

Streaming Multimedia over Wireless Mesh Networks

David Q. LIU, Jason BAKER

*Department of Computer Science
Indiana University – Purdue University Fort Wayne
Fort Wayne, IN 46805, USA
E-mail: {liud, bakejm01}@ipfw.edu*

Abstract

Wireless mesh network (WMN) research is an emerging field in network communications. However, WMNs pose several difficulties in the transmission of information, especially time critical applications such as streaming video and audio. In this paper, we provide an overview of several research papers which utilize mesh networks for streaming multimedia. We compare the results of the research and the significance they bring to the field of wireless mesh networks. We then provide possible directions for future research into wireless mesh networks as they apply to streaming multimedia.

Keywords: Wireless Mesh Networks, Multimedia, Video Quality

1. Introduction

Wireless mesh network (WMN) [1] research is an emerging field of study in network communications. Wireless mesh networks are dynamically self-organized and self-configured where the node in the network automatically establishing an ad hoc network and maintain the mesh connectivity. According to [1], there are three classes of WMNs:

- Infrastructural Backbone WMNs: mesh routers form an infrastructure for mesh clients to connect to them.
- Client WMNs: mesh clients constitute the actual network to perform routing and configuration functionalities as well as supporting end-user applications.
- Hybrid WMNs: the mesh router infrastructure is combined with client meshing as shown in Figure 1[1].

WMNs have the following characteristics:

- Support multi-hop wireless networking
- Support for ad hoc networking
- Self-form, self-heal, and self-organize
- Support minimal mobile routers and stationary or mobile clients
- Support both backhaul access to the Internet and peer-to-peer communications
- Provide power efficient protocols for mesh clients

- Interoperate with existing wireless networks such as Wi-Fi, WiMAX, Zig-Bee, and cellular networks

Recently research has been done in the area of multimedia stream over wireless mesh networks. A multi-source multipath video streaming system [2] is proposed to support concurrent video-on-demand over wireless mesh network. Network coding [3] may be used to increase throughput over a wireless mesh network. Quality of Service (QoS) support is surveyed in [4]. Paper [5] proposes QoS routing in WMNs. Real-time video stream aggregation is studied in [6]. Results from a real testbed is reported in [7] about multimedia over wireless mesh networks. The core questions for multimedia streaming over wireless mesh networks is how to establish connection, maintain transmission of multimedia data, and achieve suitable quality across such networks. This paper, in Section 2, presents several techniques that will cover the topics of path determination, adaptive quality, and cross-layer information gathering. In Section 3 we compare the varied techniques across several dimensions to determine such attributes such as usefulness and efficiency. This is to be followed by areas for future research in Section 4. Finally, we will provide a brief conclusion stating the resultant findings in Section 5.

2. Existing Techniques for Multimedia Streams over Wireless Mesh Networks

There are a variety of techniques for transmitting multimedia streams across a given wireless mesh network. However, we will limit the discussion to three general areas: path determination techniques, adaptive quality, and cross layer information gathering. These techniques try to solve the problems that occur with improving the quality and performance of multimedia transmissions across wireless mesh networks.

2.1. Path Determination Techniques

To transmit the data from the source to the destination requires the use of some form of path determination or routing algorithm. One such technique utilized in [8] is congestion-minimized routing. Congestion-minimized routing tries, as its namesake suggests, minimizing the congestion which is defined as the average delay per link. This is accomplished by dividing the total transfer rate into K sub-transfers of equal rate ΔR_k , and assigning each sub-transfer to a given route. They minimize the equation

$$\min \sum_{(i,j) \in \rho_k} \frac{C_{ij}}{(C_{ij} - F'_{ij})^2} \Delta R_k \quad (1)$$

where ρ_k is the path from source to destination, C_{ij} is the capacity of the link from node i to node j , and F'_{ij} denotes the existing network traffic plus the prior sub-transfer rates. To find the route for the sub-flow they make use of the capacity of the link and the current rate of traffic across that link. The routing problem in (1) is solved by utilizing the distributed Bellman-Ford algorithm, where each node has to store the minimum path cost from itself to the source and the link cost to all the neighbors. It is stated that solving (1) using Bellman-Ford will converge to the solution in a network of N nodes and having diameter D within D rounds of information exchange. After the path has been found the destination sends a message down the reverse path to the source, thus giving the source the routing path for route K .

Paper [9] demonstrates three different path selection algorithms with varying levels of estimation utilized: end-to-end, localized, and estimation based. For all algorithms they assume that the topology does not change during video transmission, and that there is no contention for access to the medium. To accomplish this they employ the principles of HCCA protocol as applied to an IEEE 802.11e network. The application of this technique allows the scheduling of multiple flows creating an average transmission rate for each flow. The end-to-end algorithm uses complete information of the network so it must first generate the connectivity structure P for each node which has data to transmit. P is defined as all possible paths from source to destination without loops. Using P they find the minimum cost path using an algorithm which exhaustively searches all possible paths. The algorithm then finds the path which

