A Congestion Control Framework for Handling Video Surveillance Traffics on WSN

M. Maimour
CRAN, Nancy University, CNRS
Moufida.Maimour@cran.uhp-nancy.fr

C. Pham
LIUPPA, University of Pau, France
Congduc.Pham@univ-pau.fr

D. Hoang
iNEXT, University of Technology, Sydney
dhoang@it.uts.edu.au

***Abstract*— This paper focuses on congestion control but while previous works considered scalar sensor nodes which only report events in the size of a few bytes, we are addressing congestion control for information-intensive flows such as video flows for surveillance applications in pervasive wireless multimedia sensor networks. The proposed framework that we describe in this paper tries to put several mechanisms together in order to efficiently handle information-intensive flows in a WSN. This work addresses congestion control with a multi-path routing facility. Next, an efficient congestion detection is proposed as the radio medium is most likely to introduce packet losses due to contention on the radio channel, and not only because of buffer overflow. Then a light-weight load repartition mechanism sits on top in order to take advantages to the path diversity, keeping as long as possible the sending rate constant thus keeping the video quality as high as possible. Simulations are performed in order to get insights into the performances of our proposals.**

Keywords : Congestion control, Video Sensor networks, Multipath routing

I. Introduction

The problems of transporting large amount of data in a WSN are in many aspects similar to those found in traditional networks: reliability and congestion control. This paper focuses on congestion control but while previous works considered scalar sensor nodes which only report events in the size of a few bytes, we are addressing congestion control for information-intensive flows such as video flows for surveillance applications in pervasive wireless multimedia sensor networks. In traditional sensing infrastructure, the problem of congestion control have been addressed in many works along with a transport layer protocol proposition. CODA [2], ESRT [12], RMST [11] are some of those propositions to name a few. ESRT uses the event semantic to perform congestion control: the reporting rate at the source is modulated as the network conditions vary, leading to event's reliability instead of packet's reliability. CODA uses a back-pressure mechanisms that is not very different from what have been proposed in the early age of communication networks such a Frame Relay. Congestion control and reliability for video flows are intrinsically different due to the inherently high rate of injection of multimedia packets in the network as video traffic can goes to the order of 250 kbps to 500 kbps, and because reliability by retransmission is not tractable. But the other reasons come from the video surveillance application that we are considering which leads to some design issues that are fundamentally different to the scalar case. First, the quality of the video must be kept constant, as long as possible, in order to increase the probability of identifying something. Therefore, reducing the reporting rate is not desirable, at least not as the first solution. Second, dropping packets can not be made in a blind fashion as in most video coding schemes some packets are usually more important than others.

The framework that we describe in this paper tries to put several mechanisms together in order to efficiently handle information-intensive flows in a WSN. This work addresses congestion control with a multi-path routing facility. Using multiple paths has been proposed in SAR [10] and MMSPEED [3] for QoS provisioning. However, MMSPEED focuses on reliability and none of them takes into account the fairness issues. Next, efficient congestion detection mechanisms must be implemented as the radio medium is most likely to introduce packet losses due to contention on the radio channel, and not only because of buffer overflow. Then a light-weight load repartition mechanism sits on top in order to take advantages to the path diversity, keeping as long as possible the sending rate constant thus maintaining the video quality as high as possible. We investigate various load repartition strategies where a video flow is split on multiple paths if there are some available.

The load repartition strategies vary from the simplest one which distributes uniformly the traffic on all available paths simultaneously to more complex strategies with explicit congestion notifications (CN) from congested nodes towards the sources. In these cases, on reception of a CN, a source will try to balance its traffic on available paths in order to keep its sending rate unchanged while reducing the amount of data sent on the current active paths. As a last resort, it is possible to reduce the reporting rate (decrease the number of images/s), or to drop less priority packets. This paper does not address specifically these possibilities as their implementation could trivially be realized when no more load repartition is possible. At this point, we must state that the proposed solutions does not seek to obtain the optimal load repartition on all existing paths, but rather to react as quickly as possible to congestion to avoid packet losses in very resource-constrained devices. It is our belief that a simple mechanism that limits both the number of exchanged control messages and the complexity at the sources is more suitable.

