Tentative MAC/PHY Joint Meeting Minutes
Tuesday, November 11, 1992

The meeting was called to order by Vic Hayes at 8:30 AM. Carolyn Heide secretary. Vic briefly went through the week's agenda for people who were not here yesterday.

(1) Announcements

Vic has a report from 802.11 vice-chair Rich Lee who is represented 802.11 at the Joint Experts Meeting this week in Washington DC. His presentations went well, there were a few questions and much interest was shown in private comments. TIA indicates in its NPRM that PCS spectrum, for unlicensed users, should support data and voice. Tom Stanley of the FCC reminds the ET that PCS spectrum doesn't stop at 1990 MHz. John Reed of the FCC, who will be involved in technical rule making, pointed out that the proposed rule deliberately set strict standards, like the tolerance levels, with the expectation of relaxing during negotiation. Slightly late filings should be no problem, but 802.11 NPRM comments will be important please make best effort possible in La Jolla.

Vic has, and will distribute, a draft of the CEPT recommendation - careful, this is not a final copy there are some editorial differences. IEEE P802.11-92\132 will be the number of this document.

(2) Group Chair for this morning: Tom Phinney is available and willing.

(3) Presentation of Submissions

P802.11-92\123, by Larry van der Jagt

The paper addresses issues of simulation, how to benchmark MAC proposals and what's a realistic environment in which to do that. It presents a Mathematica based framework for placing stations in a 3D environment and uses a model to calculate the path loss between stations and to execute a MAC state machine.

Proposing that a framework like this could serve for a common group model for bench marking proposals. The cost of Mathematica is such that it is available to all.

for statistical model. There are three parameters set up first: (1) the mean value of the attenuation coefficient; (2) is the variance on that; (3) is local fading variance. In this model you select the attenuation coefficient 'n', look at the distance between stations 'd', find $1/d^n$ and this is the mean for the local fading. You can get as much or little fading as you want that way. Attenuation can be calculated at many distances with this set of assumptions.

Wim Diepstraten: parameters set per station or per a per link basis?
Larry: attenuation is based on distance, a random sample from this statistical model.
Wim: typically that would be distance dependent.
Larry: the local fading has a much wider variance so you calculate the mean for the local fading then the last parameter is the standard deviation of the local fading. It is simplistic, but gives attenuation as a function of distance. Raleigh fading assumes you don't know where the stations are, we know where the stations are so that model doesn't apply here. A log normal could have been used, but this is equally valid.

Stations get defined according to their attributes which are functions - location (x,y,z); tx power; message probability. A traffic generator feeds the tx queue which is a list of stations to which this station wants to tx. The rx queue gets populated during state mach execution. An originating station has a function which calculates the arriving power level at the rx station as a result of all currently originating stations. This gets compared to a capture margin parameter for the rx station to determine success of the receive There are statistics kept for tx attempts; rx success and failure. There is a retry count which is the max. retry attempts allowed on tx before removing the packet from tx queue.

You can define as many stations as you want (5 used for this example), they can be clustered however you want.

Wim: (referring to the receive power level diagram on display) this is a snapshot of the power levels in a nominal case and the color represents the states?
Larry: color represents state of station. Bar height represents rx power level.
Wim: does not represent which station is currently txing?
Larry: in this case yes it does - any station in the originating state right now is txing. Only one guy transmitting at a time is spectrally inefficient. MAC operation in this needs to be evaluated. But the model works if only one transmits at a time too. There are parameters that can be set for the background noise and interference.
unidentified: is there a dominant signal in this picture? Can any extra attenuation, such as floor effects, be set?
Larry: the whole thing gates off an attenuation matrix. There is no floor in effect in this particular picture. The deviations are set to zero also. This MAC is unsophisticated, but the model supports more complicated MACs.

You set up a list of all stations which includes minimum power parameter (which avoids divide by zero problems) An external noise table can be set-up at any station. This allows examining the impact of a large noise source at a particular station. A capture margin for each station is used to evaluate success at that receiving station. If the signal exceeds this margin the rx is successful. The attenuation table takes the station list and generates a matrix (assuming reciprocity in this model). Generation of traffic uses a Poison model to build a tx queue size with destination for each queue entry.

This is not a MAC proposal, or analysis of a proposed MAC - a simple MAC is used to illustrate the model. For this MAC the rx queue size is one.