has the smallest estimated timing requirements for transmission. The information needed for this is transmitted using separate logical communication links between nodes. The next approach presented is a localized estimation, where only the link information of neighboring nodes is known. The rest of the path information is estimated based upon an approximation of several low layer data characteristics. The third technique presented is purely estimation based and uses an estimation technique similar to the localized version; however, it does not even keep track of the information on links between the nodes. Three path determination algorithms are discussed in [10]. One is a centralized algorithm which will not be discussed due to the fact it is almost exactly the same as the prior end-to-end algorithm from [9]. This is most likely due to the majority of authors being the same and the later publishing date of [10]. The other two are new peer-to-peer (P2P) algorithms: distributed collaborative and distributed non-collaborative. Both approaches assume that there is enough available bandwidth for an overlay layer for communication of the network conditions between peers, and they both only run when a new flow is admitted, an existing flow leaves, or the topology of the network changes. The first is a collaborative distributed algorithm, which utilizes a local greedy approach. All nodes in the network must collectively sort the sub-flows. Sorting is done to satisfy the utility function for maximizing the total quality (MTQ) or the utility function for maximizing the minimum quality (MMQ). If they are utilizing the MTQ approach, then the sub-flows are sorted in descending order of λ_x/B_x , where λ_x is a flow specific parameter that depends on video characteristics and B_x is the rate requirement for the sub-flow. For the MMQ tactic they group the sub-flows by the quality layer and then sort by λ_x/B_x . After deciding and ordering all sub-flows the sender determines if there is any path to the destination. If no path exists, do not admit the video stream transmission to send, and cancel all sub-flows that depend on the denied sub-flow. If only one exists the transmission is started. If multiple paths exist, the path that leads to the smallest amount of introduced congestion is selected.

Two methods are offered for estimating congestion: bottleneck air-time congestion and mean end-to-end air-time congestion. Bottleneck congestion works using:

$$\varepsilon_x^i = \max_{v_a \in \rho_x^i} \{1 - \rho^a\} \quad (2)$$

where ρ_x^i is the i -th potential path for sub-flow x , v_a is node a along the path, and ρ^a is the fraction of total listening at v_a . Mean end-to-end works using the equation

$$\phi_x^i = \frac{1}{|\rho_x^i|} \sum_{v_a \in \rho_x^i} (1 - \rho^a) \quad (3)$$

where $|\rho_x^i|$ is defined as the number of nodes in the path and the rest of the parameters are the same as in (2). However, it defines the non-collaborative approach has each node sorting its own flows, before scheduling any other peers are allowed to schedule their flows to pass through the node.

Paper [11] presents no noteworthy developments for path determination; as it uses the UDP/IP protocol suite for routing and transmission of data. Therefore, it can be thought to be representative of the base line.

2.2. Adaptive Quality

Adjusting the quality of the video is a must for utilizing available bandwidth, and for reducing congestion throughout the network. A technique used in [8] is to minimize the distortion of the encoded video, while limiting the congestion introduced. They achieve this by estimating the tradeoff between an increase in the rate and the decrease in the distortion. The given equation balances the rate versus the distortion

$$-\Delta D^s \approx \frac{\theta^s}{(R^s - R_0^s)^2} \Delta R^s \tag{4}$$

where $-\Delta D^s$ is the distortion reduction for the stream, R^s is the encoding rate, and θ^s and R_0^s are determined from trial encodings. At time intervals of size k , the source node increases the rate allocated to the stream and monitors the congestion versus the reduction in distortion. If the increase in rate isn't worth the reduction in distortion then the rate stays where it is.

There is a pre-agreed scaling factor λ for the congestion. This allows for a reduction in distortion as long as that is less than λ times the increase in congestion. This increase occurs until it reaches the optimal point, at which point it merely responds to the congestion present in order to raise or lower the distortion. This is guaranteed to converge for networks with fixed rates and link capacities.

In [9], we are presented with no adaptive quality changes for video transmission. Once a video is desired to be sent, the path provisioning scheme tries to find a path which meets the necessary requirements for the video transmission. If one can not be found, it does not send the video stream.

In [10], they demonstrate an interesting technique based on logical flows of data. Each sub-flow represents a partition in the actual quality of the video. This is represented by the equation

$$\hat{Q}_y(P_y) = Q_y^0 + \sum_{\substack{f_x \in \Psi_y \\ \omega_x \neq -1}} \omega_x \lambda_x \log(B_x) \tag{5}$$

For (5) Q_y^0 is a parameter dependent upon the video, encoding parameters, etc. ω_x is 1 for admitted sub-flows, 0 for non-admitted sub-flows, and -1 for rejected sub-flows. λ_x is a parameter like Q_y^0 and it is dependent on the quality layer. B_x is the bit rate for the sub-flow. When a sub-flow is not admitted, all "enhancement layers" that depend upon that flow are not admitted as well. The parameter ω_x allows ease of detection for a given non-admitted sub-flow. When a sub-flow is rejected ω_x is set to -1 recursively for all dependent