The work presented recently in [7] is the closest to our load repartition mechanism since their End-to-end Packet Scatter (EPS) split traffic on multiple paths, in an attempt to spread network load on a wider area based on the Biased Geographical Routing protocol. However, the complexity of BGR that requires location features and of the congestion control mechanism that adds an In-Network-Packet-Scatter prior to the EPS mechanism is much higher than our proposition. EPS also is much more costly in terms of control messages. We only split the flows at the source and need less control feedback messages.

Our proposed mechanisms can be used with any multi-path routing layer where explicit congestion notification, possibly implemented at a higher layer than the network layer, are available from intermediary nodes. However, in this paper, we study and propose the usage of SLiM (Simple Lifetime-based Multipath routing protocol) that was previously described in [8] and that provides multipath routing from a set of sources to a given sink with a path's lifetime criterion.

The paper is organized as follows. Section II presents the network model with the different assumptions considered in this work. The SLiM multipath routing protocol is also briefly presented for the purpose of making the paper self-reading. Section III presents how congestion can be efficiently detected in such environments. Section IV presents the various load repartition strategies for congestion control on top of SLiM. Some simulation results are then presented in section V before concluding.

II. Network Model

We consider a wireless sensor network with video sensors located in strategic locations and other non visual sensors distributed randomly in a field. A video sensor is asleep and is only waked up when alerted by other non visual sensors upon target detection or as a response to a request. In this paper we consider the case of multiple video sensors (referred to as the sources) reporting video information to the sink at relatively the same time.

Video applications are considered as semi-reliable ones where some losses are tolerated but a minimum data rate is required from the beginning of the transmission. In order to be able to satisfy this requirement, we investigate the use of multiple paths so a maximum bandwidth can be supplied. Therefore, we assume that a multipath routing protocol is available. In this paper, we use SLiM [8] but any other multipath routing protocol able to build and maintain at the same time more than one path can be used by our congestion control scheme.

SLiM, with only local topology knowledge, provides to a source and all intermediary nodes the knowledge of all available paths to the sink. It adopts the sink-initiated approach where the sink is the originator of a request. The sink floods the network with a request until the sensor, referred to as the source, having the target in its field of view is reached. With one flooding, multiple paths are built and maintained at intermediate nodes towards the sink. In SLiM, a request is identified using a path id that corresponds to the first crossed sensor's id from the sink to the source. Paths are built with respect to a quality metric specified by the application. This metric can be the path length, its available energy, an estimation of its lifetime or any other metric depending on the application requirements.

Each sensor is able to create, maintain and update a path table that records the different paths to the sink. The table contains an entry for each path with the following fields :

- *pid*, the path id,
- *inUse*, a flag, when set indicates that the corresponding path is currently in use,
- *nextNode*, the next hop towards the sink on this path,
- *quality*, an estimation of the associated quality metric for this path.

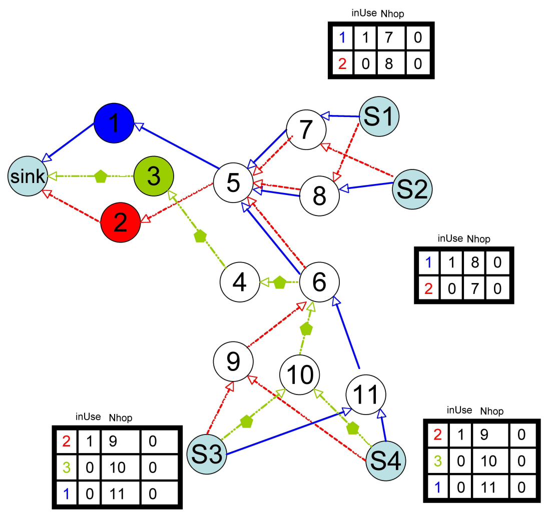

Fig. 1. Network model with multiple paths to the sink.

Figure 1 shows a configuration typically built by SLiM. This scenario shows 15 sensor nodes and 1 sink. Among the 15 sensor nodes, there are 4 sources identified as S1, S2, S3 and S4. We can see that there are 3 different paths with path id 1, 2 and 3, named after the first crossed node's id from the sink. S1 and S2 have path id 1 and 2 in their forwarding table. S3 and S4 have path id 1, 2, and 3 in their forwarding table. The right-most column keeps the rate repartition as will be explained later on. Note that SLiM avoids constructing different path id at a source with the same predecessor. This was done in order to limit congestion when no congestion control was defined on top of SLiM.

