WCCP: A congestion control protocol for wireless multimedia communication in sensor networks

Shahin Mahdizadeh Aghdam a, Mohammad Khansari b,⇑, Hamid R. Rabiee c, Mostafa Salehi b

a School of Science and Engineering, Sharif University of Technology, International Campus, Kish, Iran
b Faculty of New Sciences and Technologies, University of Tehran, Iran
c Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

ABSTRACT
The growing interest in applications of Wireless Multimedia Sensor Networks (WMSNs) imposes new challenges on congestion control protocols in such networks. In this paper, we propose a new content-aware cross layer WMSN Congestion Control Protocol (WCCP) by considering the characteristics of multimedia content. WCCP employs a Source Congestion Avoidance Protocol (SCAP) in the source nodes, and a Receiver Congestion Control Protocol (RCCP) in the intermediate nodes. SCAP uses Group of Picture (GOP) size prediction to detect congestion in the network, and avoids congestion by adjusting the sending rate of source nodes and distribution of the departing packets from the source nodes. In addition, RCCP monitors the queue length of the intermediate nodes to detect congestion in both monitoring and event-driven traffics. Moreover, to improve the received video quality in base stations, WCCP keeps the I-frames and ignores the other less important frame types of compressed video, in the congestion situations. The proposed WCCP protocol is evaluated through simulations based on various performance metrics such as packet loss rate, frame loss rate, Peak Signal-to-Noise Ratio (PSNR), end-to-end delay, throughput, and energy consumption. The results show that WCCP significantly improves the network performance and the quality of received video in the sink nodes, and outperforms the existing state-of-the-art congestion control protocols.

1. Introduction
Due to rapid development of networked video sensors in recent years, there has been a growing demand in Wireless Multimedia Sensor Network (WMSN) applications such as multimedia surveillance, traffic monitoring, and real-time object tracking systems [1,2]. The Wireless Multimedia Sensor Networks can be described as a group of connected wireless sensors that collect multimedia data (i.e. audio and video) along with scalar data from the environment and transmit them to a base station (sink node) [3]. Achieving higher video quality in base stations is an important objective in WMSNs. The main reason for low video quality in WMSN’s base station is bursty traffic which causes congestion in the network, and consequently a large number of lost packets. Furthermore, because of the small size of sensors and hence their limited battery lives, energy conservation is an important issue in WMSNs [4,5].

There are significant number of research efforts in solving the congestion problem of sensor networks [6–9]. Based on the different congestion detection and rate adjustment techniques, one can primarily classify these works into four major categories: (I) queue assisted protocols, (II) priority aware protocols, (III) topology formation protocols, and (IV) resource control protocols. The queue assisted protocols mostly concentrate on the queue length of the nodes and use a simple rate adjustment technique such as Additive Increase Multiparticle Decrease (AIMD)

[1570-8705/$ - see front matter © 2013 Elsevier B.V. All rights reserved. http://dx.doi.org/10.1016/j.adhoc.2013.10.006]
to keep the queue length of nodes as low as possible [8,10]. However, because of the simple nature of these protocols, they do not have efficient energy consumption. The priority aware protocols consider the different priority of nodes in congestion situations and try to provide equal service for nodes in the same priority class [11–14]. The topology formation protocols adjust the input rate of congested nodes by forwarding some parts of traffic from other nodes or by activating or deactivating nodes near the congested area. However, changing network topology is not always practical, and it may result in lower performance for sparse networks [7,15–17]. The resource control protocols increase the amount of resource consumption (duty cycle) in nodes near the congested area which by itself increases the energy consumption for these nodes. It may also make the interference problem more severe in the congested areas [6,9].

While the above congestion control protocols have achieved high performance in scalar sensors (sensors which sense non-multimedia data such as temperature or humidity), they do not provide high multimedia quality (video and audio) in WMSNs. The main reason for low multimedia quality (specially video) in congestion control protocols is that they are not content aware. In other words, they treat multimedia packets similar to regular data packets, whereas in multimedia communication some packets are more important than other packets. For instance, in the case of video, packets which carry I-frames have the highest priority compared to the other frame types. Moreover, the rate adjustment techniques that are deployed in congestion control protocols, only try to adjust the output sending rate of source nodes without considering the distribution of inter-arrival packets (inter-arrival process of the packets) that can have a great impact on number of lost packets in WMSNs. Recently, several cross-layer studies are presented in the scope of designing efficient protocols for WMSNs [18,19]. However, these protocols provide different congestion control techniques without analyzing or deploying any traffic model.

In this paper we introduce a two-stage WMSN Congestion Control Protocol (WCCP) as follows.

- The Source Congestion Avoidance Protocol (SCAP) is deployed in source and is responsible for predicting congestion using a proposed Group of Picture (GOP) size prediction method. Moreover, the SCAP is responsible for adjusting the distribution of the leaving packets along with the sending rate of the source nodes using the proposed traffic model. To the best of our knowledge, it is the first time in WMSN area that a protocol adjusts the distribution of inter-arrival packets to gain a better video quality.

- The Receiver Congestion Control Protocol (RCCP) is deployed in intermediate nodes and detects congestion occurrence and informs the source nodes about the congestion. RCCP uses a proposed queuing model to detect congestion in intermediate nodes and informs the SCAP protocol in source part about the congestion.

The performance evaluation of the proposed mechanism is carried out by comparing its performance against the state of the art protocols such as XLP [6], PCCP [13], CCP [12], and other classic congestion control protocols. We show the importance of considering the contents of data in multimedia communication, and the affects of using model-based approaches in adjusting the output rate of source nodes. The key contributions of this paper are as follows:

1. A two-stage protocol (WCCP) is proposed to control congestion in WMSNs; SCAP in the source nodes to avoid congestion, and RCCP in intermediate nodes to detect and control congestion.
2. A traffic model is proposed using the inter-arrival process of the packets to adjust the sending rate of the source nodes.
3. An intermediate node’s queuing model is proposed using the MMPP queuing model and is used to detect congestion in the receiver nodes.
4. The content of data in transmission is taken into account in WCCP protocol to gain higher video quality in the base station (we preserve the I-frames which are the most important frames in the multimedia communications).
5. A GOP size prediction method is proposed to predict congestion (this method is also applicable to other works such as peer to peer networks, or wireless networks).

The rest of the paper is organized as follows. Related work is presented in Section 2. The WMSN source traffic model and intermediate queuing models are presented in Section 3. Section 4 introduces the proposed protocol. Section 5 provides performance evaluation, and the concluding remarks are presented in Section 6.

2. Literature review

Recently, several studies have been performed on developing efficient congestion control protocols for WMSNs [6,20]. A typical congestion control protocol includes three phases: congestion detection in order to detect congestion in nodes, congestion notification to inform other nodes about the congestion, and rate adjustment to mitigate the congestion problem. Based on the different congestion detection and rate adjustment techniques, we have primarily categorized the congestion control protocols in four major categories: queue assisted protocols, priority aware protocols, topology formation protocols, and resource control protocols, as shown in Table 1.

From the design point of view, we can categorize the congestion control protocols to generic or cross layer protocols. The generic congestion control protocols only use the transport layer and try to solve the problem by using the functionalities of this layer. Whereas, the cross layer congestion control protocols incorporate information and functionalities of other network layers as well.

A common approach to solve the congestion problem in sensor networks is using the queue assisted protocols. These protocols concentrate on the queue length of the
nodes, and use simple rate adjustment techniques such as end-to-end AIMD or hop-by-hop rate adjustment to keep the queue length of nodes as low as possible. STCP [10], and CODA [8] are examples of this type of protocol. However, because of the simple nature of these protocols, they do not have efficient energy consumption or high performance.

Sensor Transmission Control Protocol (STCP) [10] is a generic transport protocol which uses information of transport layer in controlling congestion. Rate adjustment in STCP is achieved by using the AIMD scheme. STCP guarantees application requirements and reduces energy consumption. However, it leads to high packet loss during congestion in the network. Congestion Detection and Avoidance (CODA) protocol [8], uses three mechanisms for congestion control: congestion detection (to detect any congestion in the network by employing the buffer length or channel occupancy), open-loop hop-by-hop backpressure (to broadcast backpressure signal to other nodes when a node detects congestion), and closed-loop multi-source regulation (it starts when the source rate is more than the maximum theoretical throughput of the channel).

