17-10-2005

A novel IEEE 802.15.3 CTA sharing protocol for supporting VBR streams

Kwan-Wu Chin
University of Wollongong, kwanwu@uow.edu.au

Darryn Lowe
University of Wollongong, darrynl@uow.edu.au

A novel IEEE 802.15.3 CTA sharing protocol for supporting VBR streams

Abstract

The IEEE 802.15.3 MAC enables high-rate communications between devices in a wireless personal area network and has good support for applications requiring quality of service (QoS). To meet applications' QoS requirements, such as delay and jitter, the channel time allocation (CTA) scheduler plays an important role in sizing and positioning CTAs within each super-frame. In this paper, we first present a novel CTA sharing protocol, called VBR-MCTA that enables the sharing of CTAs belonging to streams with the same group identity. This allows our protocol to exploit the statistical characteristics of variable bit rate (VBR) streams by giving unused time units to a How that requires peak rate allocation. We then present two optimizations to VBR-MCTA, namely VBR-Blind and VBR-TokenBus. The former, by giving ownership of a CTA in a round-robin manner without consideration to traffic profiles, does not consume any valuable "air-time" with signaling overheads. The latter allows a CTA to be shared by multiple devices that take turns owning unused "air-time" from CTAs. We have simulated VBR-MCTA and its optimizations in the ns-2 simulator over an implementation of the IEEE 802.15.3 MAC. Our results show that VBR-TokenBus has the best delay and jitter as it provides a one to six milliseconds reduction in both compared to standard CTA methods. VSR-Blind, although having performing poorer than MCTA-Token or VBR-MCTA, is still significantly better than traditional CTA methods at a reduced overhead.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Chin, K & Lowe, D, A novel IEEE 802.15.3 CTA sharing protocol for supporting VBR streams, Proceedings 14th International Conference on Computer Communications and Networks (ICCCN 2005), 17-19 October 2005, 107-112. Copyright IEEE 2005.

A Novel IEEE 802.15.3 CTA Sharing Protocol for Supporting VBR Streams

Kwan-Wu Chin and Darryn Lowe
Telecommunications Information Technology Research Institute
University of Wollongong, Northfields Avenue, NSW, Australia
{kwanwu, darrynl}@uow.edu.au

***Abstract*— The IEEE 802.15.3 MAC enables high-rate communications between devices in a wireless personal area network and has good support for applications requiring quality of service (QoS). To meet applications' QoS requirements, such as delay and jitter, the channel time allocation (CTA) scheduler plays an important role in sizing and positioning CTAs within each superframe. In this paper we first present a novel CTA sharing protocol, called *VBR-MCTA*, that enables the sharing of CTAs belonging to streams with the same group identity. This allows our protocol to exploit the statistical characteristics of variable bit rate (VBR) streams by giving unused time units to a flow that requires peak rate allocation. We then present two optimizations to *VBR-MCTA*, namely *VBR-Blind* and *VBR-TokenBus*. The former, by giving ownership of a CTA in a round-robin manner without consideration to traffic profiles, does not consume any valuable "air-time" with signaling overheads. The latter allows a CTA to be shared by multiple devices that take turns owning unused "air-time" from CTAs. We have simulated *VBR-MCTA* and its optimizations in the *ns-2* simulator over an implementation of the IEEE 802.15.3 MAC. Our results show that *VBR-TokenBus* has the best delay and jitter as it provides a one to six milliseconds reduction in both compared to standard CTA methods. *VBR-Blind*, although having performing poorer than *MCTA-Token* or *VBR-MCTA*, is still significantly better than traditional CTA methods at a reduced overhead.**

I. Introduction

The plethora of digital content and the ubiquity of consumer devices with high storage and processing capabilities mandates a need for high-speed communication protocols so as to facilitate quick and easy interoperation. The IEEE 802.15.3 working group [1] is developing such high-speed communication technologies targeted at high-rate multimedia applications in a wireless personal area network (WPAN). These technologies include both medium access control (MAC) and physical layer protocols that enable WPANs to support up to 245 devices operating over an area of at least 10 square meters at speeds ranging from 11 to 55 Mb/s. Other features of these standards include power saving capabilities, security, and co-existence with interfering networks [2].

The IEEE 802.15.3 MAC also supports real-time applications. Isochronous streams are given time slots during the channel time allocation period (CTAP) wherein each stream has exclusive access to the wireless medium. This contention-free channel access enables high throughput and low delay. Although the underlying signaling protocol and message formats have been standardized, algorithms that determine CTA duration and positioning within a superframe, so as to meet a video stream's QoS requirements, have been left to implementors. Furthermore, the standard provides no explicit means to take advantage of the statistical nature of variable bit rate (VBR) video traffic.

