

Multi-hop Buffering and Adaptation for Video-Based Sensor Networking Applications

Jie Huang, Wu-chi Feng, Wu-chang Feng
Department of Computer Science
Portland State University
Portland, Oregon 97201
{jehuang, wuchi, wuchang}@cs.pdx.edu

David Romano
Intel Corporation
77 Reed Road
Hudson, MA 01749
David.Romano@intel.com

ABSTRACT

As video sensor networks become more widely deployed, mechanisms for adaptively transmitting video data within the network are necessary because of their generally large resource requirements compared to their scalar counterparts. The key features of such networks include (i) many sources can inject video into the network that is destined for the same sink and (ii) nodes that participate in routing can also potentially work collaboratively for the benefit of the entire system. In this paper, we propose a multi-hop buffering and adaptation framework for video-based sensor networking applications. We explore several approaches in this framework and compare their performance with traditional IP-based video streaming technologies. Our experiments show that these approaches outperform traditional technologies in video quality, bandwidth waste, and bandwidth sharing fairness.

Categories and Subject Descriptors

C.2.4 [Distributed Systems]: Distributed Applications

General Terms

Algorithms, Performance

Keywords

Video sensors, Video adaptation

1 INTRODUCTION

With recent advances in hardware technologies, the construction of massively scalable video sensor networks is becoming possible. Many applications that rely on video sensor networks require *video collection*, in which the video needs to be sent to a central sink (or sinks) for later analysis and processing. Often, there is no direct network connection between a video sensor and the sink in the sensor network. As such, they typically need to rely on other nodes in the network to buffer and forward data on their behalf. For example, oceanographers at Oregon State University would like to place a video camera every $\frac{1}{4}$ mile along the Oregon coast

in order to observe near-shore phenomena. This deployment will require completely autonomous video sensors that not only harvest energy from the environment for computation and networking but also cooperate somehow in order to pass data along the coast through other nodes to more power available sinks (i.e. bucket brigade style).

Because image and video data can represent a large burden on the sensor-networking infrastructure, simply passing data toward the sink, as in techniques such as *directed diffusion* [6], may result in random dropping of video data and rapid degradation of video quality. Video adaptation techniques are necessary to deal with the mismatch between video bit rates and available bandwidth, which can eventually lead to buffer overflow within the network.

Video collection in such sensor networks cannot be addressed by existing video adaptation mechanisms meant for streaming video over the Internet or other IP-style networks. First, existing adaptation mechanisms for video typically assume end-to-end semantics between them, which is not provided in most sensor networks. Second, most of the current streaming algorithms use either a one-to-one unicast or a one-to-many multicast delivery mechanism. Finally, these mechanisms have to satisfy a real-time or “just in time” delivery requirement for video streaming and might not be suitable for video collection, in which video can sit in the network for a much longer time.

In this paper, we propose a multi-hop video buffering and adaptation framework for video collection in sensor networks. In this framework, nodes in a multi-hop route collaboratively participate in video adaptation. We propose several approaches within this framework and compare them with traditional IP-based video adaptation mechanisms through trace-driven simulations. Our experiments show that these approaches outperform traditional technologies in video quality, bandwidth waste, and bandwidth sharing fairness.

The rest of the paper is organized as follows. In Section 2, we describe work related to our project. Section 3 describes the multi-hop video buffering and adaptation framework and possible approaches to implement a system within the framework. Section 4 presents the simulation setup and experimental results. Finally, we conclude with directions for future research.

2 RELATED WORK

Video streaming across intermittent, best-effort networks has been the focus of many research projects over the last decade. In

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NOSSDAV '06 Newport, Rhode Island USA.
Copyright 2006 ACM 1-55953-285-2/06/0005...\$5.00.

particular, the adaptation mechanism on each node in our framework is similar to the window-based priority-based adaptation technique proposed in [4] and [8].

Our framework has a very similar architecture to PSFQ [12]. That is, implementing a system-level task hop-by-hop. Our framework is for video adaptation while PSFQ is for end-to-end reliability. The end-to-end reliability proposed in PSFQ, however, cannot be used for video adaptation. It has been designed for small data flows from the sink to multiple sensor nodes, where the cost of extra buffer is small and justified. PSFQ also does not consider congestion, which may be a significant source for data loss for video transmission.