The priority aware protocols are another group of congestion control protocols. They assign higher priorities to the important nodes and provide fair amount of services to the nodes in the same class. Fusion [11], CCF [12], PCCP [13], and DPCC [14] protocols are examples of these protocols. These protocols mostly concentrate on priority or fairness, and often have lower throughput.

Fusion [11] uses queue length as an index to detect congestion, and uses the prioritized Medium Access Control (MAC) to assign higher priorities to the packets of sensors with full buffer which results in lower number of lost packets in these nodes. Congestion Control and Fairness (CCF) [12] is a hop-by-hop based congestion control protocol that provides a scalable algorithm to ensure the fairness in packet delivery. However, it is limited to many-to-one topologies and the fairness mechanism leads to low throughput in WMSNs. Priority-based Congestion Control Protocol (PCCP) [13] introduces an upstream and priority-based congestion control protocol. PCCP uses three components for congestion control: Intelligent Congestion Control (ICC), Implicit Congestion Notification (ICN), and Priority-based Rate Adjustment (PRA). However, it does not have any mechanism to handle prioritized mixed traffic. Dynamic Priority Based Congestion Control (DPCC) [14] assigns a dynamic priority to each node to localize traffic of nearby nodes to the base station in highly congested WMSN. Therefore in DPCC, nodes near the sink node often have higher priority and this cause lower battery life in those nodes.

The topology formation protocols are another group of congestion control protocols which use traffic redirection techniques to solve the congestion problem. In order to alleviate the congestion problem, these types of protocols try to use alternative nodes or activate existing nodes nearby the congested area to redirect some parts of the traffic. Siphon [15], TARA [7], LACAS [16], and ESRT [17] are the protocols that can be classified in this category.

In congested conditions, Siphon [15] uses some virtual sinks. Sensor nodes forward their traffic to these virtual sinks instead of physical sink to avoid congestion near the physical sink. Siphon detects congestion based on the queue size and application fidelity of the network. Topology Aware Resource Adaptation (TARA) protocol [7] tries to alleviate congestion in the network by using a resource management technique. TARA only considers node-level congestion, and in congestion situations it activates nodes around the congested area to distribute the congested traffic through these nodes. This causes TARA to have high energy consumption. Furthermore, there is no guarantee that

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Congestion detection</th>
<th>Congestion notification</th>
<th>Rate adjustment</th>
<th>Generic or cross layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue assisted protocols</td>
<td>Queue length</td>
<td>Implicit</td>
<td>AIMD-like end-to-end</td>
<td>Generic</td>
</tr>
<tr>
<td>STCP [10]</td>
<td>Queue length and channel status</td>
<td>Explicit</td>
<td>AIMD-like end-to-end</td>
<td>Generic</td>
</tr>
<tr>
<td>CODA [8]</td>
<td>Queue length</td>
<td>Implicit</td>
<td>Exact hop-by-hop</td>
<td>Generic</td>
</tr>
<tr>
<td>Priority aware protocols</td>
<td>Queue length</td>
<td>Implicit</td>
<td>Stop-and-start hop-by-hop</td>
<td>Cross layer</td>
</tr>
<tr>
<td>CCF [12]</td>
<td>Packets arrival and service time</td>
<td>Implicit</td>
<td>Exact hop-by-hop</td>
<td>Generic</td>
</tr>
<tr>
<td>PCCP [13]</td>
<td>Packets scheduling and service rate</td>
<td>Explicit</td>
<td>Exact rate control</td>
<td>Generic</td>
</tr>
<tr>
<td>DPCC [14]</td>
<td>Queue length and application fidelity</td>
<td>–</td>
<td>Traffic redirection</td>
<td>Generic</td>
</tr>
<tr>
<td>Topology formation protocols</td>
<td>Buffer occupancy and channel load</td>
<td>Explicit</td>
<td>Traffic redirection</td>
<td>Cross layer</td>
</tr>
<tr>
<td>LACAS [16]</td>
<td>Buffer occupancy</td>
<td>Implicit</td>
<td>Duty cycle and AIMD</td>
<td>Cross layer</td>
</tr>
<tr>
<td>ESRT [17]</td>
<td>Buffer occupancy</td>
<td>–</td>
<td>Duty cycle</td>
<td>Cross layer</td>
</tr>
<tr>
<td>Resource control protocols</td>
<td>XLP [6]</td>
<td>Buffer service time</td>
<td>–</td>
<td>Cross layer</td>
</tr>
<tr>
<td>ADCC [9]</td>
<td>Packet service time</td>
<td>Explicit</td>
<td>Duty cycle</td>
<td>Cross layer</td>
</tr>
<tr>
<td>Content aware protocol</td>
<td>WCCP (The proposed protocol)</td>
<td>Number of active video sources, load of each node, and the queue blocking probability</td>
<td>Explicit</td>
<td>Exact hop-by-hop and adjusting the distribution of packets</td>
</tr>
</tbody>
</table>
the protocol can work with all topologies. Learning Automata based Congestion Avoidance Scheme (LACAS) [16] tries to make the data packets arrival rate and the data packets service rate equal in all intermediate nodes. LACAS uses the drop rate of packets in intermediate nodes as an index to detect congestion. After detection of the congestion in network, the protocol tries to use other nodes as intermediate nodes to relay some part of the congested traffic. LACAS does not consider the link level congestion, and by forwarding the traffic in multihop paths causes high energy consumption problem in the sensor nodes. Event to Sink Reliable Transport (ESRT) protocol [17] considers event-to-sink reliability instead of the traditional end-to-end reliability. ESRT tries to find the appropriate level of reliability that is required in network, and activates the minimum number of source nodes to achieve this reliability. However, considering the requirement of some applications for a guaranteed end-to-end delivery service, the ESRT protocol is not always the best solution.

The final group of the congestion control protocols are the resource control protocols. In congestion situations, the resource control protocols increase the duty cycle of the nodes located in the congested areas. However, this may make the link-level congestion problem in those areas more severe. XLP [6], and ADCC [9] protocols belong to this group of protocols.

The Cross-Layer Protocol (XLP) [6] tries to solve the congestion problem by using a cross layer solution between transport layer, network layer, and medium access layer. XLP defines two types of duty for each node: source duty and router duty. During the source duty, congestion is detected based on receiving keep alive packets instead of ACK packets for a while, and is alleviated by using AIMD rate adjustment. Nodes also as a part of their router duty will decide whether to participate in communication or not. XLP suffers from a problem that it called “communication void”. It also does not consider some important measures such as priority of data flows, fairness, content of data, and traffic behavior. The Adaptive Duty-Cycle based Congestion Control (ADCC) protocol [9] uses the service time as an index to detect congestion in nodes. Based on two constant thresholds each node will decide to whether increase the duty cycle of the node or to decrease the incoming traffic by sending congestion notification messages to its child nodes. ADCC uses constant threshold which causes high energy consumption in the network, and link-level congestions are not taken into consideration in this protocol.

Table 1 shows a brief comparison of the aforementioned congestion control protocols. Almost all of these protocols focus on the transport layer and ignore the content of data or specific traffic behaviors of WMSNs. To achieve higher video quality in WMSNs, the congestion control protocols should treat video packets differently by considering their contents. As shown in Table 1, while none of the existing rate adjustment techniques use the distribution of the departing packets from the source, the proposed protocol (WCCP) is a content aware and cross layer protocol. WCCP detects congestion by monitoring the number of active video sources, load of each node, and the queue blocking probability. Moreover, it uses exact hop-by-hop rate and the video source transmission distribution parameters to alleviate the congestion problem. The details of the WCCP protocol is described in the following section.

3. WMSN source traffic model and intermediate nodes queueing model

Modeling network characteristics leads to more accurate decision in congestion situations. In this section, we propose a source traffic model and a queueing model for intermediate nodes to adjust the transmission distribution of packets and to detect congestion in intermediate nodes, respectively.

