A protocol extension to HiperLAN/2 to support single-relay networks

D. Hollos, H. Karl

Fachgebiet Telekommunikationsnetze
Fakultät für Elektrotechnik u. Informatik
Einsteinufer 25 Technische Universität Berlin
10587 Berlin
hollos@ee.tu-berlin.de
karl@ee.tu-berlin.de

Abstract:

The use of ad-hoc-based multi-hop relaying has been advocated as a means to increase the capacity and energy-efficiency of wireless networks. It is one of the goals of the IBMS^2 project to investigate the potential of such ad-hoc relaying, using HiperLAN/2 as a case study. To realize relaying in a HiperLAN/2 network, the HiperLAN/2 protocol needs to be extended. This paper describes a standard-conform extension to HiperLAN/2 that enables terminals to relay data traffic within a single HiperLAN/2 cell. This extension is flexible, imposes little overhead, and work on the time scale of a single HiperLAN/2 MAC frame.

Keywords: Wireless LAN, multi-hop wireless relaying, HiperLAN/2

1 Introduction

One of the most important problems in wireless networks is to provide a large capacity, either to access a given base station in an infrastructure-based network or to communicate directly with other users in a pure ad-hoc network. Capacity can be regarded as the total amount of bandwidth available or, similarly, as the number of users that can be supported (when the service a particular users requires is constant). This is important for a number of reasons, e.g., to keep the number of base stations small.

Some of the most important limitations to achieving a large capacity are noise and interference. Both are caused by the need of a receiver to have a certain ratio between the signal on the one hand and noise and interference (the SINR) on the other hand in order to successfully decode an incoming signal. Both noise and receiver architectures (and hence the thresholds for signal to noise and interference ratio) are either a given constant or can not easily be changed. However, interference could potentially be reduced as it stems from the system itself.

Interference is mainly caused by other terminals trying to overcome large distances, which requires large transmission power. Typically, these are terminals that are far away from their respective communication partners, e.g. their base stations. Using such infrastructure-based systems as an example, reducing the power radiated by such terminals would reduce the interference level in neighboring cells using the same frequency band.[1] However, with reduced transmission power, a terminal would no longer be able to reach its own base station. It would hence be necessary to include a relaying station in the transmission path. Using fixed setups for such relaying stations is not an attractive approach (e.g. due to cost reasons); therefore, only other mobile terminals are candidates for such relaying terminals. The usage of such relaying terminals would thus allow to reduce the transmission power levels used, improving the interference situation in the entire communication system and potentially improving the overall capacity in the system. Both the direct and the relaying case are outlined in Figure 1. The situation is similar in pure ad-hoc networks which usually do not have the notion of a cell or a base station; interference is nevertheless a severe limitation to capacity in such networks.

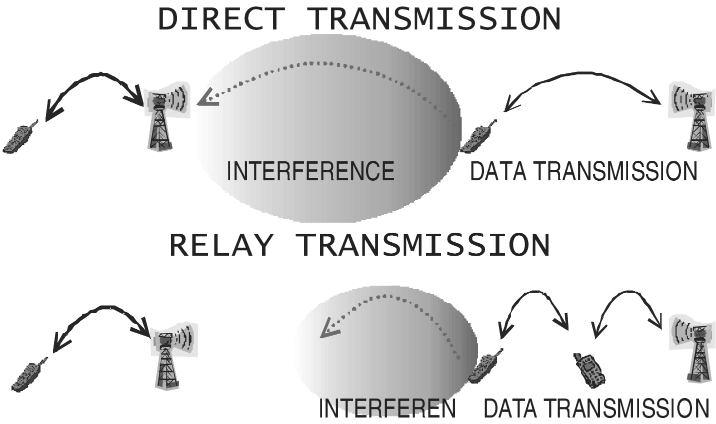

Figure 1: Conceptual view of direct and relay communication

On the downside of this system concept is the need for the relaying terminal to transport other terminals' traffic in addition to its own traffic. In order for the relaying terminal not to become a system bottleneck, the relaying terminals will have to use more capacity to transmit this traffic — but this should be made possible by the reduced interference level.

In general, this system concept trades off interference generated at the border of cells against increased traffic load in the interior of cells and investigating it is one of the main goals of the IBMS[2] project [ibm]. The main challenge is hence to determine which relaying terminals to use and how to use them. As this is ultimately a routing problem, this concept uses ad-hoc techniques to be employed as an ad-hoc extension of classical infrastructure-based networks. Unlike other, similar research projects, we are using these ad-hoc techniques to improve the capacity, not the coverage of a given cell. Additionally, we are also interested in the energy-efficiency aspects of this problem, but this work is not

---

[1] This is true even if, as is commonly the case, spatial separation is employed to avoid neighboring cells use the same frequency band. The impact of interference is then correspondingly reduced, but can nevertheless not be neglected. Also, depending on the actual technical system in use, even cells with adjacent frequency bands suffer from adjacent channel interference.

discussed here.

