Ad hoc networking with Bluetooth

Mario Gerla, Rohit Kapoor, Manthos Kazantzidis (UCLA)
Per Johansson (Ericsson)

Abstract

In this paper we explore the ability to support multimedia traffic in indoor, wireless ad hoc PANs (Personal Area Networks) using the Bluetooth technology. We first define the representative ad hoc networking applications such as wireless access to the Internet, document distribution, videoconferencing, webcasting, interaction with sensors and actuators, etc. For such applications, we define the performance requirements placed on the PAN. There are two technologies now competing for the PAN market: the IEEE802.11 "legacy" technology, and the newly introduced Bluetooth technology. By IEEE802.11, we refer to the operation of 802.11 in the DCF mode, which is the mode implemented in the commercial WaveLAN cards. In the rest of the paper, we will use the term WaveLAN to refer to 802.11 in its DCF mode. We will attempt to answer the questions: how effective is the Bluetooth technology in supporting collaborative, "virtual ad hoc networking" applications and how does it compare with WaveLAN? To answer these questions, we have developed an NS-2 model of Bluetooth. We have also developed models of adaptive applications such as voice and video. For WaveLAN, we have used the existing NS-2 models. The results show that Bluetooth provides better support for real-time applications as compared to WaveLAN. It does not exhibit the "capture" behavior observed, for example, in WaveLAN. Also, with the addition of nodes to the "indoor" space, it adds to the total "system" capacity and gives a better overall throughput.

1. Ad hoc networking and Personal Area Networks (PANs)

With the increasing dependence on the Internet in many aspects of their daily lives, users demand ubiquitous, high performance Internet access whether they are at work, at home, or on the move. Moreover, users on the move are often interested in forming "ad hoc" networks to collaborate with colleagues at conferences, or more generally to interconnect all their personal devices. This type of network, which is centered on the individual himself, is often called the **Personal Area Network** (PAN).

The PAN is defined as the collection of devices carried by a mobile, networked individual (e.g., a professional on the move, an Internet-wise tourist, a student attending "virtual classes", an avid Internet game player, etc). The devices include any subset of: cell phone, laptop, earphones, GPS navigator, palm pilot, beeper, portable scanner, etc. These devices form his/her PAN (also known as personal "bubble"). The connectivity within the bubble is wireless (using for example a low cost, low transmit power wireless LAN such as Bluetooth). The bubble can expand and contract dynamically depending on needs. The bubble may connect to wall repeaters for **access to the Internet**. It may also be dynamically stretched to include **access to sensors and actuators**. Such access is critical when the "nomad" walks into a new environment and wants to quickly become aware of what is going on, or wants to control temperature, adjust the lighting, select a particular background music etc. In some cases, the nomad himself carries sensors as part of his PAN: for example, a patient may walk around in the hospital or nursing home with several monitors which transmit to repeaters on the walls, allowing customized 24 hour monitoring.

The PAN communication infrastructure should enable efficient support of the above ad hoc networking scenarios. In essence, we need a self-configuring communications infrastructure which can: (a) provide efficient multimedia access from the PAN to the Internet; (b) permit communications with various classes of sensor/actuators, and; (c) enable voice/data intra and inter-PAN networking. The key challenges in the design of the PAN protocol architecture are: (a) the design of middleware and adaptive application protocols that provide smooth transition between different bandwidth, connectivity and mobility configurations, and; (b) the implementation of PAN MAC and network layer protocols and their interconnection with existing public (wired and wireless) network infrastructures.

In the **single PAN environment**, where nodes are all within transmission range of each other, key issues are (1) MAC protocol selection, to provide efficient transport of TCP/IP traffic and at the same time satisfy multimedia traffic requirements; (2) efficient handoff; (3) mobile/cellular IP support; and (4) end to end adaptivity, possibly via proxy agents. When **communicating with sensors**, the PAN MAC and network layer protocols must operate in a connectionless, low latency and low overhead mode. In this paper, we focus on the "single PAN environment" operation of the PAN's, where communication occurs only within a PAN, and evaluate the support of multimedia in such an environment.

