Adaptive asymmetric connections for TCP communications over the DECT air interface

Andrea Baiocchi, Maria G. Di Benedetto, Luigi Iaccarino, Marco Listanti

University of Roma "La Sapienza" - Dip. INFOCOM - Via Eudossiana 18 - 00184 Roma, Italy
Contact author e-mail: `andrea@infocom.ing.uniroma1.it`

Abstract

DECT technology provides capabilities for data and multimedia services over radio interfaces up to a few hundreds kbit/s. Be it used for local indoor environments, wireless local loop or public access, it is of interest to investigate how DECT capabilities adapt to support TCP/IP traffic: specifically, client-server applications generate strongly asymmetric traffic that can benefit of the DECT asymmetric connection capability.

The aim of this work is to define a way the DECT capabilities defined in the standard can be used to support TCP/IP asymmetric traffic and to assess the relevant performance by using a detailed simulation model of a DECT radio interface. The performance results suggest that a significant benefit is to be expected only for almost unidirectional connections, with substantial bandwidth in the main direction.

1. Introduction

The Digital European Cordless Telecommunications (DECT) [1] air interface defines a radio access intended for a broad range of applications, e.g. domestic or business private environments or cordless access to public networks. DECT technology offers quite flexible communications capabilities up to a few hundreds kbit/s for data and also multimedia services: in particular, multibearer connections [2] and asymmetric configurations can be used. This work is concerned with performance evaluation of these DECT advanced features for the support of the widely used protocol stack TCP/IP [3][4][5][6] in a local business environment.

The focus of the present work is simulation of a DECT radio air interface in an indoor environment, in the framework of the COBUCO project [6], which is going to develop, install and use a UMTS demonstrator including mobile IP hosts and exploiting TCP/IP based applications., specifically considering in full detail the MAC, DLC and TCP layers functions. The physical, IP and application layers are synthetized including only the essential functions thereof.

The aim is to assess suitability of asymmetric multibearer connections as defined in the DECT standards to support client-server TCP/IP traffic, which is known to be often unbalanced (i.e. essentially asymmetric). To this end, a background traffic is generated in a multi-cell DECT island and the entire protocol stack is simulated for a tagged user communication towards a fixed host.

As for the rest of the paper, Section 2 identifies the system scenario. Section 3 details the way the DECT capabilities are exploited to provide asymmetric multibearer connections. Section 4 describes the simulation model and Section 5 presents the performance results.

2. Reference scenario

The classical scheme of a TCP/IP communication envisages an information exchange between two hosts across an IP network (actually relying on a multiplicity of different network and on IP internetworking). Typically, the upper layer application gives rise to an essentially unidirectional data flow (e.g. client-server paradigm).

In this work we consider a mobile host, exploiting a DECT air interface, with a multibearer connection, communicating with a fixed host. The fixed part of the DECT subnetwork comprises Radio Fixed Points (RFPs) and a centralized Base Station (Central Common Fixed Part, CCFP), dealing with higher layer functions. An Interworking Unit (IWU) is required to access the external IP network: the IWU acts as an IP router interfacing the DECT subnetwork on one side and the external subnetwork on the other side. Mobility of the IP/DECT user is taken care of DECT mobility handling functions within the DECT island. A reference environment for such a configuration is a private network (e.g. wireless LAN).

The protocol data units (PDU) of the different layers are shown in Figure 1 with their respective names and sizes in bytes. We assume TCP is used as a transport layer protocol, LU2 (frame relaying) service is chosen in the DECT DLC layer and an Ip protected channel is used for user information in the MAC layer. Given L application layer bytes, the number of required MAC packets is at least (i.e. provided no retransmissions in DLC and MAC layers take place) $\mathit{ceil}\big((L+42)/29\big)$, where L is in bytes and $\mathit{ceil}(x)$ is the least integer not less than x.

In more detail, the implemented MAC procedures correspond to the Ip Correction channel [1], i.e. a part of the MAC packet B field is divided into 10 bytes subfields, each protected by a CRC field (see Figure 1). If errors are detected on at least a subfield of a packet, retransmission is repeated the next frame; each time, correct subfields are stored by the receiving side, until the MAC packet can be reconstructed without errors. A maximum number of 7 transmission attempts is allowed. It is also assumed that the undetected error probability of MAC CRCs is negligible, so that all error detection codes added by upper layers (DLC LU2, IP headers, TCP segments) have essentially no impact on performance.