We have shown in previous work [KM01, MKW01] that this trade-off depends — among other parameters — on the size and relative position of multiple cells. The main technique that achieves capacity improvement is a joint optimization of transmission power and modulation rate, exploiting the improved interference situation close to a cell's base station to use higher modulation rates to carry the added traffic load for the relaying stations. In particular, large cells where the outer terminals are relatively close to each other provide a particularly promising scenario for such relaying networks and that relaying does indeed improve the performance of such networks. Put the other way around, when keeping the total cell capacity constant, relaying allows to reduce the amount of spatial separation that would normally be required, permitting to move cells closer together.

This work was based on analytical evaluations of rather simple scenarios. Such evaluations need to be based on an underlying system architecture to provide parameters to the analysis. We have chosen the HiperLAN/2 system as an example of a rather tightly controlled system. The advantage of this choice is that HiperLAN/2 can provide a lot of information about the transmission situation within a cell (e.g., received signal strength indicators between different terminals) and its TDMA-based medium access lends itself to simple analysis and management. Also, for HiperLAN/2, predicted mappings from SINR to packet error rate are available, which are an important parameter for our analysis. The disadvantage of HiperLAN/2 is that it does not directly allow to set up a relaying network in the form outlined above. It is the goal of this paper to describe appropriate HiperLAN/2 protocol extensions that would close this gap by allowing the access point to set up such relaying relationships on a very short time scale in a simple and flexible manner; related problems like routing and detecting the potential for relaying are out of scope of this paper. Such protocol extensions also form a pre-requisite for implementing a proper simulation toolkit that would enable us to investigate larger scenarios and to make more detailed statements regarding the potential of relaying networks for the improvement of capacity.

There are also other research projects on the multi hop HiperLAN/2 [Pee01] [MHCN01] [WEK^{+}01], however, none of them deals with synchronised intra-cell relaying. The novelty of our idea is merging the advantages of multi-hop forwarding and synchronous channel access.

The remainder of this paper is organized as follows: In the following section, a brief recapitulation of the HiperLAN/2 system is provided and Section then gives an overview of the HiperLAN/2 protocol extensions that we propose to support single-relay systems. Section briefly outlines possibilities for validation of our protocol, namely, our HiperLAN/2 simulator that incorporates these extensions. Section concludes the paper and discusses future work and finally, Appendix contains a more detailed description of the protocol extensions.

2 The HiperLAN/2 system

HiperLAN/2 is a wireless LAN system standardized for the 5.2 GHz range. It is mostly aimed at low-mobility scenarios and includes modulation types of up to 54 Mbit/s. The goal of this standard is to maximize the utilization of the radio channel. The main mechanism — and the largest difference to IEEE 802.11 — is the use of a a priori scheduled medium access: Among a set of stations (a so-called cell), one station is declared to be the central controller (CC). Any station that wants to communicate with another station has to announce this to the CC, which will grant time slots in a periodically repeated frame to this station. Hence, H/2 uses a connection-oriented, centrally scheduled TDMA to organize the medium access. Main benefits are collision-free data traffic (100% efficiency, no hidden or exposed terminal problems) and simple support for priorities or QoS requirements.

Scheduling happens on the basis of a frame, which is divided into different phases (compare Figure 2), which are again divided into cells of fixed length. In the first phase, the central controller broadcasts administrative information (broadcast channel, BCH and frame control channel, FCH), in particular, which terminal is allowed to transmit to whom at what time and for how long. The FCH (Frame Control Channel) is a directory located to the first part of every H/2 MAC frame; it tells to every terminal what is going to happen in that MAC frame. The FCH consists of slots, called Information Elements (IE), where each IEs are assigned to one particular transmission for one terminal; one transmission may contain many packets. One of the basic ideas in the H/2 standard is that all devices in the cell know about all the data transfers of the MAC frame after reading its Frame Control Channel (FCH); it is important, because in the real life terminals have to prepare themselves for high-speed data reception, transmission, etc. The last part of the first phase is the so-called access channel (ACH), which provides feedback to newly registering terminals; it is not pertinent to the present discussion.