2. The scope of this study

The complete PAN architecture design is a very ambitious project and it is clearly beyond the scope of our study. In this study, we will assume that each PAN corresponds to a single user and consists of a portable device (eg, laptop, PDA, etc.). We will limit ourselves to a key application of the PAN, namely, the interconnection of PANs in virtual ad hoc networks. For simplicity, we will assume that within a virtual ad hoc network all users can hear each other, i.e., fully interconnected virtual topology and single hop communications.

In this simplified, single hop setting the performance of the network will be for the most part determined by the MAC layer. Currently, there are two leading candidates for such role: (a) the IEEE 802.11 MAC protocol and (b) the Bluetooth MAC protocol [9]. The IEEE 802.11 protocol is a rather sophisticated protocol that includes a fairly broad range of options. In particular, it includes the PCF (Point Coordination Function) mode which permits a "base station" to poll various terminal in a cellular-type environment. It also includes the DCF (Distributed Coordination Function) mode, which supports peer-to-peer, ad hoc type communications. The DCF version is a random access protocol similar to CSMA, with the addition of RTS and CTS (for collision avoidance) and of an ACK returned by the receiver after successful transmission. In our study we will assume the use of the DCF mode, which is the mode implemented in the WaveLAN cards (even for infrastructure configurations).

A couple of years ago a new MAC protocol was proposed as part of the Bluetooth PAN architecture. The Bluetooth MAC protocol is a major departure from the IEEE802.11 protocol. To start with, it uses Frequency Hopping with separate frequencies chosen dynamically for each Piconet rather than Direct Sequence Spread Spectrum or configuration based Frequency Hopping, thus exhibiting better protection from co-channel interference. Secondly, it uses time/slot synchronization. Moreover, it uses a polling type scheme to allow a "master" to poll the "slaves" in a given cluster. Bluetooth is expected to become very popular due to its low cost (in the order of a few dollars per interface). The details of the Bluetooth protocol are provided in the next section. Here, it suffices to say that the enormous commercial interest in these two PAN candidates and at the same time their markedly different characteristics warrants an in depth comparison of their performance in various realistic indoor scenarios.

In our simulation experiments we have recreated scenarios that are typical of indoor ad hoc networking. We will consider a mixed traffic environment, both with data (TCP) and with voice/video streaming (with fixed and adaptive rate). We will be interested in the throughput and delay measures, and in the fairness behavior exhibited by the two schemes. The simulation results will be reported in Sect 4. In the next section we first introduce the Bluetooth architecture and protocols.

3. Bluetooth technology overview

The Bluetooth system operates in the worldwide unlicensed 2.4 GHz Industrial-Scientific-Medical (ISM) frequency band. To make the link robust to interference, it employs a Frequency Hopping (FH) technique, in which the carrier frequency is changed at every packet transmission. To minimize complexity and to reduce the cost of the transceiver, a simple binary Gaussian frequency shift keying modulation is adopted. In order to allow efficient wideband data transmission the bit rate is 1 Mbit/s.

Two or more Bluetooth units sharing the same channel form a piconet, see Fig.1(a).

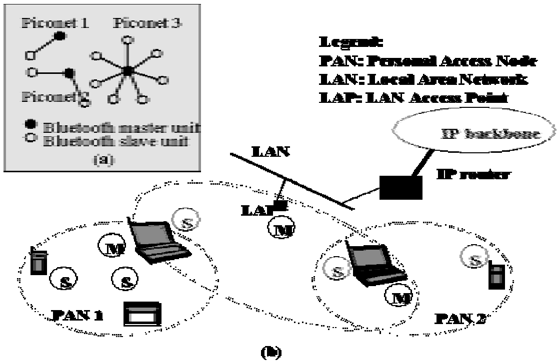


Figure 1: (a)Bluetooth Piconet (b) Bluetooth Scatternet

Within a piconet a Bluetooth unit can be either master or slave. Within each piconet there may be only one master (and there must always be one) and up to seven active slaves. Any Bluetooth unit can become a master in a piconet. Furthermore, two or more piconets can be interconnected, forming what is called a scatternet, see Fig.1(b). The connection point between two piconets consists of a Bluetooth unit that is a member of both piconets. A Bluetooth unit can simultaneously be a slave member of multiple piconets, but a master in only one, and can only transmit and receive data in one piconet at a time, so participation in multiple piconets has to be on a time division multiplex basis.