To fill the gaps, we outline a novel solution, referred to as *VBR-MCTA* that allows CTAs to be shared amongst different streams. Our solution takes advantage of the IEEE 802.15.3b's *StreamGroupID* feature and new Relinquish command that allows a CTA owner to pass control of its CTA to another device. The *VBR-MCTA* protocol requires devices to inform the piconet controller (PNC) of their traffic requirements through open management CTAs, or MCTAs, that precede each regular CTA. After collecting these requirements, the PNC then makes its decision based on a given criterion to determine which stream should be allocated the unused "air-time" in an upcoming CTA. Depending on the policy used, the PNC can choose to optimize for reductions in a VBR stream's delay, jitter or buffer requirements.

We then introduce two optimizations. The first, called VBR-Blind, is where devices gain ownership of a unused portion of a CTA in a round-robin manner. This means that the PNC does not consider the traffic requirements from devices belonging to the *StreamGroupID*. As a result, this optimization saves on "air-time" that otherwise would have been allocated to open MCTAs for signalling. The second optimization, called *VBR-TokenBus*, is similar to the Token Bus protocol [3] used on wired networks. Devices are sorted based on a given criterion, such as queue length, and the device with the highest rank is then given ownership. After this device has finished, the remaining time in the CTA is passed to the next device in the sorted list. This optimization has the lowest delay and jitter and increases CTA utilization even more than *VBR-MCTA* and *VBR-Blind* since the unused time units of a single CTA can be shared amongst multiple flows that may not individually have sufficient packets to fill the unused time.

We have evaluated *VBR-MCTA* and its optimizations via extensive simulations. We experimented with the delay and jitter impact of different number of VBR flows as well as the effect of inter-mixing high and low rates flows. Our results show that *VBR-TokenBus* offers the best delay and jitter to VBR flows, with the improvement particularly significant when there are a high number of flows. *VBR-MCTA*, which uses queue length as a criterion for determining the wining device, performs slightly worse. Finally, although the *VBR-Blind* has low signaling overheads and thereby frees up more "air-time", flows in this system experience higher delay and jitter due to the fact that the PNC does not use traffic requirement to guide its ownership assignment process. Nevertheless, *VBR-Blind* still performs significantly better than standard CTA methods.

The rest of this paper is structured as follows. We first highlight key features of the IEEE 802.15.3 MAC in Section II before describing the challenges posed by VBR traffic to a scheduler in Section III. We then outline our novel CTA sharing solution in Section IV. Then, in Section V, we present our simulation environment before discussing our results in Section VI. We then present related work in Section VII before concluding in Section VIII.

II. The IEEE 802.15.3 MAC

A. Overview

The IEEE 802.15.3 MAC [2][4] uses a master and slave model whereby a master device, called the piconet controller (PNC), controls the piconet. Figure 1 shows an example of an IEEE 802.15.3 piconet

0-7803-9428-3/05/$20.00 ©2005 IEEE.

that exists within a home network. The network is formed in an ad-hoc manner, where devices may leave or join the network at any time. Further, the piconet can support different types of traffic. For example, in Figure 1, there are audio and video streams in addition to the best-effort traffic that is being transmitted in the contention access period (CAP) of the superframe.

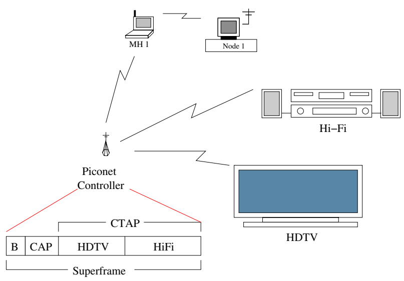

Fig. 1. Example of the IEEE 802.15.3 MAC being used to support consumer electronics devices in a home network.

A device is designated as the PNC based on a set of criteria, some of which include its preference for becoming a PNC, whether it is receiving mains power, and its ability to act as a security key originator. Once chosen as a PNC, a device's main responsibility is to coordinate channel access amongst the other devices within a piconet. Other responsibilities of a PNC include associating and disassociating devices, coordinating device wake-up times during power-save modes, and managing co-existing with other piconets or networks that share the same wireless spectrum.

The PNC coordinates channel access by periodically transmitting a beacon that defines a superframe. The superframe, as well as synchronizing all the devices in a piconet, defines two ways for devices to access the wireless channel, namely the CAP and the CTAP. Devices transmitting within the CAP must first contend for the channel through a CSMA/CA mechanism, whereas devices using the CTAP utilize a TDMA approach. For a device to receive its own CTA, it must request one from the PNC by specifying the number of desired time units. If the PNC grants the request, that device then has exclusive rights to transmit whenever its CTA occurs without needing to first contend for the channel, thus saving the overheads and delays incurred under CSMA/CA. For example, Figure 1 shows two CTAs belonging to the HDTV and Hi-Fi devices.