3.1. WMSN source traffic model

When an event occurs in the network, scalar sensors detect that event and trigger video sensors to begin video communication. Therefore, we can categorize data traffic in WMSNs into two classes: video sensors’ traffic and scalar sensors’ traffic. The video sensors’ traffic is a type of traffic which is generated by the video sensors and carries video frames to the base station. The scalar sensors’ traffic is a type of traffic which is produced by the scalar sensors and contains information such as temperature and humidity. In general, either video sensors in WMSNs are programmed to start capturing and streaming video frames periodically to the sink node, or after occurrence of an event in the network (e.g. after query by users), sensors in the event area start streaming video to the sink node. Therefore, we can also categorize the WMSN traffic into two classes of monitoring traffic and event-driven traffic. Hence, in the case of monitoring traffic, video transmission is done periodically, but in the event-driven traffic, it begins after occurrence of an event in the network. Different kinds of nodes and traffics in a simple WMSN are illustrated in Fig. 1.

The video frames captured by a camera-enabled wireless sensor node (such as $N_i$) are encoded to three types of video frames (I, P and B) in the application layer [21]. The coded video stream includes a group of successive pictures (video frames) called GOP. A GOP contains one I, multiple P and multiple B-frames. Moreover, these frame types have different impact on the overall quality of decoded videos. Since P and B frames are basically predicted from I-frames, I-frames are the most important among all three frame types, and B-frames are the least important frame type in terms of quality. Therefore, I and B frame loss has the most and the least effect on video quality in multimedia applications, respectively [21]. The encoded video frames are packetized in the transport layer and are transmitted over WMSN. Since the packet size in WMSN is too small (about 100 bytes for IEEE 802.15.4 [22]), each video frame is packetized into different number of packets based on the frame type. For example, I-frames that consume the highest number of bits per frame in a GOP are fragmented into the largest number of packets compared to P and B frames.

To gain a better understanding of the WMSN source node behavior, we try to find the distribution of interdeparture times for source node packets. To achieve this
goal, we use the general Erlang distribution [23] for the transmission distribution of packet inter-departure times. The general Erlang distribution is a combination of Hyper-exponential and Erlang-k distributions. As depicted in Fig. 2, the general Erlang distribution, which best matches with the distribution of packet inter-departure times of video sources in WMSN, has three parallel levels \( (i = 1, 2, 3) \). Level \( i \) is selected with the probability \( p_i \), where \( \sum_{i=1}^{3} p_i = 1 \). Moreover, the level \( i \) contains \( k_i \) phases with exponentially distributed time and rate \( \lambda_i \). The notations used in our formulations are summarized in Table 2. The probability density function for the general Erlang distribution is given by [23]:

\[
  f_X(x) = \sum_{i=1}^{3} p_i \frac{k_i \lambda_i (k_i \lambda_i x)^{k_i-1}}{(k_i - 1)!} \cdot e^{-k_i \lambda_i x} \quad x \geq 0
\]  

\( \text{(1)} \)

In the proposed method, at the beginning of each GOP, each source node's transport layer is able to predict the parameters of the distribution for the future packets in that GOP. In other words, in a node source at the start of the GOP \( j \), when the first frame of the GOP (the I-frame) is just transmitted from the application layer to the transport layer, the source node's transport layer predicts the parameters of the Erlang distribution of packet inter-departure times for the future packets in GOP \( j \). Therefore, we may express the probability density function of the traffic model for GOP \( j \) as follows:

\[
  f_{X_j}(x) = \sum_{i=1}^{3} p_{ij} \frac{k_i \lambda_i (k_i \lambda_i x)^{k_i-1}}{(k_i - 1)!} \cdot e^{-k_i \lambda_i x} \quad x \geq 0
\]  

\( \text{(2)} \)

The level \( i \) of the general Erlang distribution is assigned to the special group of packets corresponding to their inter-departure times:

- \( i = 1 \): The first level of the general Erlang distribution is assigned to the group of packets that have the largest inter-departure time. The first packet after an event in the network or starting of monitoring period will have the largest inter-departure time.
- \( i = 2 \): The second level of the general Erlang distribution is assigned to the group of packets that have the second largest inter-departure times. These packets are the first packets in video frames (except for the first packet after an I-frame). After delivering each video frame from application layer to the transport layer, the transport layer will send the fragmented video packets. Therefore, the time between sending the last packet from the current video frame and the first packet from the next one will create a large inter-departure time. The first packet after an I-frame is an exception, because of large size of I-frames.
- \( i = 3 \): The last level in the Erlang distribution is assigned to the group of packets that have the smallest inter-departure times. All the packets that do not belong to one of the above categories are considered to be in this level. These are normally the consecutive packets from the same video frames. This level of the Erlang distribution has the highest probability of occurrence.

\( P_{M_j} \) and \( P_{E_j} \) are defined as the probability of starting a monitoring period in GOP \( j \) and probability of event happening in GOP \( j \), respectively. Each source node can keep track of the previously happened events in the network and can certainly know whether a monitoring period will start in GOP \( j \) or not. Therefore, the \( P_{M_j} \) can only be 0 or 1. Each source node can approximate the probability of

---

Table 2
Notation and model parameters.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_i )</td>
<td>Probability of selecting level ( i )</td>
</tr>
<tr>
<td>( p_{ij} )</td>
<td>Probability of selecting level ( i ) in GOP ( j )</td>
</tr>
<tr>
<td>( k_i )</td>
<td>Number of phases in the level ( i ) of the distribution</td>
</tr>
<tr>
<td>( \lambda_i )</td>
<td>Probability of an event happening in GOP ( j )</td>
</tr>
<tr>
<td>( \lambda_{ij} )</td>
<td>Probability of starting a monitoring period in GOP ( j )</td>
</tr>
<tr>
<td>ET</td>
<td>Event time</td>
</tr>
<tr>
<td>EP</td>
<td>Event period</td>
</tr>
<tr>
<td>MT</td>
<td>Monitoring time</td>
</tr>
<tr>
<td>MP</td>
<td>Monitoring period</td>
</tr>
<tr>
<td>FR</td>
<td>Frame rate of the video</td>
</tr>
<tr>
<td>NF</td>
<td>Number of frames in GOP ( j )</td>
</tr>
<tr>
<td>( NF_{ij} )</td>
<td>Number of I-frames in GOP ( j )</td>
</tr>
<tr>
<td>( NF_{ip} )</td>
<td>Number of P-frames in GOP ( j )</td>
</tr>
<tr>
<td>( NF_{ib} )</td>
<td>Number of B-frames in GOP ( j )</td>
</tr>
<tr>
<td>( NF_{ip} )</td>
<td>Predicted number of packets in GOP ( j )</td>
</tr>
<tr>
<td>( NF_{ip} )</td>
<td>Predicted number of I-packets in GOP ( j )</td>
</tr>
<tr>
<td>( NF_{ip} )</td>
<td>Predicted number of P-packets in GOP ( j )</td>
</tr>
<tr>
<td>( NF_{ib} )</td>
<td>Predicted number of B-packets in GOP ( j )</td>
</tr>
<tr>
<td>( x_k )</td>
<td>Rate parameter of the distributions of events</td>
</tr>
<tr>
<td>( t_{GOP} )</td>
<td>Time to stream a GOP</td>
</tr>
<tr>
<td>( X_i )</td>
<td>Inter-departure times of packets in level ( i )</td>
</tr>
</tbody>
</table>
event happening in GOP $j$ in terms of exponential pattern of events happening in the network and the memory less property of the exponential distribution.

$$P'_k = 1 - e^{-\lambda_k t_{cP}}$$ (3)

The source nodes can also compute the rate of events happening in network ($\lambda_k$) by keeping track of previously happened events in the network. Moreover, $t_{cP}$ is equal to $NP/FR$.

The probability of selecting each level of the general Erlang distribution for GOP $j$ ($p'_j$) is calculated by using the following equations:

$$p'_1 = \frac{(P'_k + P'_i) - (\frac{E_{(M+1)}}{M+1}) \cdot P'_k}{NP}$$ (4)

$$p'_2 = \frac{NP + NP_i}{NP} - p'_1$$ (5)

$$p'_3 = 1 - p'_1 - p'_2$$ (6)

The source node does not know the value of parameter $NP^j$ (predicted number of packets in GOP $j$) at the beginning of the GOP $j$. Therefore, each source node should predict this number. The prediction of $NP^j$ will be explained in Section 4.1.1.

