IEEE P802.11
Wireless Access Method and Physical Layer Specifications

Title: Gear Shifting Proposal

Authors:

Pablo Brenner
LANNAIR Ltd.
Atidim Technological Park
Devorah Haneviah St. - Neve Sharet
Tel Aviv 61131 - ISRAEL
Voice: +972-3-6459127
FAX: +972-3-6487146
EMail: pablo@lannet.com

Michael Rothenberg
LANNAIR Ltd.
Atidim Technological Park
Devorah Haneviah St. - Neve Sharet
Tel Aviv 61131 - ISRAEL
Voice: +972-3-6458422
FAX: +972-3-6487146
EMail: mrot@lannet.com

Abstract:

This submission presents a proposal for the PHY and MAC hooks that will enable support of a multiple bit rate BSS. These hooks provide both a migration path towards faster bit rates, and the ability of letting each station make its own range/speed tradeoff decision.

Introduction

During the last months there has been lots of discussions about the desirability of multiple bit rates, whether this should be upon registration or gear shifting, whether the speed improvement is worth the added complexity to the MAC, and so on. This submission presents the authors' conclusions, together with a concrete proposal on what are the hooks that are needed, and how can they be used to support a gear shifting scenario.
Most of the contents of this document are not new, and were discussed over the email reflector during the last weeks.

The need for Multiple Bit Rates and Gear Shifting

There is much consensus on the need for a migration path for faster bit rates, the remaining question is whether it should be handled as different PHYs or as multiple bit rate PHYs. Leaving the semantics aside, it is hard to disagree that, from the user point of view, it is quite desirable to be able to keep using his "old" cards, while taking advantage of the new, faster cards.
To support this requirement there are 2 main approaches, allowing multiple bit-rate BSSs or to link the two different speeds across the DSS (adding a new AP), the differences on both approaches are:

- **Cost:** Seems that there is no significant difference between buying a second AP to support the new cards, and upgrading an old one, but we believe that anyhow the upgrade cost should be lower than a second AP..
- **Channelization:** Since the MAC Protocol is not very suitable for overlapping BSS on the same channel it is clear that the multiple-AP approach is not suitable. When using a multichannel PHY, the upgrading is "wasting" channels.
- **Performance:** On the Multiple APs approach we get better performance when two "high speed" stations talk to each other, but we get lower performance for "mixed stations" traffic, and higher delay.

The other requirement, is the requirement for a range/speed tradeoff, even though the user will usually make this tradeoff during the purchasing/installation phase, by making some measurements and some modeling, hence deciding how many and where to locate the APs, there will always be a couple of semi-hidden locations where "high-speed" stations will have less probability of working than lower rate stations, or some remote places where usually there is no activity but when a mobile station roams through this area it is desired to keep the connection active (even at a lower rate).
The main problem with this approach is that there is a performance penalty even when this is no needed, this is solved by defining two BSS parameters: BASIC_RATE, HIGH_RATE, so when it is not required, the user will set the two parameters to the same value.

The main question remaining is how much complexity are we adding to the MAC design, to provide these (more or less) desirable features, a simple set of hooks and a gear shifting algorithm are described in the following paragraphs.

Proposed Gear Shifting Algorithm

The basic idea is the following:

There are two (or more) predefined rates, which are the same for the whole BSS.

One of them, the BASIC_RATE MUST be supported for all the stations belonging to the BSS.

The other N rates are optionally supported by each of the stations, including the AP.

- The **Preamble** and the **PLCP Headers** are transmitted always at the **SYMBOL_RATE** (which could be the equal or less than the BASIC_RATE)
- The different **IFS**s are predefined in **SYMBOL_TIMES**.
- All the **Control, Multicast and Broadcast Messages** are sent on the **BASIC_RATE**.
- All the **Data Messages** sent **without RTS/CTS** are sent on the **BASIC_RATE**.
- A new field is added to **RTS messages**: **Requested_speed**, which indicates the maximum speed that the sender wants to transmit the data.
- A new field is added to the **CTS messages**: **Granted_speed**, which indicates the bit rate at which the Data transmission should take place, the decision about this rate depends on the **Signal_Quality** of the **received RTS**, and is less or equal than the minimum between the Requested_speed of the corresponding RTS and the maximum supported speed at the receiving station.
- **Data Messages** belonging to an RTS/CTS MPDU are sent at the **Granted_speed**, or if the Granted_Speed is not supported by the transmitting station, at the BASIC_RATE.