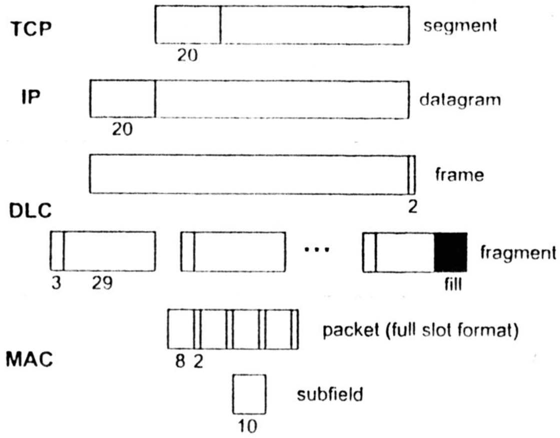


Figure 1 - Protocol Data Units of the considered protocol architecture and their relations (all lengths are in bytes).

The DLC LU2 service is considered. The protocol at the frame level is essentially void, except of a 2 bytes checksum field appended to DLC frames. The main part of the DLC LU2 service is fragmenting of DLC frames into small data units (see Figure 1), comprising a three bytes overhead, for useful payload length indication, send sequence numbers and receive sequence numbers. Sequence integrity is taken care of, by means of a Go-back-N ARQ protocol, with a transmission window of 63 fragments and two timers. A first timer is set to 4 TDMA frames (maximum lifetime of a MAC packet DIV 2 +1). When it expires and the sequence is not complete, retransmission is asked for and the second timer is started, lasting 14 TDMA frames (2 * maximum lifetime of a MAC packet). When this second timer expires, fragment recovery is given up and the next DLC frame is considered.

As for TCP, all main functions are implemented [7] (variable window management with slow start and congestion avoidance, timers, acknowledgements, fast retransmission and fast recovery algorithms, round trip time estimation). The fast retransmission algorithm envisages that a segment is retransmitted even though the retransmission time-out (RTO) is not expired yet, when at least three consecutive acknowledgements carrying the same value are detected. Fast recovery defines a faster way to increase the transmission window than allowed by slow start and can be used for retransmissions triggered by the fast retransmission algorithm (not those due to RTO timeouts). The RTO computation is made on the basis of the classical Jacobson algorithm [13].

Functions associated with IP header are neglected (e.g. header error checksum) and the fixed network segment is idealized as for information integrity and delay variation (note that if delay is almost constant, round trip time estimation mechanism of TCP manage to track it quite successfully).

3. Asymmetric connections

Asymmetric MAC connections are provided by using a different number of simplex bearer for uplink and downlink. A simple bearer is defined as the bearer capacity obtained by taking a single slot out of every DECT frame. It can be either an uplink or downlink simplex bearer. A double simplex bearer is made up of two simplex bearers whose slots are separated by half a DECT frame (5 ms), transmitting in the same direction. In case the two paired simplex bearers transmit in opposite direction, they are referred to as (full) duplex bearer.

The rules for configuring an asymmetric MAC connections are:

1) simplex bearer are always allocated in pairs, half a TDMA frame apart (double simplex bearer);
2) there are k simplex bearers in the forward direction and n simplex bearer in the reverse direction called "special" bearers, used to carry control information (n>0 and $n \le k \le 24-n$).

Alternatively, it can be said that an asymmetric MAC connections consists of at least one duplex bearer and a number of double simplex bearers. Since DECT standard actually defines only fully asymmetric connections, supporting DLC user data flow only in one direction (forward direction), all double simplex bearers share the *same* transmission direction. Moreover, a double simplex bearer setup is possible only if there is already at least one duplex bearer active in the asymmetric connection.