For what follows, the sink is never the bottleneck nor in term of bandwidth nor in term of energy. It can have its battery recharged or replaced in real applications. In contrast, all the sensors have limited energy, are supposed to be stationary but with the ability to dynamically vary their transmission power.

We assume that an addressing scheme is available. Globally unique addresses can be very expensive in terms of bandwidth and power consumption. Instead we consider a local addressing scheme as the one proposed in [1]. We use an address for a sensor that can be reused by an other one located sufficiently far away. We assume, however, a uniquely assigned address to the sink in order to distinguish it from the other sensors.

III. Congestion detection

One critical point in any congestion control mechanism is to efficiently detect when a real congestion occurs. As opposed to wired networks where packet losses and buffer accumulation are strong indicators of the congestion level, wireless communication may loose packets because of radio interference or channel saturation because of competing access to the radio medium. Therefore previous approaches to detect congestion only by monitoring a sensor's buffer queue [12] are neither efficient nor realistic. CODA [2] uses a channel sampling mechanism in conjunction with the report rate from the source which is supposed to be known in advance. The problem of using the report rate is that it is mostly a static end-to-end mechanism which is very difficult to use when there are traffic aggregation of several flows. In addition, the end-to-end requirement is incompatible with the low-response time that is desired for handling video flows from a surveillance system.

In this work we use a combination of the buffer queue size, noted Q, and the channel load indicator, noted Φ. A sensor queue will have 2 thresholds, S_{min} and S_{max} as in most queue size based congestion indicator systems. The channel load can be determined in a way very similar to the CODA proposal (computed with an Exponential Weighted Moving Average method within on N consecutive sampling epochs) and compared against a threshold ϕ. When Q is lower than S_{min} or between S_{min} and S_{max} there is no risk of congestion, except if $\Phi > \phi$. When $Q > S_{max}$, it means that packets are accumulating in the buffer thus leading to high congestion risks. However, in order to determine whether this behavior is transient or not, one should see how long Q remains greater than S_{max}. Therefore, it is better to not set S_{max} too close to the maximum queue capacity because reacting quickly (in order to not loose packets) on the queue size can lead to very unstable behavior. It is better to try to determine whether $Q > S_{max}$ is a persistent behavior or not. Therefore we introduce a third indicator T that will record, each time that Q becomes greater than S_{max} how many cycles this situation holds. A high congestion risk will be triggered when $Q > S_{max}$ and $\Phi > \phi$ and $T > \tau$, τ being a threshold on the number of consecutive cycles with $Q > S_{max}$. If Q decreases so that $Q < S_{max}$ before $T > \tau$ then T is reinitialized and there is no congestion notification. The value of S_{max} should not be set too high. It could be related to S_{min} and set approximately 3 times S_{min} as in the Random Early Drop system.

If only the queue size is used, then only $Q > S_{max}$ indicates a congestion. If only the channel load is used then we do not take into account the availability of local resources. By combining both indicators and adding the T parameter (which could be viewed as a severity indicator) we are able to distinguish congestion conditions more accurately.

Even though threshold-based systems could not be the optimal mechanism to reach a given desired utilization point, which is usually the case on Internet links, sensor networks do have strong energy constraints and a burst behavior that make target operating point approaches not suitable neither. However, the proposed system is subject to many possible optimizations. For instance it is possible to define several levels for ϕ and τ therefore leading to a multiple congestion level notification system. In the same line, in order to fully use the S_{min} parameter, congestion level could be somehow linked to the $S_{max} - S_{min}$ value. In this paper, we are not investigating these possible optimizations, leaving them for future works.

IV. Load repartition for congestion control

This paper moves on describing how load repartition mechanisms can be used for the purpose of congestion control when path diversity is available. It should be noted that some congestion control/detection loops may also provide high-level informations such as the available bandwidth or the maximum allowed rate [9], [6], [5]. These approaches usually need advanced feedback traffics and somehow complex computations which may not be possible in the context of WSN.