Our work assumes that multi-hop routes to the sink have been already established. Routing algorithms proposed for ad-hoc wireless networks [1] and sensor networks [14][15] can be used to establish such routes. We believe that reactive routing, or on-demand routing [6], is not appropriate for video collection because of the large amount of data. Furthermore, we believe that for video transmission path selection should also consider the buffer space availability in addition to energy and link quality.

The possibility of congestion in a scalar sensor network has been addressed recently [11][13]. They do not, however, address the problem of buffer management and data adaptation.

3 DESIGN OF A MULTI-HOP BUFFERING AND ADAPTATION SYSTEM

One purpose of video adaptation is to let the application prioritize the data to send to the receiver when there is insufficient bandwidth instead of having the network randomly drop data. Obviously, random data dropping in the network will degrade the video quality rapidly because of the dependencies among compressed video data. To ensure that data chosen by the sender can actually reach the receiver and are not lost in the network, existing video adaptation technologies use end-to-end semantics, either provided by the transport layer or integrated into the adaptation technologies. However, providing end-to-end reliability in a sensor networks can be very expensive, especially over lossy wireless links. In addition, the use of TDMA-like MAC protocols to save energy increases the end-to-end latency, which greatly increases the buffer space requirement to realize end-to-end reliability.

We propose adapting video hop-by-hop in a sensor network instead of adapting video at the network edges. Unlike routers in the Internet, sensor nodes can execute application-specific tasks, including video adaptation. They can drop or downsize video data in a way that can carry out the application’s adaptation policy. Even though data are dropped in the network, they are not dropped randomly. Thus, the goal of our work is to focus on providing the best data to the application as possible while minimizing the wasted communication.

3.1 Framework Design Space

For the purposes of this paper, we assume that network setup protocols exist to construct and maintain the network topology. We also assume that data loss is caused by congestion only, i.e., links between any two nodes are reliable through link layer retransmission and the adaptation mechanisms have control over data dropping.

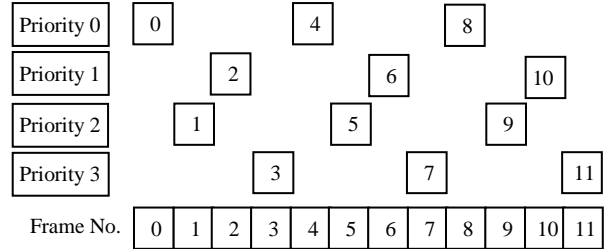


Figure 1. The basic adaptation mechanism and a simple prioritization mechanism

3.1.1 Basic adaptation mechanism

The first question in building a multi-hop adaptation mechanism is what adaptation mechanism should intermediate sensor nodes use for video adaptation? Existing video adaptation mechanisms can be classified into three categories: stream switching between streams encoded at different bit-rates and different quality parameters, transcoding that changes the bit-rate of a video stream through partial decoding and re-encoding, and selective data dropping if video is encoded in a scalable format. Stream switching is impossible because only one stream is available at an intermediate node. Transcoding is too computationally expensive especially for nodes close the sink because it has to transcode multiple streams from multiple sources and the target data rates are hard to determine. Selective dropping of data based on scalable encoding is simple and can be easily performed on intermediate nodes. Most video encodings provide at least some degree of scalability, e.g., changing the frame rate through dropping frames. More advanced scalable encoding algorithms are also available [6][9][1]. Thus, selective data dropping is a good mechanism for intermediate nodes.

Dropping for video adaptation has to differentiate subparts in a video stream and their importance and dependencies to decide what to drop. For example, in an MPEG-1 stream, a P-frame should be dropped before the I-frame it depends on; for layered encoding, an enhancement layer should be dropped before the base layer it depends on is dropped. The importance and dependencies are different for different video encodings and different application requirements. Some video adaptation technologies map importance to priorities [8]. Their adaptation mechanisms are based on the general notion of priorities and can be used for different video formats and applications. We will use priority-based data dropping in this paper because the notion of priorities separates general dropping mechanisms from application-specific prioritization mechanisms.

The dropping decision can be made for each single data item or for data items in a time window [4]. In general, the larger the time window, the better dropping decisions can be made because there is more information available and a better chance for bandwidth fluctuations to be smoothed. For a streaming application, the window size is restricted by the application’s latency requirement. For video collection, because there is typically no real-time requirement, the time window can be as large as the available buffer space allows.