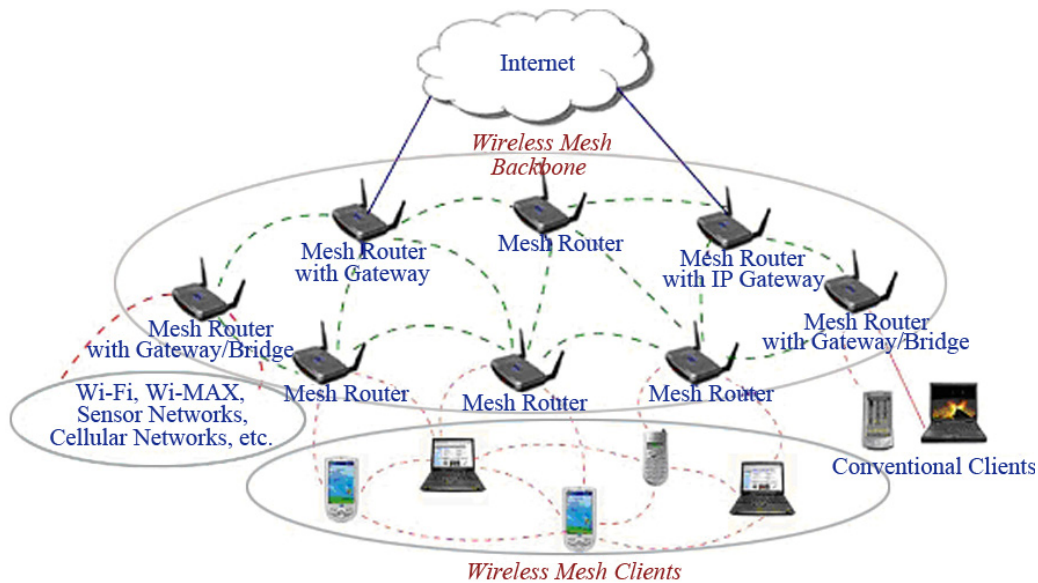


Figure 1. Wireless Mesh Network [1]

$$U_{MQ} = \sum_{y=1}^{N_p} \hat{Q}_y(P_y) \quad (6)$$

where N_p is the total number of aggregate flows. By utilizing (5) MMQ can also be defined

$$U_{MMQ} = \min_y \left\{ \hat{Q}_y(P_y) \right\} \quad (7)$$

These utility functions act as constraints, which allow the system to make quality of service decisions based upon individual sub-flows through the previously discussed path determination.

Paper [11] makes use of the standard MPEG video standard. The standard makes provisions for a quantization scale parameter (QSP), which allows the quality of the codec to be adjusted dynamically during the encoding process. A feedback formula is presented to achieve adaptive quality, which is defined as:

$$q = \Phi(s_q, a, j) = \min_{\bar{q} \in [1, 31]} (\bar{q} : s_q + R_{a,j}(\bar{q}) \leq \theta_f) \quad (8)$$

where $R_{a,j}(q)$ is the rate curve for the expected number of packets to encode for each frame type, s_q is the length of the transmission buffer before encoding the packet, and θ_f is the queue length of the transmission buffer. Equation (8) allows us to make use of the rate curve to minimize the encoding QSP and maximize the PSNR, while maintaining a full transmission buffer.

2.3. Cross Layer Information

Most of the techniques discussed in [8] utilize information from the bottom three to four layers of the OSI model. The problem is how to gather this information so that the path determination and adaptive quality algorithms can make use of it. Paper [8] gathers the busy, block and idle times which are referred to as T_{busy} , T_{block} , and T_{idle} respectively. They also keep a running average of the video payload size for each stream over the time period known as B^s . The rate for a given stream on a node is:

$$F_n^s = \frac{B^s}{T_{busy} + T_{block} + T_{idle}} \quad (9)$$

The estimation for bandwidth capacity is:

$$C_n = \sum_s \frac{B^s}{T_{busy} + T_{block}} \quad (10)$$

Employing (9) and (10), an estimation of available bandwidth for a given stream at the given node is defined as:

$$C_n^s = C_n - \sum_{s' \neq s} F_n^{s'} \quad (11)$$

Estimation of the congestion increment, denoted as ΔX_s , at a given node can be estimated using (9) and (10)

resulting in:

$$\Delta X_s = \sum_{n \in \mathbb{P}^s} \frac{C_n}{(C_n - \sum_s F_n^s)^2} \Delta R_s \quad (12)$$

which correlates the congestion increment against a given increase in the encoding rate.

Paper [9] gathers information for each link in the network. It utilizes the modulation, the bit error rate (BER), and the guaranteed bandwidth denoted as $m(l_{ij})$, $e(l_{ij})$, and $g(l_{ij})$ respectively. The modulation is defined by the physical medium, but there is no mention of how the modulation is gathered. The probability of a BER is defined as:

$$e_{l_{i,j}}(L_v) = 1 - (1 - e(l_{i,j}))^{L_v} \quad (13)$$

where L_v is defined as the size of the MSDU in bits. This is due to the assumption that the BER is normally and randomly distributed. The guaranteed bandwidth is set up so as to follow the HCCA reservation rules. These parameters are used to estimate queuing delay for each link as:

$$d_{queue}(l_{i,j}) = \sum_{\forall u \in l_{queue}^{i,j}} d_{l_{i,j}}(L_u, t_{l_{i,j}}^{mean}(T_{p_i}^{max})) \quad (14)$$

where $d_{l_{i,j}}(L_u, t_{l_{i,j}}^{mean}(T_{p_i}^{max}))$ defines the delay for the link between nodes i and j with MSDU of size L_u , with a mean retransmission time at the link and max retransmission time along the path denoted by $t_{l_{i,j}}^{mean}(T_{p_i}^{max})$. Equation (14) results in the queuing delay needed for the path determination algorithms in Section 2.1.