Wim: response from receiving station - is there an ack?
Larry: to determine if this works or not, if I send something to Joe then he has to get a message back to me to determine success. One direction working only and not other causes failure.
Wim: next packet originated from the station depends on the response?
Larry: no. There is a retry limit and if it expires the packet is deleted from top of the tx queue and a failure is counted.
Wim: model knows packet sizes?
Larry: not right now. If you propose a MAC with varying packet sizes you would build that in. Larry is currently working on other models of this that do more reasonable things. This unrealistically simple MAC simply illustrates the framework that can be used.
Wim: this is a hypothetical MAC just for demo purposes, in an environment where distance is the most important, plus all the variances of the channel model.
Larry: the channel model is a statistical model that accounts for the distance between stations and for the fact the you know where the stations are.
unidentified: attenuation is the same throughout - you have one value and keep it through the whole simulation run.
Larry: it depends - each loop generates traffic. Right now I leave it the same, but in that loop the attenuation could be changed each time through. A more reasonable a next work item is to get attenuation specified by station. I.e. for a fixed station you wouldn't update, but for a mobile station you would, and a slow or fast update could be selected. The function calculate attenuation matrix can be called from anywhere.
Wim: if you send a packet to destination 'x', then at the next transmission to 'x' do you use the same fading variance, etc.?
Larry: right now a new attenuation matrix is not calculated ...
Wim: but fading variance is fixed for every attenuation matrix constructed?
Larry: right now, yes. But the fading characteristics could by changed in the loop too by calling the set fading parameters function.
Wim: including floor or wall effects?
Larry: in the build attenuation matrix you could build in a bias on the i/j component if there was a wall between i and j. A fixed value would be added. Or you could change the fading statistics to do that.

The concept Larry is trying to present is that Mathematica can be used to form a realistic framework

Larry: yours takes average numbers (it is unrealistic). It doesn't account for specific situations in geography, external interference and internal interference. Don't know what yours are written in (BASIC?) - you can present a framework.

accessible to all

Wim: why Mathematica?
Larry: no one else has suggested anything else. I can't afford an expensive simulation package.
Wim: what is the difference between your approach and mine?
Wim: I did.
John Corey: how to we get Mathematica and how much is it?
Larry: about $700 for Macintosh extended version, $400 for not extended. Similar price for the Windows version. About $1500 for a SUN. General purpose UNIX perhaps a couple of $1000. It has great graphics capabilities.
John McGown: for $400 I can put this whole thing on my Powerbook?
KC Chen: problem: the station may move around at any point in the simulation Success does not rely on high receive power. This is different from the bit error rate.
Larry: within this framework the signal to interference ratio at an instant point in time is an indicator of bit error rate. You could derive this from not only power but by sing a ray tracing model to come up with a complex input model and apply a modelling technique to find out what really happens. This is an illustration to keep things simple. You could add your own power management so that you can't rx while you tx. That is an implementation issue - it could be changed to do that. In such a simple MAC as simulated here that doesn't work.
John C: if only power is considered, not delay spread, you are assuming all stations have effective echo cancelers. Is that as simple as it is? Mathematica is a good tool because of availability and documentation is very good. Larry writing scripts provides written documents too.
Larry: yes. This is not adequate yet, but you could expand it to be so. This submission is itself an executable program. Also it runs on many different platforms - PCs and MACs, and when the little machines run out of power simulations can be transferred to those with bigger machines to help.
Chandos Rypinski: model is primarily for peer to peer, or could a particular station be made special (such as antenna height)?
Larry: that could be added as a station parameters, affecting the attenuation factor between that station and others. Or by adding a new parameter to the list of attributes of a station. This is just solving formulae. The channel modelling group would provide input into what they would like to see in the model and we could extend the model with this is the basis of conformance testing.
Michael Rothenberg: perhaps it would make more sense to have the propagation matrix associated with a station, not a location and have the station own a location attribute. Then mobility will not change the matrix - the space will be mapped as in reality, where the station location moves. Don't change the matrix.
Larry: the metrics are a function of the station location as well.
Michael R.: this wouldn't add complexity - it's just a different way of doing the same thing.
Larry: my question is - look to see what the state of all station is. This effects what is going on at any location. The state of every station affects the matrix. Possibly that is a good approach too - the point is this is an extensible, accessible tool. Output can go from the poor to the rich. Larry ran this on a 5 Meg RAM MacIntosh-II, using virtual memory set to 8 Meg and the calculation of attenuation is immediate. When running an iteration each step comes up a line at a time with about enough time to read each line before the next comes up, on a 16 MHz 68020 with a math coprocessor..
Wim: interference - is evaluated as the sum of all interferers at a given station?
Larry: yes.
Wim: applying a fixed boundary of receive level?
Larry: you can watch interaction of failure by varying this.
Wim: performance - lines come up at readability speed. What does that mean? A couple of messages per second?
Larry: number of station define the sline printout - would run faster if you didn't print until the end. I haven't tried that yet. Don't know what this translates to in messages per second. But it is portable and