In summary, we will use a priority-based buffering and adaptation mechanism on intermediate nodes as shown in Figure 1. At any time, high priority data (priority zero is the highest and priority

three the lowest) are sent before low priority data. The prioritization mechanism shown in **Figure 1** is very simple and it tries to maintain a smooth frame rate based on the assumption that all frames are independently encoded. More complicated prioritization mechanisms can be plugged in without affecting the generality of our discussion on dropping mechanisms.

3.1.2 Composition

The next question in the design of multi-hop buffering is how can sensor nodes work together as a whole to achieve the application’s adaptation goal. We propose a collaborative framework within which adaptation mechanisms on individual nodes can be composed together in meaningful ways. The framework consists of three interactive components: buffer management, prioritization, and signaling. The buffer management component allocates buffer space among various sources, monitors buffer fill levels, and so on. The prioritization component prioritizes video data from all sources. The signaling component exchanges information among neighbor nodes to help manager buffers and make adaptation decisions. In the remainder of this section, we briefly describe the basic design parameters in such systems in more detail.

Buffer management: The buffer space on a sensor node is used by all sensor nodes using it to get data to the sink. How the buffer space is shared among multiple video sources has implications on system performance since the buffer within each node in the multi-hop network provides the room for the system to adapt the video stream size to the underlying resources. There are two primary ways to manage buffers shared by multiple sources. They can either share a single buffer in a first-come-first-serve style or explicitly partition the buffer amongst the sources. Partitioning allows a sensor node to make more informed adaptation decisions since the amount of buffer space it can use is static. However, underutilization of buffer space may happen when a partition reserved for one sensor is relatively empty and cannot be used by another sensor whose partition is overflowing.

Prioritization: Prioritization needs to account for video coding dependencies within a single stream and needs to also prioritize between multiple “events” from a single sensor. For multiple video sources, prioritization also needs to consider the relative importance among cameras and how they are to share bandwidth. One of the key problems with prioritization is ensuring that such prioritization mechanisms are provided in a distributed environment.

Signaling: Signaling within a sensor system has two purposes. First, it can help make more globally optimal adaptation decisions at the expense of signaling messages. For example, although the adaptation mechanism on each node drops data from the lowest priority, data dropped by a node might have a higher priority than data that are kept on other nodes and eventually make their way to the sink. For clarity, we refer to this undesirable effect as *priority inversion* in this paper. Exchanging the highest priority level being dropped, *dropping level* for short, among sensor nodes can help a sensor node choose the right dropping level and reduce the number of priority inversions. The other purpose of signaling is to aid in congestion control. That is, to keep data away from congested nodes. ECN-like [9] mechanisms can be used to push dropping close to sources to save network bandwidth and energy. Unlike general congestion control mechanisms in the Internet, congestion control for video in the sensor network can pass

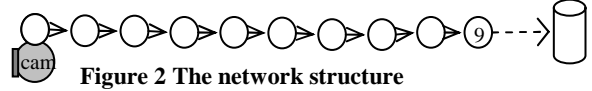


Figure 2 The network structure

specific information such as the dropping level to specify the data that can be sent to a downstream node.

3.2 Approach Descriptions

In this subsection, we propose three approaches to implement a hop-by-hop video adaptation system within our framework. We will compare these with edge-based video adaptation.

The three approaches we propose all use shared buffering. We assume all source nodes are equally important and use the same prioritization mechanisms. The main difference between them is the use of the signal in the adaptation.

Approach 1 is a simple hop-by-hop local adaptation mechanism, where the basic mechanism is applied at all nodes in the route. Each node performs adaptation independently. The collaboration among nodes is implicit through sharing the same prioritization mechanism and buffer space on each node. *Approach 2* sends messages toward the source when a buffer becomes full or becomes not full to upstream nodes. Nodes receiving the buffer full message will stop sending data to that node. This is similar to an ECN approach. *Approach 3* sends dropping levels to nodes towards the source so they will not send data that will ultimately be dropped. The dropping level for each node is determined independently.

These three approaches are basic compositions within our framework for this paper. They provide a base to understand the effects of hop-by-hop adaptation.

4 EXPERIMENTATION

To understand the implications of multi-hop buffering and adaptation for video-based sensor networking applications, we use trace driven simulation to evaluate different approaches outlined in Section 3 and compare them with adaptation at the network edge.