The IEEE 802.15.3 MAC allows an application to create, modify and tear down isochronous streams by sending requests to the PNC for time slots in the CTAP. When requesting for an isochronous stream, an application has the option of specifying various parameters such as source data rate and maximum acceptable transmission delay. These parameters are then used by the scheduler within the PNC to determine whether a stream requires sub- or super-rate CTAs in order to meets its data rate and delay requirements. Also, as part of the stream creation process, an application can specify a *StreamGroupID* that indicates to the PNC that CTAs for the stream are to be shared with streams belonging to the same group identity.

The IEEE 802.15.3 MAC supports various CTA types. Namely, regular, private and open (or management) CTAs. The type of CTA is determined by the destination address field in the request command. For example, an open CTA has the destination address field set to the broadcast address. For this protocol, a CTA type of particular interest is the open management CTA (or MCTA). MCTAs have the destination field set to the PNC's address and use a special MCTA stream index. Further, within an MCTA, devices contend for the channel via a CSMA/CA mechanism. We will see later how using MCTAs to convey traffic information, in conjunction with the *StreamGroupID* feature, allows VBR traffic to be efficiently transported over the IEEE 802.1.5.3 MAC.

A feature currently being standardized by the IEEE 802.15.3b working group is the ability for a CTA owner to relinquish its CTA to another device. One use of this feature is to allow a receiver to provide the sender with feedback on recent transmissions. For example, a TCP stream could use the Relinquish command so that, after receiving a burst of data packets, the receiver can reply with acknowledgment packets without having to wait for its own CTA to occur. Once it has control the CTA, the recipient device may do with it as it pleases, which may include using it to transmit in or passing control back to the original owner or another device. Although the standard does define the mechanism by which CTAs can be relinquished, it does not however provide details on how to select which target device should be given control. We show later how we can combine the *StreamGroupID* parameter with the ability to relinquish CTAs in conjunction with a policy for selecting the target device to handle multiple VBR streams.

III. Problem Description

A CTA scheduler for the IEEE 802.15.3 MAC must ensure that a video stream's traffic requirements are met. This task is made more difficult because the IEEE 802.15.3 MAC's allocation of CTAs at the start of the superframe makes dynamic adaptation of CTA sizes difficult, especially if the scheduler wants to leverage the statistical nature of VBR streams in order to maximize utilization and admit more flows.

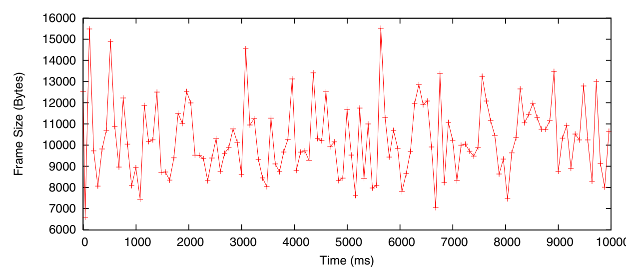

Fig. 2. An snipet of a video trace.

Figure 2 shows an example of a video trace with different sized packets arriving over time. It can be seen that the variability in frame size means that, optimally, different sized CTAs would be used for each frame. Doing this means that the scheduler at the PNC will have to adapt CTA sizes and positions to ensure that streams' QoS requirements are met without consuming disproportionate channel time for signalling. Note that for some video streams, the variability of the traffic source can be smoothed by buffering packets and waiting for the next CTA occurrence. This is more likely to be acceptable when the superframe duration is short since this means there will be less time to wait until the CTA reoccurs. However, the negative impact of using short superframes is that higher overheads are incurred due to the more frequent transmission of beacons. In sum, a scheduler will face the following challenges:

1) CTA positions. Arranging CTAs with the aim of minimizing delay and jitter is made difficult due to the need for real-time status information from devices. One solution is to shorten the superframe duration and have devices send modification requests to the PNC in order to adjust their CTAs before the

start of the next superframe. However, this approach incurs high overheads and fails to exploit the statistical multiplexing that is possible when multiple VBR flows are transmitted.

2) CTA sizes. In Figure 2, we see that a stream will require different CTA sizes at different times in order to efficiently transport the differently-sized video frames. The delay experienced by a video stream will be dependent on the match up between burst size and CTA duration since too short of a CTA means that the frame will need to wait until the next superframe. A simple solution is to allocate CTAs based on a stream's peak rate, thereby ensuring a flow will always have sufficient "air-time", however doing this is inefficient since CTAs will often go under-utilized.

