MeshVision: An Adaptive Wireless Mesh Network Video Surveillance System

Peizhao Hu[12], Ryan Wishart[2], Jimmy Ti[2], Marius Portmann[12], and Jadwiga Indulska[12]

[1] The University of Queensland,
School of Information Technology and Electrical Engineering
{marius, jaga}@itee.uq.edu.au
[2] National ICT Australia (NICTA)
{Peizhao.Hu, Ryan.Wishart, Jimmy.Ti}@nicta.com.au

Abstract. The major surveillance camera manufacturers have begun incorporating wireless networking functionality into their products to enable wireless access. However, the video feeds from such cameras can only be accessed within the transmission range of the cameras. These cameras must be connected to backbone infrastructure in order to access them from more than one hop away. This network infrastructure is both time-consuming and expensive to install, making it impractical in many rapid deployment situations (for example to provide temporary surveillance at a crime scene). To overcome this problem, we propose the MeshVision system that incorporates wireless mesh network functionality directly into the cameras. Video streams can be pulled from any camera within a network of MeshVision cameras, irrespective of how many hops away that camera is. To manage the trade-off between video stream quality and the number of video streams that could be concurrently accessed over the network, MeshVision uses a Bandwidth Adaptation Mechanism. This mechanism monitors the wireless network looking for drops in link quality or signs of congestion and adjusts the quality of existing video streams in order to reduce that congestion. A significant benefit of the approach is that it is low cost, requiring only a software upgrade of the cameras.

1 Introduction

The global market for digital surveillance cameras has increased drastically in recent years. This market has been driven by the widespread adoption of digital surveillance technologies by business owners as well as city councils seeking to reduce crime. The scale of this adoption is shown in London, England where it has been estimated that there are over 500,000 video cameras installed [5].

To simplify future camera deployments, as well as ongoing maintenance and video stream access, the major surveillance camera manufacturers have begun incorporating 802.11g wireless networking functionality into their camera product lines.

At present the feeds from such wireless-enabled cameras can only be wirelessly accessed from one hop away. To view the video feed outside of the camera's wireless transmission range requires the camera and the viewer to be connected via backbone infrastructure. This infrastructure is time consuming and expensive to deploy. Consequently, it is not practical to quickly deploy a temporary surveillance network at a crime scene or public event.

When working with wireless networks, a significant problem arises: as the wireless medium is shared it is challenging to estimate the residual bandwidth in the network at any time. This makes it difficult to provide Quality of Service by reserving bandwidth for particular flows (such as a video stream). The wireless medium is also subject to high loss rates which necessitates the use of channel coding techniques that increase delays [2]. Additionally, wireless networks have large jitter and delay characteristics to which real-time streams are very sensitive [4].

To address these problems we propose the *MeshVision* system. The approach integrates a Wireless Mesh Network (WMN) routing protocol and a bandwidth adaptation mechanism into each of the wireless surveillance cameras.

WMN are multi-hop wireless networks consisting of resource-poor, mobile mesh clients and relatively static (and comparatively resource-rich) mesh routers. The mesh routers connect together to form a backhaul network over which traffic from the mesh clients is carried.

In our approach MeshVision surveillance cameras function as mesh clients that extend the coverage of an existing mesh network or form an independent network on their own. Importantly, such cameras can route traffic, enabling multi-hop wireless communication. This multi-hop network is self-configuring (the cameras automatically discover their neighbours and find routes through the mesh) and self-healing (in that they can recover from link or node failures).

An example of a WMN created by seven MeshVision surveillance cameras is shown in Figure 1. In the Figure, a user (shown as Viewer) is able to access a video stream from camera C1 four hops away. As no cabled infrastructure is required, such a network could be rapidly deployed by emergency services or police to monitor a disaster site or crime scene.

The MeshVision system also includes a dynamic Bandwidth Adaptation Mechanism. This mechanism runs on each MeshVision camera and monitors that camera's wireless interface send queue. Elongation of this send queue acts as an indicator of poor link quality or congestion on the wireless network. When the send queue length exceeds a predefined limit the bandwidth adaptation mechanism firstly attempts to reduce bandwidth usage on the network by reducing the bandwidth consumed by video streams. This reduction is achieved by adapting the video parameters (such as frame rate, compression rate, resolution, etc.) of video streams pulled from itself as well as upstream cameras. If further adaptation is not possible, the mechanism triggers rerouting of the video streams in an attempt to rebalance the load on the network.