As this information is available to all terminals, perfect channel access can be organized: it is divided in a phase where the CC sends data to terminals (the downlink phase) and a second phase in which terminals transmit to the CC (the uplink phase). Within these phases, so-called LCH cells (long transport channel, the actual cells in which data transport happens) are exchanged between entities. While this basic frame structure and the central organization of the channel access is a very simple concept, it is the source of performance and flexibility of H/2.

Figure 2: H/2 MAC frame structure

One example of this flexibility is the so-called direct-link traffic: As it is possible for the CC to assign the same time slot to one terminal as a sender, to another terminal as a receiver, terminals can communicate directly with each other without having to send the data via the CC. Usually, the sequence of transmissions is organized such that after the frame's initial administration phase, downlink traffic is scheduled, then direct-link traffic, and uplink traffic — the order of these first four phases is chosen such that the number of send/receive turnarounds can be minimized. At the end of a frame, the fifth (and last) phase is the random access time-interval (the medium access method is slotted ALOHA) where terminals can compete for channel access; it is used to setup associations between terminals and CC and other infrequent, unscheduled requests.

On top of these basic mechanisms, a number of additional capabilities are included in the H/2 standard (e.g., automatic frequency selection, multicast, automatic CC selection, etc.). In the present context, the CC's capability to request channel measurements from any terminal, describing the channel characteristics between any two terminals, is important. Such a "radio map" allows the CC to decide how to schedule transmission and how to organize the relaying (e.g., the CC can learn whether a given terminal is "in between" itself and a third terminal as seen from the perspective of channel gains; physical location is not really important). Other possible add-ons are sleep modes for terminals or the interconnection of separate H/2 cells [Pee01].

3 Extending HiperLAN/2 with relaying capabilities

As outlined in Section , the goal of our project is to use relaying with a HiperLAN/2 cell to reduce energy consumption and improve capacity. In order to do so, terminals located far away from the CC should reduce their transmission power and communicate with the CC via a relaying terminal located closer to the CC. As the direct link phase allows terminals to communicate with each other directly, we exploit this capability to introduce a relaying terminal between the CC and the actual receiving (or sending) terminal. As the decision when to use relaying at all and which terminal to choose as the relaying point rests with the CC, the main problem to solve is how to signal to both the relaying terminal and the destination terminal to use relaying. This must happen very quickly in order to be able to accommodate changing channel characteristics.

At the moment, we only consider the use of a single relaying node, hence, we call the protocol "Single Relay" or SR protocol. The protocol extensions described below would also allow — in principle — to use more than one relaying terminal between CC and far terminal, but this would also increase the signaling overhead (even though our signaling overhead is very small: only a single FCH cell is included in the frame header for each relaying connection). The ramifications of this tradeoff on the capacity and energy efficiency problem are, however, currently not clear.

The protocol extensions described here fit into the H/2 standard and allow dynamic route path change even in every MAC frame without additional protocol overhead.

3.1 Informal protocol description

Consider the following situation: An H/2 CC has data to send to terminal MT2 and it has already decided (e.g., based on the Radio Map [BRA] and some adequate routing algorithms) to use terminal MT1 as a relaying terminal. Such a setup is illustrated in Figure 3.

The CC schedules a four-phase relaying data transfer into the next MAC frame (see Fig-

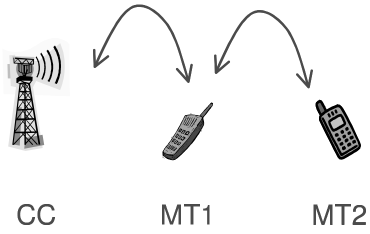

Figure 3: Basic relaying setup

ure 4). First, the CC needs to transfer the data to the relaying terminal MT1 using the ordinary Downlink Phase. When setting up this transmission at the beginning of the frame, the CC lets the MT1 know that this data is part of a relaying connection. MT1 will transmit the data to MT2 in the Direct Link Phase of the same MAC frame. The acknowledgement(s) from MT2 travels back on the same way, first from MT2 to MT1 in the Direct Link Phase, then from the MT1 to the CC in the Uplink Phase. These four phases are scheduled into one MAC frame, thus relaying does not increase the delay.

An uplink transmission from MT2 to the CC via MT1 is handled in a similar fashion. However, uplink traffic can only start in the middle of a MAC frame, as the initial downlink phase is not useful. In the direct link phase, MT2 transmits data to MT1, which in turn passes on this data to the CC in the uplink phase at the end of the frame. Acknowledgements can not be completed in the same MAC frame; their transmission is scheduled in the following MAC frame in the downlink phase from CC to MT1 and in the direct link phase from MT1 to MT2.