4.1 Simulation Setup and Metrics

For our simulations, we captured a 3,000-frame trace using a Panoptes video sensor [3]. The resolution of the video is 320x240 pixels and the average frame size is 17,282 bytes. This results in a video stream of approximately 4.14Mbps (at 30 frames per second) for each camera. **Figure 2** shows the network structure we use for most of the simulations. Because the last link to the sink is typically shared by the most sensor nodes, we assume that it is the bottleneck link. The results, however, should generalize to any network configuration where the bottleneck is between the source and the sink. We assume that each sensor has 1.5M Bytes buffer space. Each simulation runs for 100 seconds.

We compare the three approaches we propose with adaptation at the network edges, which we call *end adaptation* in this section. We have implemented a simple hop-by-hop reliability scheme, in which a video frame is kept until an acknowledgment is received from the next hop.

The goal for video adaptation is to send the most useful data to the sink with minimum waste. Therefore, the metrics we use to

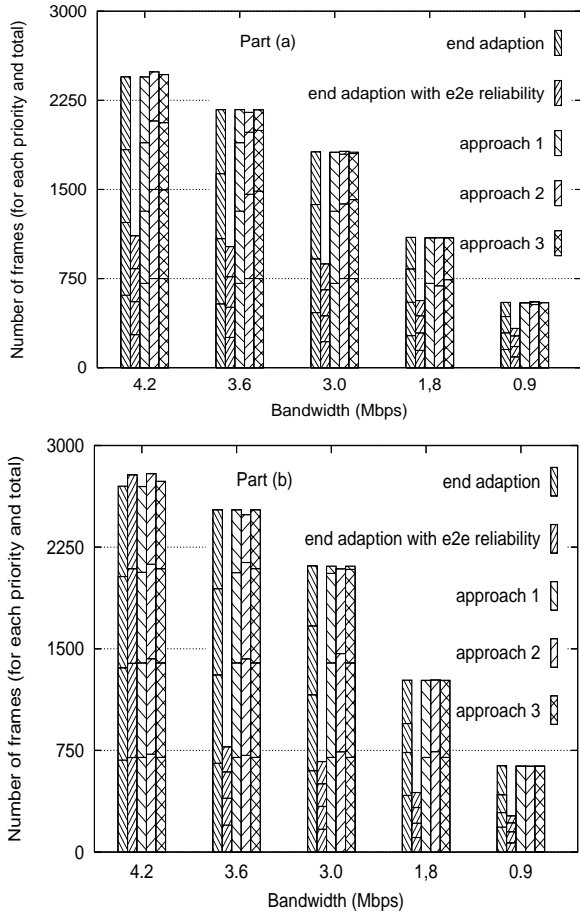


Figure 3. Priority distribution. For each approach, boxes representing numbers of frames for each priority are stacked with the highest priority at the bottom.

compare approaches are video quality and wasted bandwidth. Video quality is measured as the priority distribution of received frames and video rate smoothness. Wasted bandwidth is the number of bytes dropped after leaving their sources weighed by the distance from the sources. Signaling traffic is measured as wasted bandwidth.

In sensor networks, how networking resources are shared among multiple cameras is also important. We will use the distribution of received frames for each camera to measure bandwidth-sharing fairness.

4.2 Video Quality

In this subsection, we show the video quality delivered under different network conditions. First, we show the priority distribution of frames in the sink in **Figure 3**. In **Figure 3**, the height of a column represents the total number of frames received. There are four sub-columns in most columns and each represents the number of frames for a certain priority level, from the priority level zero at the bottom to the priority three on the top. Sub-columns on the top might be missing, indicating that frames of those (low) priority levels are all dropped.

In **Figure 3(a)**, the network is on all the time and the average bandwidth is 4.2Mbps on all links except the bottleneck link. The

