End-to-end Adaptive Multimedia over Bluetooth Scatternets

Manthos Kazantzidis, Andrea Zanella, Mario Gerla
UCLA CS WAM Lab

ABSTRACT: *Bluetooth Scatternet enabled technology today exists only in the standards not the actual chips. Even though the standard defines the primitives to support scatterneting, the architectural options are left open to research. In this paper we use a number of proposed solutions and implemented models to explore application and transport behavior in Bluetooth scatternets. We specifically look at multimedia end-to-end adaptive applications, since this type is expected to be popular in scatternets in personal area network communication as well as other commercial and military applications. We limit our study to a 'trial-and-error' adaptation mechanism, which is based on the simple RTP suggested loss rate observation. Furthermore, since mobility support for Scatternets is minimal today, we only regard cases where mobility does not effectively change the network structure. We take our study one step further by including a strictly, as much as possible, MAC layer comparison between the structured Bluetooth MAC and the totally asynchronous 802.11/DCF MAC, both suggested for infrastructure-less networks. Our study suggests that in Bluetooth even though gateways effectively limit the network capacity at a fraction of the link data rate, closed loop end-to-end adaptation can be effective in controlling congestion and improving user perceived QoS. This is attributed to the very controlled master centric polling MAC layer in combination with the time invariant inter-piconet scheduling architecture we use. On the other hand, the 802.11 limited to Bluetooth characteristics for the shake of our comparison, shows significantly less effective throughput in all cases, and a more inefficient response to adaptive traffic. We studied the traffic behavior of such applications in scatternets of up to 16 piconets and 64 nodes. We conclude that simple end-to-end adaptive closed loop congestion control can be very effective in Bluetooth piconets and scatternets, and more effective than in more asynchronous unstructured MAC layers when mobility is low. The choice between a structured and an asynchronous MAC layer should be made based on how well the more efficient structured MAC can support mobility.*

I. INTRODUCTION

Bluetooth scatternets are essential to a wide range of applications ranging from single personal area networks (PAN) to larger ad hoc networking environments e.g. sensors, large events, conferences etc. The nodes in a Bluetooth network are organized in piconets, a star formation of up to 8 time-slot and frequency synchronized nodes. Larger networks are supported through the designation of nodes that may synchronize to more than one piconets, and therefore may receive and send data between them. Any node may become a gateway if it senses the presence of a foreign master node, through the INQUIRY/PAGING procedure [1]. The gateway nodes then time-divide their presence in the piconets. An interesting situation occurs, where slave nodes in a piconet may exchange data at full bit rates [12], taking advantage of the frequency reuse from frequency hopping on different sequences, while nodes in different piconets may communicate at a fraction of this rate (and increased delays), as defined by the presence intervals of gateway nodes in the path.

Today, the most popular (and available) ad hoc networking option is based on spread spectrum radios operating in the ISM band and using the 802.11 MAC protocol. At the network layer, the favorite routing algorithms are the On-Demand routing algorithms DSR [8] and AODV [13]. The 802.11 technology is dominating the wireless LAN market today. The leading wireless LAN application is point to multipoint Internet access. In 802.11 ad-hoc mode, all the nodes can hear each other, and the interference between different conversations can become overwhelming. Moreover, it is difficult to exercise call acceptance control (on IP telephony, for example) or congestion control. In summary, today 802.11 solution provides low latency of access, no need for a priory network configuration, robustness to mobility. These characteristics are ideal for users willing to exchange quick messages among each other, or to interactively browse the web. On the negative side, there is the problem of congestion and the inability to provide efficient QoS support to real time users. Active research is currently going with the goal of providing some form of congestion control [9], fairness [10] and QoS support [11] in 802.11 ad hoc networks. But these efforts are in the very preliminary research testbed stage and still far from commercialization.