If the asymmetric connection exploits a MAC protected I-channel (Ip), the reverse bearers are also used to send acknowledgements. The Ip-channel uses a 2-state packet number in the A-field header. In an asymmetric connection, only MAC packets in the forward direction need be acknowledged by special MAC packets flowing in the reverse direction. The acknowledgement mechanism is implemented by using two bits for each logical simplex bearer of the forward direction: the ACK bit and the BCK bit. The ACK bit reports the number of the last correctly received packet; the BCK bit is used to report the number of the next expected packet. The $2k$ bits required to acknowledge all forward bearers packets are multiplexed into at least one B-subfield on at least one reverse bearer (Figure 2). If the B-subfield containing the acknowledgement bits gets lost, all of the forward MAC packets are not acked and must be retransmitted.

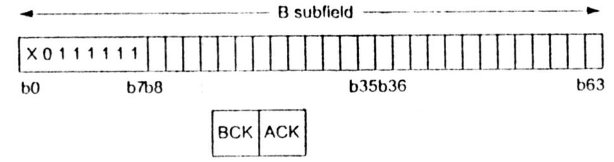


Figure 2 - Format of the ACK_Mod_2 message carried in the B-subfield of reverse direction MAC packets.

One B-subfield has also to be assigned to carry the RN (Receive Number) and the ACK/NACK bit necessary to implement the DLC-protected-service (LU2 service).

The vital role of duplex bearer(s) in an asymmetric connection is therefore evident, in that the forward direction throughput may suffer significantly from acknowledgement losses in the reverse direction and the entire connection is disrupted, if the duplex bearer(s) are not available. For all these reasons an asymmetric connection with just one duplex bearer is highly vulnerable.

In the Network Layer there is an information element called <<CONNECTION-ATTRIBUTES>>, that specifies that MAC asymmetric connections may establish 1 or 2 duplex bearers. Our choice is to use a target number of two duplex bearers and to replicate the $2k$ acknowledgement bits in two distinct B-subfield of each packet flowing in the reverse direction. Therefore, according to previous notation, we set $n=2$ and $2 \leq k \leq 22$.

Since the above defined fully asymmetric MAC connection allows a unidirectional DLC user data flow, two such connections are needed in general. The case of interest here is TCP/IP over the DECT protocol stack. Since TCP requires a bidirectional segment flow (if only to carry acknowledgements), a bidirectional DLC data flow must be provided. However, typical TCP traffic patterns are essentially unidirectional at any time (e.g. WWW traffic) and hence could benefit from adoption of asymmetric MAC connections. To overcome the limitation of fully asymmetric MAC connections, which only allow data to be transferred in one directions, two separate MAC connections can be used, as depicted in Figure 3.

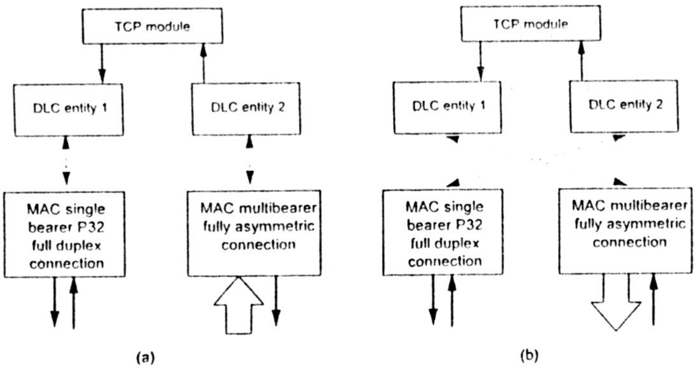


Figure 3 - Support of asymmetric TCP traffic patterns by means of fully asymmetric MAC connection facility: association between DLC entities and MAC entities.

Thus for every TCP connection there will be 2 DLC connections and 2 MAC connections of different capacity. One of the DLC connection is associated to the MAC connection with high capacity and the other to the MAC connection with just one duplex bearer. The switching of the MAC connections can be performed by exploiting the DLC layer suspend and resume functions. Moreover, the direction of the double simplex bearers of the asymmetric MAC connection can be reversed by exploiting the fast release procedure.