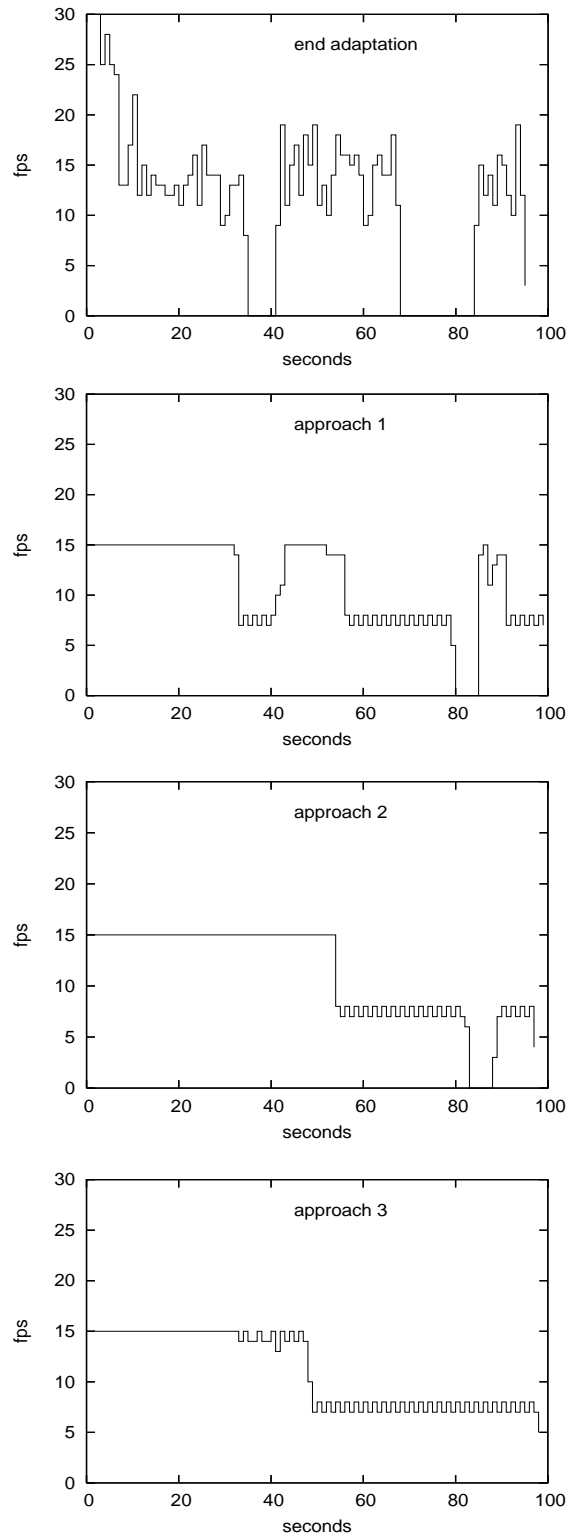


Figure 4 The frame rates

bandwidth for the bottleneck link varies as shown along the x-axis in **Figure 3(a)**. Also for the bottleneck link, there is a 6.7-second break at the 33rd second and a 16.67-second break at the 66th second in the simulation.

Figure 3 (a) shows that all three approaches we propose can get most of the important frames into the sink despite varying bandwidth. End adaptation does not work well, as expected, because it adapts to the network condition of the first hop, which is very different from that of the bottleneck link. As a result, when the bottleneck bandwidth is 0.9Mbps and allows only less than one fourth of all frames getting through, instead of sending frames with the highest priority, frames with all priority levels are sent. End adaptation combined with end-to-end reliability is even worse. Because the sent frames waiting for acknowledgement take a large portion of the buffer space, the overall throughput is halved under most conditions.

For the experiments in **Figure 3(b)**, the network is on for one second and off for one second, simulating a TDMA-like protocol. Adjacent links have opposite on/off schedules to reduce interference. The on-phase bandwidth except on the bottleneck link is 8.4Mbps so the overall bandwidth is still 4.2Mbps. The bandwidth for the bottleneck link is shown along the x-axis in **Figure 3(b)** and it has a break of 3.3 seconds at the 33rd second. The results are similar to those from **Figure 3(a)**. Together they show that our approaches are effective in different network conditions and we will show only results under the always-on network in the rest of this section. End adaptation with reliability performs fine when the bandwidth is high and degrades rapidly when the bandwidth decreases as the need for buffer space to hold unacknowledged frames grows fast. We will exclude this case in the rest of this section because of the low throughput.

Figure 4 show the frame rates of received frames over time. The frame rates are calculated based on the capturing timestamps, not on arriving time because there is no real-time requirement. The network condition is the same as that in **Figure 3(a)** and the bottleneck bandwidth is 1.5Mbps.