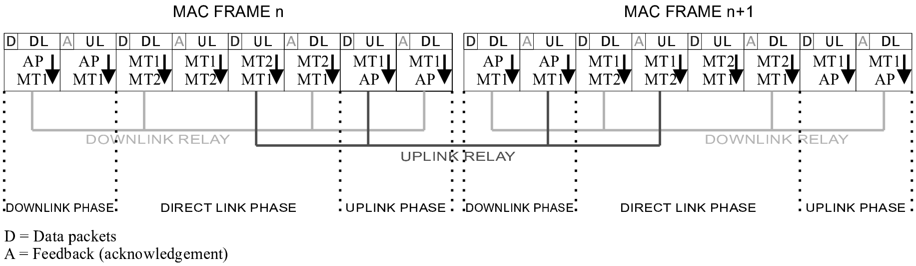

Figure 4: Relaying protocol frame format

The interesting part of this solution is the way how we let the terminals know about the type of the transmission (whether it is a relaying one or a normal one), and the signaling to the relaying terminal (MT1) where to forward the data: we complete these tasks without any protocol overhead.

3.2 Basic Data Transport of the SR Protocol

The signaling for the relaying of data happens within the FCH at the beginning of a MAC frame. Recall that the FCH is divided into information elements (IEs), each describing one or more slots of the downlink, direct link, or uplink phase.

The IEs have a Target MAC address field (it tells which terminal this IE belongs to), and many other fields, for instance the number of packets to transmit, etc. It also has a type indicator, which determines the phase of that particular transmission (Downlink, Direct Link or Uplink). Here we added four new types (which is possible to do without having to increase the size allocated for the type field in an IE): As an example, for the transmission from CC to the relaying station MT1 we added the type REL_DL1, which tells that this is the first part of a relaying transmission coming from the CC. From that point MT1 knows that it is going to receive data for relaying.

We tell MT1 where to forward that data in the following way: The CC includes in the FCH also the second part of the relaying connection, from MT1 to the destination terminal (MT2). Since this is a direct connection between these two terminals, it is placed into the Direct Link Phase. As a type of the IE for this direct transmission we use REL_DL2, and we put MT1 as the source and MT2 as the destination for that transmission, as specified in the H/2 standard for direct links. Thus, when MT1 receives the FCH, it knows that a relaying direct link transmission is scheduled from itself to MT2, after a corresponding relaying downlink transmission from the CC to MT1. Putting these pieces of information together, MT1 knows implicitly that the data coming from the CC has to be forwarded to the MT2.

As an example, we show how the Basic Data Transport works in a more complicated situation and also introduce a new trick which lets MT1 support more than one relaying connections within the same MAC frame.

Assume that the CC wants to send 3, 5, and 4 packets to MT2, MT3 (far terminals) and to MT1, respectively. We also assume that the CC wants to use relaying connections for MT2 and MT3 and uses MT1 as a relaying terminal. The CC then schedules the following transmissions: First it schedules a transmission for all the data packets (3+5+4 packets) at once to the MT1 in the Downlink Phase, using the IE identifier REL_DL1. In the Direct Link Phase of the MAC frame, it schedules the second parts of the relaying transmission, 3 packets for the MT1 → MT2, and 5 packets for the MT1 → MT3 direction.

Now, MT1 has the following knowledge concerning the structure of that MAC frame: It received 12 packets for a relaying transport, and has to transmit 3 to MT2 and 5 to MT3, as a part of the relaying transmission. So, it transmits the first 3 packets to MT2, the following 5 ones to MT3, and assumes that the rest 4 ones are for itself. Note that this mechanisms rests on the fact that MT1 can defer the identity of the packets from their point in time (the slot number in which they arrive at MT1). No additional information needs to be carried in the packet headers.

The uplink case is symmetric: in the FCH, slots are assigned to the various transmissions, and intermediate stations can reconstruct the amount of traffic to be forwarded to the CC by simple counting.

3.3 Acknowledgement handling of the SR Protocol

Although the SR protocol may be used in unacknowledged mode, acknowledged mode support is necessary for reliable connections. In the most simple solution, acknowledgements only happen "end-to-end" between CC and destination terminal; the relaying terminal has to store and may repeat packets in the case of reception failure. The Acknowledgement Handling Protocol takes care of lost acknowledgements and specifies what to do in that case in order to avoid the rescheduling of the whole relaying process. The disadvantage of simply rescheduling would be that wireless bandwidth is wasted (if a packet has already reached a relaying terminal, there is no need to send it again to this terminal) and that "end-to-end" packet error rates would increase dramatically (a packet would have to be sent successfully over two links instead of one).[2] Hence, the SR protocol takes into account the fact that packets that have reached the relaying terminal could be retransmitted locally. Currently we use the simplest "alternating bit protocol"; if an error occurs the transmission shall be repeated and the next ones are delayed (more sophisticated error control protocols are also feasible).