In this paper, we intend to address another, novel wireless LAN technology, Bluetooth, soon to be widely commercialized, which can be viewed in part as the complement, and in part as the competitor of 802.11, depending on the applications. Our focus here will be the ad hoc networking environment and the RTP end-to-end adaptive multimedia applications mentioned above. Namely, we want to examine the efficacy of Bluetooth scatternets in supporting a simple adaptation policy based on loss rate for a VBR (video) stream. We have already worked with this in the Piconet limited environment [2]. There, due to the fact that the connections endpoints are within the Piconet (2 hops maximum), and the medium is very well controlled with the Master polling, the adaptation is very efficient. It quickly reacts to changes and throws in the correct number of video layers. In that scenario, as more Piconets are added the overall system capacity is increased due to the ad-hoc frequency hopping mechanism, reaching capacities comparable to 802.11/DCF based ad-hoc systems. In scatternets, however, the situation may be different due the time division policy at gateway nodes. The gateway naturally operates at a fraction of the link bandwidth, and has different delay characteristics due to absence intervals. The Bluetooth Scatternet architecture we have used is described in [3] and is based on the assignment of rendezvous points. Rendezvous point assignment is an efficient way to define the gateway presence intervals. The rendezvous point is a pre-arranged slot that a master agrees to poll a gateway node, and the gateway node agrees to accept the poll. The master decides after how many slots it will poll the gateway and lets the gateway know. The gateway then arranges to switch back right before that poll slot. In this way poll slots are not wasted. After establishment of a rendezvous point it will be periodically honored for the duration of the connection. Rendezvous points are supported in the Bluetooth specification [1] by using the SNIFF primitive [2]. The establishment of rendezvous points allows the easy development and application of inter-piconet scheduling algorithms (IPS). For example, the algorithm in [4] that picks points that locally optimize the gateway forwarding throughput and avoids the creation of low throughput connections. This is the algorithm we use in our experiments here. We could manually choose globally optimal rendezvous points in our experiments since all traffic and topology was known in advance. But we chose to use this algorithm in order to be closer to reality, where topology and connections is not known in advance, but formed as time passes and gateway connections are established in some order. More details on the above issues can be found [4].

From the above introduction, it is evident that the Bluetooth/Scatternet environment is very different from the conventional 802.11 ad hoc network environment. In this paper we explore how a congestion control algorithm, in the context of multimedia adaptation, may perform in this new environment and how equivalent scenarios behave in the Wavelan environment. What are the performance characteristics of adaptive video in Bluetooth scatternets? Is the congestion control indeed easily handled and is it predictable in Bluetooth scatternets as in piconets? How does the asynchronous 802.11/DCF solution compares to the Bluetooth scatternet MAC with regard to effective throughput?

The paper is organized as follows. In section 2, we describe the simulation models and scenarios used. Whenever necessary we provide background information. After the environment and experiments are explained we go on in III presenting and explaining the results. We conclude in IV.

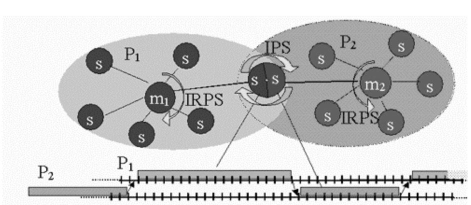

Fig. 1 A scatternet with one inter-piconet unit that divides its time between the two piconets

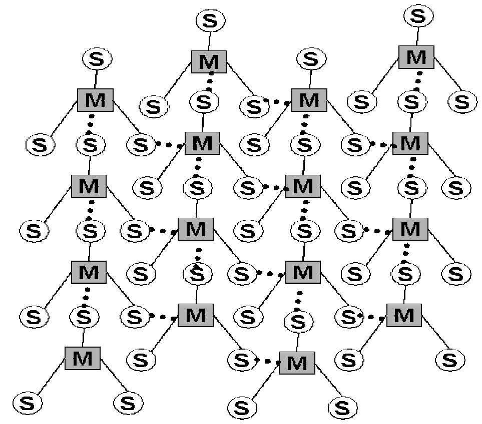

Figure 2. The canonical recursive Bluetooth scatternet topology

II. SIMULATION MODELS

a. Bluetooth Model

We base our study on our Bluetooth model implementation on the NS platform. The model implements most of the features of the Bluetooth baseband layer like Frequency Hopping, Time Division Duplexing, ARQ (Automatic Retransmission Query), Multi-Slot Packets, Fragmentation and Reassembly of Packets. Frequency hopping is modeled as a pure pseudo-random sequence. If two or more transmissions occur on the same frequency, the SIR (Signal-to-Interference Ratio) is evaluated using the gain factor g of each radio channel. The factor g is considered constant during the packet transmission and its value is obtained by considering the path loss due to distance with an exponent between 2 to 4, log-normal Gaussian random shadowing and fading with unity mean. For each portion of the packet of length L in which the SIR is considered constant the information bit error probability is evaluated by taking into consideration the modulation adopted and the FEC coding (if adopted). Details on the channel model can be found in [3]. The polling scheme used in these experiments is pure round robin. Scatternets are supported through gateway nodes exactly as described in the architecture below. In reality, piconets are not be synchronized and this results in the loss of 1 or 2 time slots when a gateway makes a switch between asynchronous piconets. In our model, we do not consider this here for clarity of results. Issues related to the scatternet architecture like “Inquiry” and “Paging” phases [1] and inter-piconet scheduling are presented in par.*d.*