Each node can keep track of the previous packets in each level of the Erlang distribution to compute the $X_i$ and $Var(X_i)$, which are the average and variance inter-departure time of packets in level $i$, respectively. $\mu_i$ and $k_i$ in Eq. (1) can be computed by using the following equations:

$$\mu_i = \frac{1}{X_i}$$ (7)

$$k_i = \left[ \frac{X_i}{Var(X_i)} \right]$$ (8)

The above source model is verified through simulations in Section 5.1.

### 3.2. Intermediate nodes queueing model

Since all sensors use low rate transceivers and have limited size of buffers, they can only handle a limited amount of load. Violating this limited load creates buffer overflow and node-level congestion in intermediate nodes. Therefore, in this section we model each node’s queue using the $\Sigma_{MMPP/D/1/S}$ queueing system [24]. We use this model to find the blocking probability and the load of each node.

As discussed in Section 3.1, the distribution of packets inter-departure time in WMSN is assumed to be general Erlang. To find the buffer overflow probability in each node with inter-arrival distribution of general Erlang, we model the arrival process as Markov Modulated Poisson Process (MMPP) [24,25].

MMPP can be used to model sources which have time-varying arrival rates. In MMPP the source node generates arrivals according to an L-state continuous-time Markov process. That means the source node has $L$ states and would spend an exponentially distributed time in each state. When the source node is in state $i$ ($i = 1,2,\ldots,L$) the arrivals are generated according to a Poisson process with the rate of $\lambda_i$. We model the WMSN source queue which generates packets with an L-level general Erlang distribution, using an L-phase MMPP. Each level in general Erlang is associated with a state in MMPP model. The transition-rate matrix ($Q$) of the MMPP model contains the rate of departing from one state to another state:

$$Q = \begin{pmatrix} -q_1 & q_{12} & \cdots & q_{1L} \\ q_{21} & -q_2 & \cdots & q_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ q_{L1} & \cdots & \cdots & -q_L \end{pmatrix}$$ (9)

where

$$q_i = \sum_{j=1, j\neq i}^{L} q_{ij}$$ (10)

The steady state vector ($\pi$) of the transition-rate matrix is given by:

$$\pi = [\pi_1, \pi_2, \ldots, \pi_L]$$

$$\pi Q = 0 \text{ and } \pi_1 + \pi_2 + \cdots + \pi_L = 1$$ (11)

The diagonal matrix $\Delta$ represents the arrival rate at each state:

$$\Delta = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_L \end{pmatrix}$$ (12)

where $\lambda_i = 1/\mu_i + \lambda_s \mu_i$ is the average time between departure of packets in level $i$ of the general Erlang distribution and $\lambda_s$ is the departure rate of scalar traffic (we assume that a source node can have both scalar and video traffic). The superposition of $n$ independent MMPP process is again an MMPP process and it’s $Q$ and $\Delta$ parameters are given by:

$$Q = Q_1 \oplus Q_2 \oplus \cdots \oplus Q_n$$

$$\Delta = \Delta_1 \oplus \Delta_2 \oplus \cdots \oplus \Delta_n$$ (13)

where the $\oplus$ operator denotes the Kronecker sum.

$e$ is the column vector of all ones, and $m$ is single message transmission time. Thus, the average arrival rate of the process ($\lambda$) and load ($\rho$) are respectively given by:

$$\lambda = \pi (\Delta \times e)$$ (14)

$$\rho = \frac{\lambda}{m}$$ (15)

The steady-state probabilities for the number of messages left by a departing message ($P_k$) is given by [24,25]:

$$P_0 = k/k\nu$$

$$P_k = (P_0B_k - \sum_{j=1}^{k-1} P^j_kA_{k-j})(I - A_1)^{-1}$$ (16)

for $k = 1,2,\ldots$

where

$$A_k = \sum_{i=k}^{\infty} A_i B_i e^{-\lambda} \quad \text{and} \quad B_k = \sum_{i=k}^{\infty} B_i e^{-\lambda}$$ (17)
The steady-state probability distribution of queue length at an arbitrary points in time \( P \) is given by:

\[
P_0 = -\frac{1}{C_0} \lambda P_0 (Q - \Delta)^{-1}
\]

\[
P_k = (P_{k-1} - \lambda (P_k_{k-1} - P_k))(Q - \Delta)^{-1}
\]

for \( k = 1, 2, \ldots \).

The blocking probability \( P \) for a queue with the length of \( K \) is equal to:

\[
P = P_k e
\]

Therefore, the loss probability of packets \( P_l \) is given by:

\[
P_l = 1 - \frac{1 - P_0 e}{\rho}
\]

The above queueing model is verified through simulations in Section 5.2.

4. WCCP: WMSN Congestion Control Protocol

WCCP is a two-part protocol. In the source-part, it uses SCAP to adjust the sending rate and distribution of the leaving packets (refer to Fig. 3). The goal of SCAP is to begin congestion avoidance from the source node. SCAP uses GOP size prediction method to predict congestion occurrence and adjusts the sending rate and the distributions of leaving packets from the source by considering WMSN traffic model. In the receiver-part\(^1\) the RCCP protocol is used to detect the congestion occurrence and to inform the source nodes about the congestion. The RCCP detects congestion using the WMSN queueing model and when it detects congestion, calculates the allowable input rate of the node and sends it as a feedback to the SCAP protocol in the source node. Thus, the SCAP protocol can adjust the output rate of the source node by using the feedback information from the RCCP protocol. In the following we elaborate on SCAP and RCCP protocols.

4.1. SCAP: a Source Congestion Avoidance Protocol

As it can be seen from Fig. 3, the SCAP protocol uses the GOP size prediction to predict the future sending rate of the source node and based on this information adjusts the sending rate and the transmission distribution of packets in the source nodes.

4.1.1. GOP size prediction

As we mentioned in Section 3, each source node should predict the number of packets at the beginning of the GOP. By the term “at the beginning of the GOP”, we mean the time when the first frame of the GOP (the I-frame) has already received at the transport layer from the application layer. Therefore, when a source node tries to predict the number of packets in the GOP \( j \), it already knows the exact number of I-packets in the GOP but it should predict the number of P and B packets in the GOP:

\[
NP^j = NP^j_I + NP^j_P + NP^j_B
\]

Fig. 3. The structure of WCCP.

The source node will predict the number of P-packets in the GOP using two components:

- The number of P-packets in previous GOPs. To predict the number of P-packets in current GOP, the source node can either use the average number of P-packets in the previous GOPs \( \bar{NP}_P^j \) or the number of P-packets in the last GOP \( NP^j_P \). Therefore, the source node can use Exponential Weighted Moving Averages (EWMA) (with parameter \( 0 < a < 1 \)) of the two values \( (1-a)NP^j_P + aNP^j_P \). \( a \) is a constant number which adjusts the weight between \( NP^j_P \) and \( NP^j_P^{-1} \).
- The number of I-packets in current GOP. In some points of the video there might be sudden increments in the size of the video frames. Therefore, the source node also uses the fraction of the number of I-packets in the current GOP to the number of I-packets in the previous GOP \( NP^j_I/NP^{j-1}_I \) to estimate the increments of the P-packets in the current GOP. In this type of prediction we implicitly assume that there is a relation between the number of I-packets and the number of P-packets in the same GOP.

The mentioned prediction method is also true for predicting the number of B-packets in GOP.

In Eqs. (22) and (23) we compute the number of P and B packets in the current GOP by using EWMA, respectively. The EWMA (with parameter \( 0 < b < 1 \)) is used to create a weighted average of the two mentioned elements.

\(^1\) The terms intermediate nodes and receiving nodes are used interchangeably from now on.
The parameters $a$ and $b$ are adaptively changed and learned from the pervious predictions in Eqs. (24) and (25) (see Appendix A).

$$a_{\text{new}} = \frac{N_{P}^{j} - N_{P}^{p}}{N_{P}^{j}/N_{P}^{p}}$$

$$b_{\text{new}} = \frac{N_{P}^{j} - ((1 - a)N_{P}^{p} + aN_{P}^{j-1})}{(N_{P}^{j}/N_{P}^{p}) \times N_{P}^{j-1} - ((1 - a)N_{P}^{p} + aN_{P}^{j-1})}$$

The above GOP size prediction method is verified through simulations in Section 5.3.