The major contributions of our MeshVision approach include:

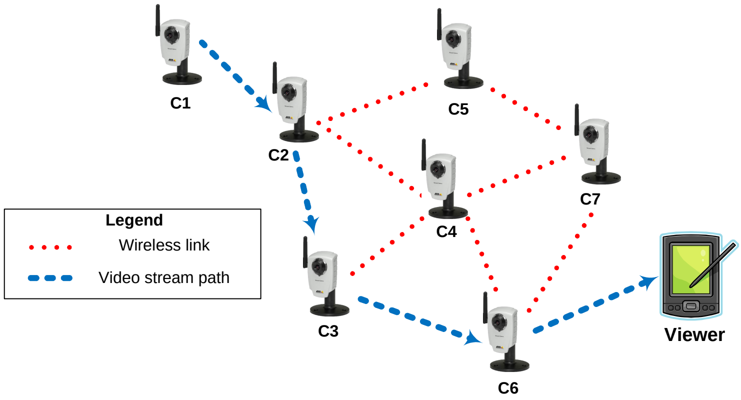

Fig. 1. An example of wireless mesh video surveillance network.

- MeshVision is a low-cost solution as it requires only a software upgrade of the surveillance cameras. Commercial products intending to fulfill a similar role (such as Motorola's MeshCam) bundle a camera with a dedicated wireless mesh router increasing the size, complexity and power requirements of the system.
- MeshVision includes an adaptation mechanism that intelligently adjusts the quality of video streams pulled from cameras on the network to ensure fair allocation of network bandwidth between cameras.
- The system includes a mesh network routing protocol that attempts to create high-capacity routes by routing around sections of the network experiencing interference and congestion.
- The bandwidth adaptation mechanism is integrated with the routing protocol so that when adaptation is not possible, rerouting of the video streams is attempted.

The remainder of this paper is structured as follows. In Section 2 we present background information relevant to our approach. An overview of related work in the field is then discussed in Section 3. We follow this in Section 4 with a discussion of MeshVision, our mesh networking and bandwidth adaptation approach for wireless surveillance cameras. An implementation is then presented in Section 5 and evaluated in Section 6. The paper is then concluded in Section 7.

2 Background

In this section of the paper we present information on the Ad hoc Distance Vector (AODV) routing protocol [7] and discuss our extended, congestion-aware version of AODV, which we refer to as AODV-CA [8]. It is this AODV-CA routing protocol that is used in our MeshVision implementation.

2.1 AODV

In the standard AODV routing protocol [7], route discovery is initiated when a node has a packet it wants to send but has no route to the packet's destination. This results in the node broadcasting a Route Request message (RREQ) to all its one-hop neighbours.

Nodes that receive a RREQ that are (1) not the requested destination, or (2) do not have a fresh route to the destination forward the RREQ to all their one-hop neighbours. It should be noted that a node only forwards a RREQ if it has not received that RREQ before, or the metric associated with the RREQ is better than the metric it currently has for the route to the RREQ source. In this way the RREQ is flooded through the network with its spread controlled by a time to live field (decremented on each hop). Once this time to live value reaches zero, the RREQ is dropped.

When a RREQ is received by the requested destination node, or a node with a fresh route to the destination, a Route Reply (RREP) message is sent. This RREP travels back along the reverse path over which the corresponding RREQ was received. Each node that receives the RREP creates a route back to the sender of the RREP. When the RREP is received by the source of the RREQ, a bi-directional route exists between it and the requested destination.

2.2 AODV-CA

In our previous work we developed a congestion-aware variant of AODV referred to as AODV-CA [8]. Improvements over the standard AODV include a new routing metric to measure the "goodness" of routes as well as support for multi-radio nodes.

The new metric is known as the Channel Diverse Congestion Aware (CDCA) metric and has two components capturing the channel diversity and expected channel congestion along the route.