3) Utilization. Due to the statistical nature of video streams, the scheduler has to ensure that CTAs are allocated so that each stream gets an appropriate amount of "air-time" without wasting resources. This means the scheduler has to balance CTA durations, positions, transmission rates, super-frame length, super- or sub-rate allocations along with the flow's delay, jitter and throughput requirements.

In the following section, we show how the above challenges can be efficiently addressed by using a solution that employs the IEEE 802.15.3b MAC's new Relinquish command [4] and *StreamGroupID* features. A point of this approach is that it does not require alteration to the IEEE 802.15.3b MAC. The key goal of our solution is for a device to relinquish its CTA when it has finished transmitting, thereby giving any remaining time units to a another device. In the next section, we show how our solution simplifies the task of arranging CTAs to meet efficiently meet the requirements of VBR streams.

IV. Proposed Solution

In the previous section, we have shown that in order to efficiently support VBR traffic and promote high utilization of CTAs, it is necessary to dynamically adapt the size and position of CTAs. To accommodate the varying needs of a VBR flow and ensure that a flow receives appropriate transmission time, we propose a CTA multiplexing solution that allows streams to transmit packets in another stream's CTA in a contention-less manner.

The aim of our solution is to extend the use of the Relinquish command. Specifically, our solution takes advantage of the fact that multiple VBR streams are statistically uncorrelated so that we can aim for a flow with a large video frame to use an under-utilized CTAs belonging another flow. Doing this means that we increase slot utilization and allow devices to acquire additional "air-time" to better meet their QoS requirements. The key issues are determining how devices advertise their instantaneous requirements and specifying the criteria by which a device is chosen as being most in need of an opportunity to transmit. By allowing streams belonging to a common *StreamGroupID* to multiplex packets in each other's CTAs, a PNC only needs to allocate a stream's mean rate, as opposed to its peak rate. This results in superior CTA utilization which in turn allows for more flows to be accommodated in a given system.

Our solution requires all VBR streams to set the *StreamGroupID* flag to inform the scheduler that CTAs belonging to VBR streams with a common *StreamGroupID* are sharable. The scheduler then schedules open MCTA slots and positions one of them before the CTAs of flows belonging to the relevant *StreamGroupID*.

We first present the basic protocol, *VBR-MCTA*, before describing the two optimizations. Figure 3 shows the concept of *VBR-MCTA* in that open MCTAs will precede each regular CTA. At each open MCTA, devices send traffic information describing their respective flows to the PNC with the aim of acquiring extra "air-time" in the upcoming regular CTA. Clearly, whether a flow obtains additional "air-time" or not is dependent on an upcoming CTA being under-utilized by its owner and the relative priority of the flow requesting it.

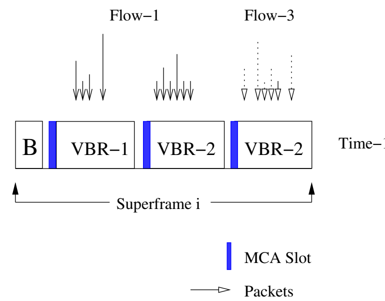

Fig. 3. This figure depicts our idea conceptually. At Time-1 we see that packets from Flow-1 are transmitted in Flow-2's CTA (or VBR-2). Without the protocol presented in this paper, the first occurrence of the VBR-2 CTA would be wasted.

From Figure 3, we see that Flow-1 is able to make use of Flow-2's empty CTA, namely VBR-2, to transmit some of its packets. To do this, a protocol needs to be developed that will allow a device to contend for any unused air-time. Further, this protocol needs to be aware that not all devices may be within range of each other even though they belong to the same piconet. Our proposed algorithm, *VBR-MCTA*, uses the PNC to manage contention since it is, by definition, a common neighbor to all devices in the piconet. Figure 4 shows graphically *VBR-MCTA*'s protocol steps.

- Step 1. Each device (apart from the original owner of an upcoming CTA) composes a set of traffic information descriptors. This information could be queue length/size, stream priority, head-of-line packet expiration time, etc.
- Step 2: Each device then sends its traffic information to the PNC in an MCTA via a CSMA/CA mechanism, similar to how the channel is accessed during the CAP. The PNC then stores each device's traffic information for processing in Step-6 below.
- Step 3: The owner of an upcoming CTA, D_{src}, determines whether it has anything to transmit. If it doesn't, then the protocol skips directly to Step-6. If it does, the protocol continues to Step-4.
- Step 4: D_{src} transmits in its CTA as usual.
- Step 5: Once D_{src} has finished transmitting, it determines the remaining time units in its CTA. If the time units left are sufficient to transmit at least two packets, including inter-frame spacings and acknowledgment overheads, D_{src} sends a Relinquish command to the PNC. This gives control of the CTA to the PNC.
- Step 6: The PNC then uses the requirements gathered in Step-2 to determine which device will be given the remaining time in D_{src}'s CTA.
  The PNC can use a variety of criteria for choosing D_{win}. In our implementation of *VBR-MCTA*, the PNC is able to choose D_{win} based on queue length/size, expiration date of the head-of-line packet, and stream priority. However, we will only be presenting results based on queue length due to its superior performance over other criteria.
- Step 7: After determining a winning device, D_{win}, the PNC forwards the Relinquish command received in Step-5 to D_{win}.
- Step 8: Once D_{win} receives the Relinquish command, D_{win}