Information is gathered at the physical and network layers in paper [10]. The BER is also used in this paper, and is defined as

$$e(\theta_x^a) = \frac{1}{1 + e^{\mu(s-\delta)}} \quad (15)$$

where μ and δ are constants, s is the signal to noise ratio (SINR), and θ_x^a is the physical layer mode at the given node a on stream x . This gives the expected goodput (throughput without errors) defined as:

$$\bar{G}_x^a = (1 - \varepsilon_x^a(L_x, \theta_x^a)) T_x^a(\theta_x^a) \quad (16)$$

where the function ε_x^a is the probability of a bit error on a packet of size L_x bits (same equation as (14), and the function T_x^a is the physical layer transmission rate on the physical mode θ_x^a . They then utilize the transmission service interval, listening service interval, and transmission service interval denoted as t_{SI} , $t_{SI}^{(RX)}$, and $t_{SI}^{(TX)}$ respectively. The assumption is made that the transmission service is much greater than the receiving service, implying that congestion is due to transmissions. Also recorded is the fraction of the listening time given

to a stream for a node denoted as r_x^a . Using the gathered cross-layer information the admission of sub-flows is decided based on the inequality:

$$\overline{G}_x^a r_x^a \frac{t_{SI}^{(RX)}}{t_{SI}} < B_x \quad (17)$$

where if this is true the sub-flow can be allowed, otherwise it must be dropped.

3. Comparison

Given the wide variety of techniques previously described, this section will be devoted to comparing the overhead, benefits, and disadvantages of the techniques as they apply to situations dealing with streaming media.

3.1. Path Determination Techniques

The variety of path determination techniques explained in Section 2.1 have distinct advantages and disadvantages. To more objectively discuss these techniques they will be compared by the attributes over the domains: information complexity, algorithmic complexity, network overhead, congestion avoidance, and dynamic adaptation. These domains will be assigned ratings qualitatively from very bad, bad, average, and good, to very good.

Information complexity covers the storage and updateability of information, which widely varies depending on the technique used for each algorithm. Congestion-minimized routing is very good with regarding to informational complexity. This is due to the fact that it only requires estimations on the available bandwidth and utilized bandwidth on links between nodes as shown in (1). This information can easily be stored with the link information, and can be automatically updated by traffic passing through the node. End-to-end requires perfect knowledge of the entire network at each node and therefore has a very bad information complexity rating. This requires a massive amount of information to be stored at each node. Also, when the information changes all nodes must be updated. Localized attains a rating of good due to the fact that it only needs to store the information associated with the nodes one hop away in the paper, although since this grows as it looks farther away until the complexity reaches end-to-end. If it is just 1 hop away, the information is easily stored as in congestion-minimized routing. The estimation based approach receives a rating of very good for the fact that it stores only one piece of information regarding the link between nodes and estimates everything off of that one piece of information. Distributed collaborative path determination obtains an average rating. This is due to the fact that it must store information for each sub-flow passing through it; including listening times, and keep this information in

sorted order to determine when to drop quality layers. The distributed non-collaborative approach also receives an average rating, because it must also store the same amount of information as the non-collaborative. UDP/IP receives a very good rating due to the fact that no additional information has to be stored or updated.

Algorithmic complexity encompasses the run-time and ease of implementation, which again varies widely depending on the technique. Congestion-minimized routing obtains a rating of good; because of the distributed nature of the algorithm it can easily replace the distance metric used Bellman-Ford algorithm. However, it requires the use of an overlay network to keep nodes current on changes in topology, and this must be implemented to update the routes and their tables. End-to-end achieves a rating of very bad for algorithmic complexity due to the fact that it performs an exhaustive search of the path space from source to destination, and performs computations along the complete path. This results in a triple summation along different dimensions of activity to find the minimum path. The localized algorithm collects an average rating. This is due to the fact that it requires an exhaustive search of all neighbor nodes. However, it too requires an overlay network to convey information between the nodes, which is used to relay the information of the path up to the current determined node while determining which path to take. Estimation based gets a good rating due to the fact that it only requires the use of the overlay network to pass messages between nodes in the wireless mesh network. It is a simple algorithm of estimating the total path deadline and then chooses the route that minimizes the deadline criteria. The distributed collaborative algorithm gets a rating of good. This is due to the fact that the path provisioning algorithm only runs to meet significant changes in the network. However, each time a path is determined they have to see if the path can support the bit flow, incurring a tremendous amount of overhead. The ease of implementation is about the same between this and other algorithms presented due to the overlay layer used. The distributed collaborative algorithm scores a good rating. This is caused by the imposed requirement of an overlay network for notifications and path admission. However, it does not incur the complexity of communicating with its neighbor about what it is doing. UDP/IP receives a very good rating because it follows the simple pre-established algorithms provided on all routers.

