This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2008 proceedings.

Adaptive Split Transmission For Video Streams In Wireless Mesh Networks

Wanqing Tu, and Cormac J. Sreenan
Department of Computer Science, University College Cork, Ireland
{wt1,cjs}@cs.ucc.ie

Abstract— **Wireless mesh networks hold great promise in the wireless transmission of video flows, particularly if the problem of providing sufficient network capacity can be addressed. For this reason, schemes which help to address this difficulty in capacity-limited wireless networks are of great interest. This paper presents a novel and simple algorithm,** *adaptive split transmission algorithm*, **for achieving real-time, and quality-guaranteed video transmission in wireless mesh networks. The algorithm utilizes the unused capacities of multiple channels rather than trying to transmit the flow over just one overloaded channel. The flow is efficiently split into several sub-flows in a capacity-aware manner, each sub-flow then being transmitted through different channels in parallel. The** *adaptive split transmission algorithm* **controls flows dynamically in response to changes in the states of the available channels, thereby avoiding the overloading of any one channel. We evaluate the algorithm through simulations. The results show that the** *adaptive split transmission algorithm* **achieves synchronized, quality-guaranteed, and real-time wireless video transmission. The proposed algorithm can be used for interactive real-time wireless video applications without changing current wireless hardware, MAC protocols and upper-layer protocols.**

I. INTRODUCTION

The interest of wireless mesh networks (WMNs) has been greatly spurred by a number of potential commercial applications. Currently, most of these applications [1,2,3] focus on providing last mile data connectivity to the Internet. In this paper, we study video transmission in WMNs. A WMN replaces access points in WLAN and WiFi and base stations in cellular networks with inexpensive mesh routers that have minimal mobility. It shows that the WMN has stable connectivity and capacity, and enables efficient transmission under two different access points through equipping mesh access points (i.e., mesh routers) with multiple radios to perform routing among mesh routers and access of mesh clients separately. The WMN is therefore suitable for transmitting video traffic as compared to other wireless networks.

Video applications, such as online games, wireless video conferences, real-time monitoring of activities at homes and in offices, and online exchange, generate high rate traffic and have stringent requirement for short delay and small delay jitter performance. Wireless links have limited bandwidth and the throughput is always decreased because of interfering and fading. Hence, high rate video traffic challenges WMN for overload-free transmission. Overload causes longer delay and larger delay jitter. A number of researchers has advocated the use of layered transmission as a solution to transmit high rate video flows with acceptable performance in the Internet. Layered transmission adapts to congestion through encoding a video flow onto multiple layers and deciding to transmit a layer suiting to link capacity. Different layers introduce different transmission rates. A basic layer is such a layer that has the lowest rate to guarantee acceptable performance. It is easy to extend layered transmission to a wireless world [14-15]. But, the capacity in a wireless network is limited. When several traffic coexists, a wireless node even has difficulty in transmitting a basic layer video. This paper addresses the problem of quality-guaranteed video transmission when a basic layer transmission causes channel overload.

Recent development in wireless technologies (e.g., IEEE 802.11) provides several non-overlapping radio interfaces that each such interface can have at least one channel. Multiple radios provide an opportunity to explore WMN transmission. A line of research [5,6,7] focuses on utilizing multiple channels through switching channels on the same radio. P. Bahl *et al.* [5] presented SSCH in which end hosts hop on different channels through slotting time. J. So *et al.* [6] designed a medium access control protocol that solves the multi-channel hidden terminal problem through dynamically using temporal synchronization. S. Wu *et al.* [7] designed a new MAC protocol that employs a RTS/CTSlike reservation mechanism to dynamically assign channels to mobile nodes in an "on-demand" way. An alternative approach [8,9,10,11,16,22] is to study channel assignment and channel hopping through assigning different channels to different multiple radios. P. Kyasanur *et al.* [16] presented a multi-interface channel assignment protocol in which each node is assigned some fixed channels for long time intervals and dynamically assigned other channels over short time. A. Adya *et al* [8] presented the multi-radio unification protocol (MUP) to optimize local spectrum usage via intelligent channel selection. A. Raniwala *et al* [10] studied a centralized greedy solution in which links are visited in the decreasing order of link loads. In [11], A. Raniwala *et al* extended their study to a distributed algorithm - *Hyacinth*. M. Shin [22] *et al* presented SAFE which is a channel assignment scheme that utilizes all independent channels in the system while attempting to distribute links sharing a particular channel evenly throughout the network. Both of these two lines of research presented ways to search anticipated channels or paths. Once a channel or path is found, traffic is transferred to this channel or path. In this paper, we explore multiple channels in a new point of view. While multiple channels provide opportunities of additively individual channels, they also generate more unused network capacities. By the unused network capacity, we mean the capacity in a channel that is currently unoccupied by traffic transmitting through the channel. For example, if a 64Kbps flow passing through a channel with the bandwidth of 128Kbit/s, the unused capacity of the channel is 64Kbit/s. Usually, the unused capacities are wasted because they are not enough for transmitting a whole flow. In this paper, without complex channel hopping and assignment, we simply and fully utilize the unused capacities in the wireless network to transmit quality-guaranteed video flows when the transmission of basic layer video causes overload in any individual channel.