The CC schedules transports for the acknowledgement(s) coming from the MT2 through the MT1. These parts are the third and the fourth parts of the relaying transport. The connection between acknowledgements and packets happens via appropriate SCH types in the ACKs (for details, compare the appendix).

In order to spare some bandwidth we apply the following idea: The acknowledgement coming from the MT2 conveys administrative information about the transmission, the error result is only one bit. The relaying terminal MT1 forwards this acknowledgement but adds one more error indication bit, which is its own error result. Using this information the CC is able to decide whether it needs to reschedule the whole SR transport (this is the case when MT1 had a reception error, since then it could not forward the data to MT2), or repeat just the second part, the MT1→ MT2 transport (when MT1 received correctly, but MT2 did not). The protocol is also capable of handling lost acknowledgements; please refer to the appendix for details.

A possible design alternative would have been to use two separate exchange cycles: send data to the relaying terminal until acknowledged from the relaying terminal, then send data from the relaying terminal to the destination terminal (or the CC in the uplink case) until acknowledged. While simple, this approach has two main disadvantages: it adds detail as these two cycles could not be scheduled in the same MAC frame, and an acknowledgement from an intermediate terminal does not mean that the data has actually been delivered.

---

[2] How to choose transmission power and modulation schemes to compensate for such effects is a different issue, which we consider is other parts of the IBMS^2 project.

Hence, a failure of a relaying terminal (with data loss) could only be detected by higher layer protocols such as TCP. In our approach, a failing relaying terminal can in principle also be detected by the CC and rerouting could potentially be performed faster.

Note that using the ARQ protocol defined in the H/2 standard creates also so-called ARQ reply packets, which are replaced with the SR protocol acknowledgement packets in the case of relaying.

3.4 Conclusion of the SR Protocol

The protocol is able to handle two-hop relaying connections using the possibilities of the flexible H/2 MAC frame structure. Moreover, the CC is allowed to change the route from MAC frame to MAC frame according to network load or mobility situations without any protocol overhead. Since all trensmissions are controlled by the CC, the phases of the relaying happen synchronised, the transmission delay and delay variance is not increased.

Including further terminals in the relaying path should be possible in principle. Adding the required additional hops by using more IE in the FCH is straightforward, additionally, the semantics of the acknowledgements would have to be extended accordingly.

4 Validation

In order to practically validate our protocol extensions, we plan to use them both in a simulation environment as well as in a real implementation of a HiperLAN/2 stack.

4.1 Validation using a simulator

These protocol extensions form a part of the IBMS2 project, were we are interested in the use of ad-hoc based routing and relaying approaches to improve capacity and energy efficiency in wireless LANs, using HiperLAN/2 as a case study. In order to investigate these questions, we are currently developing a simulation model of HiperLAN/2 that incorporates our protocol extensions. This simulator is fairly complete and we expect to be able to test it before the conference.[3] We plan to make this simulator available to the public domain as soon as it is fully tested.

There do exist other HiperLAN/2 simulation models as well, however, none of these models quite met our requirements: The performance of H/2 under a link-layer perspective has been investigated by a group at Ericsson, resulting in C/I (channel to interference ratio) to packet error rate mappings [KJMST00]. While these results are used in our simulator, our interest rests with higher layer functionalities. A fairly complete simulator of H/2 has been developed in SDL [KEM], yet is (to our knowledge) only commercially available. Also, the use of SDL does not make statistical performance evaluation an easy task, which is critical for the assessment of routing and relaying protocols. The level of detail of

---

[3] Currently, only the implementation of the direct link phase is missing; the other protocol mechanisms are fully implemented and have been tested already.

this simulator appears to be very similar to ours. The CC selection process has also been studied by simulation [Kes00], yet the goal of this simulator is different from ours.

4.2 Validation using implementation

One of the goals of the IBMS^2 project is the construction of a real testbed, in which relaying protocol on the basis of HiperLAN/2 can be tested and evaluated. To this end, a joint effort is undertaken by the project partners: the hardware has been developed by Infineon, Systemonic, and IHP; the protocol implementation is currently underway at IHP, with the inclusion of the relaying protocol extensions described here. We expect to have a working prototype at the latest by the end of 2002.