uses the the PNC's beacon that was received at the beginning of the superframe to determine how much time remains in D_{src}'s CTA, and, if it is sufficient, schedules the next packet for transmission.

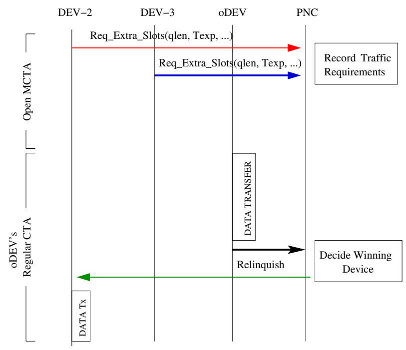

Fig. 4. A message sequence chart for the proposed CTA multiplexing protocol. oDEV denotes the owner of the upcoming CTA.

The *VBR-MCTA* requires an MCTA to precede each regular CTA to allow devices transmit their traffic requirements to the PNC. The cost of this approach is that the MCTAs themselves consume valuable "air-time" which could otherwise be given to the VBR flows. The magnitude of this signalling overhead is based on the size and the frequency of the MCTAs. For *VBR-MCTA*, the duration of the MCTAs is determined by the number of VBR flows that need to convey information to the PNC while the frequency of MCTAs is set by the number of CTAs.

In order to reduce the signalling overheads, *VBR-MCTA* can be optimized by reducing the number of the MCTAs so that the PNC is updated as to the status of the VBR flows less often. In other words, the update of one MCTA is used for multiple CTAs. The negative impact this optimization is that the information that the PNC uses to determine the device in most demand of air-time is less current. In order to define the boundary condition for the performance of this approach, we define an entirely blind technique, named *VBR-Blind*, wherein no MCTAs are used. Under this approach, the PNC has no information about the queue status of devices so the PNC attempts to fairly share unused channel time using a round-robin mechanism. In other words, the PNC awaits until the arrival of D_{src}'s Relinquish command indicates that the current CTA is free. The PNC then determines if there is sufficient transmission time units remaining, and, if there is, the next device the list of possible candidates is selected. In our implementation, this list is randomly shuffled at the beginning of each superframe.

Our second optimization called *VBR-TokenBus* which allows the Relinquish command to be transferred to another device after D_{win} has finished transmitting, provided that D_{src}'s CTA still has unused time units. This optimization is achieved by having D_{win} return the Relinquish command to the PNC when it is finished transmitting, thereby allowing the PNC to choose the next device in need of transmission time. Figure 5 shows a message sequence diagram of this technique. A slight variant of this approach would be one where the PNC returns a list of devices awaiting extra slots to D_{win}, allowing D_{win} to bypass the PNC and pass the Relinquish command to the next device directly.

V. Simulation

We have implemented the proposed protocol and its optimizations in the *ns-2* simulator (*ns-allinone-2.26*)[5] over our IEEE 802.15.3

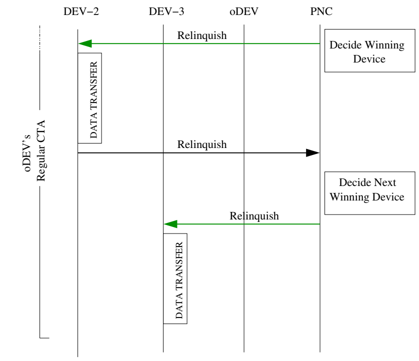

Fig. 5. A continuation of the message sequence chart in 4 using a Token Bus like implementation.

MAC. Our MAC implementation sits on top of an Ultra Wideband (UWB) physical layer based on the DS-UWB proposal currently before the IEEE. The DS-UWB physical layer [6] uses sequences of sub-nanosecond pulses spread over several gigahertz of spectrum. Depending on the distance between transceivers, DS-UWB offers data rates of 28, 55, 110, 220, 500, 660, 1000, 1320 Mb/s.