can be speeded up by using better machines. Larry would like to see us choose this the simulation modelling tool
unidentified: a station based model for hopping can be added?
Larry: yes, by changing the capture margin to be parameterized. Mathematica is hard to get used to, but good when you are used to it. Most of the time spent so far has been in doing the basic functions, which are done now. This took 4-5 weeks.
John C: will you make the function library available to the group?
Larry: not now, but will think about it - haven't been able to find anyone willing to pay for it, but not sure if it should be given away yet. Maybe.
B Ichikawa: there is currently a $99 special being run for Mathematica Windows version.
Larry: the output can be exported in encapsulated Postscript and put into desktop publishers. A note to the issues editor - issues addressed by this submission are noted in the document. The summary point is that a framework needs to be chosen for simulation work.
Bob Crowder: is sensitive to the issue of giving things away, but if the functions aren't part of the framework, then how can this be used to compare MACs?
Larry: the details of what the functions are is in the paper. The details of how you do code - you would benefit by having my code, but you can write your own. This is a framework because it describes the functions.

P802.11-92\125, by Larry van der Jagt

This is an implementation of the parameterized MAC/PHY interface addressed in IEEE 802.11-92/99.

The transmit scenario: the MAC in station 1 decides, based on a table he has, to set-up the PHY in this way to talk to this destination address. For instance, use 1 watt to talk to this guy. The MAC puts this in the remote control parameter as part of the MPDU and sends it to the PHY. The PHY does as told, and transmits the MPDU. The receiving PHY does everything it has to make the receive succeed and provides the local control parameter to his MAC - for instance saying I just received at 100 mW. The MAC decodes the MPDU and now knows the transmit power level. The MAC now stores the remote and local parameter vectors and the source address. Now he has an picture of the path loss between him and this source station. There is also s parameter mapping function that has been uploaded at the beginning of time. So the MAC can take the stored remote and local control parameter vectors, plug them into the parameter mapping function and get the parameters for transmitting to this station.

This means any station that wants to come on and listen promiscuously and hear path loss and the antenna diversity information, and the stations per channel etc. - these can all be determined by listening. This can achieve PHY independence.

Colin Lanzl: this assumes that communication has been established between these two stations. If station 1 doesn't know station 2 is there, the first time he may be at wrong setting.
Larry: or be promiscuous for a while first. This is intended to optimize media access control.
John C: if you listen first to the station to whom you want to transmit, then determine that you must set power level to 'x', then feed that information into that vector and into the PHY - are you actually plugging the value in or a vector that gets looked up?
Larry: whatever you agree goes in there.
Dave Bagby: whatever is agreed - by whom?
Larry: 802.11, if we want interoperability. If not, a manufacturer can do his own thing, but he would not conform to our standard. For a given PHY we should specify this.
Tom Phinney: for remote parameters you have to *have to* agree in the standard. For local you *can* agree in the standard. The upper part has to learn what the lower part is saying.
Wim: remote control parameter vector - if it in every packet as a header, or just in some kind of initial session?
Larry: it is overhead in every packet.
Wim: as a suggested power level and silence level, etc.
Larry: yes, this is a general format for doing that. This is a good mechanism for adaptive power control
Dave B: round trip aspect - what are Larry's thoughts about the validity of the information by the end of round trip?