The Bluetooth system provides full-duplex transmission using a slotted time division duplex (TDD) scheme where each slot is 0.625 ms long. Master-to-slave transmissions always start in an even-numbered time slot, while slave-to-master transmissions always start in an odd-numbered time slot. An even-numbered time slot and its subsequent odd-numbered time slot together are called a frame. There is no direct transmission between slaves in a Bluetooth piconet; transmission is only between a master and a slave, and vice versa.

The communication within a piconet is organized such that the master polls each slave. A slave is only allowed to transmit after the master has polled it. The slave will then start its transmission in the slave-to-master time slot immediately following the packet received from the master.

Each Bluetooth unit has a globally unique 48-bit IEEE 802 address. This address is permanently assigned when the unit is manufactured. In addition to this, the master of a piconet assigns a local active member address (AM ADDR) to each active member of the piconet. The AM ADDR is three bits long, is dynamically assigned and reassigned, and is unique only within a single piconet. The master uses the AM ADDR when polling a slave in a piconet.

Bluetooth packets can carry either synchronous data on synchronous connection oriented (SCO) links mainly intended for voice traffic, or asynchronous data on asynchronous connection-less (ACL) links. To ensure reliable transfer of data, a fast acknowledgment and retransmission scheme is used, only for ACL links. In addition, a forward error correction (FEC) scheme may be used to further improve reliable packet transmission.

4. Case studies and Simulation results

In this Section we present simulation results based on a set of representative traffic scenarios. One of the main goals was to evaluate achievable Bluetooth throughput taking into account interference between different coexisting piconets. The simulation environment used in our experiments is NS-2 [5]. NS-2 already includes several wireless network models. In particular, it supports the IEEE 802.11 WaveLAN standard. We have augmented NS-2 with the Bluetooth model. The Bluetooth model has support for defining multiple piconets which may overlap with each other causing interference. The model contains most of the standard features of Bluetooth like Frequency Hopping, Multi-Slot Packets, Fast ARQ (Automatic Retransmission Query). It also contains a channel and collision model for an indoor environment.

4.1 Conference Hall Case Study

Our aim is to compare the performance of Bluetooth and WaveLAN in a totally adhoc environment, where no infrastructure in the form of access points is available. This would typically model the scenario of a large conference, where a number of Bluetooth or WaveLAN devices may be talking to each other. The traffic in such a scenario is heterogeneous and multimedia in nature, i.e., TCP, voice and video. It is assumed here that any two devices wanting to communicate are close enough to be in the same piconet and thus communicate through the master. This will be a realistic model for ad-hoc group collaboration where members of the same team will be sitting nearby and will interact with each other by exchanging files and engaging in videoconference
.

In the experiment, we consider a 50m * 100m room, in which nodes are distributed according to a uniform random distribution. In the case of Bluetooth, piconets are formed by clustering the nodes close enough to each other. The number of slaves present in each piconet is chosen randomly. Also, some piconets overlap with each other incurring a certain fraction of collisions. The traffic consists of a mix of TCP, Voice and Video. The TCP data connections are always active large file backlogs, with 500-byte packets. The voice connections are modeled according to the Brady model [2]. In particular, the voice connections are "on-off" sources. The on and off times are exponentially distributed, with mean 1 s and 1.35 s respectively. The voice coding rate is 8 kbit/s and the packetisation period is 20 ms, which gives a payload size of 20 bytes. Header compression is assumed for voice packets in Bluetooth and the total packet size is 30 bytes. Voice packets are sent using RTP over UDP. Each experiment lasts 32 seconds of simulation time. In order to probe the sensitivity of performance to population size and to the number of simultaneous connections, we perform different experiments choosing different values of number of nodes and connections.

The slave polling strategy in Bluetooth is of our own creation [3]. It tries to assign slots to slaves based on their traffic history and activity. The topology is totally static, which means that nodes are not mobile and piconets are set up at the beginning of the simulation and do not dynamically change. Again, it is important to note that connections are only 1 or 2-hops, as in intra-piconet communication. No inter-piconet communication takes place.

