digital connection

BY DON ROTOLO,* N2IRZ

# Building a Decent RF Network

As the ARRL's HSMM (High Speed Multi-Media) project builds up steam, I sense that we'll be building more RF networks in the near future. Networks are nothing new, but it's surprising to me that many people who haven't been involved in the digital side of amateur radio for a long time try to reinvent the wheel when it comes to building a network. I wrote about this topic nearly ten years ago, and while much of what I wrote back then still applies, perhaps it's time to take a fresh look at building a decent RF network.

A brief word about HSMM: Many hams make the mistake of equating HSMM with "802.11 under Part 97," but it's much more than that. HSMM is a unified approach to making data compatible across many different platforms. While a decent network can be built with 802.11 equipment, a good portion of the HSMM working group's focus is on more efficient and robust HF communications as well.

Before we really get started, you should understand that good network design has nothing to do with the kind of networking software you use. Any software will perform poorly on a poor network. In addition, understand that this topic can fill a thick book, so what I'm covering here is necessarily brief and limited.

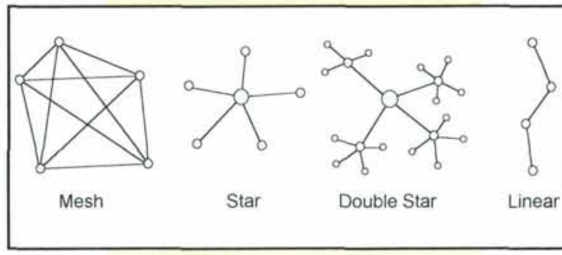


*Fig. 1– Common network topologies. A mesh network is highly reliable, due to redundancy, and has the greatest capacity, but costs the most. A star network covers wide areas inexpensively, but a failure can cut off service and link capacity can be quickly saturated. The linear network covers long distances inexpensively, but with lowered reliability. Cost savings for star and linear can be used to "harden" the network to improve reliability somewhat, and carefully chosen alternative links can be cost-effective. Most real networks are a combination of all three types.*

## Network Facility Classes

To examine the two network functions more carefully, we can break down each function as either a *user port* or a *backbone*. Each of these types of facility has unique requirements and needs.

A user access facility is shared by a number of stations, all accessing a single *User Port*. These individual stations may or may not be able to hear one another, and they all have highly variable signal qualities, parameter settings, and data throughput requirements.

A *backbone,* in contrast, is a point-to-point link with only two stations on the channel. The two stations trade data only with one another, with good signal quality, and link parameters are set to optimize link performance. Subsets of the backbone are the *wormhole,* a link that passes through a non-amateur service (such as the internet), and the *gateway,* which provides direct access to another type of network, usually non-amateur. With many gateways, non-amateur access is possible, so access control must be considered.

Now we'll take a look at user ports and backbones, and how to best design them.

## User Ports

First, a little background. In the RF world, when two stations try to transmit at the same time, they will interfere with one another if they are close enough in frequency and geography. This interference, known as a collision, distorts the signal enough to make the voice or data unusable. When two stations *do* transmit at the same time, there is a data collision, which renders both transmissions unusable. It should be clear that data collisions are bad, since they waste channel time, and must be avoided.

In the digital world, we simply listen on the channel for data. If we detect data, we wait until the channel is quiet to transmit. This is called Data Carrier Detection, or DCD, and it's a key part of CSMA (Carrier Sense Multiple Access), a common channel-sharing scheme used by many data networks, including AX.25 Packet and 802.11. Some other sharing methods include CDMA (Code Division Multiple Access) and TDMA (Time Division Multiple Access), used by some digital cellular telephones.

When someone puts up a user port, he (or she) may be thinking like a repeater owner, trying to maximize the port's coverage area. Installing a user port atop a tall building with a 100-watt transmitter may offer great coverage, and possibly boost the sysop's ego a bit, but this is poor networking practice.

The first problem encountered is the *Exposed Transmitter Syndrome* (ETS). This is where the user port hears so much RF activity that the DCD never turns off, effectively shutting it down. The channel is simply so saturated with data (at least from the port's perspective) that nothing works at all. One option for such a site is to install an attenuator in the receive side, so that fewer stations are heard, but this creates another serious problem, the *Hidden Transmitter Syndrome* (HTS).


*P.O. Box 114, Park Ridge, NJ 07656
e-mail: <n2irz@cq-amateur-radio.com>


 