b. Support for Ad-hoc Routing

Ad-hoc routing protocols like AODV rely on a broadcast at the MAC layer to propagate the routing packets. In Bluetooth, packets can be broadcast inside a single piconet using the piconet broadcast address of –1 [1]. In order to support broadcast of routing packets to outside of a piconet, we define the rules: (i) If a non-gateway slave node needs to broadcast a packet, it sends the packet to the Master of the Piconet it belongs to.(ii) If a Master node needs to broadcast a packet, it broadcasts the packet to all the non-gateway slaves in its piconet using the piconet broadcast address of -1 and unicasts the packet to each of the Gateway slaves when they visit the piconet. A Gateway slave node wanting to broadcast a packet sends it in a unicast manner to the Master of each piconet it belongs to when it visits the piconet.

c. Topology & Traffic

In this paper's experiment we will use the canonical recursive scatternet topology shown in Figure 2. Each piconet (e.g. person in the PAN paradigm) has a master and three slaves, as distant as possible. If the slaves are numbered starting at the top and clockwise, then slave 0 may connect to the piconet on top, and slave 1 may connect to the piconet to the right, while slave 2 will not become a gateway. This recursively may result in piconets with up to 5 slaves, 4 of which gateways. In our experiments we give an address (IP or MAC) to each node starting at the piconet to the left and bottom and continuing in rows. The Master is assigned first and then the slaves clockwise starting at the top.

We experiment with up to 16 piconets, in 1 piconet, 2 by 2, 3 by 3 and 4 by 4 configurations. The connections we throw in go vertically and horizontally. In the 3 by 3 case for example we have 6 connections. Our application is adaptive or non-adaptive VBR video based on real traces gathered from the Star Wars IV trailer clip, encoded in Mpeg of 256, 128, 80, 64 and 48 Kbps average target rates. The smoothing process is limited to one frame time and uniformly spreads the frame bytes into apprx. 220 byte payloads into the inter-frame interval. The adaptation is performed through monitoring of loss rates at the client from sequence numbers. The monitoring and the reporting goes on in 1 second intervals and according to RTP. The QoS feedback packet, has to traverse the network back to the server. We use a UDP based protocol so that we can implement delay and timeout algorithms. The timeout we use is employed at the server. When a few QoS packets are lost the server assumes a congested reverse path and will react pessimistically lowering the rates. In our implementation whenever we detect *TOTimesThreshold* lost QoS packets every *TOMaxTicks* (5 and 7 respectively). Hysterisis is also implemented, having different low and high loss thresholds (5 and 20%).

d. Scatternets and Inter-Piconet Scheduling

In this paragraph we introduce the inter-piconet scheduling we use based on the locally maxmin forwarding throughput optimal rendezvous assignment algorithm. We use this instead of assuming a perfect inter-piconet gateway presence synchronization. Perfect inter-piconet gateway synchronization is not possible, since gateways are in reality established one after another, without possibly having the future, whole picture. Details can be found in [4].

A slave of piconet *i*, that recognizes another master's (piconet *j*) INQUIRY, may become a gateway between its home piconet and the visiting piconet. There may be other gateways already present in the home piconet as well as in the visiting piconet. Also, the prospective gateway node may already be a gateway (i.e. connecting its home piconet with some other piconets). Therefore, the parameters that affect the forwarding throughput (FT) of the gateway are the existing rendezvous point in home, visiting piconet and gateway. Note that a rendezvous point is the number of slots after which the master will poll. In order to use absolute numbers to refer to rendezvous points (rather than relative to the establishment time) we also introduce the notion of the superframe cycle (measured in slots). A superframe cycle defines the period after which the rendezvous point will re-occur. Therefore all operations are assumed to be in modulo superframe arithmetic. Note that the superframe notion is logical only, and does not imply any piconet synchronization. The possible forwarding throughput from one piconet to another based on the RV points of its gateways. An intuitive depiction of the situation is attempted in Fig 3. It is defined by the presence interval of the gateway as a fraction of the superframe, and the number of other slaves and gateways that it has to share the date rate during that time. The algorithm we use calculates the possible FTs between the visiting piconet and all piconets it connects to, as if the proposed connection had already been established. For each possible rendezvous slot it looks at the minimum among the FTs. It then keeps the point that results in the maximum of the minimum FTs. For details and equations in this complex problem please see [4].