In this paper, simple notifications are used to trigger load repartition. In figure 1, after SLiM has constructed the path configuration and forwarding tables, we assume that S1 and S2 use path id 1 as the default path whereas S3 and S4 use path id 2. This information is stored in the source's forwarding table with the $inUse$ field. An additional field in the forwarding table keeps the current data rate (or an estimation if the exact data rate is not known) sent on the path. We assume that each source stores paths in the order of decreasing quality.

A. Load repartition strategies

We study 3 load repartition strategies for congestion control, from mode 1 to mode 3. For the purpose of comparison, mode 0 refers to the no load repartition scenario in which a source uses the same path (the best path in term of lifetime with SLiM) without any congestion control concerns. In mode 1 the source uses all the available paths to a sink from the beginning of the transmission. The traffic is then uniformly load-balanced on these paths. Mode 0 and mode 1 therefore represent the 2 end-points in the load repartition strategies design space.

In modes 2 and 3, explicit congestion notifications are used. At every intermediary node, when a congestion is likely to occur, according to the previously proposed congestion detection, a Congestion Notification (CN) message is sent back to the sources for each path id known by the node. A CN message contains the node id and the path id: CN(nid, pid). For simplicity, we assume that each source sends 1 data flow identified by the source id. A source S_i should react to a CN message if the path id contained in the CN message corresponds to an active path in its local forwarding table. The basic principle behind these load repartition strategies is to make each source aware of a congestion on path i and reacting to it by load-balancing the current traffic on this path on a larger number of paths. Selected paths at the source are then marked as active with the $inUse$ flag, and the data rate

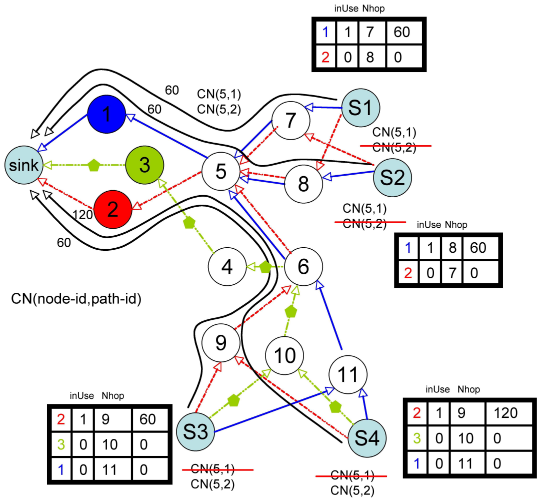

Fig. 2. Initial configuration, then congestion notification from node 5.

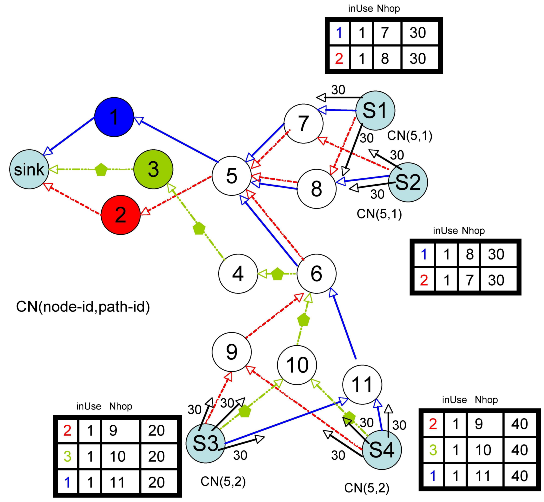

Fig. 3. After congestion notification from node 5.

repartition for each path is kept in the forwarding table. In the following paragraphs, we will describe mode 2 and 3.

- **Mode 2**. The source starts initially with one path. For each CN(nid, pid) message received, the source adds a new path (the first available path different from pid that is non active) until all available paths are marked as active. The load is uniformly distributed on the number of active path. It is therefore an incremental approach.
- **Mode 3**. The source starts initially with one path. Upon reception of a CN(nid, pid) message the source will uniformly balance the traffic **of path** pid on all available paths (including path pid in order to limit oscillations). Therefore depending on the number of CNs received for each path, the transmission rate is not the same on all the active paths as opposed to mode 2.