Larry: depends on the time between the beginning and the end of the trip. TDMA will say that you use the same channel, and FDMA uses separate channel. The validity of the information depends on how static the environment is, the time since you last used that path, and whether you are listening constantly. You can get information from other stations - messages don't have to be for you for you to learn from them.
Dave B: this assumes that the environment is sufficiently stable that the information is of use by the time you need it.
Wim: remote control parameter vector only being of use to the destination station - in power level this is not necessarily so, but for a thing like bit rate that is different.
Larry: promiscuous listeners can find out where to take the first guess at a starting point - baud rate for instance.
Wim: unless the baud rate was chosen to suit the last person to whom he was talking.
Tom P: it can never hurt to have this information, although an ageing factor might be useful. This information may be misleading, but it can never make you worse off than having no information.
Larry: perfect information about the transmitter would increase capacity - this helps get there.
John C: do you show the management function here just because of the standard interface drawing? Or could this table be queried from other stations through the management function.
Larry: this option is available to the MAC designers. This could be used for all kinds of things, like deciding whether to use the AP or go directly to a peer. Or as a MAC/PHY independence tool. Or to allow individual vendors to do things in the PHY without burdening the MAC - the MAC stores, executes functions and puts the vector in the MPDU ignorantly.
unidentified: this is better for AP use, more than end node. It is asking a lot for a cheap little node, but an AP can do lots of things.
Larry: there are a lot of problems, trying to balance the complexity of what we could do - to limit what we can do in an end node.
John M: nothing is too complicated to be cheap - anything made in large quantities will be cheap.
Tom P: we may find that a local address header is not appropriate for end to end (i.e. all the way through the distribution system,) but is for AP to end, trying to look at a local environment. Use a local 8 bit label in the PHY layer or something. But the basic approach has merit - if the transmitter knows what the receiver did it will help.
Bob C: if a PHY didn't need them, these parameters could be present but null values or always the same used. This keeps it simple if needed. Letting the MAC build a more complex picture of world is good.
Wim: in a mixed environment receivers don't have to use all the information.
Dave B: is concerned if lots of information goes into the MPDU - watch out for growing overhead.
Larry: but it increases capacity too.
Wim: specific parameters?
Larry: not ready to say. The groups need to make proposals. Needs time to get proposals that use this tool first.

P802.11-92\99, by Jonathon Cheah

This document maps out the overall generic idea of the parametric MAC issues. Larry defined a detailed use of it, this document describes the theory..

We have 'n' different OSI upper layer function and, we have 'm' different parameters overcoming different PHY structures, requiring diversity in the MAC from both sides. But having to use a huge MAC for a simple PHY to handle all this is unnecessarily large and expensive.

The MAC can be separated into 2 halves. (1) OSI type management functions, the parameters coming form the top layer. This is a limited set. (2) Then there is the bottom part which implements the PHY layer independence. Define groups of parameters, where each PHY can then define its own parameters to make the MAC work by loading its own state machine. with a loadable set of equations. In the PHY independent portion - what parameters exactly requires a limits test. Individual PHYs think of different sets - there needs to be some universal set.

Jonathon has grouped these parameters into categories:

mac_driven_phy_passive: we define 'n' of these for all possible things we can think of to cover all possible PHYs.

mac_driven_phy_auto - channel related things that PHY can take care of itself. MAC tells PHY do something, PHY decides how to do it.

mac_driven_phy_response: PHY says OK, I did what you asked.

phy_driven_flags: parameters transferred to MAC but without MAC instructions, i.e. notification of an event.

signal_quality: a category by itself because in radio bit by bit integrity of the transfer needs to be taken care of. Also necessary for FEC. Just leave it blank if you don't need it.

phy_kill: anything that goes wrong in radio, the gospel is kill that transmitter. This is not a parameter this is an essential command.

If you keep these lists small, the MAC will be easy and cheap.

Implementation: structure borrowed from the field bus standard. Jonathon likes designing around a DTE/DCE interface because its straight forward. You have a MAC, choose a PHY, primitive driven instructions for the PHY live in silicon that comes with the PHY. The local management translates the upper level parameters.

Discussion
Colin: signal quality and PHY kill - aren't they special cases of the others?
Jonathon: absolutely. I broke them out because if implemented in a slow speed environment with FEC, if they are together everything has to be fast.