Network overhead is related to informational complexity for any information transferred over the network. Congestion-minimized routing obtains an average rating for network overhead due to the fact that again it requires an overlay network which performs status message passing. It also requires that the Bellman-Ford algorithm be re-run each time a path is introduced. In addition, the algorithm takes a number of iterations of the Bellman-Ford algorithm equal to the diameter of the

graph to converge. End-to-end routing is the most costly in terms of overhead for the network. It must obtain the complete connectivity structure and all link information at each node for path determination. The amount of network overhead introduced by this is phenomenal and grows at an astonishing rate as nodes are added to the network, leading to its low rating of very bad. Localized only needs to transmit information between the neighboring nodes, so the amount of network overhead is low and the use of the overlay network does not significantly impact this technique. Estimation based acquires a favorable score for network overhead due to the fact that it transfers no information between nodes. It utilizes only link information that is set a priori to determine what path to take. Thus, estimation based gets a very good rating for network overhead because there is no overhead aside from the actual transfer. Distributed collaborative procures a rating of average for network overhead, because it is necessary to retrieve information from the network to find viable paths, and admit them into the network. This process has a high overhead in network message passing using the proscribed overlay network. The distributed non-collaborative technique receives a good rating because it is similar to the distributed collaborative in many respects, such as the use of an overlay network and path determination sans congestion avoidance. However, the individual nodes don't spend time messaging back and forth to allocate the flows reducing the overhead. UDP/IP has no network overhead aside from normal transmission costs; it also has no retransmission due to failed packets.

Congestion avoidance is a necessary feature for streaming video; if you stream video through a congested node then you will have jittery video or even lost streams. Congestion-minimized routing boasts a very good rating for congestion avoidance. The algorithm is designed to avoid the inherent problem by estimating the network congestion and finding the path through the network, described in (1), that minimizes congestion. End-to-end has a side-effect of congestion avoidance since it tries to minimize the queuing delay along the possible paths. By thus avoiding the heavily queued nodes, it avoids congestion. Therefore, end-to-end receives a very good rating for congestion avoidance. The localized estimation uses the same technique as end-to-end; however, it does not possess perfect knowledge of the network, and thus bad estimations may result in a quick build up in congestion. This method will nevertheless work for local congestion avoidance, so it receives a rating of good. The estimation based approach has no knowledge of network conditions and makes estimations for all links set up and congestion from (2) and mean end-to-end congestion from (3). The bottleneck congestion is useful for networks with bottlenecks, and mean end-to-end is useful in all other cases, both techniques provide excellent feedback on the network from the point view of reserving air space. This

leads to routing into congested areas quite easily, so the congestion avoidance for the estimation approach has a rating of very bad. The distributed collaborative technique receives a rating of very good for congestion avoidance. It gives two techniques for congestion avoidance bottleneck. Therefore, the necessity for a priori knowledge about bottlenecks detracts from the possible perfect score in congestion avoidance. Distributed non-collaborative gets a rating of very bad for implementation due to the fact that it does not implement any kind of congestion avoidance mechanism. UDP/IP also gets a low rating for congestion avoidance; by itself UDP/IP supports no inherent means for congestion avoidance. Since no other congestion avoidance means were discussed in [11], UDP/IP receives a rating of very bad.

Dynamic adaptation encompasses how responsive and accurate the algorithm works with changes to topology and data flows. Congestion-minimized routing adjusts the paths of video streams over time based on the conditions in the wireless mesh network. This automatic adjustment earns congestion-minimized routing a very good rating for the dynamic adaptation of the environment. End-to-end, localized and estimation based all assume that network topology is fixed while transmitting and that time is reserved for communication before transmission starts. These three algorithms are sorely lacking in the ability to respond to dynamic changes in the wireless mesh network. They do not reduce the current stream's bandwidth to accommodate new ones, and if a node becomes unresponsive during transmission there is no recovery for this happenstance. However, the overlay network provides a possibility of recovery if implemented in the future. As a result we assign all three of these techniques a rating of very bad. Distributed collaborative and non-collaborative both do not perform dynamic adaptation of packet routing. Again, although the algorithms used and the overlay network potentially allow for extensibility into this area. Therefore, distributed collaborative and non-collaborative both receive a rating very bad also. UDP/IP inherently routes around areas where nodes are malfunctioning or have left the wireless mesh network; however, it does not take into consideration data flows already going through the network. Therefore UDP/IP receives a rating of good.

Table 1 shows the overall associated ranks for each technique and the critiqued areas. As can be easily seen some path determination techniques fared better than others. However, congestion-minimized routing, distributed collaborative, and the standard UDP/IP suite fared the best overall.

3.2. Adaptive Quality

To compare the adaptive quality techniques from Section 2.2, we utilize the dimensions of scalability, algorithmic

Table 1. Path determination overview

Technique	Critiqued Areas				
	<i>Information Complexity</i>	<i>Algorithmic Complexity</i>	<i>Network Overhead</i>	<i>Congestion Avoidance</i>	<i>Dynamic Adaptation</i>
Congestion-Minimized Routing	very good	good	average	very good	very good
End-To-End	very bad	very bad	very bad	very good	very bad
Localized	good	average	good	good	very bad
Estimation based	very good	very good	very good	very bad	very bad
Distributed collaborative	average	good	average	very good	very bad
Distributed non-collaborative	good	good	good	very bad	very bad
UDP/IP	very good	very good	very good	very bad	good

complexity, and smoothness. Again they will be appraised on the same scale as used in Section 3.1.