We have modeled in C++ a set of objects that describe the nodes, piconets and gateways and interface both with NS (see Fig. 4). We define an event as a new node that is to become a gateway between a new pair of Piconets (home Piconet and some other one). The list of events is sorted based on a random *eventId*. This simulates the Beaconing process described in [1], as it is random which node pair becomes aware of their proximity first. Then the RVs are calculated based on the algorithm proposed.

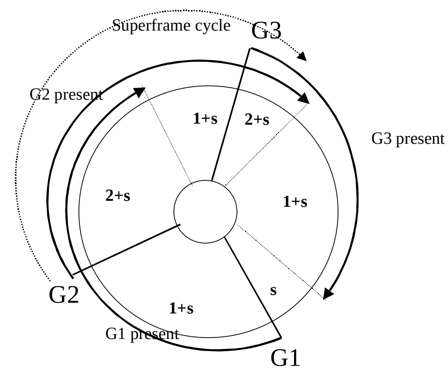

Fig 3. The number of nodes active in the polling cycle during the superframe cycle of a piconet with s non-gateway slaves and 3 gateway nodes.

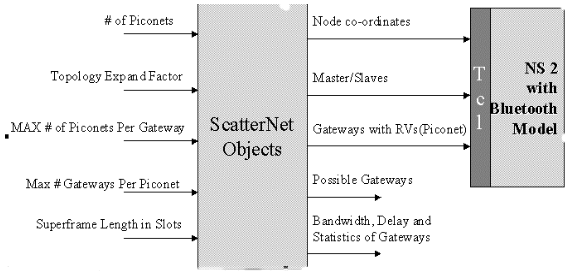

Fig. 4. Scatternet Simulator and Interface with NS and GlomoSim Bluetooth model

III. RESULTS

In this section we are presenting the results obtained with the previously described simulation models and scenarios. First for the 16 piconet case and the 8, 4-piconet hop connections we show the server throughput and the goodput (client received throughput) in the case of adaptive and non-adaptive stream and in Bluetooth and 802.11.

In the latter case the 8, 256Kbps average coding rate connections do not even need to adapt as no loss rate occurs (in Wavelan at 11Mbps with the 100m range all nodes are within a single hop). In Bluetooth the connections need to adapt. Two connections are carried on each middle piconet and forwarded to the edge piconets by the gateways in the middle. The gateways can forward at half the rate ideally and looking at the topology we see that piconets in the middle have to forward two streams. As bottleneck links, they define the received throughput to be 25% of the link data rate, with ideal IPS. Two questions remain to be answered in order to understand the rates achieved in the graphs. One, how much is the link data rate? It is well known that when connections are two-way the effective data rate in Bluetooth is *340bytes * 8 / .625msec*5 slots= 870.4Kbps.* Two, how close to the ideal IPS case are we? Because of the normality of our topology (same number of slaves per piconets) the max-min optimal IPS we used will almost ideally assigned rendezvous points. Looking at the rendezvous points these are all indeed from 0-4, 24-26, 50-54 and 74-76 which are quite close to the optimal 0,25,50 and 75 for our topology. (If we wished to calculate this we would do the following, looking at the topology and the presence intervals of the gateways: for 4 slots, 1 more gateway will be present than in the optimal case. This is a 4*slots*/100*slots* that will be divided among 3 gateways instead of 2, so this inefficiency can be roughly calculated at *2.7%* lost throughput.) There is also one slave present that wastes two slots per polling round in the bottleneck middle piconets. This at least another *2/(2+40) = 4.8%* decrease (assuming always perfect 5 slot packets both ways case). The above means that if our connections were CBR they would ideally each get at most 201Kbps (92.5% of 217Kbps). There are more wasted slots because when gateways are close to switching they cannot fit the packets to the available number of slots. The VBR shape and the packet size of the traffic causes a significant amount of the aforementioned fragmentation. As mentioned, our video smoothing results in many packets around 210 bytes. The 5-slot packets (full utilization 870Kbps case) correspond to 340 bytes payload. In fact our packet size happens to be a ‘bad’ size for the slotted Mac since it needs a 3-slot packet and many packets have a remainder of few bytes. Also in the few first seconds of the 36 of the connections the buffer utilization is not full. These cause the lower throughput, in the orders of 125-175Kbps, seen here.