The switching of the DLC-MAC entities association can be activated by monitoring the congestion status of the DLC entities. More in detail, a threshold is set in the DLC entity buffer: whenever such a threshold is exceeded, the DLC entity whose buffer is congested requires the higher capacity MAC connection, hence causing a MAC connection switching. Conversely, a DLC entity releases the high capacity MAC connection (hence allowing the switching of the MAC connections) whenever there is an explicit request from the other end peer entity and the DLC buffer content level is below the threshold.

4. Simulation model

We consider a system consisting of three RFPs (cells) with Portable Terminations (PTs) of different types. The RFP belong to the same MAC cluster (no external handover is considered), mobility is assumed to be marginal (i.e. PT remain stationary during connections). Intra-cell (bearer) handover is considered, since it is due to changing propagation conditions that might occur even for a stationary PT.

Each PT is capable of supporting only one connection at a time. PTs engaged in a connection can only belong to two types in the considered scenario: a) PT exploiting a single full duplex bearer connection (P32 bearer); b) PTs exploiting a multibearer connection. Type b) PTs are briefly referred to as "Data" PTs and more precisely they use a combination of two MAC connections, one of which is a fully asymmetric one, as described at the end of Section 3. For each type of PT, call arrivals are modelled as Poisson processes. Call durations are exponentially distributed for PTs of type a), while PTs b) call holding times can take the values 1, 2 or 3 min with the same probability.

The simulation model comprises two main parts: i) a "background" environment, loading the DECT air interface and made up by narrowband type a) PTs and wideband Data PTs; for these PTs only DECT MAC functions are implemented in the simulation software; ii) a single special Data PT, referred to as "TCP user", for which MAC, DLC, IP and TCP functions are also implemented.

4.1. Simulation of the DECT MAC functions

The simulation software runs on a TDMA frame by TDMA frame basis and emulates the major MAC layer procedures through the evolution of a state machine for each MAC entity of each PT (Figure 4).

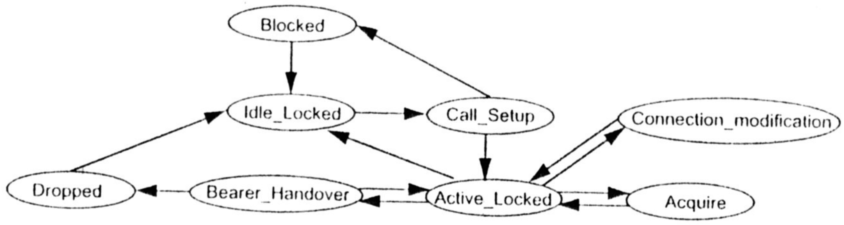


Figure 4 - MAC layer PT evolution state diagram.

At the beginning of the simulations, all PTs are set to the Idle_Locked state and they are uniformly distributed in the three cells. While in this state, PTs are just allowed to pass to the Call_Setup state. The Call_Setup state is not defined as a part of the DECT standard but has been included in this work to indicate that a given PT has a connection setup procedure in progress.

At call set up and each time the DECT connection is to be modified, the mobile terminal uses the so called channel list, where physical DECT radio channels are labelled as being quiet (measured signal level < –93 dBm) or busy or somewhat in between the two extremes. Hence, we define three channel labels, quiet (Q), busy (B) and interfered (I). The channel list is updated by the mobile terminal itself. To account for measure imperfections and inter-cell interference, we assumed the following distribution for the channel list updating outcomes: P(Idle→Q)=0.8; P(Idle→I)=0.2; P(Idle→B)=0; P(Reserved→Q)=0; P(Reserved→I)=0.5; P(Reserved→B)=0.5.

Immediately after a PT enters the Call_Setup state, the corresponding Call_Setup timer (3 s) starts. A channel list generation algorithm provides the PT with an up to date channel list. Then the PT selects a channel from its channel list and requires the channel to the RFP it is locked to. The FP decision logic evaluates if the channels selected by a PT are idle or have already been reserved by other PTs and detects if request collisions take place. Frame by frame, idle channels at any RFP are assigned to the PTs that select them if no collision occurs. When a PT selects a channel that was previously reserved, the corresponding bearer setup attempt is considered to have failed. The PT can repeat the above procedure up to a specified maximum number of times before the expiration of its setup timer. For multibearer connections (Data users) after each successful bearer setup, a new bearer setup can be attempted, subject to not exceeding the target number of bearers.