To acquire the link congestion information, the metric uses the IFQ (Interface Queue) length information from the node's wireless card driver. This queue contains all outbound layer-2 frames to be transmitted by the physical layer. A build up of frames in the IFQ indicates congestion, either due to traffic load or due to low link quality. Our routing metric incorporates the IFQ length as it also reflects a number of link parameters including: link quality, link capacity, interference, and background noise. Furthermore, the IFQ length is information that is locally available at the data link layer, and does not require communication with other nodes or any expensive operations such as active probing (an approach whereby probe packets are sent to all neighbours at regular intervals).

In order to make the IFQ lengths comparable between links with dissimilar data rates, we divide the IFQ length by the current data rate (BW) of a link to compute the estimated time required to empty the queue, which we refer to as the Queue Discharge Interval (QDI).

$$QDI = \frac{IFQ}{BW}$$

The QDI represents the minimum time a packet has to remain in the IFQ before being transmitted on to the physical medium. By normalizing the QDI we ensure that the QDI of different nodes with varying channel bandwidths are made comparable. The Cumulative QDI (CG) of a path consisting of n links is the sum of the QDIs of all links forming that particular path.

$$CG = \sum_{i=1}^{n} QDI_i = \sum_{i=1}^{n} \frac{IFQ_i}{BW_i}$$

The computed CG is included in the Route Request header of AODV-CA routing protocol and stored at each hop along the route. Subsequently received Route Request packets with a lower QDI value are updated to remain the lowest QDI possible. To select a route with the lowest QDI value, the CG value is also retained in the routing table as the cost of the Reverse Route to the source.

3 Related Work

In the literature a number of publications have examined the problem of flow control for multi-media streams as well as video streaming over wireless networks.

For example, Akyildiz et al. [1] developed the Rate Control Scheme (RCS) that works in a TCP-friendly way to control the transmission rate of video streams over lossy network links. Their approach uses *dummy* messages initially sent at the application's expected data, and then sent periodically, to gauge the remaining bandwidth and round trip time along a path. This information can be fed into an adaptive encoder to adjust video stream quality and thus the bandwidth consumed by the video stream. RCS is able to differentiate between link congestion and packet transmission errors, and apply different transmission control algorithms for each case. For example, packet losses due to link congestion cause the transmit rate at the sender to drop. Packet losses due to errors initially cause the transmission rate to be cut and a burst of probes to be sent to the receiver. Should these be acknowledged then the transmission rate resumes at its previous level. By treating congestion and packet errors separately, RCS is able to support higher bandwidths over lossy links than other TCP-style transmission control protocols can. A downside of this approach is that it is concerned with flow-control over a particular route and has no mechanisms to rebalance traffic in the network.

Zhu et al. also applied a similar rate allocation approach to minimize playback distortion while limiting the increase in network congestion [9]. In their work, each node has a MAC layer monitor, a congestion minimizing routing agent and an application layer video streaming agent. The MAC layer monitor records the time spent sending, the time spent waiting to send and the actual time receiving frames. This information is used to estimate link usage and residual bandwidth on the links. The routing agents distributively decide on routes through the multi-hop network for video streams while the streaming agent dynamically adapts the video feed (using information provided by the lower layers) in order to manage the handoff between playback distortion and network congestion.

A significant drawback of the approach is that it requires nodes to exchange a large amount of information about their existing video streams and current delay information whenever a new route needs to be discovered.

Licandro and Schembra presented a mechanism to support video feeds over a multi-hop wireless mesh network in [4] . In their approach, wireless surveillance cameras attach to an infrastructure mesh network. Each camera encodes its packets using MPEG-4 such that the quality (and hence packet size) does not cause the transmit buffer to grow more than a set length. The approach assumes a static backbone of mesh routers over which the surveillance traffic can be carried. Routing over the network is done using a proactive, multi-path approach in which multiple paths are proactively discovered between all nodes.

A significant issue with this approach is that it relies on a static mesh router backbone as the wireless cameras are not able themselves to form a mesh network. The use of multi-path routing also introduces unnecessary overheads in that it requires multiple independent routes to be discovered (and then maintained). Additionally a packet reordering and a packet jitter buffer are needed at the viewer side to remove the effects of the multi-path routing (whereby the same packet can arrive at different times at the receiver). A further problem lies in the use of a proactive routing protocol. Such protocols perform poorly when the network topology changes frequently (e.g., because of high node mobility).