There are 2 parameters that are important to be transferred per frame through the MAC/PHY interface and they are:

- **Speed**: which is used by the MAC to indicate which bit_rate should be used, and on receive indicates the bit_rate at which it arrived, and
- **Signal_Quality**: which is passed from the PHY to the MAC, and is used by the MAC to decide whether a gear shifting is appropriate or not.

Note that it is highly desirable that the CCA function will be such that a station supporting only the BASIC_RATE will be able to recognize channel activity at higher speed rates, this is pretty simple if we use the same SYMBOL_RATE.

The NAV vector will be updated as following:

When an RTS is seen, the NAV will be updated according to the Requested_Speed.

When a CTS is seen, the NAV will be updated according to the Granted_Speed.

There is special case when a station "heard" the RTS, but didn't "heard" the CTS, and there is a difference between them (as specified before, the Granted_Speed can only be lower or equal to the Required_Speed), hence this outsider station will set a shorter NAV than required, this will not be a problem, since the station that "heard", and succeeded to decode the RTS message, is supposed at least to recognize channel activity when the same station is sending the Data message, and the result of or'ing the Carrier Sense and the NAV, will be exactly the same as if the CTS was heard.

Performance Analysis

A throughput capacity analysis was performed, and shown below, a real performance analysis would require simulation of traffic patterns and channel modeling.
Note that the described algorithm performance adapts to the number of station supporting the high rate. Only the Control Messages, Multicast messages, RTS and CTS will be always in the BASIC_RATE regardless of the Source and Destination Stations. The Data Messages which should be the majority (that's the throughput provided to the upper layers) will be in the highest speed supported by the two stations.

For the capacity analysis the following parameters were used:

SYMBOL_RATE = 1 MSymb/s
BASIC_RATE = 1 MBit/s
HIGH_RATE = 2 MBit/s
SIFS = 23 symbols = 23 microsec
Preamble+PLCP Header: 112 symbols = 112 microsec.
RTS = 16 Bytes
CTS = 8 Bytes
ACK = 6 Bytes
Data Header = 24 Bytes

The following equation gives as the time required for transmitting a Data Message of "length" bytes length:

t(length) = 4 * (Preamble + PLCP Header + SIFS) + 8/Basic_rate * (RTS+CTS+ACK) +
8/High_rate * (data_header + length)

This gives the following results:

Data length	Time in 1 Mbit/s	Time in 2 Mbit/s	Time in 1+2 Mbit/s		% Improvement 2 Mbit/s	% Improvement 1+2 Mbit/s
100	1580	1060	1180		49	33
200	2380	1460	1580		63	50
500	4780	2660	2780		79	71
1000	8780	4660	4780		88	83
1500	12780	6660	6780		91	88
Maximum Throughput	**0.93**	**1.8**	**1.76**			
Through @ 500 byte	**0.83**	**1.5**	**1.43**			
Parameters	**Value**	**Units**				
Preamble + PLCC	112	micro				
Window slot	23	micro				
RTS	16	byte				
CTS	8	byte				
ACK	6	byte				
Data header	24	byte				

Performance Analysis Conclusions

From the above shown figures we see that for packets bigger than 500 bytes, we get a significant performance improvement using the above specified algorithm.
Since throughput is usually important for file transfers and similar operations, and these use the bigger packet length available, even if we define an MTU of 500 byte, the higher rate stations will get a significant improvement.
Even though, the figures show that from the performance point of view, the MTU size is of extreme importance, and that an 1500 byte MTU is recommended, in the case that fragmentation is needed to support such packets, the fragmentation algorithm, should be such that there is a single RTS/CTS transaction for the whole MPDU.