Our channel model assumes a uniform distribution of errors over time, i.e. no burst errors are considered. Note that this assumption will bias the results towards a worst-case estimate since, while keeping the average bit error rate (BER) constant, any aggregation of individual bit errors into bursts will effectively concentrate the errors into fewer packets. In other words, by distributing the bit errors evenly over time, we produce the worst-case packet error rate that could be perceived by the MAC at a given BER. Readers who are interested in other aspects of our implementation are referred to [7]. Parameters relevant to our simulation studies are outlined in Table I.

Our network topology has, depending on the experiment, a PNC managing a piconet with from 2 to 18 devices. The scheduler at the PNC positions CTAs on a first-come first-serve basis, i.e., the device that first requests for a CTA will get its CTA first. In addition, an MCTA of length $250\mu s$ is scheduled before each CTA to give devices needing extra "air-time" an opportunity to send their requests to the PNC for consideration.

Parameters	**Values**
Superframe length	15 msec
Durations of CAP	500 msec
Desired CTA time units	1 msec.
Receiver and sender buffer size	Unlimited
Bandwidth	1 Gb/s
Simulation runtime	10s
Number of runs	50
Duartion of MCTAs	$250\mu s$

TABLE I
Parameters used in simulation studies.

The PNC has the option of employing various criteria when choosing which device should be given ownership of a relinquished CTA. In this paper, we consider only queue size. That is, that the device that has the largest queue size wins. In all experiments, we keep the superframe duration constant at 15 milliseconds.

We used the MPEG-4 model developed by Matrawy [8] to generate VBR traffic. Each VBR flow is started randomly and is controlled by two parameters, *InitialSeed* and *rateFactor*. The former is used to start the frame generation process and the later scales the video transmission, i.e., increases the video transmission rate. In the first experiment, we keep the *rateFactor* constant at 10 which equates to a video transmission rate of around 2 Mb/s for all flows. We later introduce flows with *rateFactor* of 50. In all experiments, we initialize *InitialSeed* to a random value between 0.8 and 0.9 at the start of each simulation run.

In all experiments we compare *VBR-MCTA*, *VBR-Blind* and *VBR-TokenBus* to two standard CTA methods:

- Standard CTA (*No-MCTA*). Each flow gets a one millisecond CTA block.
- Super-Rate CTAs (*No-MCTA-SuperRate*). Each flow is allocated four CTAs, each having a length of 250 μs, that are spread evenly across the superframe.

VI. Results

In the following section we present results from our experiments, namely the average delay and jitter experienced by all flows. Further, we also show the average number of additional packets that flows were able to transmit due to the new protocol.

Figure 6 and 7 show the average delay and jitter experienced by each flow. We see that as the number of flows increases, there are more opportunities for a flow to obtain relinquished channel time. This reduces the average delay significantly. For example, when there are nine flows, the average delay is only 1.27ms for *VBR-MCTA* compared to around 7ms when our scheme is not enabled, i.e., *No-MCTA*. Further, as the number of flows increases, a flow has a higher chance of winning a CTA, thus improving jitter, as shown in Figure 7. For example, a flow's jitter can improve by up to 600μs.

In Figures 6 and 7, we see that having super-rate CTAs help reduce delay. Indeed, when there are only two flows, we see *No-MCTA-SuperRate* has a lower delay compared to our solution. This is because when there are only 2 flows, *VBR-MCTA* there is only one CTA before the next superframe which could potentially have unused time units. In the super-rate case however, given that a flow has multiple CTAs that are spread out over a superframe, packets arriving later in the superframe will still have a CTA that they can use. However, as the number of CTAs increase when more flows are present, our solution is able to take advantage of the extra CTAs and gives a much smaller delay than the super rate case. Further, there is nothing preventing these techniques to be combined to give high performance in all cases.

Apart from the above, we see that the jitter experienced by flows using *VBR-MCTA* and those using *No-MCTA-SuperRate* are similar. However, as the number of flows increase, the transmission opportunities offered by other flows' slots outweigh the multiple CTAs in the super rate case, thereby reducing jitter even further.

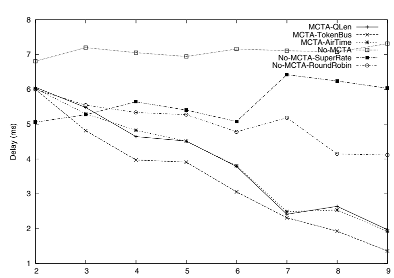

Fig. 6. Average Delay versus Number of Flows. We see that as the number of flows increases, there are more transmission opportunities, resulting in a lower transmission delay for MCTA-based solutions.