Huang et al. [3], presented an approach to real-time media streaming designed for wireless networks. In their work they use the H.263 codec for video conferencing over the Internet. The video stream produced by the codec groups frames into a Group of Pictures (GOP). Within this GOP, there is one I-frame and a further 14 P-frames which reference the I-frame. One GOP is transmitted every second (i.e. the frame rate is 15 fps by default). In Huang et al.'s work P-frames are selectively dropped depending on the amount of congestion in the network (determined using packet Round Trip Time). Three levels of response are proposed with congested networks dropping all of the P-frames, lightly loaded networks dropping half the P-frames and unloaded networks dropping none. A significant problem with the approach is that with only 3 levels of adaptation available, it is very coarse-grained. Lastly, the approach was primarily developed for the MPEG family of codecs (as it depends on the use of P-frames) and cannot be widely applied to other encoding protocols, such as Motion JPEG.

4 MeshVision

The MeshVision system we introduce in this paper incorporates a wireless mesh network routing protocol as well as a dynamic bandwidth adaptation mechanism. This section covers each of these separately.

4.1 Wireless mesh network routing protocol

The mesh networking functionality that MeshVision brings to wireless surveillance cameras is enabled by a wireless mesh network routing protocol. While our approach is routing protocol independent, the protocol must meet certain requirements, particularly:

- The routing protocol must be low overhead.
- The routing protocol should discover routes on an as-needed basis.
- The routing protocol should consider congestion information when choosing a path between the source and the destination nodes in the mesh network.

The first two requirements are significant because the cameras typically come with a single wireless interface and have processors optimized for video processing. This limits the amount of network-related processing that can be done. Consequently, the routing protocol used on the camera must not require a high level of communication between nodes. This suggests that the protocol should use a reactive approach (like the AODV protocol) whereby routes are only discovered when needed.

The final requirement is for the protocol to be congestion-aware. As video streams are sensitive to network congestion it is important that a low-congestion route be discovered for each new video stream. MeshVision is then able to dynamically adapt the video stream quality to cope with changing network conditions and application requirements.

4.2 Dynamic bandwidth adaptation mechanism

The second function of the MeshVision software is to balance video stream quality against network congestion. When the level of network congestion increases, the mechanism reduces the quality of video streams pulled from cameras on the network. Conversely, a drop in network congestion results in the mechanism attempting to increase the quality of video streams pulled from the surveillance cameras.

To facilitate this in MeshVision, each camera runs a bandwidth adaptation mechanism (BAM) daemon. The mechanism operates as follows.

Firstly, whenever an application pulls a video feed from a camera, it must register a Quality of Service window for the feed with that camera's BAM. This window contains maximum and minimum values for the video stream parameters (such as frame rate, resolution and compression ratio) and is stored by the BAM for the duration of the video feed.

Once video streams are being pulled from the camera, the BAM monitors the congestion on the wireless network surrounding the camera. This congestion is approximated by monitoring the length of the outgoing packet queue on the camera's wireless interface. In the Linux operating system this queue is referred to as the IFQ (Interface Queue), a term which we adopt in this paper. The IFQ length increases if packets cannot be sent fast enough, if the wireless medium is in use or if packets must be retransmitted.

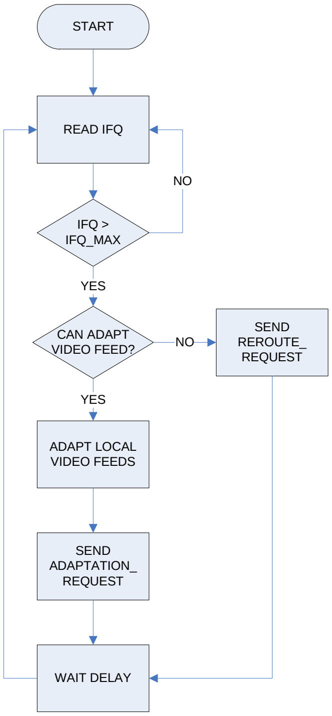

Fig. 2. Flowchart showing MeshVision bandwidth reduction process.

The monitoring procedure and subsequent behaviour of the BAM is shown in the flowchart in Figure 2.

As shown in the flowchart, the BAM continuously monitors the IFQ value of the camera's wireless interface. When this value exceeds a set threshold, shown as IFQ_MAX, the BAM initiates the adaptation process. To distinguish this BAM from other BAMs we refer to it as the *adaptation initiator*.