Scalability refers to how well the technique scales in terms of the network size and the amount of traffic present in the wireless mesh network. The technique presented in [8] is extremely scalable well due to the fact that the technique only worries about its own traffic so each transmitting node is self monitoring. There is an overhead incurred by using the overlay network to transmit the needed information about congestion along the path. This is somewhere between constant and linear in terms of the growth of the network. This can be said because the addition of a node may or may not impact the path taken. If the path is lengthened then, the impact is linear; if the path stays the same, then it is constant. Also, (4) provides a way to calculate the given decoding distortion, which can then be utilized to optimize the system against the increase in congestion. This leads to even the network layer adapting to the introduction of streams of data as described in (1). This appears to be extremely scalable because it is also done per sending node, with overhead in the overlay network. Therefore, this technique receives a rating of good for scalability. Paper [9] presents no techniques for adaptive quality and thus obtains a rating of very bad for this section. In [10], distributed collaborative algorithm only, the flows are split into hierarchical layers equivalent to layers of quality or “enhancement layers.” To make use of this two possible utility functions are described in (6) and (7) which do determine the MMT and MMQ respectively. Both of these equations utilize (5), which allows the wireless mesh network to determine the aggregate quality of a video stream across the network. Then, when utilizing the MMT or MMQ utility functions the system can possibly self-optimize the flows through utilization of the overlay network. However, the overhead incurred on the overlay network would be quite extreme. This is because that both utility functions require perfect knowledge of the number of video streams through the network and all rejected “enhancement layers” need to

be kept track of to reintroduce these video streams into the system. This can be deduced from both utility functions requiring perfect knowledge of the number of video streams moving through the network to make any decisions and from the fact that all rejected “enhancement layers” need to be kept track of to reintroduce them into the system. The first problem grows tremendously with respect to the network, whereas the second problem is only the responsibility of the sender nodes. This leads to a rating of average for scalability. Paper [11] presents a feedback approach local only to the system encoding video through (8). This leads to each individual encoder only adjusting its encoding rate without communicating with other nodes in the wireless mesh network. This problem scales perfectly as there is no communication between the nodes, therefore it receives a rating of very good.

Algorithmic complexity is defined as ease of implementation and speed of response when performing adaptive quality. Paper [8] has a quick initial response due to the nature of how the congestion and distortion converge. In the beginning, the decoding distortion rapidly decreases as you raise the encoding rate. Whereas, the introduced congestion in the network increases slowly in the beginning and increases rapidly after the initial low rates. Thus, the network quickly converges upon the goal encoding rate given the maximal network congestion. Ease of implementation is again a problem with the techniques described in [8]. For the algorithm to efficiently and quickly converge, the overlay network must be implemented and quickly pass the requisite knowledge to the appropriate nodes. This information passing overlay network is a significant burden to overcome, which is why it receives a rating of good for algorithmic complexity. Again, [9] obtains a meager rating of very bad for not having implemented any kind of adaptive quality. Paper [10], with respect to the collaborative approach, has a quick response due to several key features dealing from queue admittance discussed in path determination to the actual use of

MMQ and MTQ to ensure quality of service. This quick convergence is due to the fact that for admittance into the network, a given sub-flow is ordered by the benefit to the aggregate video stream. Thus, all base flows will be added; then, “enhancement layers” will be added by order of contribution. However, this is detracted by the need for an overlay network to enforce the utility functions, ensure admittance of new sub-flows, and update all nodes in the wireless mesh network. These detractions result in a rating of good for algorithmic complexity. Paper [11] presents an adaptive quality technique which is apparently simplistic to implement and extremely responsive to the criteria presented. It is simple to implement because the algorithm runs at each sender and encodes at the given quality rate based upon the current transmission buffer capacity as determined by (8). It responds instantaneously to any change in the buffer. These attributes give this technique a rating of very good.

Before we can discuss smoothness, it must be defined with respect to streaming media over wireless mesh networks. Smoothness will be defined as the continuity of the quality as it changes over time. In [8] the smoothness is somewhat ideal. This can be attributed to the small changes in (4). The congestion is only incremented by a given amount for each rate increase. This algorithm works on k time steps, such that it only increases or decreases the rate over time based on the current congestion in the network. This leads to a very smooth curve of quality for small k and a very discontinuous curve for large k . For most applications, though, it would be safe to assume that k is much smaller than 2 seconds. Therefore, it is safe to say that a rating of very good is very accurate and deserved for this technique. Again, [9] receives a rating of very bad due to not implementing any techniques. For [10] — we once more only consider the distributed collaborative — it appears to have an ordered but non-uniform smoothness. This can be said because the system implicitly orders the quality layers by contribution to the quality dependent upon the chosen utility function. Ergo, the quality is as smooth as the quality layer added or removed based upon network conditions. The problem associated with this is the varying delta for the improvement in quality. For more important base flows, the delta is much higher than for less important “enhancement layer” flows. This is still a good technique; it’s just more prone to non-uniform increases in visual quality. Thus it has a resultant rating of very good. Paper [11] implements the feedback formula in (8). However, this leads to problems in quality where the algorithm tries to ensure the transmission buffer capacity is constant. By ensuring the buffer capacity is constant, the quality of the frame may change suddenly if the buffer rapidly increases and decreases in size radically in an alternating fashion. This will lead to an oscillating digital waveform where the resulting video will have extremely high quality images

and then drop to low quality images. Because of this severe deficiency the algorithm scores a rating of bad.