4.2 Video Traffic

The traffic consists of a mix of video, voice and TCP. The video sources are represented by real traces. We use the Star Wars trailer clip encoded using Intel's H.263 compatible codec. The traces have been smoothed using a simple technique, namely a frame as returned by the codec is distributed uniformly in time within the frame interval using a target of 200 byte packets. There is no other smaller time scale transport mechanism and the generated packets are simply sent through the network with UDP. A few seconds from the resulting sources for the codec is shown in Fig 2. A description of the framework used in the experiments can be found in [4].

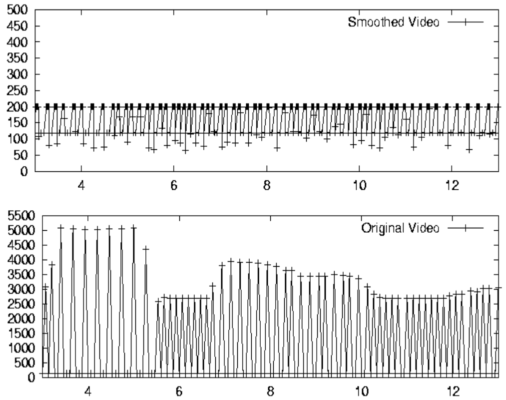


Figure 2: A few seconds from the H263 source trace (sec, bytes)

We investigate adaptive as well as non-adaptive video streaming. The former uses average rates of 48, 64, 80, 128 and 256Kbps for the two codecs, while the non-adaptive cases use the 256Kbps trace. The adaptation mechanism is based on an end-to-end, periodic (1 sec) feedback that contains the number of packets received during the feedback interval. This feedback is used by the server to compute the RTP loss rates. The server then changes its rate using a min/max loss threshold. Below the minimum packet loss rate (5%) the server attempts to additively increase its rate. When the loss rate is above the maximum loss threshold (15%) the server reduces the sending rate, choosing a rate among the 5 available rates that is appropriate to support the reported loss rate. For example, if the current rate is 128Kbps and the loss rate is 50% the sending rate drops to 64Kbps.

The following graphs (Fig 3(a) and (b)) illustrate this behavior in a random 30-node 60-connection experiment. This initial experiment targets at showing the adaptive behavior with the two MAC protocols. The scenario is generated as mentioned above, but now 90% of the voice, video and TCP connections are created at 8.6s and finish at the 16.6s. The goal is to study the adaptive behavior of a video connection that lives throughout the experiment (i.e. from 0s to 32s). As the feedback indicates, the server downgrades the sending rate or attempts a higher rate. We show the loss rates as calculated when a feedback packet is received, the per-packet delays and the server selected average rates for the two cases. First, we note that when the additional connections enter the network (from 8.6s to 16.6s) in WaveLAN, the video connection downgrades to the lower possible transmission rate because the loss feedback goes beyond the threshold. On the other hand, in Bluetooth the loss rates are lower, the transmission rates remain higher and the downgrading is in all cases gradual (one layer at a time). These indicate that the network response is more regular allowing for efficient feedback control with less oscillations. This is true not only for the connection shown but for the other competing adaptive connections as well. It is interesting to note that in the congested network, less packets get lost in the Bluetooth case, but their delay is significantly increased, mainly due to the creation of longer link queues. Since in this experiment we are using a loss feedback this is exactly what the network is expected to do, send as much as possible by trading off bandwidth with increased delay. In WaveLAN the delay experienced by packets is doubled immediately when the new connections enter and does not increase gradually as in Bluetooth. This is because in WaveLAN the connections have downgraded to the lowest rate and the 10Mbps bandwidth does not allow the link queues to fill up. Instead, packets are dropped due to collisions.

