senders Multi Paths Variable Hops (a) (Ratio of real time data delivery), (b) (Ratio of additional data sends from node), and (c) (End-to-end delay in second)

B. *Discussion of Results*

In this paper, comparison of proposed RRVTP protocol is made with two transport protocols (TCP and UDP) to evaluate the performance of the proposed protocol by focusing on several important parameters like reliability and real time.

The performance of the proposed RRVTP transport protocol is studied depending on three evaluation metrics: real-time packet delivery ratio, average throughput and end-to-end delay between sender node and sink.

Considering Fig. 11(a) for scenario one, it shows the difference in ratio of additional traffic to original traffic between proposed protocol RRVTP (for with and without congestion control mechanisms) and TCP, and how can this parameter effects other parameters such as reliability and real time.

Fig. 11(b) shows packet ratio that is delivered in real time comparison between RRVTP (for with and without congestion control mechanisms) and UDP. As shown in Fig. 11(b) RRVTP with congestion mechanisms get best packet delivery ratio from others protocols.

Fig. 11 (c) shows end-to-end delay among UDP and RRVTP before and after using a proposed congestion control mechanism. One can note that there is a slight difference between RRVTP after and before applying that mechanism and both with UDP.

Scenario two shows a study of the same parameters as shown in Fig. 12(a, b, c). However, it differs from scenario one in that it presents throughput change from (100-1000kbps) while in the first scenario the throughput change is from (50-400 kbps). This difference results from the difference in arranging the nodes in each of the two scenarios. It is noted from Fig. 8 and Fig. 9 that in the first scenario the packet pass through three nodes to arrive to sink while in the second scenario the packet is passed directly from sender3 and sender4 to arrive to sink or packet pass through one node from sender1 and sender2 to arrive to sink.

As a result of difference between scenario one and scenario two, scenario three has been created to discuss the different packet ratio that has been delivered in real-time, end-to-end delay and additional packets ratio that must be sent to ensure that all packets have arrived to sink, when number of nodes increases the packets must pass through them all to arrive to sink.

As shown in Fig. 13 (a), it is noted that packet ratio decreases for the packets that are delivered in real time when increasing the number of nodes between sender node and sink. The end-to-end delay increases when number of nodes increase as shown in Fig. 13 (c). As well as additional packets ratio increase when number of nodes increase between sender node and sink as shown in Fig. 13 (b).

CONCLUSIONS

The following can be concluded from this paper :

1) RRVTP do not have full reliability in real-time. When event happen and the visual sensor node begin to send the stream of video, RRVTP does not ensure complete reliable arrival of video to sink in real-time. However, with playback the video will arrive fully.

2) The use of congestion control mechanisms that proposed in this paper decreases congestion on network and increases packet ratio that arrive in real-time to sink.

3) When the number of hops between sender node and sink increases, it is noted that RRVTP's performance decreases. Hence, in network design, make sure to reduce the number of hops between sender nodes and sink in order to ensure high performance with RRTVP protocol.

REFERENCES

[1] I. Akyildiz, and M. Vuran, "Wireless Sensor Networks", John Wiley & Sons Ltd., 2010.

[2] K. Obraczka, R. Manduchi, and J. Aveces, "Managing the Information Flow in Visual Sensor Networks", The 5th International Symposium on Wireless Personal Multimedia Communications, Vol. 3, pp.1177 – 1181, October, 2002, ISSN 1347-6890.

[3] S. Misra, M. Reisslein, and G. Xue, "A Survey of Multimedia Streaming in Wireless Sensor Networks", IEEE communications surveys & tutorials, Vol. 10, No. 4, Fourth Quarter, pp. 18-39, 2008.

[4] I. Akyildiz , T. Melodia and K. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks" , ELSEVIER B.V., Computer Networks, Vol. 51, pp. 921-960, November, 2006.

[5] D. Meneses, A. Grilo, and P. Pereira, "A Transport Protocol for Real-time Streaming in Wireless Multimedia Sensor Networks", 7th EURO-NGI Conference on Next Generation Internet (NGI), June 2011.

[6] C. Omidyar and G. Pujolle, "Introduction to Flow and Congestion Control", IEEE Communications Magazine, Vol. 34, Issue: 11, pp. 1-8, November, 1996.