Table 2 presents the final results for the papers adaptive techniques versus the critiqued areas. Most of the adaptive quality techniques are of above average quality overall.

Table 2. Adaptive quality

Technique	Critiqued Areas		
	Scalability	Algorithmic Complexity	Smoothness
Media-aware [1]	good	good	very good
Cross-layer [9]	very bad	very bad	very bad
Resource-Ex [10]	average	good	very good
Multipath-Rt [11]	very good	very good	average

3.3. Cross Layer Information Gathering

The criteria used to compare the cross layer information gather techniques are estimation accuracy and ease of implementation.

Estimation accuracy involves the underlying assumptions and the accuracy of the actual estimation formula. Paper [8] gathers information from nodes and congestion. Equation (9) utilizes the busy, block, and idle times of the wireless card along with the payload size of the stream to calculate estimation on the rate for a given stream. This seems like a reasonable estimation due to the fact that it represents the amount of actual transmitted data over the entire time it took to transmit said data. The equations (10) and (11) deal with capacity of actual nodes. For (10), we assume that the estimated capacity is extremely accurate because the blocking time isn’t normally large, and the busy time represents the time to send. This is a fairly accurate estimate on the capacity, as long as the packet sent is of an average size. Equation (12) provides a direct correlation between rate and congestion. This common observation is that an increase in rate leads to an increase in congestion. This is a very apt model for congestion increments. If the amount of expected bandwidth left is minuscule then the congestion blows up quickly for even small rate increments. However, if a plethora of bandwidth is left which results in a much smaller increase in congestion. The only possible problem in this formula is that the sum of the stream rates is larger than the expected bandwidth, which is impossible due to the lack of idle time being present in the estimated capacity. The validity of assumptions and accurate estimates results in a rating of very good. Paper [9] gathers information about low level transmission and buffer issues with respect to cross layer information gathering. They utilize the BER and the packet size in (13) to determine the probability of an error happening during transmission across a given link. This is an accurate assessment of an error happening on the link for a packet of size L_v bits. Due to the equation

calculating the probability of an error not occurring for one bit, and taking that to a power equal to the number of bits, this is not a problem. This gives the probability of an error not occurring for the transmitted data. The assumption they made was that the BER was normally and randomly distributed which is a fair assumption for normal network transactions. Using (13) they derive (14) which represents expected queuing delay on a given link. This formula sums all the queuing delays for all MSDU passing through the given link, which gives an approximate estimation of the queuing delay. However, this assumes that the delay per packet is accurately calculated. Given the amount of resources poured into devising this calculation in the paper, we assume it is valid without further verification. Therefore, this paper obtains a rating of very good for estimation accuracy also. Paper [10] gathers information on expected throughput and the BER. Equation (15) calculates the expected error rate given the physical layer mode. Again, this equation will have to be assumed correct due to the lack of resources for further verification. Equation (16) is much more feasible in its intentions. It calculates the expected valid throughput for the packet size, and multiplies this against the expected physical rate on the given physical mode. The assumptions that the error rate is accurate make this a good estimation of the available goodput. However, this is detracted by the fact that it does not subtract bandwidth for retransmissions caused by errors in the estimation. As for (17), this is a simple admission metric which accurately represents the ability to transmit a given packet. This assumes that goodput, service intervals, and the listening time given to the stream are known values. Due to the high number of assumptions and some flaws in the estimation we would assign this paper a rating of good.

Ease of implementation covers the ease of gathering this information and the availability of the actual information using modern systems. For the information that [8] gathers it is safe to say that not all of it is accessible by hardware. It is challenging to believe that the busy, idle, and wait time for each packet is accessible via a hardware interface. If the hardware does not support retrieval of this information, it has to be estimated from a custom device driver sitting on top. This device driver could perform a timing operation when called to send a message across the network interface, which can then be stored for future retrieval. The packet size is obviously easily calculated. Thus, the cross layer information for this paper is easily implemented. Due to the one major deficiency, the score obtained is only a rating good. Paper [9] has issues about where the majority of the cross layer information comes from. The modulation for the transmission at the physical layer is needed for many calculations. However, where and how the modulation is obtained is never discussed. This is a serious shortcoming for the algorithm. The error rate and throughput are easily

calculated and implemented. Thus, I would say there are not any issues with the rest of the low level gathered statistics. Therefore, this paper obtains a rating of good. The last information gathered is from [10]. Here they gather the BER from the physical layer mode. Again there is no mention of how the physical layer mode is determined as in [9]. If this mode is determined a priori then all nodes need to be set up in advance. Otherwise, the system needs to be able to identify the physical layer mode automatically and adjust accordingly. The MSDU size is easily calculated from known information at the application level. Other low level information like the length of the reservation time is easily transmitted across the overlay network and must already be known to actually send the video. Thus, we assign a rating of good also for the same reasons as the other sections.

Table 3 provides a quick overview of how cross layer information gathering fared for each paper. All papers essentially estimate accurate low level information or gather it directly. There do not appear to be any major problems with how the information was gathered or utilized.