If the Call_Setup timer expires and the PT has not yet established the minimum acceptable number of bearers, it enters the Blocked state. Otherwise, it enters into the Active_Locked state. When a PT enters the Blocked state, it remains there for a TDMA frame before finally returning to the Idle_Locked state. In the Blocked state all bearer previously established for the connection are released. The return of a PT to the Idle_Locked state from the Active_Locked state represents the natural termination of its connection.

A PT enters the Bearer_Handover state each time it has at least one of its bearers needing to be changed to another physical channel. This situation is generated in a probabilistic way, considering that for a particular bearer the time between bearer handovers is an exponentially distributed random variable. For a PT, only one bearer handover at a time is processed. A completely updated channel list is immediately provided to a PT as it passes from the Active_Locked state to the Bearer_Handover state. The channel list is periodically updated every 30 frames until the PT leaves that state. A single bearer handover procedure ends when either the PT obtains the reservation of a new physical channel or the respective timer expires. At the expiration of each timer the number of surviving bearers in the connection is checked . If it falls below the minimum acceptable number, then the PT enters into the Dropped state and in the following frame all old bearers of the connection are released. On the other hand, each time a channel is abandoned by a PT after a bearer handover, it is no longer "reserved" in the respective RFP.

At the beginning of each frame, an Active_Locked PT checks if its number of bearers equals the target number (it cannot be less than the minimum number though). If not, the Acquire state is entered and acquisition of new bearers up to the target number is attempted. A new bearer acquisition is tried out up to 16 times, with attempts being 50 ms apart of one another. If all attempts fail (i.e. after a timeout of 750 ms), a random back-off time is scheduled between 10 and 20 s. The Acquire procedure is invoked until the target number of bearers is attained. For asymmetric connections, restoring the target number of duplex bearers has priority over double simplex bearers.

The Connection_Modification state is entered from the Active_Locked state when the association between DLC and MAC entities of a Data PT has to be switched.

4.2. Simulation of the TCP user

The TCP user is of Data type and exploit a DLC/MAC arrangement as discussed in Section 3 (two MAC connections, one single bearer full duplex connection and one multibearer fully asymmetric connection). The asymmetric connection has a target of 2 duplex bearers and 5 double simplex bearers. So, overall the bandwidth assigned to a Data user amounts to 8 double bearers: 1 full duplex bearer for the narrowband MAC connection and 2 full duplex bearers plus 5 double simplex bearers for the asymmetric wideband connection. The minimum number of bearers for each MAC connection is assumed to be 1 duplex bearer. Should the MAC connection of the TCP user be dropped, it is assumed that a new MAC connection setup is started anew immediately (Call_Setup state).

The BER and bearer handover processes are modelled as follows. A two state Markov chain is defined, associated to each bearer of the TCP user, the two states being labelled as B (bad) and G (good). Transitions between the two states occur at TDMA frame rate, according to the probabilities $P(B \rightarrow G)=0.8$ and $P(G \rightarrow B)=0.16$. The BER (probability of bearer handover required) is 1.5e-4 (1e-4) in the G state and 9e-2 (1e-3) in the B state. Bit errors are assumed as independent of one another; hence the subfield error ratio (SER) can be easily computed, yielding 0.012 in the G state and ≈ 1 in the B state. The Markov modulation approach aims at modelling the effect of variable quality transmission channel typical of radio interfaces [2][8][9][10].

The TCP Maximum Segment Size (MSS) is chosen to be 536 bytes. With the assumed protocol stack, this requires 20 MAC packets at least (i.e. without any retransmission) to carry a TCP segment. New TCP segments are generated according to two traffic models: a) Poisson arrival process of fixed size TCP segments (=MSS) at both ends of TCP connections; b) a request-response traffic pattern aimed at reproducing WWW type of traffic. Type b) traffic consists of a request process originated by the mobile host towards a fixed host, each request with uniformly distributed length between 40 and 50 bytes. For each request, a file is issued by the fixed host having an average size of 18 kbytes, with a special heavy tailed distribution with should represent actual Web pages size [11].