4.1.2. Adjusting the sending rate and transmission distribution of packets

In order to understand the importance of adjusting the sending rate and the transmission distribution of packets in a source video node we have performed an experiment. Fig. 4 shows the departure rates of different types of video packets from a video source node. As it can be seen, the I-packets have the highest average departure rates. This is because the I-frames are the largest frames and are fragmented into the larger number of packets. The higher sending rate of I-frames will cause higher probability of congestion (both link-level and node-level congestions) in the network, and consequently higher rate of I-packets loss (in this example almost 28% of I-frames are lost, while 17% and 14% of P and B-frames are lost, respectively). The I-frames are the most important frames in video and the higher rate of I-frames loss causes lower quality in the received video.

There are two reasons behind the high rate of frame loss (specially high rate of I-frame loss).

1. The inter-departure time between two consecutive packets in the same video frame (packets in the third level of the general Erlang, Section 3.1) are very small, and create burst traffic when the source node is sending I-packets. The burst traffic can cause congestion and high chance of packet loss and consequently high chance of video frame loss.

2. As mentioned earlier I-frames usually have bigger size and are fragmented into the larger number of packets in the transport layer. Therefore, there is higher chance of one or more packet loss in I-frames and consequently high chance of the I-frame loss.

To solve the above problems SCAP\(^2\) defines Packet Inter-Departure Time ($PIDT$) as a measure that computes the inter-departure time of two consecutive packets at the transport layer. The $PIDT$ of I-packets is lower than $PIDT$ of P or B packets. This leads to bursty traffic during sending of I-packets. To solve this problem in the proposed protocol, the video packets are sent in the distributed intervals of GOP. To this end, we send I, P and B packets in equal packet inter-departure times that result in equal $PIDT$ for all these packet types in a GOP $j$.

$$PIDT = \frac{Ps \times NF_{s}}{NoP_{s}}$$

where $OR_{s}$ is the maximum output rate of the source $s$ and $\frac{Ps \times NF_{s}}{NoP_{s}}$ shows the predicted sending rate of the source $s$.

The Source Congestion Avoidance Protocol (SCAP) is a cross-layer transport protocol that uses MAC and application layers information to distribute packet arrivals in a GOP (Fig. 5). Application layer sends the number of packets in each GOP along with the frame type to the transport layer. Moreover, MAC layer sends Maximum Transmission

\(^2\) Preliminary version of the SCAP was appeared in [20].
When a congestion notification message is received by a source node, the source node should decrease its output rate to the allowable sending rate which is informed by the congestion notification message. In the SCAP protocol, the rate adjustment is done by ignoring less valuable packets from the source nodes (say B and then P frames). In this way, we not only have prevent congestion and packet drops in the network but also achieve energy conservation in the source nodes.

During this step, the source node recalculates $PIDT^j$ for GOP $j$ with the informed allowable output rate ($OR_s$) using the following equation:

$$PIDT^j = \frac{PS}{OR_s}$$

where $PS$ is the packet size. Source $s$ also should calculate the number of packets that have to be ignored ($IP^j$) in GOP $j$ using Eq. (27).

The pseudo code of SCAP is shown in Algorithm 1.

Algorithm 1. Pseudo code of SCAP protocol

```
1: for all GOPs of source $i$ do
2: Predict number of packets in GOP $j$ by using Eqs. (21)-(23).
3: Find the packet inter-departure time in GOP $j$ ($PIDT^j$) by using Eq. (26).
4: Ignore $IP^j$ packets with lower priority in GOP $j$ by using Eq. (27).
5: Send packets in GOP $j$ with $PIDT^j$ intervals.
6: end for

1: for all nodes in the network do
2: if A congestion notification message was received then
3: Recalculate $PIDT^j$ for GOP $j$ using Eq. (30).
4: Ignore $IP^j$ packets with lower priority in GOP $j$ by using Eq. (27). (rate adjustment)
5: end if
6: Send packets in GOP $j$ using SCAP protocol.
7: end for
```

4.2. RCCP: a Receiver Congestion Control Protocol

Two types of congestion may occur in WMSN: Node-level congestion and Link-level congestion. Node-level congestion occurs when the input rate of a sensor node become more than its output rate and mostly it causes buffer overflow in the sensor nodes. Link-level congestion occur because of interference, when two nodes are in the same range and start sending simultaneously. The RCCP protocol consists of two phases to deal with any type of congestion in the network. In the congestion detection and prediction phase, congestion is detected in the network by using the intermediate nodes queueing model of the nodes. In the rate adjustment phase, the exact rates at which the source nodes are allowed to send data are determined.

```
Fig. 6. Architecture of the RCCP protocol.
```
4.2.1. Congestion detection

Nearly all the congestion control protocols have the congestion detection phase. As it can be seen from Fig. 6, RCCP protocol uses number of active video sources, the load of each node, and the blocking probability of each node’s queue as criteria to detect congestion in the network.

Packet loss in wireless data communication is hardly avoidable because of source nodes interference and link level congestion. As the number of source nodes in the network increases, the probability of link level congestion will increase. Therefore, we can use the number of active video sources as an index to detect link-level congestion.

In the RCCP protocol, congestion detection and prediction are implemented in all sensor nodes. In order to detect or predict congestion, node \( i \) compares its system load and blocking probability matrix \( (N_i = [\rho_i, Pi]) \) with a threshold matrix \( (T_i = [\rho_{Ti}, Ti]) \). In some types of networks, little congestion may be tolerable or sending high number of congestion notification messages may not be pleasant. To address this issue, a matrix for Hysteresis Margin \( (H = [\rho_m, Pm]) \) including elements with values between 0 and 1 is defined. Node \( i \) predicts or detects congestion using the following equation:

\[
\begin{align*}
\text{Congestion} & \quad \text{if } T < (N_i \circ H) \\
\text{No Congestion} & \quad \text{if } T \geq (N_i \circ H)
\end{align*}
\]  

(31)

where

\[
\rho_{Hi} = \begin{cases} 
\rho_T & \text{for } NC_i = 0 \\
\rho_T - NC_i - 1 & \text{for } NC_i = 1, 2, 3, \ldots 
\end{cases}
\]  

(32)

The \( \circ \) operator denotes the element-wise product of the matrices. \( NC_i \) is the number of active child nodes of node \( i \). To avoid link level congestions, we subtract \((NC_i - 1)\rho_s\) from the threshold load \( \rho_T \) when node \( i \) has more than one active child nodes. In this way, when we have more than one active source nodes, it is possible to activate MAC layer acknowledgment.

4.2.2. Congestion notification

After detection or prediction of any type of congestion in a sensor node, a congestion notification message is sent to its child nodes. In the proposed congestion control protocol, congestion notification message carries the exact rate value at which each node is allowed to send data. The exact rate is calculated with respect to the parent node’s allowable output rate.

As mentioned in Eq. (31), congestion is occurred because of high system load or high blocking probability. If the system load is high, regardless of blocking probability; the rate at which each child node of the node \( i \) is allowed to send data, can be calculated by using Eq. (33):

\[
CR_{ri} = \frac{\rho_{Ti} \times m}{NC_i}
\]  

(33)

where \( m \) is the single message transmission time (we assume that all child nodes of node \( i \) have the same priority). If the system load is not high but the blocking probability is high, the source node should adjust the threshold value for the system load \( \rho_T \) by using Eq. (34):

\[
\rho_T = (1 - (Pi - Pi)) \times \rho_T
\]  

(34)

By using Eq. (34), each node gradually finds its proper load threshold value. Any child node that receives congestion notification accepts the notified rate as its maximum allowable output rate. Therefore, we have applied a hop-by-hop congestion control mechanism in which no end-to-end congestion notification is sent.

Algorithm 2. Pseudo code for RCCP protocol

1: Estimate the congestion status using Eq. (31) (congestion detection)  
2: if Congestion detected then  
3: for all the child nodes do  
4: Notify it about congestion and send the child node its maximum allowable output rate using Eq. (33).  
5: Adjust the threshold value for system load Eq. (34).  
6: end for  
7: end if

Algorithm 2 shows the pseudo code for the RCCP protocol. The algorithm shows congestion detection and congestion notification phases for any node \( j \) of WMSN.