B. Detailed example of mode 3

We consider that each sensor can capture 120x125-pixels images with 16 gray levels coded as a bitmap image on approximately 60000 bits. Therefore a rate of 1 image/s would need 60kbps of capacity and 2 images/s would need 120kbps. We assume that all link capacities are 250kbps and for simplicity we will assume that on such 250kbps links there can be theoretically (when there is no contention on the radio medium and no significant overheads associated to headers) up to 4 1-image/s video flows.

Now, in the scenario depicted by figure 3, each source sends a video flow to the sink. Therefore, according to the forwarding tables in each source, node 5 sees 4 video flows. Assuming that flows from S1, S2 and S3 are 1-image/s flows (60kbps each) and the flow from S4 is a 2-image/s flow (120kbps), as shown in figure 3, node 5 has to relay a total data rate of 300kbps on a 250kbps link.

If we assume that such a data rate triggers 2 CN messages, CN(5,1) and CN(5,2), from node 5, sources S1 to S4 will receive them by means of the intermediate routing nodes. Upon reception of CN messages, each source will determine which CN message, if any, announces a congestion on an active path in its local forwarding table. For each active path a, S_i will load-balance its current traffic on a on all the available paths. In the scenario of figure 3, S1 and S2 will use path id 2 in addition to path id 1 on reception of CN(5,1), ignoring CN(5,2), and will send on each of these paths $60/2 = 30$kbps of data. S3 and S4 will use 3 paths on reception of CN(5,2), ignoring CN(5,1), and will respectively send $60/3 = 20$kbps and $120/3 = 40$kbps of data on each of those. Finally, we will end up with the data rate repartition shown in table I. Node 5 sees a total data rate of $30 + 30 + 30 + 30 + (20 + 40) + (20 + 40) = 240$kbps instead of 300kbps. Nodes 3, 4 and 10 (on path id 3) relay $20 + 40 = 60$kbps from sources S3 and S4 (figure 4). At this stage, with this scenario, mode 3 gives the same repartition than mode 2.

path id	S1	S2	S3	S4	total
path id 1	30	30	20	40	120
path id 2	30	30	20	40	120
path id 3			20	40	60
total	60	60	60	120	300

TABLE I

RATE REPARTITION AFTER PROCESSING CN(5,1) AND CN(5,2).

Now, in figure 4, let us continue to assume that for some reason node 2 becomes congested (not shown in the figure). At this point, node 2 is relaying $30+30+20+40 = 120$kbps from 4 flows. In this case, S1 to S4 will receive a CN(2,2) that will trigger a new rate repartition. S1 and S2 will then load-balance uniformly their current traffic on path id 2 (30 for each from previous steps) on the 2 available paths. S3 and S4 will also load-balance uniformly their current traffic on path id 2 (20 for S3 and 40 for S4 from previous steps) on the 3 available paths. We have now the repartition illustrated in table II. Node 5 that previously sent a CN message now have a total data rate of $170+50 = 230$kbps instead of 240kbps without issuing any CN message. We see that path id 3 now carries a total traffic of $20 + 20/3 + 40 + 40/3 = 80$kbps instead of 60kbps.

path id	S1	S2	S3	S4	total
path id 1	30+15	30+15	20+20/3	40+40/3	170
path id 2	15	15	20-2*20/3	40-2*40/3	50
path id 3			20+20/3	40+40/3	80
total	60	60	60	120	300

TABLE II

RATE REPARTITION AFTER PROCESSING CN(2,2).

V. SIMULATION RESULTS

The routing protocol and the different load repartition strategies were implemented with TOSSIM, the bit level simulator for TinyOS platform. We considered a square sensor field of size $1000 \times 1000m^2$ where a given number of static sensor nodes ranging from 50 to 250 with a step of 50 are randomly deployed. Each node has a maximum radio range. All the sensors have same processing capability and maximum transmission speed is 250kps. 2 separate queues are used for data packets (DP) and CN packets, with priority to the latter. In the current sets of simulation, the DP queue size is 14 packets, $S_{max} = 7$ and $S_{min} = 2$. The CN queue size is set to 4 which is not really a problem. We adopted the energy model of [4] for transmission. The energy dissipation due to processing was neglected in our simulations. The sink is located at the upper right corner (coordinates 1000,1000) and an event occurrence is simulated at the opposite quarter of the field. Every video sensor located close enough to the event will start sensing and transmitting information towards the sink. Experiments were performed and averaged over 100 simulations with different randomly generated topologies (with radio range of 400m) and initial energies at the sensor nodes which are generated following a uniform distribution between 0 and 0.4 Joules.