TCP segments waiting for first emission or for acknowledgement are stored in a so called TCP buffer; also DLC fragments waiting for emission or acknowledgement are stored in a buffer. Both buffers are assumed to have infinite size (i.e. buffer overflow is neglected).

5. Performance evaluation

Extensive simulation experiments have been carried out. The results presented here refer essentially to throughput delay trade-off of the TCP user and aim at a comparison between symmetric and asymmetric connections. For a fair comparison, the symmetric connection consists of 8 full duplex bearers. Accounting for layer overhead from TCP down to MAC, each simplex bearer has a nominal maximum throughput of about 2.7 kbytes/s. Hence, the maximum theoretical throughput of each direction of the TCP connection based on the symmetric MAC connection amounts to 21.6 kbytes/s (8 simplex bearers), whereas the same quantity foe the main direction of the fully asymmetric connection is 32.4 kbytes/s (12 simplex bearers). The actual TCP throughput is reduced because of: i) MAC state machine evolution (e.g. bearer handovers); ii) MAC and DLC retransmissions. The TCP net throughput is denoted by Λ in the following.

Figure 5 shows average end-to-end delivery delays of TCP segments W as a function of Λ by assuming a Poisson traffic: i) only on the downlink direction (leftmost picture); ii) alternating with 90 s downlink bursts and 10 s uplink bursts (middle picture); iii) alternating with 900 s downlink bursts and 100 s uplink bursts (rightmost picture). As it can be expected, asymmetric MAC connections allow a higher throughput to be achieved, but throughput-delay trade-off advantage diminishes as the main traffic direction switches, the more the more frequent the switching.

Figure 6 compare symmetric and asymmetric connections by plotting W and the application data blocks (e.g. WWW pages) delivery delays T as a function of Λ, by assuming the request/reply traffic described in Section 4, with deterministic application file sizes of 18 kbytes (traffic type D). As for Figure 5, both pure downlink traffic and mixed uplink downlink traffic are considered, 90% of the overall application data being generated by the fixed host (downlink) in this last case. Figure 7 extends the results of Figure 6 to the case of heavy tailed application data blocks distribution. The common result is that asymmetric connection entail a significant performance improvement if application traffic is unidirectional or main traffic direction switching occurs with much longer times than TCP traffic dynamics.

Finally, Figure 8 shows that a significant amount of radio link capacity is needed to perform MAC packet retransmissions that guarantee a negligible residual error ratio in information delivered to upper layers.