If the adaptation initiator has video feeds that can be reduced in quality (i.e. the video feeds are not at their minimum quality specified in the Quality of Service window), the initiator firstly downgrades the quality of its own video feed and then sends an $ADAPTATION\_REQUEST$ message to all upstream cameras (i.e. cameras that route their video stream traffic through the adaptation initiator's camera).

If adaptation of the feeds is no longer possible (because video feeds drawn from the adaptation initiator are all at their lowest QoS level), rerouting of traffic is attempted. To do this the initiator sends out a $REROUTE\_REQUEST$ message. This message is sent to the furthest away camera (where distance is measured in hops). If, after a set delay period, the IFQ value on the initiator node has not reduced, a $REROUTE\_REQUEST$ is sent to the next furthest away camera. This process is repeated until all upstream cameras have been sent a $REROUTE\_REQUEST$.

When the BAM running on a camera receives a $REROUTE\_REQUEST$ it causes the routing protocol to relaunch the route discovery process. If a better path is discovered the video feed is switched over to the new path.

Here we have described how the BAM is able to reduce the quality of video feeds in order to balance quality against reducing network bandwidth. In situations where the available bandwidth increases, the above procedure can be run in reverse. That is, if the IFQ length on a camera remains higher than a set threshold, referred to as IFQ_MIN (where $IFQ\_MIN < IFQ\_MAX$), the BAM can adjust its own video feed quality in small increments. Each increment would be followed by a waiting period to see if any downstream cameras send an $ADAPTATION\_REQUEST$ indicating that video quality should be reduced again.

With such an approach it is possible that video quality reductions following an $ADAPTATION\_REQUEST$ free up too much bandwidth in the network which causes nodes to increase their video quality again. This may trigger another $ADAPTATION\_REQUEST$ from downstream cameras. This process could repeat itself indefinitely. To prevent this occurring, the video quality increase should include a hysteresis component so that increases following an $ADAPTION\_REQUEST$ become less aggressive.

We leave a full investigation of this bandwidth recovery process for future work and concentrate on the bandwidth reduction process in the rest of this paper.

5 Implementation

To evaluate the viability of the MeshVision system we created a proof-of-concept implementation. This implementation used the Axis 207W/WM and 211W wireless surveillance cameras from Axis Communications (shown in Figure 3). These camera models are equipped with an ARTPEC-A CPU, 32 MB (64 MB on the 211W) of RAM and a wireless interface with a 1.6 dBi antenna. The dimensions of the cameras are 85 x 55 x 34 mm and 44 x 88 x 200 mm respectively. The cameras use a custom build of Linux kernel 2.6.

Fig. 3. The Axis 207W/MW and 211W cameras on which the prototype MeshVision implementation was constructed.

The MeshVision system incorporates a routing protocol and also a Bandwidth Adaptation Mechanism.

To provide the routing functionality we employed our AODV-CA routing protocol (described in Section 2). The implementation of this protocol was based on the AODV-UU 0.9.5 open source code base from Uppsala University [6]. We used AODV-CA as it fulfills all the routing protocol requirements listed in Section 4.1.

The BAM component was implemented as a user-space daemon process written in C. For our proof-of-concept, the BAM applied adaptation to video streams by modifying the frame rate and MPEG-4 compression ratio only. These two parameters were used as they were the most easily changed on the camera platform, though in a full implementation other video parameters such as camera resolution could also be modified.

When performing an adaptation to reduce video quality, the BAM calculated the video stream's new compression $c_{i'}$ and frame rates $f_{i'}$ using the equations below. The term i is taken as the number of hops from the adaptation initiator:

$$c_{i'} = c_i + d \times (hc_i + 1) \times \alpha \times (c_{max} - c_{min}); \tag{1}$$

$$f_{i'} = f_i - d \times (hc_i + 1) \times (1 - \alpha) \times (f_{max} - f_{min}); \tag{2}$$

The terms in these two equations are to be interpreted as follows:

- c_i and f_i are the current values of compression ratio and frame rate before the adaptation;
- d is the percentage of the current video quality to which camera at each hop should be reduced; this should be set to a relatively small value (around 1-2%) to maintain the best video quality possible throughout the system;
- hc_i is the hop count from the initiator camera (e.g., hc to itself is 0);
- α is a weight parameter that biases the system toward either better image quality of individual frames (i.e., lower compression ratio) or better awareness of changes in a set of image frames (i.e., higher frame rate);
- c_{max} and c_{min} are the upper and lower bound of compression rate supported by the camera; and
- f_{max} and f_{min} are the upper and lower bound of frame rate supported by the camera.