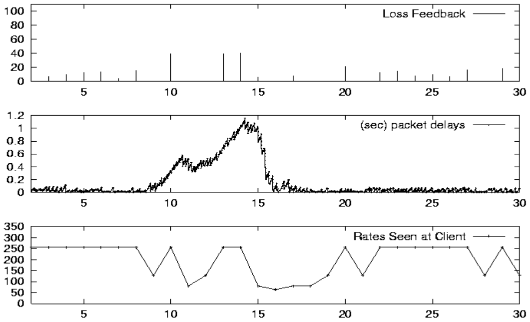


Figure 3(a): WaveLAN End-to-End Adaptation

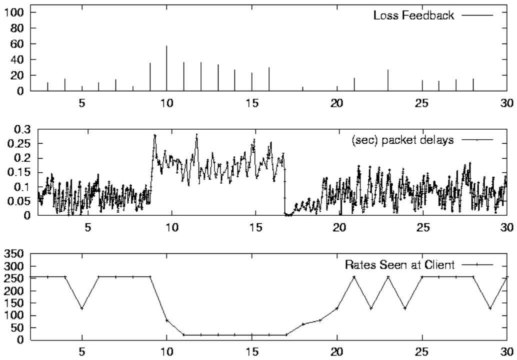


Figure 3(b): Bluetooth end to end adaptation

4.3 Aggregate results –Non-Adaptive Video, TCP and Voice

4.3.1 Video and TCP

The aggregate throughput is the same for all experiments for WaveLAN. Bluetooth, however, manages to grow the aggregate throughput as the number of nodes increases. The smaller range and formation of additional piconets adds capacity to the network.

A generic difference in the way the two source types, TCP and Video, share the network bandwidth is illustrated in Fig 4. In WaveLAN, individual TCP connections are allowed to grow their window and 'capture' the channel. When this happens, video connections suffer increased loss rates. On the other hand, the presence of polling in Bluetooth allows the video connections to share the channel with the TCP. In fact, with Bluetooth, the video achieves its full rate for different configurations. Several measurements are reported in Fig. 4 as shown by the caption below each sample (Number of nodes; number of connections; WL vs BT). It can also be noted from Fig. 4 that the total throughput for WaveLAN is higher than that of Bluetooth fot the 30nodes, 30 connections case. As the number of connections increases, the total throughput for Bluetooth increase with respect to WaveLAN since larger connections means more collisions in WaveLAN. Also, a larger number of nodes also causes the total throughput for Bluetooth to be higher since it leads to formation of more piconets and hence, addition to system capacity.

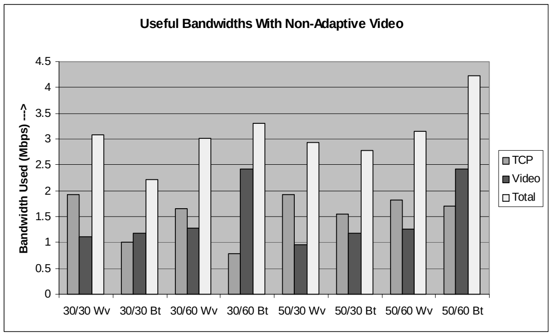


Figure 4 : H.263 Non adaptive video and TCP connections aggregate throughput.

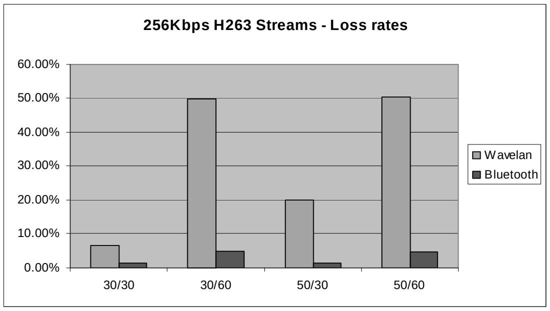


Figure 5 : Loss Rates for video connections for H.263.

From Fig. 5, we see that the loss rates for video are much higher in the WaveLAN case where the contention between the connections, especially TCP and video, allows some TCP connections to increase their window and capture the channel, locking out packets from the 256Kbps video connections. On the other hand Bluetooth shows less than 5% packet loss in all cases. Due to less packet retransmissions the Bluetooth case will save a significant amount of power which is particulary important in battery powered devices.

4.3.2 Voice

The significant parameter that needs to be studied for voice is the delay. The complementary cumulative delay distributions for voice in Bluetooth and WaveLAN for 30 nodes and 60 connections with non-adaptive MPEG Video are shown in Figs. 6(a) and (b).