As shown in **Figure 4**, the frame rate for end adaptation varies significantly and when there are breaks, the frame rate drops to zero. The frame rate for *approach 1* is much smoother but still responsive to bandwidth breaks. *Approach 2* smoothes the frame rate over the first break, with the help of congestion indication, but the frame rate drops to zero during the second break, which is longer than the first one. This is due ECN message stopping the transmission of data. *Approach 3* survives both breaks because it allows high priority data to get through even though a buffer is full. This shows that exchanging application-specific information, the dropping level in this case, can help with adaptation.

It is worth mentioning that the advantage of our approaches is more significant if there are dependencies among frames such as in MPEG. A dropped high priority frame can cause many low priority frames un-decodable thus the numbers of usable frames for the end adaptation case are lower than those shown in **Figure 3**.

4.3 Wasted Bandwidth

In this subsection, we compare wasted bandwidth. In sensor networks, energy is a precious resource and wireless networking is the major consumer of energy. Thus, it is very important to reduce bandwidth wastage. There are two sources for bandwidth

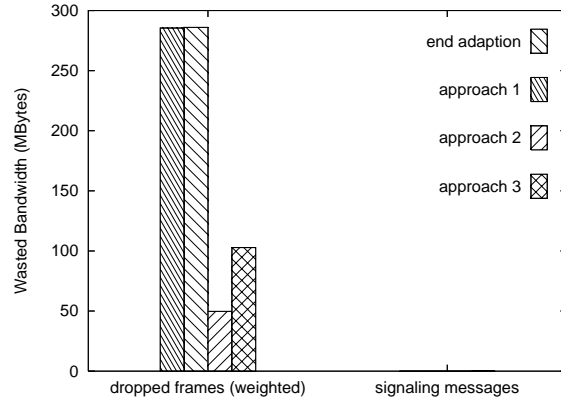


Figure 5 Wasted bandwidth

wastage (i) data dropped between the source and the sink and (ii) messaging overhead.

Figure 5 shows the wasted bandwidth when the network condition is the same as that in **Figure 3(a)** and the bottleneck bandwidth is 1.5Mbps. The dropped data are weighted by the distance (in hops) from its source. *Approach 2* and *approach 3* greatly reduce the amount of data dropped in the network, 82.7% and 67.1%, respectively. The price they pay is negligible: 7620 and 8850 signaling messages. Assuming 20 bytes per signaling message, the wasted bandwidth is negligible. **Figure 5** clearly shows the benefit of explicit signaling among sensor nodes.

4.4 Fairness

To compare the bandwidth sharing fairness, we use two network structures. One is the line structure shown in **Figure 2** with a camera attached to each sensor node. The other is shown in **Figure 6**. The bottleneck link in this structure is also the last link to the sink. All links have 10.5Mbps bandwidth and the bottleneck links have two breaks the same as those in **Figure 3(a)**. Both structures have ten cameras.

Figure 7 shows the numbers of received frames for each camera. End adaptation cannot do system-wide prioritization so nodes closer to the sink get more bandwidth than nodes farther away because they take more buffer space on *node 9*, which is the last node to the sink. Prioritization can offset some of the bias because low priority data from closer nodes on *node 9* are dropped to make room for high priority data from nodes farther from the sink. In *Approach 1*, cameras share the bandwidth pretty fairly. *Approach 2* stops sending when a buffer is full thus high priority data from farther nodes cannot get into *node 9* when it is full. *Approach 3* fixes this by allowing high priority data to be sent to a full buffer, forcing low priority data in the full buffer to be dropped. *Approach 3* does not send to a full buffer aggressively thus the bandwidth sharing is not as fair as in *Approach 1*. However, this can be easily changed by tuning parameters such as the signaling message frequency and threshold to update a dropping level.

5 CONCLUSION

In this paper, we propose a multi-hop buffering and adaptation framework for video-based sensor networking applications. We have shown that adapting video in the network is more effective