The important result, however, from these experiments is that the simple end-to-end adaptation worked very efficiently and achieved only slightly lower rates. If we compare the goodput of non-adaptive connections and the server throughput of the adaptive connection we see that they are almost equal! Furthermore, the loss rates are less than 3% in all cases, which means that the adaptation was also timely.

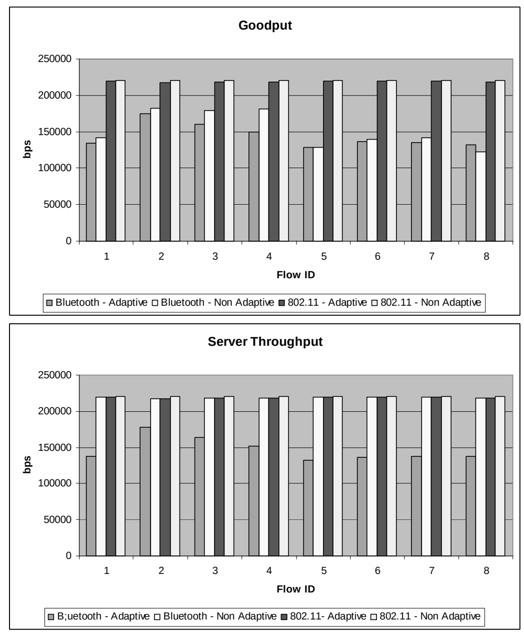

Figure 5. Per Connection Server Send Throughput and Goodput for Bluetooth and 802.11, adaptive and non-adaptive for the 4 by 4 piconet case.

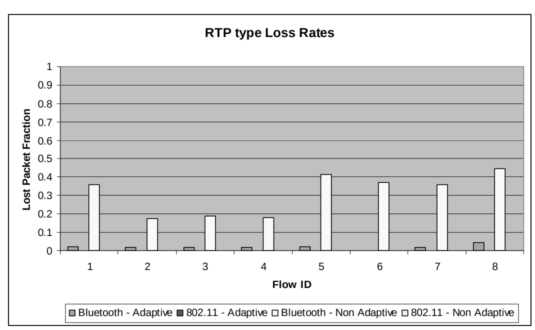

Figure 6. The loss rates for the cases of figure 5

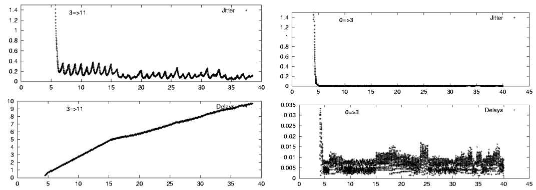

Figure 7 Jitter and delays for a typical connection. Left: from the 3 by 3 piconets case, Right: From the single piconet case.

a. Non Adaptive Bluetooth

In this paragraph we look closer at a connection in the 3 by 3 non-adaptive case. As we calculated above the forwarding throughput of the gateways is less than the (256Kbps) connection rate and therefore, the link layer buffers get filled and delay is increasing up to the point that the link layers will be full and start dropping packets (we have experimented with large link buffers in order to observe delays). When the same connection does not go through gateways (source and destination are within the same piconet) its rate is acceptable. The slave to slave delays seen range from 1.2 msec to 16 msec or 2 to 26 slots.

b. Adaptive Bluetooth

In this paragraph we look at the experiments that the video connections adapt. We look at typical connections (remember our topology and traffic is almost symmetrical so all connections are typical) from 2 by2, 3 by 3 and 4 by 4 piconets case. Besides delays now we also look at the reported loss rates, the layer from which each frame is picked and transmitted and the received rate averaged over 1 sec.

The first thing we note is that the delays and the delay jitters are determined by the link layer buffering rather than the superframe size and IPS related parameters. This is because they have similar shape and magnitude in all cases.

The second notable thing is the very low frequency oscillatory nature of the delays. Because the connections operate at capacity the receiving rates are not affected by our oscillation in choosing layer. Our simple control, even if it implements hysterisis (different low and high loss thresholds) cannot avoid oscillations. As the loss rate is low a higher rate will be attempted. This fills up the buffers until packets are dropped. Since inter-piconet interference is not significant in our experiments it does not affect loss rates. Packets are dropped from the link layer at the gateways. The application notices the drop and lowers its sending rate. Following that, delays for subsequent packets start falling, as the queues are emptied in the master round robin. This indicates that the closed control system has a strong and immediate impact on the scatternet. By time invariant, we refer to the fact that the rendezvous points do not change (and are not random) in our architecture.