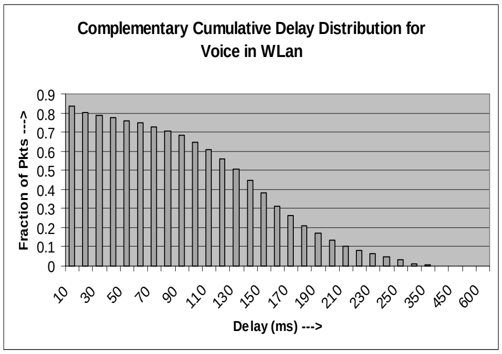


Figure 6(a): Voice Delay Distribution for WaveLAN

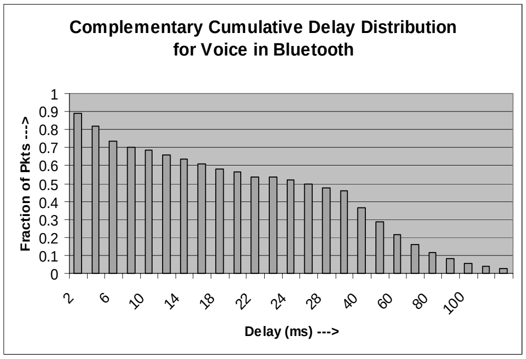


Figure 6(b): Voice Delay Distribution for Bluetooth

It is seen that the delays suffered are much lower in Bluetooth than WaveLAN. From the complementary cumulative distribution graph, we note that a packet loss ratio of less than 5% can be obtained for a play-out buffer of about 80 ms in the case of Bluetooth, whereas a play-out buffer of more than 350 ms is required to achieve the same effect with WaveLAN. Typically, a delay in excess of 300ms is considered unsuitable for interactive voice communications. To explain the very high WaveLAN delays, recall that the scenario considered here is of a very congested network with large number of connections. In such a network, the uncontrolled access to the channel and the large number of collisions and retransmissions in case of WaveLAN leads to large delays. Bluetooth, on the other hand, has a very controlled access to the channel determined by the polling scheme. This keeps the delays low and well-bounded.

4.3.3 Adaptive Video and TCP

In this section we repeat the experiments of Section --- with adaptive in place of non-adaptive video. The video sources adapt through the use of a periodic end-to-end feedback containing the RTP loss rates, as described in Section 4.2.

First, we show the aggregate source sending rates in Fig 7. This quantity is influenced by the loss rates reported to the server and represents the application requested bandwidth. If high loss rates are reported the server drops layers and uses less bandwidth. If loss rates are misreported or not reported then the server continues streaming at the present rate unaware of adverse network conditions. The sum of the feedback packets received in the WaveLAN configurations was 1215 whereas in Bluetooth 1216 feedback packets were received. WaveLAN tends to transmit less, especially at high connection density and load.

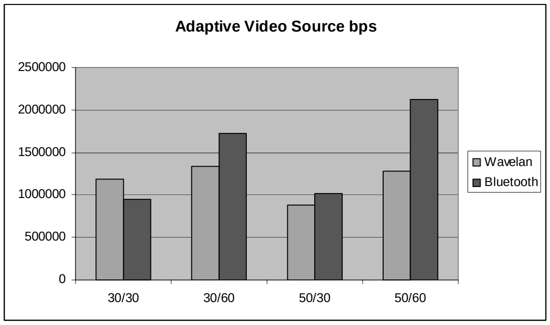
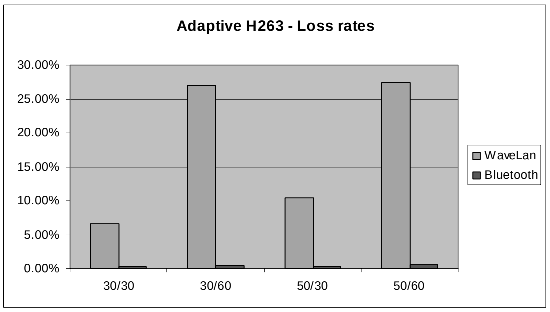


Figure 7: H.263 aggregate server sent rates
Figure 8: Loss Rates for Adaptive H.263 Video



Next we look at the loss rates with adaptive video in Fig 8. The controlled, adaptive polling environment of Bluetooth, with less reverse channel problems managed to eliminate the video loss rates almost completely in the adaptive case in our experiments. The highest aggregate loss rate in Bluetooth is 1.32%. In WaveLAN too the loss rates are reduced to half with respect to the non-adaptive case shown in Fig 5.