5 Conclusions and future work

We have described a protocol extension to HiperLAN/2 that allows to use a terminal of HiperLAN/2 cell to be used as a relaying terminal in the communication between a cell's central controller and any other terminal. One main advantage of this extension are its principle conformance with the HiperLAN/2 standards as only a few additional packets are required to implement the relaying protocol — a means of extension which is provided for in the HiperLAN/2 standard. The additional implementation complexity and the protocol processing overhead are also expected to be small; preliminary results of an ongoing real protocol implementation look promising.

Functionally, the main benefits of our extension are the small protocol overhead combined with a large flexibility in scheduling relaying communications on the order of single MAC frames. While we are mostly interested to use relaying to improve capacity and energy efficiency, the protocol mechanism presented here should also be applicable to extending the coverage of a given HiperLAN/2 central controller (as the far terminal has to receive the CC's BCH and FCH transmission, asymmetric transmission power assignments for terminal and CC would be required to do so). Moreover, the mechanisms lend themselves to an extension to multiple relaying nodes within a single cell.

Based on this protocol implementation, we will study the capacity and energy efficiency problem when relaying is used. We expect to obtain results from simulation-based investigations, where the open problem is how to select relaying terminals on the basis of actually available channel measurements — effectively, an ad-hoc routing problem needs to be solved. Additionally, it will be interesting to study how the relaying protocol presented here can be integrated with existing inter-cell relaying protocols [Pee01]. Finally, once the protocol implementation and HiperLAN/2 hardware are completed, actual performance and power-consumption measurements will be conducted.

References

[BRA] ETSI BRAN. ETSI HiperLAN/2 extension for home environment,101761-4v1.1.1. `http://www.etsi.org`.

[ibm] Webpage of the IBMS^2 project. http://www.ibms-2.de.

[KEM] A. Kadelka, N. Esseling, and S. Mangold. Warp simulator. Project homepage at `http://www.comnets.rwth-aachen.de/~warp/what_about.html`.

[Kes00] C. Keszei. Hiperlan 2 CC selection. Master's thesis, Technical University of Budapest, 2000.

[KJMST00] J. Khun-Jush, G. Malmgren, P. Schramm, and J. Torsner. HIPERLAN type 2 for broadband wireless communication. *Ericsson Review*, 2:108–119, 2000. `http://www.ericsson.com/review/2000_02/files/2000026.pdf`.

[KM01] H. Karl and S. Mengesha. Analyzing capacity improvements in wireless networks by relaying. In *Proc. IEEE Intl. Conf. Wireless LANs and Home Networks*, pages 339–348, Singapore, December 2001.

[MHCN01] S. Mangold, J. Habetha, S. Choi, and C. Ngo. Co-existence and interworking of IEEE 802.11a and ETSI BRAN HiperLAN/2 in multihop scenarios. In *The 3rd IEEE Workshop on Wireless LANs: Technical Sessions*. IEEE, 2001.

[MKW01] S. Mengesha, H. Karl, and A. Wolisz. Improving goodput by relaying in transmission-power-limited wireless systems. In *Proc. of Informatik 2001-31. Jahrestagung der GI, OCG*, Austria, Vienna, September 2001.

[Pee01] J. Peetz. HiperLAN/2 multihop ad hoc communication by multiple-frequency forwarding. In *Proc. of Vehicular Technology Conference (VTC) Spring 2001*, Rhodes, Greece, May 2001.

[WEK$^+$01] H. Wijaya, N. Esseling, O. Klein, W. Zirwas, Hui Li, J. M. Eichinger, and A. Vidal. H/2 DLC protocol in hard real time systems. In *The 3rd IEEE Workshop on Wireless LANs: Technical Sessions*. IEEE, 2001.

A Single relay protocol description for HiperLAN/2

This section contains a more detailed description of the HiperLAN/2 relaying extensions. In particular, it defines the additional packet types (information elements and SCH types) as well as describes the protocol behavior.

A.1 Abbreviations

CC	Central controller (access point)
MT1	Terminal near to the CC
MT2,MT3	Terminals far from the CC

Table 1: Entities used in the protocol description

The entities that are used in the following discussion are defined in Table 1. We assume that:

- MT1, MT2 and MT3 are already associated to the CC. It implies that the CC has a database about the link capabilities of the terminals.
- The CC measures (and keeps updated) the Radio Map of its H/2 cell (see ETSI 101761, 6.5: Link Quality Calibration for DM operation)
- Based on these measurements, the CC has (somehow) decided to use MT1 as a relay terminal for MT2 and MT3 in all our examples.
- In the case of uplink Relay the data path is: MT2 (and MT3) → MT1 → CC
- In the case of Downlink Relay, the data path is: CC→ MT1 → MT2 (and MT3)

We added the Information Element types as described in Table 2 to be used in the Resource Grants in the Frame Control Channel; additional SCH types are described in Table 3 and 4.