Figure 5 shows the mean drop rate at the sensor queues as a function of the number of sensors for the various load repartition modes. In mode 0, the different sources transmit data with a fixed rate using only one path without any congestion control. Intermediate nodes, when overloaded, drop packets and hence the number of dropped packets is the largest compared to the other modes. Mode 1 gives the best performances with a dropping rate not more than 35%. This is due to the fact that the sources distribute their flows on all available paths from the beginning hence reducing the probability of overloaded queues. The other modes appears to have similar performances but less than mode 1. This is due to the fact that a source sends data on an other path only when it receives a CN. Meanwhile some packets can be dropped. However, we see that mode 3 that tries to balance the load of a congested path on the other paths does not succeed in reducing the drop rate when compared to a simpler approach such as mode 2, at least for small network size.

We also looked at the fairness among the sources in term of transmission rate when performing congestion control. The

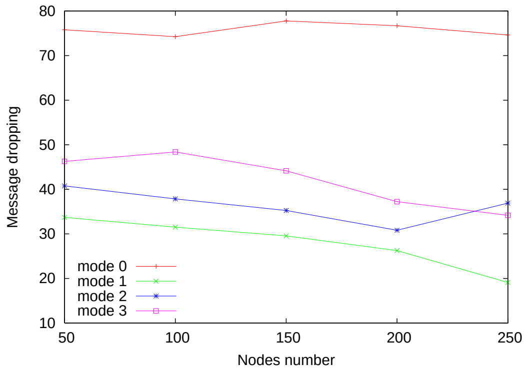

Fig. 4. Message dropping rate at sensor queues

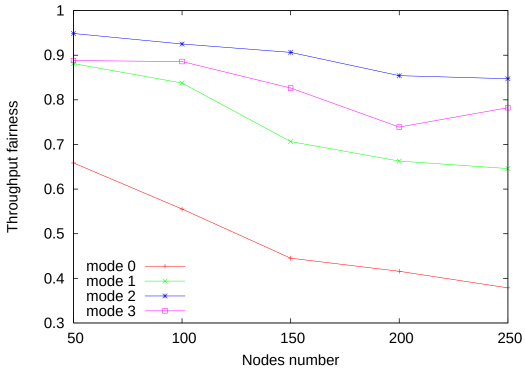

Fig. 5. Rate fairness among sources

following commonly used formula:

$$\frac{(\sum_{i=1}^{N_s} r_i)^2}{N_s \sum_{i=1}^{N_s} r_i^2} \quad (1)$$

gives a fairness metric where r_i is the success rate achieved by source i and N_s is the number of sources. Figure 6 shows the achieved fairness among the sources for the different modes as a function of the number of nodes. It appears that when using only one path per source (mode 0), fairness among sources is the worst. This is due to the fact that every source sends without any coordination since there is no congestion control. When distributing the flows on all available paths from the beginning (mode 1) without assessing the congestion situation, we eliminate any coordination between the sources and fairness among them is difficult to achieve. In modes 2 and 3, a form of implicit coordination is created among the sources since a congestion control mechanism using CN is carried out. We see that in mode 2 for example we achieve a fairness of more than 80% even for a large number of nodes.

We also evaluated the load distribution among active sensors (i.e. those taking part in data forwarding). We used the same fairness metric but replaced the transmission rate by the amount of processed data at a given node. Figure 7 draws the load fairness among active sensors for the different modes. Mode 0 achieves the best fairness since there is only one path and the load fairness is computed only for sensors belonging