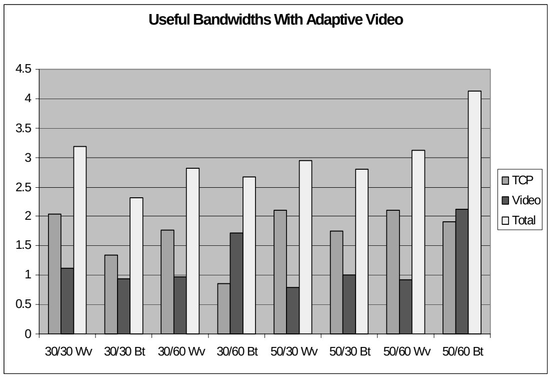


Figure 9: H.263 adaptive video and TCP connections aggregate throughput.

Next we examine adaptive video throughput in Fig 9. 1Mbps Bluetooth throughputs are again comparable to the 10Mbps WaveLAN total throughput. As video connections adapt they allow TCP connections to get more bandwidth. In Bluetooth video adaptation reduces loss rate to less than 1% in most cases whereas in WaveLAN adaptive video connections suffer 25% to 30% loss rates.

The total throughput is higher in WaveLAN than in Bluetooth for lower number of nodes. As more piconets are formed, Bluetooth adds bandwidth and surpasses the constant WaveLAN bandwidth, which is independent of the number of nodes.

5. Conclusions

In this paper we have evaluated the efficacy of the Bluetooth technology in supporting ad hoc indoor communications. The simulation results show that Bluetooth performs very well in mixed data and real time traffic scenarios typical of such environments. In particular, it guarantees service quality to multimedia streams while providing fair share of capacity to TCP users. It does not suffer from the TCP capture behavior exhibited by WaveLAN. Though the total system throughput is larger for WaveLAN for small number of nodes, Bluetooth can exceed the WaveLAN throughput when number of nodes becomes large, by using multiple, overlaid piconets. Adaptive video applications fare better with Bluetooth than WaveLAN, in part because the polling schedule of Bluetooth seems to offer a more stable service to adaptive video, precluding oscillations. It is to be noted again that these experiments were performed with the DCF mode of 802.11. In the future, we plan to repeat some of the experiments using the PCF mode.

Work is currently in progress in several directions. A Scatternet model is being developed, to allow the interconnection of piconets. Sensor interaction experiments are planned, with various mobility models. Finally, a Bluetooth testbed is planned jointly with Ericsson in order to validate the simulation results and develop hybrid simulation/emulation experiments with H/W in the loop.

References:

1) M. Gerla , P. Johansson , R. Kapoor , and F. Vatalaro: “*Bluetooth: last meter technology for nomadic wireless internetting*”, 12th Tyrhennian International Workshop on Digital Communications.
2) P.T. Brady: *" A model for generating on-o speech patterns in two-way conversation"*, Bell System Technical Journal, Sept. 1969, pp. 2445-2471.
3) A. Capone, R. Kapoor and M. Gerla: *“Efficient Polling Schemes for Bluetooth Picocells”*, Submitted to ICC 2001.
4) M.I Kazantzidis., I. Slain, Y. Romanenko, T. Chen, M. Gerla: *“Experiments on QoS adaptation for improving end user speech perception over multi-hop wireless networks”*, IEEE International Conference on Communications IEEE, 1999. p.708-15
5) *“Network Simulator (NS-2)“*, www-mash.cs.berkeley.edu/ns/
6) M.I. Kazantzidis, L. Wang, M. Gerla: *“On fairness and efficiency of adaptive audio application layers for multihop wireless networks”*, 1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99) IEEE, 1999. p.357-62. xii+390 pp.
7) S. Zurbes: *“Considerations on Link and System Throughput of Bluetooth Networks”*, PIMRC 2000, pp. 1315-1319.
8) S. Zurbes, W. Stahl, K. Matheus, J. Haarsten: *“Radio Network Performance of Bluetooth”*, ICC 2000, pp. 1563-1567.
9) J. Haartsen, *"BLUETOOTH - the universal radio interface for ad hoc wireless*

connectivity", Ericsson Review, n.3, 1998, pp. 110-117.