A.2 Downlink Single Relay

Assuming that the CC has some LCH packets to send to the MT2, it uses the Radio Map to decide whether relaying is used or the ordinary downlink mode. In case of relaying, it schedules the following data transfers in the next MAC frame (see also Figure 4):

- Data transmission for the LCH packets, CC → MT1, in the downlink phase, with IE type RL_DL,
- Data transmission for the LCH packets, MT1 → MT2, in the direct link phase, with IE type RL_DDL,
- One SCH for acknowledgement, MT2 → MT1, in the direct link phase, with IE type RL_DUL. MT2 will send an RLC_ACK_1 message in this slot, which contains the DLCC_ID of the relaying transmission, the MAC address of the far terminal (MT2), the result of the transmission (1 bit, everything was received correctly or not),
- One SCH for acknowledgement, MT1 → CC, in the uplink phase, with IE type RL_UL. MT1 will send an RLC_ACK_2 message in this slot, which contains the

DLCC_ID of the connection (which is valid for only MT2, so copied from REL_ACK_1), the MAC address of the far terminal (MT2), and the result of the transmission which is one of the following cases (2 bits):

1. All reception is OK,
2. MT1 reception is OK, MT2 reception has failed,
3. MT1 reception has failed,
4. MT1 reception is OK, No ACK received from MT2.

The CC continues in the next MAC frame based on the acknowledgement from MT1 as the following, consequently:

1. No error, the same procedure repeats as long as there are packets to send,
2. The CC schedules one downlink SCH for MT1 and the same number of LCH packets for MT1 → MT2 as in the previous MAC frame. The CC sends an SR_ACK_REP message in the scheduled SCH to MT1, and MT1 repeats the last transmission for the MT2. The IE type of the SCH shall be RL_DL,
3. The CC schedules the same number of LCH packets as in the previous MAC frame for a complete SR downlink procedure, and transmits the same data again,
4. The same procedure as in 2.

When MT1 receives a grant with IE type=RL_DL, it prepares for acting as a relay. In the case when MT1 is scheduled to relay for more than one far terminals, the CC shall schedule all LCH packets in one RL_DL IE. Since the CC also schedules the second phase of the SR transmission to the direct link phase, MT1 knows how many packets it should send and to whom.

For example, CC schedules SR transmissions to MT2 (2 LCH packet) and to MT3 (3 LCH packets) through MT1. In the Downlink Phase it sends 5 LCH packets to MT1, and schedules 2 LCH packets from MT1 to MT2, and 3 LCH packets from MT1 to MT3 with IE type=RL_DDL. Since MT1 also reads the FCH it knows that the first 2 LCH packets should be sent to MT2, and the rest three to MT3.

MT1 waits for an acknowledgement (REL_ACK_1) from MT2 (and MT3) after sending data to them. This acknowledgement is scheduled for the direct link Phase, after the data packets sent by MT1. The scheduler shall ensure a time gap for MT2 for calculating CRCs and constructing a REL_ACK_1.

After the time slot scheduled for the MT2 to send the REL_ACK_1, MT1 constructs a REL_ACK_2 message and sends it to the CC in the uplink phase, as scheduled.

A.3 Uplink Single Relay

Since the direction of the data transmission is reversed, one complete uplink SR cycle will be overlapped on two adjacent MAC frames.

Assuming that MT2 has data to send to the CC, it sends Resource Request(s) first. Basically MT2 has to send RRs directly to the CC. However the CC may schedule USR SCHs for MT2 periodically to avoid the direct communication. The CC uses also these SCH grants to poll an acknowledgement from MT2, MT2 shall differentiate between these two possibilities using the following rule:

- If the MT2 has at least one packet which it didn't send positive acknowledgement about, MT2 shall use the scheduled uplink SR SCH to send the REL_ACK_2 message,
- If the MT2 has acknowledged all packets it received, positively, it may send RR in the uplink SR SCH. In the case when MT2 has not any data to send, it sends dummy SCH in the uplink SR SCH.

When the CC polls the MT2 for a new Resource Request, it shall also schedule an RL_UL slot for the MT1. MT1 shall relay the SCH received from the MT2 to the CC.

Note that in the case when REL_ACK_2 was not received correctly by the CC, CC polls MT2 with scheduling USR SCHs for the MT2 (with IE type RL_DUL). If MT2 responds a dummy SCH packet or a new Resource Request, the previous lost acknowledgement was positive. In the case of lost negative acknowledgement, MT2 shall repeat that acknowledgement.