Table 3. Cross layer information gathering

Technique	Critiqued Areas	
	<i>Estimation Accuracy</i>	<i>Ease of Implementation</i>
Media-aware [1]	very good	good
Cross-layer [9]	very good	good
Resource-Ex [10]	good	good
Multipath-Rt [11]	N/A	N/A

4. Possible Future Research

There are a variety of areas for potential research for streaming media over wireless mesh networks, due to the new and their emerging nature and the problems they present.

4.1. Dynamically Adapted Path Determination

From the path determination techniques presented in this paper it is clear that dynamically adapted routing algorithms need to be researched for wireless mesh networks. The majority of the presented algorithms had problems when faced with working on topologies that changed.

4.2. Congestion Avoidance

Congestion in a wireless mesh network leads to dropped packets, reduced throughput, and dropped video streams. To avoid this cross layer techniques need to implement congestion avoidance. Once implemented, this will help increase the total network utility and save on wasted

transmissions.

4.3. Adaptive Quality to Algorithms

Adaptive quality is requisite to meet the demands of the changing network. Without adjustments to quality, video transmissions may be jittery or even dropped. All that is needed is a dynamic adjustment of the video quality with respect to the state of the network. Several algorithms presented in this paper are good starting points for continuing adaptive quality algorithms.

4.4. Network Overlays

Several algorithms utilized network overlays in this paper. However, the overhead incurred by the network overlays is extreme in some cases. Optimizations need to be made to existing algorithms, or overlay networks need to be redesigned to reduce the overhead incurred by overlay networks.

4.5. Information Reuse

Cross-Layer optimized methods were the bulk of the presentation in this paper. A majority of the papers here only used the information gathered for one very specific purpose and not as many ways as could possibly be done.

4.6. Expansion of Described Algorithms

The current algorithms presented in these papers could be improved in a variety of ways such as congestion avoidance. A majority of the papers had the capability for congestion avoidance and simply not yet implemented it.

5. Conclusions

This paper compared five papers which streamed video across wireless mesh networks. All papers were decomposed into sections dealing with path determination, adaptive quality, and cross layer information gathering. Commonly an overlay network was used to ensure transmission quality, and route information to nodes in the network. Path determination was lead by the newer congestion-minimized, localized, and distributed collaborative algorithms on average. Although, these techniques are harder to implement and have a higher overhead they are more flexible in the long run. The congestion-minimized routing approach is a good example which is more flexible than UDP/IP, especially in areas of congestion avoidance. Adaptive quality — which is important for the fact that quality must be dynamically adjusted during transmission to deal with the state of the network — was planned into the algorithms in some way. Wireless mesh networks are

new and emerging field of study, but this entails that strong research needs to be focused on this area. If this is done it will allow for new robust algorithms for transmitting large amounts of data wirelessly independent of any centralized control.

6. References

- [1] Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, pp. 445–487, 2005.
- [2] D. Li, Q. Zhang, C.N. Chuah, and S.J. Ben Yoo, "Multi-Source Multi-Path Video Streaming over Wireless Mesh Networks," *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2006.
- [3] H.Seferoglu and A.Markopoulou, "Opportunistic Network Coding for Video Streaming over Wireless," the 17th Packet Video Workshop, November 2007.
- [4] P.S. Mogre, M.Hollick, and R. Steinmetz, "QoS in wireless mesh networks: challenges, pitfalls, and roadmap to its realization," 17th International workshop on Network and operating systems support for digital audio & video, June 2007.
- [5] V. Kone, S. Das, B. Y. Zhao, and H. Zheng, "QUORUM— Quality of Service Routing in Wireless Mesh Networks," *IEEE/ICST International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine)*, August 2007.
- [6] V. Navda, A. Kashyap, S. Ganguly, and R. Izmailov, "Real-time video stream aggregation in wireless mesh network," *IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–7, September 2006.
- [7] V. Chavoutier, D.Maniezzo, C.E. Palazzi, and M. Gerla, "Multimedia over wireless mesh networks: results from a real testbed evaluation," *The Sixth Annual Mediterranean Ad Hoc Networking Workshop*, pp. 56–62, Corfu, Greece, June 12–15, 2007.
- [8] X.Q. Zhu and B. Girod, "Media-Aware Multi-User Rate Allocation over Wireless Mesh Network," *IEEE First Workshop on Operator-Assisted (Wireless Mesh) Community Networks*, pp. 1–8, September 2006.
- [9] Y. Andreopoulos, N. Mastronarde, and M. Van Der Schaar, "Cross-layer optimized video streaming over wireless multihop mesh networks," *IEEE*, vol. 24 pp. 2104–2115, November 2006.
- [10] N. Mastronarde, D.S. Turaga, and M. Van Der Schaar, "Collaborative resource exchanges for peer-to-peer video streaming over wireless mesh networks," *IEEE*, vol. 25, pp. 108–118, January 2007.
- [11] F. Licandro, A. Lombardo, and G. Schembra, "Applying multipath routing to a video surveillance system deployed over a wireless mesh network," the 2nd ACM International Workshop on Wireless Multimedia Networking and Performance Modeling (Terromolinos, Spain, October 06–06, 2006), *WMuNe'06, ACM, New York, NY*, pp. 19–26, 2006.