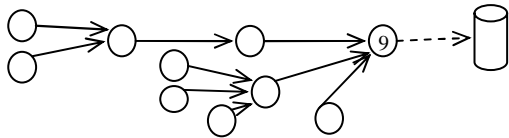


Figure 6 The tree network structure

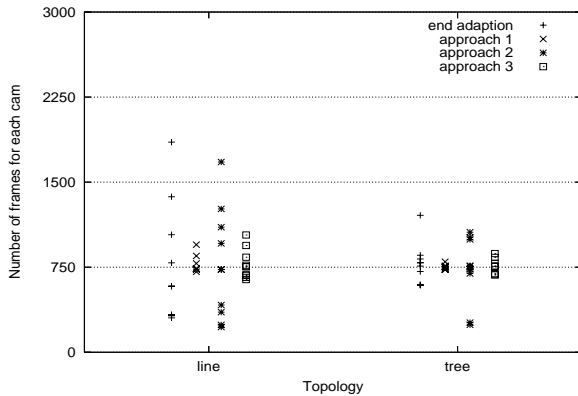


Figure 7 Fairness. Received frames for each camera.

in collecting high quality video than adapting video at the network edges. We also show that explicitly sharing information among sensor nodes can achieve smoother frame rates and reduce bandwidth wastage. Sharing of application-information among nodes can also maintain fair sharing of bandwidth.

Future research includes effects of different approaches in real sensor networks. The parameters we choose for *approach 2* and *approach 3*, such as the frequency of signaling messages and thresholds for buffer status changes, are heuristic. Systematic study of tuning these parameters can improve their performance further. More complicated buffer management and signaling protocols can be designed to further exploit the benefit of multi-hop buffering and adaptation.

6 ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. RR-0423728. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

7 REFERENCES

[1] Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens and J. Cornelis, "Fully- scalable wavelet video coding using in-band motion compensated temporal filtering," *Proc. IEEE International Conf. on Acoustics Speech and Signal Processing, ICASSP'03*, Hong Kong, CN, vol. 3, pp. 417-420, March 2003.

[2] Benjie Chen, Kyle Jamieson, Hari, Balakrishnan, Robert Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *MOBICOM 2001*, Rome, Italy, July 2001.

[3] Wu-chi Feng, Brian Code, Ed Kaiser, Mike Shea, Wu-chang Feng, Louis Bavoil, "Panoptes: A Scalable Architecture for Video Sensor Networking Applications", in *Proceedings of the ACM Multimedia 2003*, Nov. 2003, pp. 562-571.

[4] W. Feng, M. Liu, B. Krishnaswami, and A. Prabhudev, "A Priority-Based Technique for the Best-Effort Delivery of Stored Video," In *Proceedings of SPIE/IS&T Multimedia Computing and Networking*, San Jose, January 1999.

[5] Deepak Ganesan, Ramesh Govindan, Scott Shenker and Deborah Estrin, "Highly Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks", In *Mobile Computing and Communications Review (MC2R)* Vol 1., No. 2. 2002.

[6] Barry G. Haskell, Atul Puri, and Arun N. Netravali. "Digital Video: An Introduction to MPEG-2". Chapman & Hall. ISBN 0-412-08411-2.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in *Proc. of the Sixth Annual MOBICOM*, August 2000.

[8] C. Krasic, J. Walpole, and W. Feng, "Quality-Adaptive Media Streaming by Priority Drop," In *Proceedings of NOSSDAV 2003*, Monterey, California, June 2003.

[9] Weiping Li. "Overview of fine granularity scalability in MPEG-4 video standard". *IEEE Transactions on Circuits and Systems for Video Technology*, Volume 11, Issue: 3. March 2001

[10] Ramakrishnan, K.K. and Folyd, S. "A Proposal to add Explicit Congestion Notification (ECN) to IPv6 and to TC". Internet draft draft-kksjf-ecn-oo.txt, work in progress, November 1997.

[11] Sankarasubramaniam, Y., Akan O.B., and Akyildiz I.F. "ESRT: Event-to-Sink Reliable Transport for Wireless Sensor Networks," *Proc. ACM MOBIHOC 2003*, June 2003, pp.177-188.

[12] C. Wan, A. Campbell, and L. Krishnamurthy. "PSFQ: A reliable transport protocol for wireless sensor networks," *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pages 1--11, 2002.

[13] Wan, C.-Y., Eisenman S.B., and Campbell A.T, "CODA: Congestion Detection and Avoidance in Sensor Networks," *Proc. ACM SENSYS 2003*, Los Angeles, CA, November 2003.

[14] Alec Woo, Terence Tong, and David Culler. "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks". *SneSys '03*, Nov. 5-7, 2003, Los Angeles, CA.

[15] Ya Xu, John Heidemann, and Deborah Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," *MOBICOM 2001*, Rome, Italy, July 2001.