(4) General Discussion

Dave B: would like to get some feel for various kinds of parameters that PHY group has considered.
Larry: in François's outline, channel select, power and rx level are the only common uniform parameters determined. The PHY group is not there yet.
John C: what parameters might be affected by mobility?
Jonathon: fast reacting decisions may be required from a PHY autonomous layer in the mobile environment. Don't want to go all the way up to the MAC for these decisions in a mobile station. Keep the mobility responsibility in the PHY as much as possible - or no higher than the PHY independent layer. Long term things can propagate further up the stack.
John C: concept of multiple service areas - the PHY will be able to detect a new service area? John starts to see mobility imposing a lot of information exchange between management and MAC.
Jonathon: this is one of many things, but this is one of the slowest problems in a mobile unit. PHY can only pass this information up and have the MAC make the decision. This is a slow response.
Nathan S.: these issues fall into 2 categories: one has to do with reliable channel; and the other to do with access the medium. The latter has to be in MAC or above. The former should be in the PHY otherwise the MAC becomes PHY dependent.

Dave B: asks for clarification of DTE/DCE interface subject. These diagrams don't make clear what this thing labelled DTE/DCE interface is. Brings to mind modems - 2 ends, one to the media and one that is thought of as the DTE/DCE interface. How is that analogous to the pictures we see here now.
Tom P: motivation comes from experience in other groups (802.4) wanting to not expose the variable part of the PHY to the fixed part of the PHY.
Dave B: concept is there is a logical partition embodied in a physical split, in the PHY? But at a different place from the MAC/PHY interface
Tom P: this is not a layer boundary, you unplug a part of the PHY layer. DTE/DCE interface is used to take the variability and separate it. You want to plug replace the variable part.
Jonathon: likes the DTE/DCE interface because it allows the PHY independence concept. Allows plug replaceable media - DS to IR for instance.

Larry: returns to the simulation submission. Note the pros and cons for issues 29.1: functions on a lot of platforms; low cost, accessible to all; upward compatible to group members with more power; links to C programs; Postscript output for document preparation; supports self documenting submissions directly from Mathematica

Chan: has no quarrel with Mathematica as a tool. The issue is what algorithms and assumptions are used. That is harder to agree on. Everyone should use the tool of his choice. No simulation is any better than the assumptions that went into it, and we are a long way from that. It is, however, obviously desirable to have one common package.

François: pros for issues 29.1 - does anyone object to simply reference Larry's paper [sec note: no one does.]

Tom P: everyone has the right to use their own tools, but distribution is easier in a form everyone can use. There has been an immense amount of work done by several people to date. This is a joint decision - MAC people need physical attribute models but can't develop them themselves. What is the sense of the group - whether it's useful to have an agreed on tool for sharing of simulation results?

unidentified: the program is not the framework, the framework is the procedures, the modelling assumptions. The program by itself is not that useful. If we have the framework people can modify it for their own tools.

Jim: is in support of a common tool. We thought that might be Extend, a DOS version is not planned for release, so that is probably not a valid alternative.

Nathan S.: is in favor of this approach - we have a framework and a tool. Let's move forward. Let's just use it, and if it turns out not to be good enough, replace it.

Colin: isn't the PHY group responsible for saying these are the parameters that must be in the framework.

Larry: we are working on a channel model and were going to put it into some format so that MAC state machines can be put on top of it.

Wim: the framework should contain the approach to be followed to model the environment. My simulation approach is similar to Larry's suggestion - see document 92/26. Tool is a separate issue - Wim used power basic.

Tom P: it would be useful to have a way of sharing executable things. This implies tool, language, framework. This is not necessarily that last framework. MAC needs to inherit simulations from the PHY group for channel modelling. There may be others, but this may be a good one. Larry's work could be translated by any volunteer.

John C: Larry has done lot of work in building up these functions. We have to get those standards is a common domain. The assumptions need to be agreed on.

Tom P: this is a 2 or 3 phase problem. First agree on a common tool so we can exchange files. Next address framework using that tool, how do develop the underlying assumptions.

Straw poll: people willing to use Mathematica (30 for, 1 against with reservations).

John McGown: this is a communication tool rather than development tool. Keep your high performance simulators to yourself, but this our common transfer of information language.

Jonathon: simulation presentations in the past have had no common denominator to argue. This gives us a common language in which to argue.

KC Chen: in scientific research we simply assume everyone does simulation correctly. The mode, framework is the part that must be common. We must get the same results from each simulation - that is what's important. KC likes the idea of a common communications language, but the model is what's important, not the tool.

Larry: everyone will have their own tools behind this. But given the same set of state machines and equations we will not come up with the same results - this is too dynamic a scenario.

Tom P: at least we have a common subject for discourse.

Meeting adjourned: 11:45 AM.