We present a new WMN traffic control algorithm, the *adaptive split transmission algorithm*, that fully utilizes unused capacities in the WMN system to transmit a basic layer video when the basic layer transmission is not suitable for any single channel. The algorithm splits the basic layer video into several sub-flows that are then transmit through using the unused capacities in the network. That is, the sub-flows transmit in parallel through the minimum number of

channels that have enough aggregative unused capacities suiting to the basic layer video. More specifically, the *adaptive split transmission algorithm* holds the following characters.

- Pro-activity. We propose the *overload detection* for the algorithm that detects a coming overload in some channel based on the flow input rate. Therefore, traffic control can be implemented before overload occurs which greatly decreases the possibility of unacceptable performance.
- Adaptivity. Mesh nodes employ the *adaptive split transmission algorithm* under heavy load network status adaptively. The algorithm enables the basic layer video to transmit through adaptively splitting the traffic into several sub-flows whose transmission rates suitable for channels' unused capacities.
- Efficiency. By efficiency, we mean that the number of sub-flows is small and the packet delays are short. As few as possible channels that have enough aggregative capacities are selected to transmit short delay sub-flows. To keep the number of selected channels small is to enable other flows to have opportunity to use channels and also decrease the number of generated sub-flows which is good for continuous and light-overhead communications.
- Deployability. The proposed algorithm can inter-operate with current available hardware and MAC protocols without modification, and existing upper-layer protocols including current work on multiple channels. Therefore, the algorithm doesn't complicate wireless systems and is readily deployed using current commodity wireless mesh equipments.

The paper is organized as follows. Section II presents the *adaptive split transmission algorithm* in detail. Section III evaluates the algorithm through simulations. Section IV concludes the paper.

II. Adaptive Split Transmission

Adaptive split transmission algorithm is presented for quality guaranteed video when a basic layer transmission causes overload. Before the description of the algorithm, we first introduce the network model that the algorithm employs.

A. Network Model

We use an undirected graph $G = (V(R), E)$ to represent a wireless network, where $V(R)$ is a set of nodes, R is a set of radio interfaces used by wireless mesh nodes, and E is a set of wireless links. Without loss of generality, we denote any node in the wireless mesh as $v(r)$, where $v(r) \in V(R)$ and $r \in R$. In our model, at any time, $v(r)$ can only use one channel at each radio interface. All of the channels in the network will be assigned into two categories: the control channel set and the data channel set. The control channel set includes two channels, and the data channel set includes all other channels. We will explain the reason for such channel assignment soon. Without loss of generality, we assign the channel $N-2$ and the channel $N-1$ as control channels, and all other $N-2$ channels (i.e., from the channel 0 to the channel $N-3$) as data channels.

We now see how each mesh node chooses its control and data channels. We know that each mesh node $v(r)$ has r interfaces. $v(r)$ will use $r-1$ interfaces (from the interfaces 0 to $r-2$) for data channels and 1 interface (the interface $r-1$) for control channel. If we arrange the channels in an increasing order of channel rate, two channels $N-1$ and $N-2$ are assigned as control channels in the WMN system. $v(r)$ selects either the channel $N-2$ or the channel $N-1$ as its control channel through using the following criteria: $v(r)$'s control channel won't incur interference to its neighbors when the neighbors are using their control channels at the same time. The control channel assignment is to decrease interference and confliction when mesh nodes detect network status simultaneously. We use an example in Fig. 1 to illustrate the assignment of control channels in the wireless system. When the node 0 selects the channel $N-2$ as its control channel, its neighbor (i.e., the node 1) will select the other channel (i.e., the channel $N-1$) as its control channel. For the node 2, since its neighbor occupies the channel $N-1$, it uses the channel $N-2$ as its control channel.

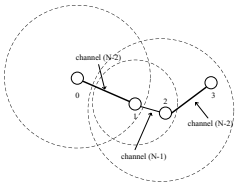

Fig. 1. An example of assigning listening channels.

B. Overload Detection

One of the key problems of the *adaptive split transmission algorithm* is to detect a coming overload to conduct effective traffic control. Sending probes (e.g., packet pair and packet train) [20-21] is a popular way to detect channel/link status. Light-weight probes make sense when measuring the status of a path that contains multiple hops. However, overheads are generated. And, the status returned by probes have already happened. Hence, it is a re-active behavior. *Overload detection* adopts a pro-active way to detect overload for effective traffic control.