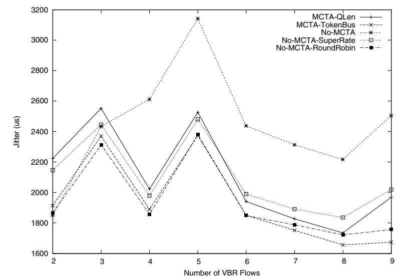

Fig. 7. Average Jitter versus Number of Flows. We see the advantage of having transmission opportunities spread out over the superframe, using either the super-rate CTA method, or using our MCTA-based solutions.

The VBR-TokenBus scheme has the best performance in terms of delay and jitter, as shown in Figures 6 and 7. The reason is simply that there are more opportunities given to devices awaiting transmission. Further, given that each flow is allocated a generous CTA block of one millisecond, the CTAs are long enough that multiple devices can transmit in a single allocation.

Figure 8 shows the number of additional packets that were transmitted by each of the flows during the extra transmission time. This graph shows only the case when there are nine flows in the piconet. We see that all flows received an equal number of opportunities, which demonstrates that the queue length is an unbiased criterion.

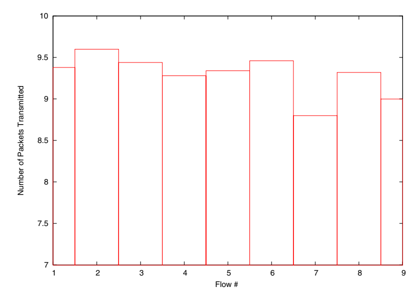

Fig. 8. Average additional packets transmitted for a nine flows scenario. This shows that using devices' queue length enables all devices to obtain a fair amount of transmission opportunities.

Figure 9 and 10 show the results from an experiment where the piconet has nine flows running at around 2 Mb/s (a *rateFactor* of 10 in the VBR model) and we increase the number of high rate flows at 10 Mb/s (a *rateFactor* of 50) after each iteration. We see that as we increase the number of high rate flows, both delay and jitter increase linearly for all schemes, with MCTA-based schemes performing significantly better in all cases. The linear increase in jitter is due to the increase in competition for unused time units. Given that there are limited CTAs, flows have to wait for their CTA to occur, perhaps in the next superframe, before they can clear their queue.

From Figure 9 we see that *VBR-TokenBus* has a lower delay and jitter compared to *VBR-MCTA*. However, as the number of high rate flows increase, *VBR-TokenBus* records a higher delay because a flow has to relinquish the ownership of a CTA after clearing its queue or a burst of packets, effectively reducing its transmission opportunities. Therefore, subsequent packet bursts can only be transmitted in a subsequent CTA. In the *VBR-MCTA* case, since a flow retains control of the whole CTA slot without relinquishing it to another device, it is able to transmit the burst of packets that arrive some time before the end of the current CTA.

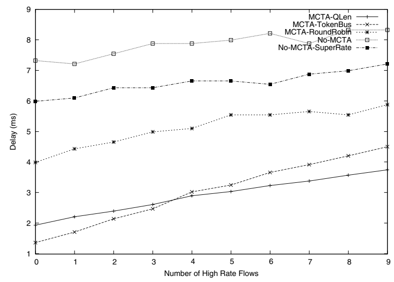

Fig. 9. This shows the effect of high rate flows on flows' delay.

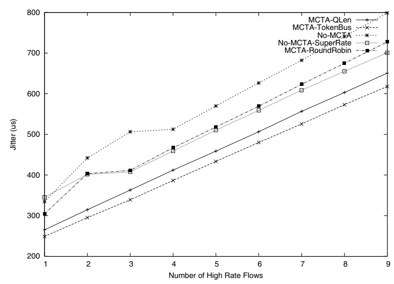

Fig. 10. This shows the effect of high rate flows on flows' jitter.

VII. Related Work

To date there has only been one published work [9] on supporting VBR streams over the IEEE 802.15.3 MAC. Török et al. [9] introduce a hierarchical superframe whereby the PNC transmits mini-beacons in addition to the regular beacon sent at the start of a superframe. The mini-beacons solicit devices' queue sizes and provide additional CTAs for those devices that have asked for extra time in the previous superframe. Our scheme and theirs are similar in that both provide devices with opportunities to inform the PNC that additional time is required. However, our scheme uses a superior method to realize this concept. First, we do not transmit mini-beacons and instead offer devices MCTAs that precede each regular CTA. This approach is less complex and has lower signalling overheads. Second, the additional CTAs are assigned immediately rather than having to wait for the next superframe. Thirdly, our scheme facilitates sharing of CTAs so it reduces the need to adapt a flow's CTA durations and positions. Finally, our solution is compatible with the current IEEE 802.15.3b specification.