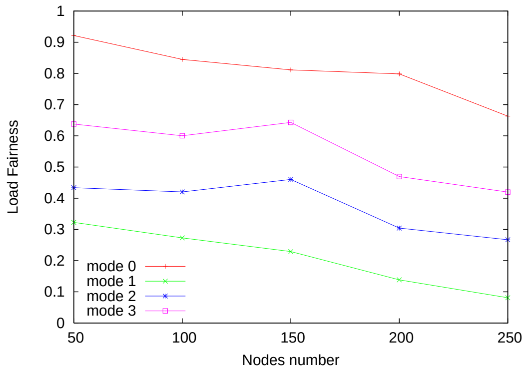

Fig. 6. Load fairness among active sensors

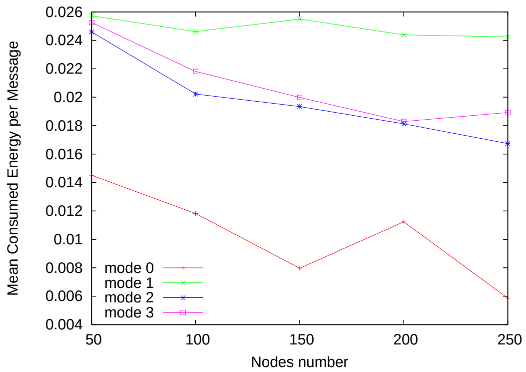

Fig. 7. Mean consumed energy per received packet

to this unique path. Distributing the flows on all the paths from the beginning (mode 1) does not appear to be interesting from a load distribution perspective. Mode 3 appears to have the best load distribution since we take into consideration the load on a per-path basis and then adjust accordingly the data rate on each active path.

Finally, we looked to assess the energy requirements of the different modes. Figure 8 shows the amount of consumed energy per correctly received packet at the sink. Naturally, mode 0 consumes less energy since, only one path is used and a small amount of data is received by the sink. Mode 1 consumes more energy per received packet than the other modes where load repartition mitigates congestion.

VI. Conclusions

In this paper, we proposed a framework for congestion control of video flows in a wireless sensor network for surveillance applications. The framework comprises a network level multipath support, an efficient congestion detection mechanism and load repartition strategies on top of the multipath support. The motivation of load repartition is to maintain the video quality unchanged by splitting a video flow on multiple paths instead of decreasing the transmission rate thus affecting the effectiveness of the surveillance application. Various load repartition strategies are therefore evaluted. The preliminary results show that load repartition does improve congestion control by reducing the packet drop probability. Regarding fairness, which is a key factor in congestion control, the preliminary results show that even simple load repartition strategies can have a very high impact on performances. However, depending on the targeted video applications on the sensor network, one may choose to prefer either rate fairness among sources or load fairness among active sensors. More importantly, it has been shown that distributing the traffic on all the available path from the beginning is not efficient in term of energy nor in term of fairness.

References

[1] M. Ali and Z. A. Uzmi. An energy-efficient node address naming scheme for wireless sensor networks. In *IEEE INCC*, 2004.

[2] S. B. E. C.-Y. Wan and A. T. Campbell. Coda: Congestion detection and avoidance in sensor networks. In *Proceedings of ACM Sensys*, 2003.

[3] E. Felemban, C.-G. Lee, E. Ekici, R. Boder, and S. Vural. Probabilistic qos guarantee in reliability and timeliness domains in wireless sensor networks. In *IEEE INFOCOM*, 2005.

[4] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of HICSS*, 2000.

[5] D. Hoang, Q. Yu, M. Li, and D. Feng. Fair intelligent congestion control resource discovery protocol on tcp based network. In *IFIP TC6/WG6.2 6th Interworking*, 2002.

[6] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwidth-delay product networks. In *ACM SIGCOMM*, 2002.

[7] I. S. L. Popa, C. Raiciu and D. Rosenblum. Reducing congestion effects in wireless networks by multipath routing. In *IEEE ICNP*, 2006.

[8] M. Maimour. Multipath routing protocol for layered video transport in wireless sensor networks. In *Workshop on Wireless Sensor Networks in conjunction with NOTERE 2007*, Marrakesh, Morocco, 2007.

[9] J. P. Sally Floyd, Mark Handley and J. Widmer. Equation-based congestion control for unicast applications. In *ACM SIGCOMM*, 2000.

[10] K. Sohrabi, J. Gao, V. Allawadhi, and G. Pottie. Protocols for self organization of a wireless sensor network. *IEEE Personal Communication*, 5(7), 2000.

[11] F. Stann and J. Heidemann. Rmst: Reliable data transport in sensor networks. In *Proceedings of IEEE SNPA*, 2003.

[12] O. B. A. Y. Sankarasubramaniam and I. F. Akyidiz. Esrt: Event-to-sink reliable transport in wireless sensor networks. In *Proceedings of ACM Mobihoc*, 2003.