Suppose there are $F(F \in N)$ flows that $v(r)$ needs to send/forward through a data channel $n(n \in [0, N-3])$. When a new video flow f whose basic layer rate is r_f enters in and wants to be output through channel n, $v(r)$ checks whether f's transmission will cause overload or not. Based on $v(r)$'s knowledge about F flows' incoming rates and the current length of the queue attaching to channel n, the *overload detection* calculates channel n's instantaneous available capacity C_n by (1) when the output cannot catch up the input packets, i.e., the queue length $l > 0$.

$$l = \int_{0^+}^{t} \sum_{j=0}^{F-1} r_j dt - \int_{0^+}^{t} C_n dt, \tag{1}$$

where r_j is the jth flow's incoming rate, and t is the time at which the new flow f inputs. We use 0^+ to represent the time at which at least one of the F flows begins occupying channel n. If $C_n \geq r_f$, $v(r)$ thinks that channel n is able to carry f; otherwise, if $C_n < r_f$, $v(r)$ employs the *adaptive split transmission algorithm* to release overload. Without introducing overhead, *overload detection* implements detection in a distributed pro-active way. The calculation of channel capacity is based on flows' incoming rates which shows that the achieved network status will take place in channel n. A subsequent control therefore makes sense in avoiding channel overload. Furthermore, *overload detection* observes each channel's status instead of bottleneck in a multi-hop path. It holds the advantage of fully utilizing each channel's capacity.

C. Adaptive Split Transmission

When $v(r)$ detects that channel n is going to carry heavy traffic, it adopts the *adaptive split transmission algorithm* to avoid a coming overload. The basic idea of the *adaptive split transmission algorithm* is to aggregate the unused capacities of the data channels to transmit f. More specifically, $v(r)$ listens to the status of the data channels, and then selects the minimum number of channels that won't cause

confliction and have an aggregative capacity suiting to f's transmitting rate. After selecting the available channels, f's basic layer is divided into several sub-flows. Each sub-flow has a rate suiting to the unused capacity of one of the selected channels. Then, all of the sub-flows transmit to next-hop mesh nodes through the selected channels in parallel.

It can be seen that selecting channels is a key step for the algorithm. To evaluate channel quality, we define a measurement η.

$$\eta = \hat{C(i)}A(i), i \in [0, N-3], \tag{2}$$

where $\hat{C(i)}$ is the ith channel's unused capacity, and $A(i)$ is the availability of the ith channel. In the algorithm, channels with larger η value have the priority to be selected as transmission channels. To calculate η, $\hat{C(i)}$ and $A(i)$ in (2) are necessary. We now present the *channel capacity collection* and the *channel availability detection* to achieve these two goals.

1) Channel Capacity Collection: Channel capacity collection is proposed to achieve $\hat{C(i)}$. $v(r)$ classifies the data channels into two groups: *occupied channels* and *unoccupied channels*. As we have introduced, $v(r)$ can have $r-1$ *occupied channels* at most. Then, $v(r)$ has $(N-2)-(r-1) = N-r-1$ *unoccupied channels* at least. $v(r)$ uses different ways to achieve the unused capacities of its *occupied channels* and *unoccupied channels*.

For the *occupied channels*, the same way as *overload detection* does is employed to calculates unused capacities. $v(r)$ maintains a queue for each *occupied channel*, and calculates the unused capacity of each *occupied channel* by the following equation.

$$C(\hat{i'}, t) = \sum_{j=0}^{F-1} r_j - \frac{d(l(i', t))}{dt}, i' \in [0, r-2], \tag{3}$$

where r_j is the jth flow's incoming rate, t is the time at which $v(r)$ collects capacities, $l(i', t)$ is the length of the queue for the i'th channel at the time t, and F is the number of flows currently sent/forwarded by $v(r)$ through the i'th channel.

For the unused capacities of the *unoccupied channels*, $v(r)$ collects them through the information exchanging with its neighboring nodes. Each mesh node in our WMN system calculates its *occupied channels*' unused capacities by (3). Then, the mesh nodes share their achieved results through exchanging. $v(r)$ sends the unused capacities of its *occupied channels* to its neighbors through its control channel, and receives the information about unused capacities of its *unoccupied channels* from its neighbors through the neighbors' control channels. The information exchanged includes not only the mesh node's unused capacities in its *occupied channels* but also the *unoccupied channels*' unused capacities that the mesh node knows. Through this way, all mesh nodes know about the status of the $N-2$ data channels. The exchanged information sent by a mesh node is an entry list in which each entry corresponds to a data channel that the mesh node knows about its situation. Each entry contains a few numbers of fields: *channel ID*, *unused capacity*, and *the IDs of the nodes occupying this channel*. And, to avoid large overhead, the list sent by each mesh node doesn't include the entries reflecting the channels that the mesh node's neighbors have commonly.