c. 802.11

In the 11Mbps, 100 meters 802.11/DCF case all nodes (64 in the 16 piconet, 4 by 4 case) connections are within one hop. 8 connections of 256Kbps on average, need 2 Mbps and therefore adaptation is not issue in 802.11 when the exact same scenario is tried. We have studied adaptation in 802.11, end-to-end in [7], and using MAC level, AODV propagated bandwidth measurements as application feedback in [5]. We have found that even though end-to-end adaptation is mostly effective and lowers loss rates, due to the large end-to-end variance in response it can easily become unstable. End-to-end measurements are also not accurate in presence of hidden terminals. MAC and network feedback based on link by link measurements is therefore suggested there. This is our motivation to compare the two systems on how well they allow a scalable end-to-end transport architecture.

Here, we merely want to look at the 802.11 equivalent of our Bluetooth experiment. In keeping with our target, we only look at traffic, not power consumption issues, which are obviously very different in the two wireless technologies. Due to the higher capacity and the single hop situation, 802.11 does not even have hidden terminal problems in the experiments Bluetooth had multiple hops, gateway fractioned bandwidth and needed adaptation. The delays are also only due to the virtual carrier sense, rarely reaching a back-off situation. Therefore, as the traffic is more, in the higher node/connection number experiments delays are proportionally higher. We can see that all delays are lower than 70 *msec*, 35 *msec*, 12 *msec* and 2 *msec* in the 64, 32, 16 and 4 node experiments (which actually correspond to the exact Bluetooth topology of 4 by 4, 3 by 3, 2 by 2 and 1 piconet arrangement studied here).

d. Comparison with 802.11 MAC

In this section we are attempting a direct comparison of the 802.11 DCF MAC layer with the Bluetooth MAC layer. This comparison is interesting because these two MACs are intended to work both in infrastructure-less environments and represent two essentially different approaches. The former is a totally asynchronous approach, which uses physical and virtual carrier sense and collision avoidance techniques relying on a common code or frequency hopping pattern physical layer. The latter is a structured synchronous polling MAC layer. It is natural that in face of mobility 802.11/DCF, not requiring on any structure to define its spreading codes or hopping patterns, will have an a-priori advantage over Bluetooth. Since the mechanisms under which Bluetooth is going to support mobility are not yet researched and developed (again, the functions are defined in the standard but their efficient use is a new research problem) we will look at the case without mobility: Given a network topology and adaptive traffic which of the two MAC approaches are more efficient? Since in this section we wish to focus on the MAC layer mechanics directly, we are equalizing physical layer variables. Namely, we lower the 802.11 bit rate to the 1Mbps Bluetooth bit rate, and adjust the radio power to effect a maximum of 10meters range.

The topology we use in this section is still the same (Figure 2), only now the 802.11 case requires as many wireless hops as Bluetooth and has to deal with hidden terminals in its collision avoidance functionality. We use 4 and 9 piconets, which is 16 and 36 nodes respectively. The traffic we use is CBR, instead of VBR so that the results are clearer. We progressively (in different experiments) throw in the network more connections, up to the number of nodes in number. The connections are either adaptive in 7 layers (180, 128, 64, 32, 16, 8 and 3 Kbps) or non-adaptive at 180Kbps. We use connections that follow an even hop distribution; in Bluetooth terms for example in the 9 piconet case, this is one/third within the same piconet, one/third within neighboring piconets and one/third with 2 piconet distance. The same in 802.11 is 2 hops, 4 hops and 6 hop connections.

Figure 10 summarizes these experiments by showing averages over all connections in each experiment. Bluetooth is able to deliver the most throughput to its connections in the larger network and larger connection numbers case. 50Kbps are delivered on the worst case average with Bluetooth, while 802.11 is deteriorating its effective throughput as more connections are added. Up to a certain point (14 connections on 16 nodes and 18 connections on 36 nodes) the adaptation works positively in 802.11 as denoted from the decreasing server consumed throughputs. After that adding more connections, creates reverse path problems and increased delays which construed the closed-loop adaptation. In Bluetooth this is significantly less obvious. In both cases, adaptation achieved decreased loss rates but it did not always manage to stabilize the network into low loss rate operation. Note however that the loss rates reported in the figure are averaged over all connections and for the lifetime of the connections (which is 30 secs) and therefore a large portion of them are due to the high loss rates before adaptation could converge to acceptable rates. This initialization time is quite large, as denoted also by the delay graphs. The average delays where high in both MAC cases but much higher and increasing more rapidly in the 802.11 case. This is due to the nature of delay in the asynchronous MAC versus the nature of delay in polling MAC. In the former the more packets thrown in the network, the higher the virtual carrier sense and back-off timers for each MAC re-transmission. In the latter, due to the non-exhaustive polling and the periodic time division multiplexing in gateways the delay maintains a value defined by the link buffers and the superframe. Therefore, adaptation decreased the high delays in the 802.11 case.