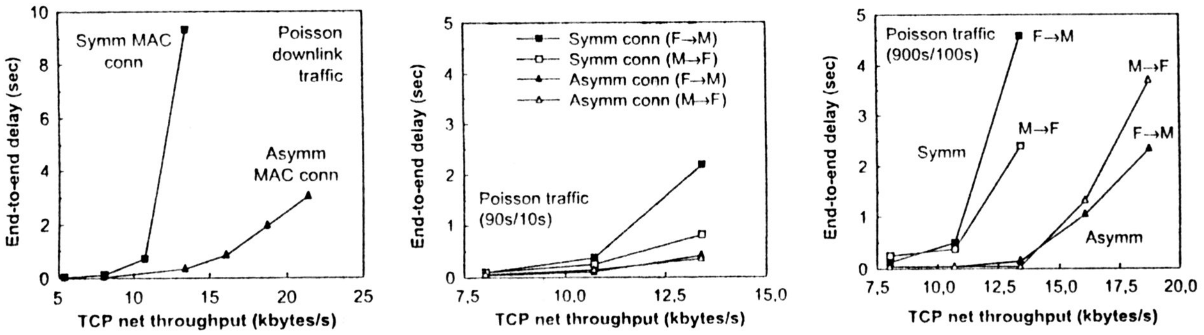


Figure 5 - TCP segments end-to-end delays

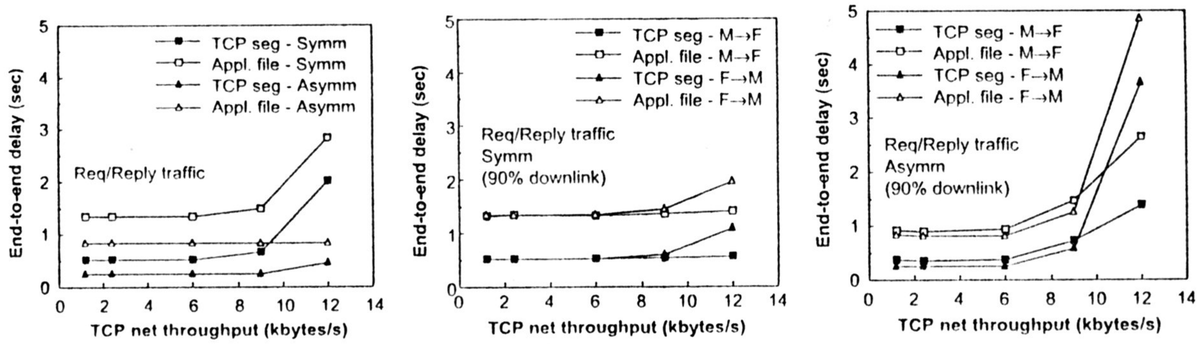
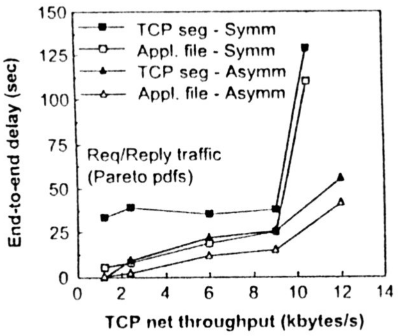


Figure 6 - TCP segments and application layer data delivery delays for request/reply traffic type D.



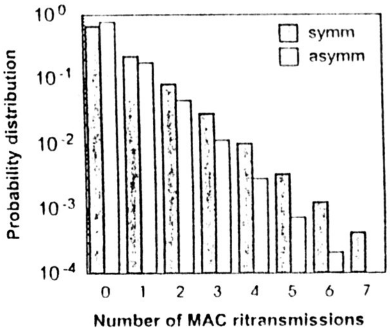


Figure 7 - TCP and application data delivery delays for request/reply traffic type P.

Figure 8 - Probability distribution of MAC packet retransmissions.

References

[1] "Digital European Cordless Telecommunications", ETSI Standard ETS 300 175 Parts 1÷9, 2nd edition, 1996.

[2] P. Wong, A. Lasa, F. Halsall, G. Schultes: "Performance of multibearer connections for varied data services in a TDMA system", proceedings of the IEEE ICC'94, New Orleans, 1-5 May 1994, pp. 593-597.

[3] G. Bao: "Performance evaluation of TCP/RLP protocol stack over CDMA wireless link", Baltzer Wireless Networks Journal, March 1996, Vol. II, pp. 229-237.

[4] R. Caceres, L. Iftode: "Improving the performance of reliable transport protocols in mobile computing environments", IEEE Journal on Selected Areas in Communications, June 1995, Vol. 13, n. 5, p. 850.

[5] P. Manzoni, D. Ghosal, G. Serazzi: "Impact of mobility on TCP/IP: an integrated performance study", IEEE Journal on Selected Areas in Communications, June 1995, Vol. 13, n. 5, pp. 858-867.

[6] H. Herbrig, R. Rheinschmitt: "DECT/ATM based UMTS demonstration and trial system", Proceedings of the first Mobile Summit, Granada (Spain), 27-29 November 1996, Session B3.

[7] W.R. Stevens: "TCP/IP illustrated. Vol. 1: the protocols", Addison Wesley, 1994.

[8] H. S. Wang, P. Chang: "On verifying the first-order markovian assumption for a Rayleigh fading channel model", IEEE Transactions on Vehicular Technology, May 1996.

[9] N. Benvenuto, A. Salloum, L.Tomba: "Performance of digital DECT radio links based on semianalytical methods"; IEEE Journal on Sel. Areas in Comm., May 1997.

[10] J. Chuang: "The effects of time delay spread on portable radio communications channel with digital modulation"; IEEE Journal on Sel. Areas in Comm., June 1987.

[11] M.E. Crovalla, A. Bestavros: "Explaining World Wide Web traffic self-similarity", Boston University Tech. Rep. TR-95-015, October 1995.