2) Channel Availability Detection: Channel availability detection is to check which channels can be used without incurring confliction. Wireless media is a shared and scarce resource. For reliable transmission, before sending data, wireless nodes need to detect the channels' availability. Obviously, each mesh node $v(r)$ can employ its *occupied channels* without incurring confliction. That is, $v(r)$ sets the variable A in (2) as 1 for all its *occupied channels*. However, for the *unoccupied channels*, confliction may be caused if employing them to transmit data. The *channel availability detection* is designed to avoid confliction when using the *unoccupied channels* to transmit data.

When $v(r)$ detects a coming overload, it checks the availability of its *unoccupied channels* with its neighboring nodes. More specifically, through the control channel, $v(r)$ sends CONFLICTION DETECTION to its neighboring nodes. After receiving $v(r)$'s request, all neighboring nodes check the data channels from which they receive data. And then, each neighboring node feedbacks an acknowledgement to $v(r)$ that includes the information of the neighboring node's receiving channels. Once receiving the acknowledgements from all of the neighboring nodes, $v(r)$ combines the information to decide the *unoccupied channels'* availability. $v(r)$ sets the variable A in (2) as 1 for all the *unoccupied channels* that $v(r)$'s neighboring nodes are not occupying to receive data; otherwise, if an *unoccupied channel* is currently used to receive data by one of $v(r)$'s neighbors, $v(r)$ sets $A = 0$ for the channel to show the unavailability.

3) Algorithm Description: When a basic layer transmission at $v(r)$ is going to cause overload, $v(r)$ implements the *channel capacity collection* to collect the unused capacities of all data channels and checks the availability of all data channels through the *channel availability detection*. After knowing the unused capacity $\hat{C(i, t)}$ and the channel availability $A(i)$ ($i \in [0, N-3]$), the *adaptive split transmission algorithm* directs $v(r)$ to transmit quality-guaranteed video in the WMN.

$v(r)$ first aggregates the unused capacities of $r-1$ *occupied channels*. If the summarization of the unused capacities of these *occupied channels* is enough or more than transmitting the basic layer of f, $v(r)$ selects f's transmission channels from the *occupied channels*. In this situation, the selected channel number m should be $m \leq r$. Otherwise, if the aggregative unused capacities of *occupied channels* are not enough for the basic layer video, including the *occupied channels*, $v(r)$ selects more transmission channels from the unoccupied available channels to meet the capacity requirement of the basic layer video. To decrease the number of channels that $v(r)$ will occupy, the *adaptive split transmission algorithm* assigns the minimum number of channels that has an enough aggregative unused capacities to $v(r)$. Assigning the minimum number of channels to $v(r)$ is to provide channel opportunity for other mesh nodes. And also, as we will introduce soon, it guarantees the flow to be split as few as possible. The channel number, $m \geq r$, in the selected transmission channel set is calculated by

$$\begin{cases} \hat{C(0}, t) + \hat{C(1}, t) + ... + \hat{C(r}, t) + ... + \hat{C((m-1)}, t) \geq r_f, \\ \hat{C(0}, t) + \hat{C(1}, t) + ... + \hat{C(r}, t) + ... + \hat{C((m-2)}, t) \leq r_f, \end{cases}$$

where t is the time at which $v(r)$ collects unused capacities, and r_f is the basic layer transmission rate of the flow f. The algorithm sorts the *unoccupied channels* in a decreasing order of their unused capacities. In the above equation, the capacities from $\hat{C((r+1)}, t)$ to $\hat{C((m-1)}, t)$ are the detected available capacities of the first $m-r-1$ *unoccupied channels*.

After deciding m transmission channels, $v(r)$ splits the basic layer video into m sub-flows. The sub-flow sizes are decided by the unused channel capacities. We use the following equation to calculate the size of the jth sub-flow $S_j (j \in [0, m-1])$.