5. Performance evaluation

We analyzed the performance of the congestion control protocol in terms of packet loss, frame loss, Peak Signal-to-Noise Ratio (PSNR), throughput, delay and energy consumption. We used the NS-2 simulator [26] to evaluate the performance of the WCCP protocol. Moreover, we used the Evalvid [27] to enable video transmission simulation in NS-2.

IEEE 802.15.4 is the most common MAC and physical layer protocol for WMSN [28]. Therefore, the NS-2 simulator was setup to use IEEE 802.15.4 standard in MAC and physical layers of each node. The sending rate of MAC layer

---

**Table 3**

Simulation framework.

<table>
<thead>
<tr>
<th>Simulation parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel bandwidth</td>
<td>250 Kbps</td>
</tr>
<tr>
<td>MAC layer</td>
<td>IEEE 802.15.4</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>30</td>
</tr>
<tr>
<td>Simulation time</td>
<td>800 s</td>
</tr>
<tr>
<td>Initial power</td>
<td>100 J</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>AODV</td>
</tr>
<tr>
<td>Transmit power</td>
<td>25 mW</td>
</tr>
<tr>
<td>Receive power</td>
<td>15 mW</td>
</tr>
<tr>
<td>Idle listening power</td>
<td>13 mW</td>
</tr>
<tr>
<td>Sleeping power</td>
<td>15 µW</td>
</tr>
<tr>
<td>Transmission range</td>
<td>40 M</td>
</tr>
<tr>
<td>Packet size</td>
<td>100 Byte</td>
</tr>
<tr>
<td>Video format</td>
<td>QCIF</td>
</tr>
<tr>
<td>GOP size</td>
<td>9 frames</td>
</tr>
<tr>
<td>Number of video frames</td>
<td>2000</td>
</tr>
</tbody>
</table>
was set to 250 Kbps [28]. The packet size was 100 Bytes. The initial power of each sensor is set to 100 J. In transmitting, receiving, idle listening and sleeping states, nodes consume 25 mW, 15 mW, 13 mW and 15 μW, respectively. Simulations were done during 800 s over 30 WMSN sensors. All sensors were assumed to have 5 KB RAM memories and AODV [29] was employed as the routing protocol in our simulations.

Two types of traffic were generated in WMSN: monitoring and event-driven traffic. Monitoring time and monitoring periods were set to 5 and 100 s, respectively. Moreover, events occurred in the network by using the NS-2 exponential random generator with \( \mu = 70 \) s. In each event, the sensors that were close to the event area, streamed the video in 10 s. The video files used in simulations have 2000 frames and were picked from [30,31]. We used eight video sequences Akiyo, Bridge, Claire, Container, Highway, Mobile, Mother, and Salesman which have different characteristics in terms of motion, frame size, and quality, at QCIF resolution [28]. All video files were appended or trimmed to create 2000-frame video files.

Table 3 shows the simulation parameters used in this paper. The framework of our simulation environment is depicted in Fig. 7. First we created encoded video using the raw YUV video. FFmpeg encoder [32] was used to create encoded video files. By using mp4trace tool of Evalvid and Tcpdump [33] programs, we created sender trace files and sender dump files. These trace files were used as the simulator input and after experiencing route noises, delays and packet loss, receiver trace files are created. Receiver trace files were used to create the reconstructed mp4 video files by using the Evalvid ET program. The Evalvid FY program was also used to create damaged YUV files. Peak Signal-to-Noise Ratio (PSNR) of the original YUV file and the reconstructed YUV files were computed and compared to measure the effectiveness of the proposed protocol.

We compared the performance of the WCCP protocol with the following schemes:

- CCF [12]. CCF protocol uses simple congestion control mechanism and it uses exact rate adjustment technique.
- PCCP [13]. PCCP protocol uses priority assigning techniques to adjust each source nodes sending rate.
- XLP [6]. XLP is a complicated congestion control protocol that uses duty cycle adjustment and new routing protocol to solve the congestion problem.
- WCCP. The proposed protocol was compared by using three different Hysteresis Margin Eq. (31):
  \[ H = [0,0], \]
  \[ H = [0.1,0.1], \]
  \[ H = [0.3,0.3]. \]
- No congestion control. This is the baseline and no congestion control scheme is used. In this section we call this scheme NoCC (No Congestion Control).

5.1. Evaluation of the source nodes traffic model

To verify the source traffic model in WMSN (Section 3.1), we performed the following experiment. Fig. 8 compares the simulation data and the model results for the inter arrival time distribution of packets where \( k_1 = 1, k_2 = 46, k_3 = 2, \) and \( \mu_1 = 0.0251, \mu_2 = 13.8889, \mu_3 = 163.9344. \) The simulations were performed in three classes of configurations based on different sets of probability of selecting each level of the Erlang distribution. As is clear from Fig. 8, the proposed source traffic model follows the simulation data which verifies the accuracy of the proposed traffic model.

5.2. Evaluation of the intermediate nodes queueing model

To verify the intermediate nodes queueing model in WMSN (Section 3.2), we performed the following experiment. Fig. 9 compares the simulation data and the model results for the packet loss probability (because of buffer
overflow) in a WMSN network with three different buffer length. As it is clear from Fig. 9, the proposed queueing model follows the simulation data which verifies the accuracy of the proposed queueing model.

5.3. Evaluation of the GOP size prediction

Fig. 10 shows the predicted number of packets in GOP compared to the actual number of packets in GOP in the Highway video file with both $a$ and $b$ equal to 0.01. As we can see, our prediction is close to the actual numbers. This demonstrates the validity of Eqs. (21)–(23).

We used the KS test (Kolmogorov Smirnov test) [34] to compare the distributions of the values in actual and predicted numbers. The KS test proves the null hypothesis that predicted values and actual values are from the same continuous distribution. The value for KS test ($h$) would be 0 if the null hypothesis is correct at the 5% significance level; and 1 otherwise. The KS test returns the test statistic ($k$) which is the maximum difference between the curves. The test statistic is equal to $\max(|F_1(X) - F_2(X)|)$, where $F_1(X)$ is the proportional to $X1$ values when it is less than or equal to $X$, and $F_2(X)$ is proportional to $X2$ values when it is less than or equal to $X$ [34]. The KS test also returns the asymptotic $p$ value which reports if the numbers differ significantly or not.

The KS test results are listed in Table 4. The KS test for our experiment returned the $h$ value equal to zero which proves that the predicted numbers and the actual numbers have the same distribution. Moreover, small values of $k$ are returned from the test, which shows the accuracy of our predictions.

5.4. Number of packet loss and frame loss

Fig. 11 shows the average number of lost packets and frames for different frame types in our simulations. The $x$-axis shows the frame rate of video clips and $y$-axis shows the number of lost packets or frames. We performed the experiment in different frame rates, and we marked the number of lost packets and frames for each protocol.

Fig. 11(a) shows the average number of lost I-packets for our simulations. The higher video frame rate results in higher sending rate in source nodes, and consequently higher number of packets are lost. Therefore, in Fig. 11(a) the number of lost packets increases as the frame rate increases. We can see a rapid growth in number of lost I-packets for NoCC. Also NoCC has the highest number of lost I-packets except at frame rates between 15 and 25 frame/s. However, because of unsolicited rate regulations in CCF and PCCP protocols, they perform worse than NoCC in these frame rates.
CCF, PCCP, and XLP protocols have lower number of lost I-packets in comparison with NoCC. The XLP protocol uses a duty cycle adjustment technique in congestion situation which is the main reason of better performance for this protocol in comparison with the CCF and PCCP. Moreover, after the frame rate of 20 frame/s we can see slower growth of lost I-packets in these protocols. In the proposed protocol (WCCP) we ignored some of B and P packets in congestion situations to save I-packets. In addition, we increased the inter-departure time of two consequent I-packets in WCCP. Therefore, as we can see in Fig. 11(a) the WCCP has the lowest number of lost I-packets. The higher hysteresis margins allow the protocol to ignore higher degree of congestion in the network. Therefore, in lower hysteresis margins WCCP protocol has lower lost I-packets. Considering Fig. 11(a), on average, WCCP shows about 900 and 2000 lower packet loss compared to XLP and other protocols.