6 Evaluation

To test the viability of our MeshVision scheme, we created an experimental testbed consisting of one 211W and five 207W/MW wireless surveillance cameras from Axis Communications. A chain topology was used for the network configuration as shown in Figure 4. The cameras were connected using 802.11g, wireless channel 1. As all cameras were within range of one another during the testing, layer-2 MAC filtering was used to enforce the network topology.

The roles of Client 1 and Client 2 within the testbed were performed by two Dell Latitude D610 laptops running Linux 2.6.24. The machines were equipped with one 802.11b/g interface each.

Fig. 4. Mesh network-enabled testbed topology.

Three different experiments were performed as part of the evaluation which looked: at the maximum throughput the surveillance cameras could achieve over multiple hops, the maximum number of video streams that could be supported without the BAM running, and the number of video streams supported with the BAM active on each camera within the testbed.

6.1 Throughput evaluation

In this first experiment, our goal was to measure the average maximum throughput that could be achieved using wireless mesh-enabled cameras arranged in a chain topology. The offered load for this experiment was produced using the *iperf* utility. Client 1 was set up to be an iperf client, while Client 2 functioned as an iperf server. It should be noted that the bandwidth adaptation approach was not used during this experiment.

We varied the number of cameras (i.e., hops) in the chain and measured the maximum throughput achieved between Client 1 and Client 2.

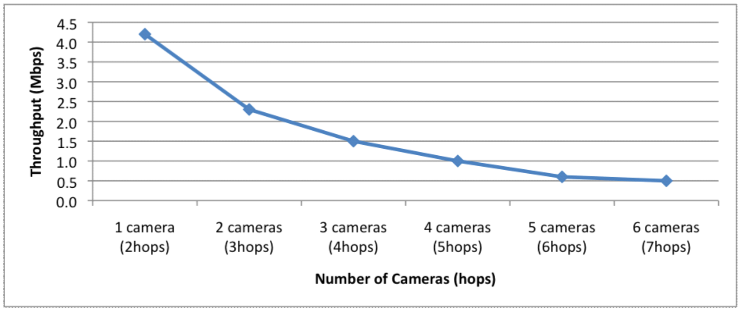

Fig. 5. Quantitative performance evaluation results.

The results of the experiment are plotted in Figure 5. From the Figure it can be seen that the maximum throughput obtained for these experiments was 4.2 Mbps in the 2 hop scenario, dropping to approximately 500 Kbps in the 7 hop scenario. The drop in bandwidth as the number of hops increases can be explained by the shared nature of the wireless medium. Because of this characteristic, nodes can not send and receive at the same time; and each node is competing to send its own packets.

These results demonstrate that the MeshVision concept of loading routing software onto the wireless surveillance cameras is feasible. In particular, the resource-limited cameras can run the AODV-CA routing protocol, form a mesh network and are capable of sustaining hundreds of kilobits per second data rates over multiple hops.

6.2 Determination of number of supported video streams

In our initial testing we attempted to determine the number of video feeds that the network could support *without* the MeshVision adaptation mechanism. The tests were performed with the camera resolution set to 480x360. The MPEG-4 codec was used with 30% compression and a frame rate of 30 fps for transferring the video. The results of the test are shown in Figure 6 which plots the number of concurrent video streams against the number of lost layer-2 frames. The number of lost frames was determined using the Wireshark[3] packet sniffing utility. As can be seen in the Figure, the percentage of lost packet frames rises quickly as more video streams are added, reaching nearly 80% for 4 concurrent video streams. The network would not support any additional video streams beyond four.