$$S_j = \frac{\hat{C_j}}{r_f} S_f - H, \tag{4}$$

where $\hat{C_j}$ is the unused capacity of the jth channel in the selected transmission channel set, S_f is the total amount of video packets queueing at $v(r)$, and H is the header added to each sub-flow to show

the information of the sender, the destination, the flow, etc. When $v(r)$ divides a sub-flow from the video flow, it packs the sub-flow with a header and then transmits the packed sub-flow to next-hop neighbors through the jth channel. Hereafter, $v(r)$ immediately generates the second sub-flow and transmits it through the corresponding channel using the similar way above. We use an example in Fig. 2 to illustrate such split transmission. Channels illustrated in the figure are the m selected channels. Flow f is split according to the unused capacity of each selected channel. Different sub-flows transmit through different channels in parallel. Hence, sub-flows can reach a next-hop node with much less delay difference. It shows the synchronization reception at different channels. To organize the sub-flows at each next-hop node, apart from the IDs of the sender, the destination, and the flow, the header of each sub-flow includes two other fields: *split flag* and *sub-flow ID*. Flows whose *split flag* are 1 shows that the flows are sub-flows. And, the number in the *sub-flow ID* field gives the sequence of the sub-flow. After receiving these sub-flows, $v(r)$'s next-hop nodes unpack the sub-flows and assemble them based on the sequence in the *sub-flow ID* fields.

The *adaptive split transmission algorithm* is only employed by the mesh nodes who are going to suffer from overload. During the procedure of the split transmission, the mesh node $v(r)$ checks the original channel that was used to transmit f. When the channel status becomes light loaded, $v(r)$ stops splitting the video flow and transmits the whole basic layer through the original channel instead. Therefore, the occupied multiple channels can be released and then used by other mesh nodes. The algorithm is described below.

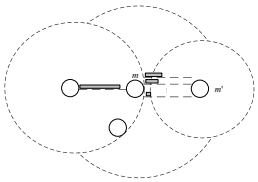

Fig. 2. An example of the *adaptive split transmission algorithm*. m channels are selected by $v(r)$ to transmit f's sub-flows in parallel. Channels $0 \sim r-2$ are $v(r)$'s *occupied channels*. Channel $r-1$ has the largest η value and channel $(m-1)$ has the smallest η value among other $m-r+1$ channels.

Adaptive Split Transmission Algorithm

Input: A sending/forwarding node $v(r)$, and the basic layer rate r_f of a video flow f;

Output: $v(r)$ outputs f to its next-hop node without causing over load;

1. $v(r)$ uses Equation (1) to estimate the current channel n's available capacity C_n ;
2. If $C_n \geq r_f$, $v(r)$ outputs flow f through channel n;
3. Else if $C_n < r_f$
4. For $i = 0$ to $r-2$
5. $v(r)$ employs Equation (3) to estimate the unused capacities of its *occupied channels*;
6. $v(r)$ sends the achieved results to its neighbors through the control channel and achieves capacity information of its *unoccupied channels* from its neighbors;
7. If the *occupied channels* have enough aggregative capacities for transmitting at least the basic layer
8. $v(r)$ sorts *occupied channels* in a decreasing order of unused capacity, and selects the first m channels who can satisfy the basic layer transmission into the transmission channel set;
9. Else if the *occupied channels* have no enough unused capacities for the basic layer video
10. $v(r)$ checks the availability of *unoccupied channels* through *channel availability detection*; and $v(r)$ sorts the available *unoccupied channels* in a decreasing order of unused capacity;
11. $v(r)$ picks up the first $m-r+1$ *unoccupied channels* into the transmission channel set;
12. For $j = 0$ to $m-1$
13. $v(i)$ splits the jth sub-flow from f based on (5) and transmits it to the next-hop nodes through the jth channel in the transmission channel set.

The *adaptive split transmission algorithm* is simple and easy to be developed in real-world networks. Without implementing channel hopping and assignment, an overloaded channel is avoided through flow splitting and multi-channel transmission. The minimum number of channels are employed to transmit sub-flows without incurring confliction, and next-hop nodes achieve synchronized reception from $v(r)$. Both the *channel capacity detection* and the *channel availability detection* limit overhead in terms of amount and range. It shows that the *adaptive split transmission algorithm* is useful for short delay and quality-guaranteed transmission in the interactive and real-time wireless video applications. Further, the algorithm operates in a distributed manner and can easily inter-operate with current wireless mesh hardware, MAC layer protocols and routing protocols without modification.

III. Simulation Evaluation

In this section, we use a set of simulations run in *ns*-2 to evaluate video transmission performance with and without the *adaptive split transmission algorithm*.

A. Simulation Metrics

We first introduce the metrics that we are going to measure in the simulations.