We assume that CC has received Resource Request(s) from MT2. CC shall schedule the following data transmissions to the next MAC frame:

- Data transmission for the LCH packets, MT2 → MT1, in the direct link phase, with IE type RL_DUL,
- Data transmission for the LCH packets, MT1 → CC, in the uplink phase, with IE type RL_UL,

At the beginning of the next MAC frame:

- One SCH for acknowledgement, CC → MT1, in the Downlink phase, with IE type RL_DL. The CC will send a REL_ACK_2 message to the MT1, which contains the DLCC_ID of the relaying transmission (which is valid for only MT2!), the MAC address of the far terminal (MT2), the result of the transmission (2 bits)
- One SCH for acknowledgement, MT1 → MT2, in the Direct Link phase, with IE type RL_DDL. MT1 will send a REL_ACK_2 message to the MT2, which contains the DLCC_ID of the connection (which is valid for only MT2!), the MAC address

of the far terminal (MT2), and the result of the transmission which is copied from the REL_ACK_2 received from the CC, and one of the following cases:

1. All receptions are OK,
2. MT1 reception is OK, CC reception has failed,
3. MT1 reception has failed,
4. MT1 reception is OK, No ACK received from the CC.

The CC continues in the second half of the MAC frame based on the acknowledgement from MT1 as the following, consequently:

1. No error, the same procedure repeats as long as there are packets to send,
2. The CC schedules the same number of LCH packets for MT1 → CC as in the previous MAC frame. The MT1 repeats the last transmission for the CC. The IE type of the SCH grant shall be RL_DL,
3. The CC schedules an RL_DDL SCH message for MT1 → MT2, and the same number of LCH packets (and signaling) as in the previous MAC frame for a complete SR uplink procedure.

   MT1 shall send an REL_ACK_2 message to the MT2, which contains reception errors. MT1 can decide whether to send error indication in that REL_ACK_2 message, since it knows if the reception from MT2 was OK or not.

   After receiving the negative REL_ACK_2 message from MT1, MT2 shall send the same data packets as in the last SR cycle. The CC knows about MT1 reception errors, since MT1 sends dummy packets in place of not correctly received ones,
4. In this case MT1 and MT2 assumes that the CC had reception errors.

   If the CC schedules the same amount of LCH packets in the next SR cycle, Both MT1 and MT2 (and MT3) repeat the last data transmission.

   If the CC schedules space for less or more LCH packets than in the previous SR cycle, MT1 and MT2 (and MT3) assumes that the reception was OK, and continues transmitting the next packets in their queues.

A.4 Data formats

The following Tables 2 to 5 describes additional protocol messages and bit positions.

Identification	Bits (IE type)	Description
RL_DUL	1011	Uplink relaying Direct data transfer: MT2 → MT1
RL_UL	1001	Uplink relaying data transfer: MT1 → CC
RL_DL	1000	Downlink relaying data transfer: CC → MT1
RL_DDL	1010	Downlink relaying direct data transfer: MT1 → MT2

Table 2: Additional Information Element types

Identification	Bits (SCH type)	Description
SR_ACK	1011	Acknowledgement message for Single Relay connections

Table 3: Additional SCH type

Identification	Bits	Description
SR_ACK_1	000	Acknowledgement sent after the first phase (from MT1)
SR_ACK_2	001	Acknowledgement sent after the second phase (from CC or MT2/MT3)
SR_ACK_REP	010	Request for repeating the last SR transmission
Future use:	011-111	

Table 4: Additional SR_ACK Type fields

Identification	Num. of bits	Description
SR_ACK_TYPE	3	Sub-type, see table 'SR_ACK Type Field'
MAC	8	MAC address of the far terminal (MT2, MT3)
RELAY_MAC	8	MAC address of the relaying terminal (MT1)
DLCC_ID	6	DLCC_ID of the data connection of the far terminal. This DLCC_ID is always the DLCC_ID established between the CC and the far terminal (MT2, MT3). The relaying node is hidden from the point of view of DLCC_IDs in the far terminals.
DIR	1	Direction of the relaying (data packets): 0=downlink, 1=uplink relaying
RES	2	Result of the Reception. In the case when SR_ACK_TYPE=SR_ACK_1, only the first bit is valid: 0: Reception is OK 1: Reception has failed SR_ACK_TYPE=SR_ACK_2, 00: All receptions are OK 01: MT1 reception is OK, client reception has failed 10: MT reception has failed 11: No ACK from the client SR_ACK_TYPE=SR_ACK_REP, can be set to any values
Future use:	24	
Total:	52	

Table 5: Contents of SR_ACK_xxx (000 - 010)