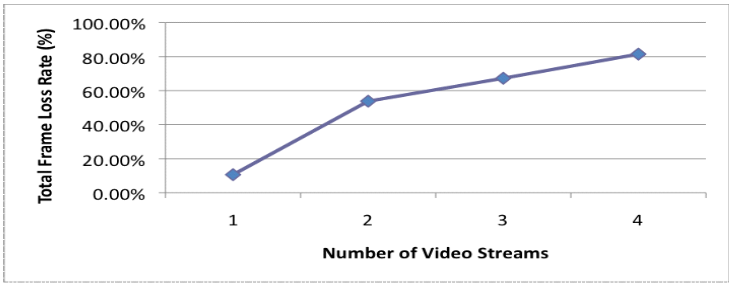

Fig. 6. Packet loss rates with no adaptation.

These results suggest the need for video stream quality adaptation, particularly in dynamic networks where the video settings that give the best quality while supporting as many cameras as possible cannot be determined *a priori*.

6.3 Bandwidth adaptation mechanism evaluation

To evaluate the performance of the bandwidth adaptation mechanism (BAM), the testbed was configured into the chain topology shown in Figure 7. This topology was enforced with MAC layer filtering.

---

[3] http://www.wireshark.org/

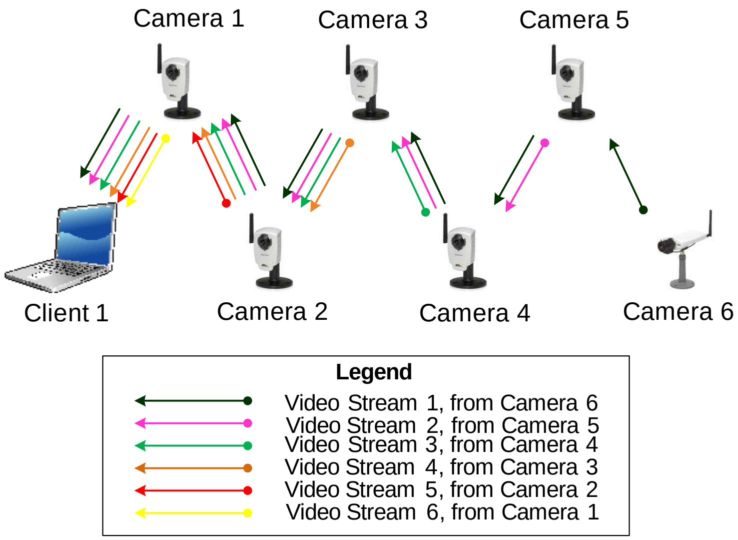

Fig. 7. Testbed topology showing direction of video flows from the cameras.

As with the previous test, the cameras were set to use a resolution of 480x360 pixels with the MPEG-4 codec at 30% compression and 30 fps. The value of parameter d from equations 1 and 2 was set to be 1%.

In our testing, a video stream viewer application running on Client 1 was used to pull concurrent video feeds from each of the six cameras in the chain topology. The video streams were initiated starting with Camera 6, then in descending order to Camera 1. Arrows indicating the path taken by each of the video streams can be seen in Figure 7.

After a video feed was started, the BAM on Camera 1 evaluated its IFQ length. If this value was more than the IFQ_MAX, the bandwidth adaptation procedure described in Section 4 was run. Only when the IFQ value reduced below IFQ_MAX was a new video session started. The value of IFQ_MAX used for the experiments was particular to the wireless cameras driver implementation and determined based on prior experimentation.

The number of network-wide adaptations required before the next video stream could be initiated is plotted in Figure 8.

As can be seen in the Figure, one video stream required two adaptations, while for 5 streams the adaptation processes needed to be run 12 times to reduce the IFQ value on Camera 1 to below IFQ_MAX. The corresponding frame loss rates are also given in Figure 9. As can be seen by comparing Figure 6 and Figure 9, the BAM was able to cut the layer-2 frame loss rate from nearly 80% to 5% (for four concurrent video streams).

Further tests were conducted with 7 to 12 video feeds. The 7th feed was pulled from Camera 6, the 8th from Camera 5 and so on. The number of adaptations

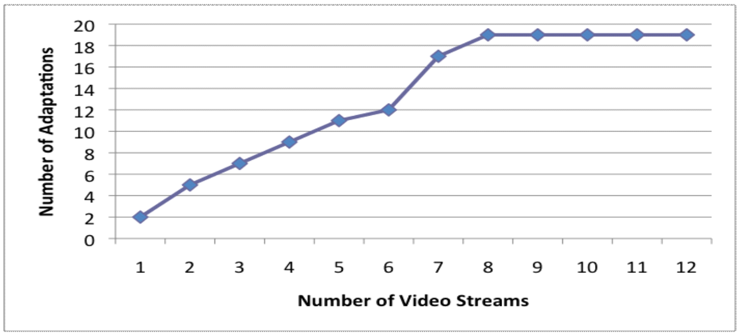

Fig. 8. Number of adaptations required to support the desired number of video streams.