- Average packet delay (APD). Average packet delay at the jth receiver is calculated by $\bar{D}_j = \frac{\sum_{i=0}^{p_j-1} d_i}{p_j}$, where p_j is the number of received packets, and d_i is the delay of the ith packet. Then, APD for all receivers is calculated by

$$APD = \frac{\sum_{j=0}^{n-1} \bar{D}_j}{n}, \tag{5}$$

  where n is the number of receivers in the network. APD shows whether most of the receivers are satisfied with the delay performances or not.
- Improved quality (IQ). The best video quality that the network transmission can guarantee is measured by the maximum video rate without incurring overloaded channels and unacceptable delays. IQ is calculated by

$$IQ = \frac{\tilde{Q} - Q}{Q}, \tag{6}$$

  where $\tilde{Q}$ and Q are the best video qualities with and without the *adaptive split transmission algorithm* respectively.
- Average delay jitter (ADJ). Delay jitter is the delay variance between consecutive packets which is calculated by $J_{j,i} = |D_{j,(i+1)} - D_{j,i}|$, where $D_{j,(i+1)}$ and $D_{j,i}$ are the delays of the $(i+1)$th and the ith packets at the jth receiver, and $J_{j,i}$ is the ith delay jitter at the jth receiver. The average delay jitter at the jth receiver is $ADJ_j = \sum_{i=0}^{p_j-2} \frac{J_{j,i}}{p_j-2}$, where p_j is the number of

packets received by the jth receiver. Average delay jitter ADJ in the network is

$$ADJ = \frac{\sum_{j=0}^{n-1} ADJ_j}{n}. \quad (7)$$

B. Simulation I: Single Receiver

Fig. 3 shows the network topology. The wireless network includes 2 mobile nodes (s and r). s is the traffic sender and r is the traffic receiver. They have an identical set of four radio interfaces. Each interface has one channel. One channel is used for control channel, and the other three channels are used for data channels. Under the "good" network status, s transmit video traffic f to r through channel 1. When wireless links become overloaded, s uses the unused capacities of more than one channel to transmit f to r. Wireless communication adopts 802.11 protocol. Channel bandwidth is set as 2Mb. Video transmission rate is set as 128Kbit/s. In the simulation, we import disturbance traffic to generate network load.

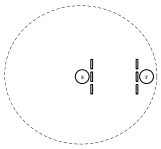

Fig. 3. Network topology for the single receiver simulation.

Fig. 4 gives the average packet delay curves. In this figure, each point is an average value of 20 runs of the simulation. The curves illustrate that the *adaptive split transmission algorithm* decreases packet transmission delay greatly when network traffic load becomes larger than 600Kbit/s. It shows that the *adaptive split transmission algorithm* improves performance much when network traffic becomes heavy. We use $\frac{A\tilde{T}D-ATD}{ATD}$ to evaluate the degree of delay decrement, where $A\tilde{T}D$ and ATD are average packet delays with and without the *adaptive split transmission algorithm*. The lowest decrement is 19% when network traffic load is 600Kbit/s, while the highest decrement is 95% when network traffic load is 1800Kbit/s. The improved performance is achieved by dispersing traffic to avoid overloaded channels through multiple non-confliction channels.

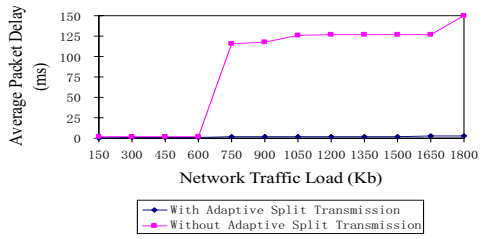

Fig. 4. Performance of average packet delays in the single receiver network shown in Fig. 3.

The first line in Table I gives the comparison of the highest video qualities (represented by data rate) that guarantee acceptable delays in the single receiver WMN. The *adaptive split transmission algorithm* aggregates capacities of multiple non-interfering channels to guarantee higher quality video transmission. According to the results, IQ in this simulation is 2.4.

Fig. 5 illustrates the average delay jitter performance in this simulation. ADJ increases with the increasing of network traffic load. Traffic controlled by the *adaptive split transmission algorithm* suffers from a bit larger ADJ when network traffic load becomes heavy

TABLE I
COMPARISON OF THE HIGHEST VIDEO QUALITY (KBIT/S).

	With the *adaptive split transmission algorithm*	Without the *adaptive split transmission algorithm*
Single receiver	5100	1500
Multiple receiver	850	150

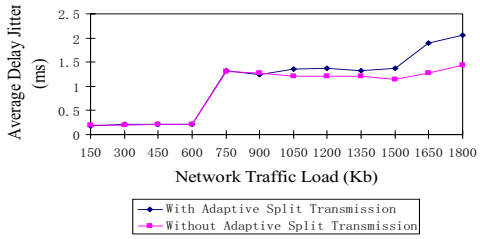

Fig. 5. Average delay jitter performance in single receiver network.

(heavier than 950Kbit/s in our simulation). It is because splitting f into different sub-flows and then transmitting them through different channels causes the variance of the time that sub-flows reach the destination. However, according to [23] and [24], delay jitter within 10ms is acceptable for video flows with the compressed TV quality. It show that the delay jitter generated by the *adaptive split transmission algorithm* is low enough to guarantee continuous and synchronizing reception.