Our methods bear resemblance to some previous work in the area of wireless ATM, for examples [10][11] and also those related to the IEEE 802.11e standard [12]. For example, Lee et al. [11] presented a "2-level-scheduling algorithm" where devices are allocated a minimum number of slots, and they may request from the base station (BS) more slots on demand. Dyson et al. [10] extended the packet reservation multiple access (PRMA) [13] protocol to support VBR in addition to CBR traffic. The extensions entail having devices request from the BS multiple slots instead of one slot in the original PRMA. The BS then determines whether the request can be met through tracking of available slots allocated to each device in the network. The above works assume the existence of a contention period which devices use to reserve one or more upcoming transmission slots. Although this is similar in principle to *VBR-MCTA*, existing works have not considered the possibility of relinquishing reserved slots to another device nor the policies for doing so.

In a different work, Ansel et al. [12] presents a scheduling scheme for the IEEE 802.11e. Their scheme entails devices piggybacking their queue length information to the QoS access point (QAP), where the QAP estimates each device's queue length using past reported values, and variations from what was reported by the device and the allocated transmission opportunities. With these information, the AP then schedules the appropriate number of transmission opportunities at each service interval. Although their scheme will allocate the correct amount of transmission opportunities per device the tradeoff however is transmission delay especially when a device requires peak rate allocation. Our approach on the other hand, the PNC can afford to allocate each device peak rate and rely on schemes such as VBR-MCTA to ensure CTAs are used efficiently.

VIII. Conclusion

We have presented a novel CTA sharing mechanism that takes advantage of the IEEE 802.15.3b *StreamGroupID* and Relinquish CTA features, thus requiring no changes to the standard. By allowing CTAs to be shared, flows get additional transmission opportunities, thereby resulting in lower average delay and jitter. Further, by allowing streams to share CTAs a scheduler is offered more flexibility in terms of CTA durations and their positions with respect to meeting video traffic requirements.

From our simulations, we find *VBR-TokenBus* to have the best delay and jitter, followed by *VBR-MCTA* and finally *VBR-Blind*. Although *VBR-Blind* has poorest delay and jitter due to the inability of the PNC to consider devices traffic requirements, it does not consume additional time units or incur any signaling overheads. Therefore, depending on application types, the delay and jitter may still be within their desired bounds.

References

[1] IEEE, "802.15 WPAN task group 3 (TG3)." http://www.ieee802.org/15/pub/TG3.html.

[2] IEEE 802.15.3 Working Group, "Part 15.3: Wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPAN)." IEEE Draft Standard, Draft P802.15.3/D16, Feb. 2003.

[3] IEEE, "Token-passing bus access method and physical layer specifications." IEEE p 802.4/82/40 Project 802 Local Area Network Standards, Draft D, Dec. 1982.

[4] IEEE 802.15.3b Working Group, "Part 15.3b: Wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPAN): Amendment to MAC sublayer." IEEE P802-15-3-DF8 Draft Ammendment, Jan. 2005.

[5] S. McCanne and S. Floyd., "ns network simulator-2." http://www.isi.edu/nsname/ns/.

[6] IEEE 802.15.3 Working Group, "DS-UWB physical layer submission to 802.15 task group 3a." IEEE Working Group Submission, July 2004.

[7] K.-W. Chin and D. Lowe, "A simulation study of the IEEE 802.15.3 MAC," in *Australian Telecommunication Networks and Applications Conference*, (Sydney, Australia), Dec. 2004.

[8] A. Matrawy, "MPEG4 Traffic Model for the ns-2 Simulator." ns-2 Source Distribution. http://www.sce.carleton.ca/ãmatrawy.

[9] L. Vajda, A. Torok, Y. Kyu-Jung, and J. Sun-Do, "Hierarchical superframe formation in 802.15.3 networks," *IEEE ICC*, 2004.

[10] D. A. Dyson and Z. J. Haas, "A dynamic packet reservation multiple access scheme for wireless ATM," *ACM/Balzer Mobile Networks and Applications*, vol. 4, pp. 87–99, May 1999.

[11] S. Lee, Y.-J. Song, D.-H. Cho, Y.-B. Dhong, and J.-W. Yang, "Wireless ATM MAC layer protocol for near optimal quality of service support," in *IEEE GLOBECOM'98*, 1998.

[12] P. Ansel, Q. Ni, and T. Turletti, "An efficient scheduling scheme for IEEE 802.11e," *WiOpt'04: Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks*, 2004.

[13] D. Goodman, R. A. Valenzuela, K. Gayliard, and B. Ramamurthi, "Packet reservation multiple access for local wireless communications," *IEEE Transactions on Communications*, vol. 37, Aug. 1989.