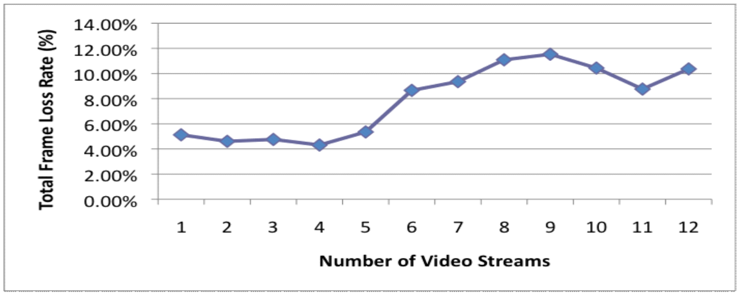

Fig. 9. Packet loss rates for video streams when using BAM.

required before the next stream could be initiated is shown in Figure 8, while the corresponding frame loss rates are shown in Figure 9.

This experiment has shown that up to 12 concurrent video streams (2 pulled from each camera) can be supported with our adaptation approach. Running 12 concurrent video streams requires the adaptation procedure to be repeated 19 times. In comparison, when not using our adaptation approach a maximum of four video streams could be supported.

7 Conclusion

The current generation of wireless network cameras can only be wirelessly accessed one hop away. The cameras must be connected to backbone infrastructure before they can be accessed outside of the cameras wireless transmission range. To overcome this problem we introduced the MeshVision system. MeshVision combines wireless mesh network functionality with intelligent bandwidth adaptation. Cameras running MeshVision are able to route and forward traffic for one another. This means that video feeds can be accessed across multiple hops. The bandwidth adaptation mechanism running on every MeshVision camera seeks to balance video feed quality with available bandwidth in order to get the best performance out of the wireless mesh surveillance network. Testing of the MeshVision Bandwidth Adaptation Mechanism showed that it increased the number of concurrent video feeds pulled over a testbed network from 4 to 12.

Acknowledgement

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program; and the Queensland Goverment.

References

1. I.F. Akyildiz, O.B. Akan, and G. Morabito. A Rate Control Scheme for Adaptive Real-Time Applications in IP Networks with Lossy Links and Long Round Trip Times. *IEEE/ACM Transactions on Networking*, 13(3), June 2005.
2. W. Feng, J. Walpole, W. Feng, and C. Pu. Moving towards massively scalable video-based sensor networks. In *Proceedings of the Workshop on New Visions for Large-Scale Networks: Research and Applications*, pages 12–14, 2001.
3. C.M. Huang, Y.T. Yu, and Y.W. Lin. An Adaptive control Scheme for Real-time Media Streaming over Wireless Networks. In *Proceedings of the 17th IEEE International Conference on Advanced Information Networking and Applications (AINA'03)*, pages 373–378, 2003.
4. F. Licandro and G. Schembra. Wireless Mesh Networks to Support Video Surveillance: Architecture, Protocol, and Implementation Issues. *EURASIP Journal on Wireless Communications and Networking*, 2007(1), 2007.
5. Michael McCahill and Clive Norris. Cctv in london. *Working paper No.6 for UrbanEye - the 5th Framework Programme of the European Commission*, page 20, 2002.
6. E. Nordstrom. University of Uppsala open source implementation of AODV.
7. C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, 2003.
8. A.A. Pirzada, R. Wishart, M. Portmann, and J. Indulska. A wireless mesh network routing protocol for incident area networks. *Journal of Pervasive and Mobile Computing, Special Issue on Global Security, Elsevier*, 2009.
9. X. Zhu and B. Girod. Media-aware multi-user rate allocation over wireless mesh network. In *Proceedings of the IEEE First Workshop on Operator-Assisted (Wireless Mesh) Community Networks, (OpComm-06)*, September 2006.