C. Simulation II: Multiple Receivers

Fig. 6 shows the network topology. In the wireless mesh network, there are 25 nodes who have an identical set of six radio interfaces. Each radio interface has one channel. Among the 6 channels, two of them are used as control channels, and four of them are used as data channels. Node 0 is the sender, and nodes 8, 11, 12, and 24 are receivers who are randomly selected by the program. Node 0 sends one video flow with the rate of 128Kbit/s to each receiver as shown by the arrowed lines in the figure. Hence, there are 4 video flows in the wireless mesh network. Wireless communication adopts 802.11 protocol. Channel bandwidth is set as 2Mb. During the simulation, we import disturbance traffic to generate network load. Under the "good" network status, node 0 sends video flows to the receivers through channel 1. But, when the network suffers from overload, node 0 employs the *adaptive split transmission algorithm* to guarantee the basic layer video transmission.

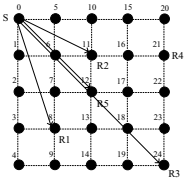

Fig. 6. Topology of the multiple receiver simulation. The dotted line between each pair of nodes shows that they are within each other's transmission range.

Fig. 7 gives the average packet delay curves in the simulation. Each point in the curves is an average value of 20 runs of the simulation. The figure shows that the *adaptive split transmission algorithm* achieves stable variance in the average packet delay, and also it decreases packet transmission delays greatly when network traffic load becomes heavy (heavier than 144Kbit/s in the simulation).

The delay curves of the communication without the algorithm are not plotted when network traffic load becomes heavier than 144Kbit/s. It is because the unshown delays are much longer than the correspondingly delays of the communication with our algorithm. To make the comparison clearly, we do not show those points. For the degree of delay decrement $\frac{ATD-ATD}{ATD}$, in this simulation, the *adaptive split transmission algorithm* achieves the lowest decrement, 65%, when network traffic load is 120Kbit/s, and the highest decrement, 98%, when network traffic load is 192Kbit/s. Compared to the single receiver performance, the *adaptive split transmission algorithm* works more efficiently in decreasing packet transmission delays in a multiple receiver network. It means that our *adaptive split transmission algorithm* controls traffic better in complex network situations.

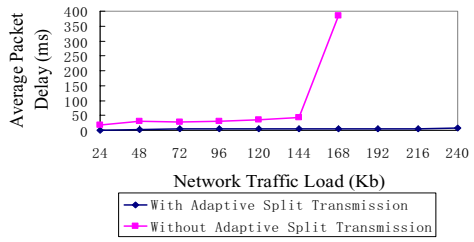

Fig. 7. Performance of average packet delays in the single receiver network.

The second line in Table I shows the comparison of the highest video qualities (represented by data rate) that guarantee acceptable delay transmission in the multiple receiver WMN. IQ in this simulation is 4.67. Compared to the performance in the single receiver WMN, our algorithm is more effective in complex network situations. Fig. 8

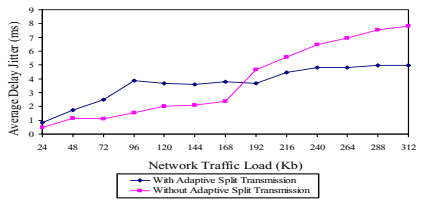

Fig. 8. Average delay jitter performance in multiple receiver network.

illustrates the average delay jitter performance in the multiple receiver network. Similar to Fig. 5, ADJs achieved by both algorithms increase with the increasing of network traffic load. But, when network traffic load becomes heavy (heavier than 180Kbit/s in this simulation), the *adaptive split transmission algorithm* generates lower delay jitters. Therefore, transmission with the *adaptive split transmission algorithm* achieves lower delay jitter than transmission without the *adaptive split transmission algorithm* when network traffic load becomes heavy. Such trend is mainly because, without the *adaptive split transmission algorithm*, the heavy network traffic cannot be controlled to generate shorter packet queue and low packet loss rate.

IV. Conclusion

We studied a novel and simple algorithm, the *adaptive split transmission algorithm*, to distribute real-time and quality-guaranteed video flows in wireless mesh networks. Our algorithm fully and efficiently utilizes the unused capacities and non-interfering simultaneous transmission of multiple channels attaching to different individual radio interfaces. Without conducting complex channel hopping and assignment, the algorithm aggregates unused channel capacities to transmit a basic layer video flow when its transmission may cause overload in individual channels. The algorithm is a complimentary traffic control scheme for the layered transmission. It defines a new measurement η to select the minimum number of available channels and transmit a basic layer video flow in a splitting way. We then use computer simulations to evaluate the algorithm. Simulation results prove that the *adaptive split transmission algorithm* enables receivers to receive quality-guaranteed and short delay wireless video transmission in a synchronization way. The algorithm has no requirement for underlying network architecture and can be easily developed on top of current wireless hardware and MAC protocols. We believe that the algorithm is useful for wireless interactive real-time video applications (e.g., online-games, online-business, and online wireless conference).