Fig. 11(d) shows the average number of lost I-frames for our simulations. Fig. 11(d) similar behavior to Fig. 11(a). The reason is that packet loss in Fig. 11(a) causes frame loss in Fig. 11(d). Again NoCC has the highest number of lost I-frames. In frame rates equal or higher than 32 frame/s, NoCC has lost approximately all the I-frames.

**Table 4**

<table>
<thead>
<tr>
<th>Video</th>
<th>p</th>
<th>k</th>
<th>Video</th>
<th>p</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akiyo</td>
<td>0.1553</td>
<td>0.2647</td>
<td>Highway</td>
<td>0.2160</td>
<td>0.0987</td>
</tr>
<tr>
<td>Bridge</td>
<td>0.00020</td>
<td>0.2242</td>
<td>Mobile</td>
<td>0.4223</td>
<td>0.2059</td>
</tr>
<tr>
<td>Claire</td>
<td>0.3035</td>
<td>0.1786</td>
<td>Mother</td>
<td>0.0853</td>
<td>0.2941</td>
</tr>
<tr>
<td>Container</td>
<td>0.0853</td>
<td>0.2941</td>
<td>Salesman</td>
<td>0.0024</td>
<td>0.3529</td>
</tr>
</tbody>
</table>

Fig. 9. Packet loss probability with respect to buffer overflow.

Fig. 10. Predicted number of packets in GOP for Highway video.
CCF, PCCP, and XLP protocols have lower number of lost I-frames in comparison with NoCC, respectively. WCCP protocol has the lowest number of lost frames among the other protocols. Especially in lower hysteresis margins, WCCP has the lowest number of lost I-frames. In frame rate of 38 frame/s, WCCP ($H = [0,0]$) lost about 100 and 40 I-frames lower than NoCC and XLP protocols, which are equal to 44% and 18% of all I-frames in the video file, respectively.

Fig. 11. Average number of packet loss and frame loss in various protocols.
Fig. 11(b) shows the average number of lost P-packets in the simulation. Similar to Fig. 11(a), because of high chance of congestion at the high sending rates, number of lost packets increases as the frame rate increases. WCCP protocol ignores P-packets in congestion situation to save I-packets. Therefore, WCCP and NoCC have the highest number of lost P-packets. CCF and PCCP protocols have lower number of P-packets, and XLP because of using duty cycle adjustment and more efficient rate adjustment technique has the lowest number of lost P-packets among the other protocols. In lower hysteresis margins, WCCP protocol has higher lost P-packets (unlike the I-packets). This is due to the fact that in lower hysteresis margins, WCCP ignores higher number of P-packets. As illustrated in Fig. 11(b) the WCCP protocol on average has about 1000 packets loss, the XLP protocol has about 800 packets loss, CCF, PCCP, and NoCC have about 960, 870, and 1110 packets loss, respectively.

Fig. 11(e) shows the average number of lost P-frames during the simulation. Similar to Fig. 11(d), because of high chance of congestion in high sending rates, number of lost frames increases as the frame rate increases. XLP protocol has the lowest number of lost P-frames and WCCP has the second lowest number of lost P-frames. As explained in Fig. 11(b), WCCP had the highest number of lost P-packets. However, in Fig. 11(e) WCCP shows lower number of lost P-frames in comparison with NoCC, CCF, and PCCP protocols. The reason is that when WCCP adjusts the sending rate, it ignores all packets of the less important frame. WCCP shows the benefit of using cross layer information to minimize the frame loss in transport layer. Therefore, behavior of CCF, PCCP and XLP protocols are similar in both Fig. 11(e) and (b) but the position of WCCP curves were changed compared to others.

Fig. 11(c) shows the average number of lost B-packets. Similar to Fig. 11(a) and (b), because of high chance of congestion at the high sending rates, the number of lost packets increases as the frame rate increases. As illustrated in Fig. 11(c), the WCCP protocol on average has about 1500 more packet loss compared to other protocols. Therefore, WCCP protocol has the largest number of lost B-packets among the other protocols. Actually, in congestion situations B-packets are the least important packets and WCCP ignores many of them to save the more important I and P packets. Similar to P-packets, in lower hysteresis margins, WCCP protocol has higher lost P-packets. Similar to Fig. 11(b), XLP protocol has the lowest number of lost B-packets.

Fig. 11(f) shows the average number of lost B-frames for our simulations. Similar to Fig. 11(d), because of high chance of congestion on the high sending rates, the number of lost frames increases as the frame rate increases, and Fig. 11(f) is similar to Fig. 11(c). XLP protocol has the lowest number of lost B-frames and WCCP has the highest number of lost B-frames. The only difference between Fig. 11(f) and (c), is that in Fig. 11(f) there is a lower difference between WCCP and NoCC. This is again because of centralizing packet loss in WCCP. In frame rate of 38 frame/s, WCCP protocol has lost about 200 B-frames more than NoCC, which is about 17% of all B-frames.

In summary, in Fig. 11(a)–(c), number of lost I-packets in WCCP are the lowest among all other protocols. However, WCCP has the largest number of lost P and B packets. This is because of adjusting the sending rate of the source node in WCCP by ignoring B-packets or sometimes P-packets, in bursty traffic. As the hysteresis margin increases, higher level of the congestion is allowed. Therefore, in higher levels of hysteresis margin we have higher number of lost I-packets but lower number of lost P and B packets. XLP acts better than PCCP and CCF protocols because of using an efficient rate adjustment technique. For Fig. 11(d)–(f), lower number of lost packets results in lower number of lost frames. Without deploying a congestion control protocol, packet loss may occur at each video frame and consequently the loss of a packet leads to dropping of the corresponding frame for that packet (frame loss).

5.5. Video quality

Clearly, lower number of I-frame loss leads to a better video quality in the receiver node. Fig. 12 shows the comparison between the WCCP, CCF, PCCP, XLP, and NoCC in terms of video quality.

Peak Signal to Noise Ratio (PSNR) was used to measure video quality. The x-axis shows the frame rate of video, and y-axis shows the quality of the received video. Because of higher number of lost frames at the high sending rates in Fig. 11(d)–(f), the video quality is decreased as the frame rate increases. NoCC has the lowest video quality. Since XLP, PCCP, and CCF protocols have lower lost frames, they show better video quality, compared to NoCC.

WCCP has the lowest number of lost I-frames, but has higher lost P and B frames. However; apparently WCCP ($H = [0, 0]$) has the highest PSNR and has improved the received video quality by about 13 dB and 6 dB in comparison with NoCC and XLP, respectively. Although WCCP ($H = [0, 0]$) has lower number of lost I-frames and higher number of lost P and B frames in comparison with WCCP ($H = [0.1, 0.1]$) and WCCP ($H = [0.3, 0.3]$), the WCCP ($H = [0, 0]$) has better video quality as illustrated in Fig. 12. This shows the importance of saving I-frames in video communication protocols, and also suggests that
content aware congestion control protocols may have a better performance in WMSNs.

Higher number of nodes in network increases the chance of congestion in the network. Fig. 13 shows the average received video quality of WCCP \((H = [0.1,0.1])\) in different network densities. We have compared WCCP \((H = [0.1,0.1])\) for different number of nodes (e.g. 15, 20, 30, 50, 90). The x-axis shows the frame rate of video, and y-axis shows the quality of the received video. In Fig. 13 WCCP has the highest received video quality when the number of nodes are equal to 15, and it has the lowest received video quality when the number of nodes are equal to 90. In particular, WCCP protocol has higher video quality in lower network densities and as the network density increases the quality of received video decreases. This was expected, since in higher number of nodes we will have the higher risk of interference or buffer overflow.

5.6. End-to-end delay

To provide real-time video transmission in WMSNs, controlling the delay of the network is crucial. Fig. 14 shows the average delay of the network for different protocols. The x-axis shows the frame rate of the video, and y-axis shows the delay of the network. Different hysteresis margins in WCCP did not make a significant differences in the End-to-End delay of the network. Therefore, only WCCP \((H = [0,0])\) was considered in the shown results.

Fig. 14 shows that NoCC has the highest delay in the network. WCCP protocol has the lowest delay until frame rate of 22 frame/s, but after frame rate of 22 frame/s, XLP has the lowest delay. This comes from the fact that WCCP protocol assumes equal delays (PIDs) between consecutive packets in GOP. In frame rate of 38 frame/s, the WCCP protocol has 45 ms more delay than XLP, but 38 ms less delay than the NoCC. As stated in description of Fig. 11, at this rate WCCP has 18% and 44% lower I-packet loss compared to XLP and NoCC, respectively. Moreover, WCCP has never passed the 400 ms delay upper bound which is necessary for real time communications.