IV. CONCLUSION

We have experimented with issues of congestion control in Bluetooth scatternets. We have used a scatternet architecture based on rendezvous points, and advanced inter-piconet scheduling algorithms to build a state-of-the art Bluetooth ad-hoc network, and used realistic models to experiment with it. We looked at the congestion control as performed by an end-to-end RTP based adaptive video application. Bottleneck points are necessarily created in the scatternet where gateway nodes divide their time to different piconets. They operate at a fraction of the link data rate that determines the effective allowable rate of end-to-end connections. Buffers at these points should be higher as packets are gathered there and waiting for gateway switches. In fact, end-to-end delays are mostly defined by these link queuing delays. We discussed the issues that limit the end-to-end goodputs in such an environment and studied the dynamic adaptation. As was known to happen in piconets, in scatternets too, when using static window and static rendezvous points the control medium lends itself to a very efficient and timely adaptation. The adaptive connections managed to achieve the goodputs that non-adaptive connections achieved but at less than 3% loss rates.

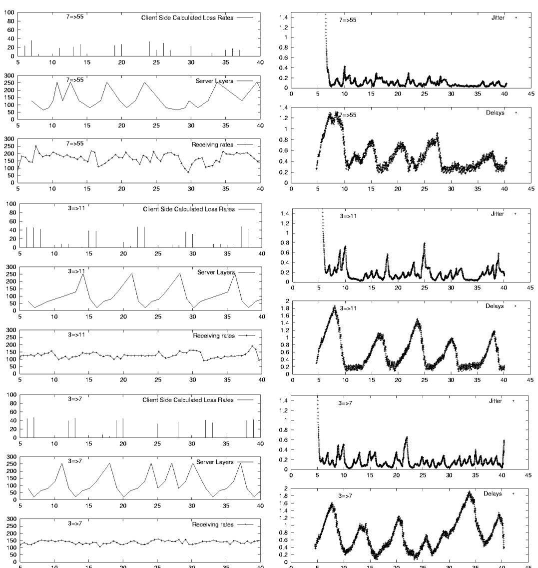

Figure 8. Adaptive connections in the (top) 4 by 4, (middle) 3 by 3 and (bottom) 2 by 2case. Left column shows (top) reported RTP loss rates (middle) layer from which frame is picked for transmission (bottom) received rate averaged over 1 second. Right column shows jitter and delay.

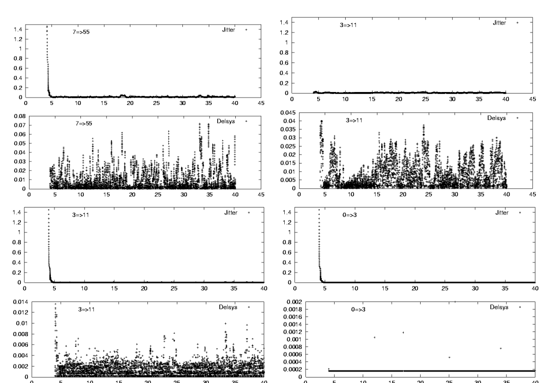

Figure 9. The corresponding 802.11/DCF exact topology cases (top, left) 64 node, eq. to 4 by 4 (top,right) 36 node, eq. to 3 by 3 piconets (bottom, left) 16 node, eq. to 2 by 2 and (bottom, right) 4 nodes, eq. to 1 piconet Bluetooth case.