Acknowledgments

This work is supported by Embark Postdoctoral Fellowship of Ireland with the funding code: 501-et-504 4890.

References

[1] Mesh Networking, http://research.microsoft.com/mesh
[2] Tropos Networks, http://www.tropos.com
[3] IEEE 802.11s Working Group, http://grouper.ieee.org/groups/802/11/Reports/tgs-update.htm
[4] L. Loeb, Roaming Charges: Wireless Video Transmission Gets Real, http://www-128.ibm.com/developerworks/library/wi-roam47.html, 21 June, 2006.
[5] P. Bahl, R. Chandra, and J. Dunagan, SSCH: Slotted Seeded Channel Hopping For Capacity Improvement In IEEE 802.11 Ad-Hoc Wireless Networks, in Proc. of MOBICOM 2004.
[6] J. So, and N. H. Vaidya, Multi-Channel MAC For Ad-Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver, in Proc. of MOBIHOC 2004.
[7] S. Wu, C. Lin, Y. Tseng, and J. Sheu, A New Multi-Channel MAC Protocol With On-Demand Channel Assignment For Multi-Hop Mobile Ad-Hoc Networks, in Proc. of ISPAN 2000.
[8] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks, in Proc. of Broadnets 2004.
[9] M. K. Marina, and S. Das, A Topology Control Approach to Channel Assignment In Multi-Radio Wireless Mesh Networks, in Proc. of Broadnets 2005.
[10] A. Raniwala, K. Gopalan, and T. Chiueh, Centralized Channel Assignment And Routing Algorithms For Multi-Channel Wireless Mesh Networks, in Proc. of ACM SIGMOBILE MC2R, 8(2):50C65, 2004.
[11] R. Raniwala, and T. Chiueh, Architecture And Algorithms For An IEEE 802.11 Based Multi-Channel Wireless Mesh Network, in Proc. of INFOCOM 2005.
[12] L. Popa, C. Raiciu, I. Stoica, and D. S. Rosenblum, Reducing Congestion Effects in Wireless Networks by Multipath Routing, in Proc. of *The 14th IEEE International Conference on Network Protocols, November 12-15, 2006, Santa Barbara, California.*
[13] S. Lee, and M. Gerla, Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks, in Proc. of IEEE ICC 2001.
[14] M. Khansari, and M. Vetterli, Layered Transmission Of Signals Over Power-Constrained Wireless Channels, in Proc. of the 1995 International Conference on Image Processing (ICIP'95), Volume 3.
[15] K. Lisimachos, S. Deepika, P. Dimitris, and B. Stella, Layered Video Transmission Over Wireless Multirate DS-CDMA Links, IEEE Transactions On Circuits And Systems For Video Technology, Page 1629-1637, No. 12, Volume 15, 2005.
[16] P. Kyasanur, and N. Vaidya, Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks, in Proc. of *WCNC 2005*, 2005.
[17] UC Berkeley, LBL, USC/ISI, and Xerox PARC. Ns Notes and Documentation. October 20, 1999.
[18] G. Ahn, A. Campbell, A. Veres, and L. Hsiang, Supporting Service Differentiation for Real-Time and Traffic in Stateless Wireless Ad Hoc Networks (SWAN), IEEE Transaction on Mobile Computing, 2002, 1(3): 192-207.
[19] Y. Li, K. Long, W. Zhao, and C. Wang, Analyzing the Channel Access Delay of IEEE 802.11 DCF, in Proc. of *IEEE GLOBECOM 2005*, St. Louis, MO, 28 Nov.-2 Dec., 2005.
[20] W. Huang, and N. Hwang, An Embedded Packet Train and Adaptive FEC Scheme for Effective Video Adaptation over Wireless Broadband Networks, Journal of Zhejiang University (SCIENCE A), Volume 7(5), Page 811-818, 2006.
[21] Q. Liu, J. Yoo, B. Jang, K. Choi, and J. Hwang, A Scalable VideoGIS System for GPS-Guided Vehicles. Signal Processing Image Communication, 20(3):205-208.
[22] M. Shin, S. Lee, and Y. Kim, Distributed Channel Assignment for Multi-Radio Wireless Networks, in Proc. of *IEEE MASS*, Vancouver Canada, October 2006.
[23] Q. Zheng, and R. Li, Performance parameter calculation in the distributed multimedia synchronization, Journal of China Institute Communication, Vol.2, Page 53C57, October 1999.
[24] W. Cai, Multimedia Communication Technology, The Publishing House of Xian Electronics Technology University, 2000.