5.7. Energy consumption and overhead

Fig. 15 shows the average number of ignored packets in the network for WCCP protocol. The x-axis shows the frame rate of the video, and y-axis shows the average number of packets that were ignored.

According to Fig. 15, B-packets are ignored more than P-packets, and as the frame rate of the network increases, more packets are being ignored. Ignoring packets leads to lower I-packet loss and lower energy consumption in the source nodes. Lower energy consumption is one of the main goals of developing a cross layer transport layer protocol for WMSNs.

Fig. 15 shows the percentage of overhead packets for WCCP protocol in various frame rates. Feedback packets correspond to congestion notification in the network. As it is clear, percentage of feedback packets are low compared to the ignored packets.

Fig. 16 shows the average energy consumption of source nodes for our simulations. The x-axis shows the frame rate of the video, and y-axis shows the average consumed energy by the source nodes in Joule. Results for different hysteresis margins of WCCP were similar. Therefore,
only WCCP \( (H = [0, 0]) \) was considered. As the frame rate of the video increases, the traffic intensity is also increases. Therefore, in Fig. 16 as the frame rate of the video increases, the average energy consumption also increases. NoCC has the highest energy consumption, and XLP, because of using an energy conservative mechanisms, has lower energy consumption in comparison with PCCP and CCF protocols. WCCP has about 2.3 J and 1.4 J lower energy consumption in comparison with NoCC and XLP protocols which are about 19% and 12.5% of overall energy, respectively. The reason for the lower energy consumption of WCCP protocol is ignoring less important packets in the congestion situations.

5.8. Network throughput

The network throughput is defined as the average successful message delivery rate over the network. Fig. 17 shows the average throughput of the network for different protocols. The x-axis shows the frame rate of the video, and y-axis shows the throughput of the network in kbps. Throughput of the network is increased as traffic of the network increases. However, congestion can cause throughput degradation in higher traffics. As illustrated in Fig. 17, until the frame rate of 28 frame/s the throughput of the network is increases as the frame rate increases, but after that rate, the throughput of the network decreases as the frame rate increases. Therefore, NoCC protocol has the highest throughput for frame rates lower than 28 frame/s, and for frame rates above 28 frame/s XLP, PCCP, and CCF protocols show higher throughput because of using rate adjustment techniques and congestion avoidance. WCCP has the lowest throughput and it is because of ignoring the B or P frames in congestion situations. On average WCCP protocol has about 7 kbps lower throughput than the other protocols. It is important to mention that the higher throughput in WMSNs does not necessarily means a better performance. In general, WMSNs require better video quality in the receiver side and lower energy consumption in the network. In fact, higher throughput in WMSNs shows higher power consumption in the sensor nodes, which is not of interest in many applications.

6. Conclusion

In this paper, a new content-aware cross layer protocol (WCCP) was proposed to minimize the packet loss in WMSNs by considering the traffic characteristics, inter-arrival pattern of packets, and video packets priority. WCCP employs intermediate nodes queueing, and source traffic model to detect and remedy the congestion. A two part scheme was introduced in WCCP. In the source part, SCAP, uses a GOP size prediction method to predict future sending rate and congestion occurrences. It, then adjusts the source traffic to avoid congestion. In the receiver part, RCCP, detects congestion in the network by using a queuing model and sends feedback to the SCAP protocol. Extensive simulations were performed to show the accuracy of the traffic model, queuing model and GOP size prediction method. We demonstrated the effectiveness of WCCP by performing simulations and comparing WCCP with four other protocols, i.e. XLP, PCCP, CCF, and NoCC (No Congestion Control). The experimental results showed good improvement in terms of video quality (6–13 dB), and energy consumption (12.5–19%). WCCP has slightly higher delay compared to XLP, but nevertheless, it does not pass the delay upper bound which is necessary for realtime applications. Furthermore, WCCP has slightly lower network throughput compared to the other protocols which is not as important as quality and energy consumption metrics in WMSNs. Simulation results also showed that content-aware congestion control protocol are preferred over other type of protocols in WMSNs, because they achieve higher video quality at the receiver side.

Appendix A

At the beginning of each GOP, a source node should calculate new a and b values by considering the pervious predictions:

\[
NP_p = \left(1 - a_{new}\right)NP_p + a_{new}NP_p^{b-1}
\]  

(A.1)

\[
a_{new} = \frac{NP_p^b - NP_p}{NP_p^{b-1} - NP_p}
\]  

(A.2)
At the beginning of GOP $j+1$ the source node has the exact number of $NP_j$. Based on this, the source node can compute the precise value of $a$, by using Eq. (A.2).

The computation of $b_{new}$ is similar to $a_{new}$, as follows:

$$NP_j = \left(1 - b_{new}\right) \left(1 - (1-a)NP_j + aNP_{j+1}\right) + b_{new} \left(NP_j/\left(NP_j^{1/2}\right) \times NP_{j+1}\right)$$  \hspace{1cm} (A.3)

$$b_{new} = \left(NP_j/\left(NP_j^{1/2}\right) \times NP_{j+1}\right) - \left(1 - (1-a)NP_j + aNP_{j+1}\right)$$  \hspace{1cm} (A.4)

References


Shahin Mahdizadeh Aghdam received his M.Sc. degree with honors in Communication and Computer Networks from Sharif University of Technology in October 2011, under supervision of Prof. Mohammad Khansari. He received his B.Sc. degree in Information Technology from Nabi Akram University in 2009. He was a Research Assistant at Digital Media Lab (DML), Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. His current research interests include wireless multimedia sensor networks, adaptive and cross-layer protocol design for wireless sensor networks, and stochastic modeling and performance optimization in wireless multimedia networks.

Mohammad Khansari received his B.Sc., M.Sc., and Ph.D. degrees in Computer Engineering from Sharif University of Technology, Tehran, Iran, in 1996, 1999, and 2008, respectively. Formerly, he has been the director, member of steering committee, and head of technical committee of the Iran Free/Open Source Software national project, and a faculty member and head of Information Technology department in Iran Telecommunication Research Center, Tehran, Iran. He is currently a faculty of New Sciences and Technologies at University of Tehran. His current research interests include network science and complex networks, wireless multimedia health sensor networks, Peer-to-Peer multimedia networks, and Free/Open Source Software.

Hamid R. Rabiee (SM’07) received his BS (1987) With Great Distinction and MS (1989) degrees in Electrical Engineering from CSULB, his EEE (1993) degree in Electrical and Computer Engineering from USC, and his Ph.D. (1996) in Electrical and Computer Engineering from Purdue University, West Lafayette, USA. From 1993 to 1996 he was a Member of Technical Staff at AT&T Bell Laboratories. From 1996 to 1999 he worked as a Senior Software Engineer at Intel Corporation. He was also with PSU, OGI and OSU universities as an adjunct professor of Electrical and Computer Engineering from 1996–2000. Since September 2000, he has joined Sharif University of Technology, Tehran, Iran. He is the founder of Sharif University Advanced Information and Communication Technology Research Center (AICT), Advanced Technologies Incubator (SATI), Digital Media Laboratory (DML), and Mobile Value Added Services Laboratory (VASL). He is currently a Professor of Computer Engineering at Sharif University of Technology, and the Director of AICT, DML, and VASL. He has been the initiator and director of international and national level projects in the context of UNDP International Open Source Network (IOSN), and Iran National ICT Development Plan. He has received numerous awards and honors for his industrial, scientific and academic contributions. He has acted as chairman in a number of national and international conferences, and holds three patents.

Mostafa Salehi received his B.Sc. (2005) degree in Computer Engineering from Iran University of Science and Technology, his M.Sc. (2008) degree in Computer Architecture from Amirkabir University of Technology, and Ph.D. (2012) degree in Computer Engineering from Sharif University of Technology. On February 2013, he joined the Faculty of New Sciences and Technologies at University of Tehran as an assistant professor. He was also involved in a number of large scale network design and consulting projects in the telecom industry. His research interests include multimedia networks, and complex networks.