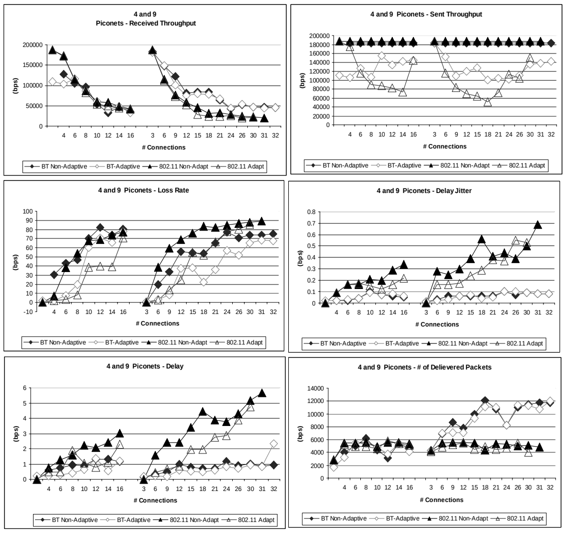

Figure 10. Direct Comparison of Bluetooth and 802.11 MAC using an end-to-end adaptation mechanism. (top) Average Recv throughput, Sent Throughput (middle) Loss rates, Delay Jitter (bottom) Delay, Packets Delivered.

Comparing to an 802.11, 11Mbps ad-hoc scenario, we noted the big difference in capacity of the two technologies. When only piconets are used, say as Internet connection clients, the network capacity is comparable to the 11Mbps 802.11 case. This is due to the ad-hoc frequency hopping pattern exchange that effectively adds capacity (at Bluetooth data rate increments) as more piconets are introduced. However, in the scatternet case this is not true, because the capacity of the network is now defined by the fraction of the Bluetooth data rate at the gateway nodes. In general, from our experiments it is shown that end-to-end adaptation works very well in scatternets. Generalizing, this indicates that congestion and admission control, measurements and delays may be efficiently handled in the controlled Bluetooth medium. We took our comparison further and looked at how the 802.11/DCF (CSMA/CA with virtual carrier sense, ACK and RTS/CTS) asynchronous MAC compares with the structured poll controlled Bluetooth MAC in terms of effective throughput. We showed that the Bluetooth MAC is offers significantly more effective throughput as well as uses the throughput much more efficiently through a simple adaptation procedure than the asynchronous MAC. Therefore, the trade-off between the two types of MAC should be determined by how efficiently a structured poll controlled MAC can actually maintain and support its structure through mobility without significant overhead. In conclusion, the Bluetooth MAC offers a higher ratio of effective to link throughput while adaptive transports and applications may monitor better and perform their function better providing for low loss rates and efficient QoS adaptation.

REFERENCES

[1] “Specification of the Bluetooth System”

[2] Mario Gerla, Rohit Kapoor, Manthos Kazantzidis (UCLA), Per Johansson (Ericsson) “Ad hoc Networking with Bluetooth” *WMI at Mobicom 2001*

[3] Per Johansson (Ericsson), YZ lee, Mario Gerla, Manthos Kazantzidis (UCLA) – “Bluetooth an Enabler of Personal Area Networking” - *IEEE Network, Special Issue in Personal Area Networks, Sept-Oct 2001*

[4] M. Kazantzidis, *Locally optimal forwarding throughput Bluetooth scatternets– UCLA CS Technical report #010033*

[5] M. Kazantzidis, End-to-end versus Explicit Feedback Measurement in 802.11 Networks UCLA CS TECHNICAL REPORT # 010034

[6] T. Salonidis, P. Bhagwat, L. Tassiulas, R. La Maire “Distributed topology construction of Bluetooth personal area networks.” *Infocom 2001*

[7] M Kazantzidis, L Wang, M Gerla “On Fairness and Efficiency of Adaptive Multimedia in MANETs” – IEEE MOMUC 1999

[8] Johnson, D.B., and Maltz, D.A., "Dynamic Source Routing in Ad-Hoc Wireless Networks," in Mobile Computing, edited by T. Imielinski and H. Korth, chapter 5, pp. 153-181, Kluwer, 1996

[9] Fair Sharing of MAC under TCP in Wireless Ad Hoc Networks.Ken Tang and Mario Gerla.Proceedings of IEEE MMT'99, Venice, Italy, October 1999.

[10] A New Model for Packet Scheduling in Multihop Wireless Networks Haiyun Luo, Songwu Lu (UCLA, USA) and Vaduvur Bharghavan (University of Illinois at Urbana-Champaign, USA)

[11] C. R. Lin and J.-S. Liu, "QoS Routing in Ad Hoc Wireless Networks", IEEE Journal on Selected Areas in Communications, V. 17, No. 8, pp. 1426-1438, Aug. 1999.

[12] Kalia, M.; Bansal, D.; Shorey, R. “MAC scheduling and SAR policies for Bluetooth: a master driven TDD pico-cellular wireless system.” *1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99)*

[13] Ad-hoc on-demand distance vector routing**,** Perkins, C.E.; Royer, E.M. Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on , 1999 , Page(s): 90 –100.