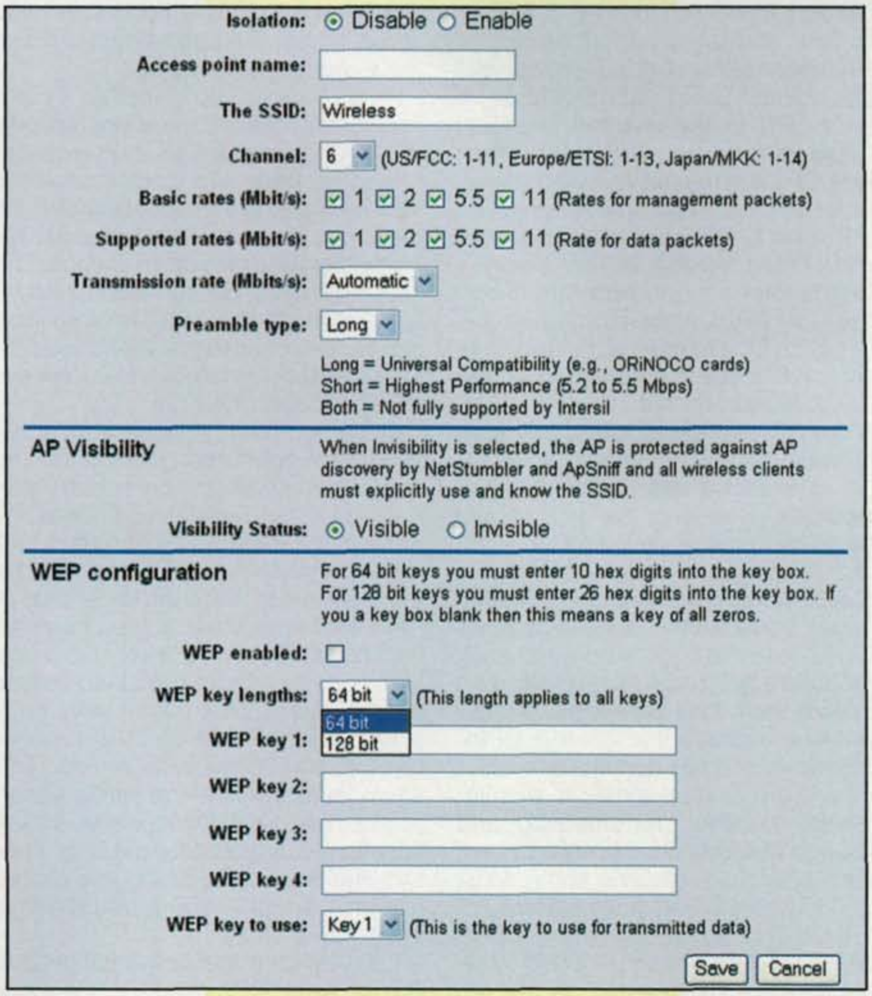


*Fig. 2– The main configuration screen for my 802.11b Access Point (AP). Note that WEP encryption is not intended to limit access; instead it serves to prevent eavesdropping, not useful for amateur radio applications. A simple way to prevent casual access is to set the AP Visibility to Invisible, but then you need to worry about station identification (normally done by setting the SSID to your callsign) if you're operating under Part 97.*

Hidden Transmitter Syndrome is the absolute worst thing that can occur on any shared data channel. In order for a CSMA channel to work properly, everyone on the channel must be able to hear everyone else. If this is not the case, a station might transmit while another station (not heard by the first station) is also transmitting, meaning the second station is a hidden transmitter to the first station. A CSMA channel that has hidden transmitters on it will always perform poorly, since a very large percentage of channel time is wasted on collisions. It should be painfully obvious that ETS and HTS are very bad things and must be avoided, even at considerable cost.

What can you do? Installing a full-duplex data repeater will solve HTS problems, but ETS can still be a problem where more than few hams can hit the repeater—not to mention the expense of a repeater. There's also the issue of using up two channels, plus the impracticality of a repeater for really wideband modes such as 802.11. The solution is really much less expensive, though, and de-ceptively simple—*Cellular User Ports.*

## The Cellular Solution

Instead of creating a small number of wide-coverage user ports, the better way is to create a large number of relatively small cells, each supporting perhaps 10 to 20 users (of which only two or three may be active at a given time). Cells are placed near the users and designed so that adjacent cells cannot hear one another. Data is then transported on a backbone channel, where there are no users. Cellular phone companies use this scheme quite successfully.

One cardinal rule of networks is that you *never* carry traffic on a user access frequency (something 802.11 likes to do, in its native version). When you do so, the channel capacity is immediately cut in half: half the time is used for the user to get the data to the user port, and the other half is used by the port in sending the data somewhere else. Sure, it costs a little more, but are you building a high-performance network here or just fooling around?

## Backbones

The single most important thing you can do for backbone links is to insist that each one is a dedicated point-to-point link (DPPL). That means only two radios on any given frequency (hopefully not on a band where user channels can exist), with high-gain directional antennas pointing at one another. As soon as you introduce a third station to a backbone channel, throughput drops like a rock.

I really can't emphasize this enough. The reason behind this throughput loss is in the nature of the data transfer protocol. Packets of data (not only AX.25 uses packetized data) are sent, and a brief packet is returned to acknowledge the data packet, allowing the next group of packets to be sent. Because of the way the timing is set up, a third station causes additional waiting time between data and acknowledgement. That waiting time is wasted performance. For user channels it isn't as critical, but large-volume users ("large volume" is about 10% of the channel capacity) should be encouraged to set up their own DPPL into the network backbones.

It is far better to set up a lower speed DPPL than a higher speed shared backbone. When loading is very heavy—which is when capacity becomes important—the DPPLs will outperform the shared "zoo" channel. Again, it does cost a little more to do things right, but the results are worth it.

Given the realities of the typical amateur radio budget, it can be unreasonable to expect a large initial outlay of resources. However, the goal should always be to migrate towards the ideal, even if it is presently unattainable. If your present network already suffers a shared backbone channel, carefully monitor the data activity between each site. Often two sites pass more data between them than between any other two sites. Upgrade that link to a DPPL when the resources become available.

There are tools available to monitor wireless network performance, and although I haven't found one that's free, many offer free trial versions. Most 802.11 access points have some built-in functionality, at least for signal strength, while tools are generally available for other RF gear. As an alternative, the wired side of a link can be monitored, or test files can be sent across a link to gauge performance.

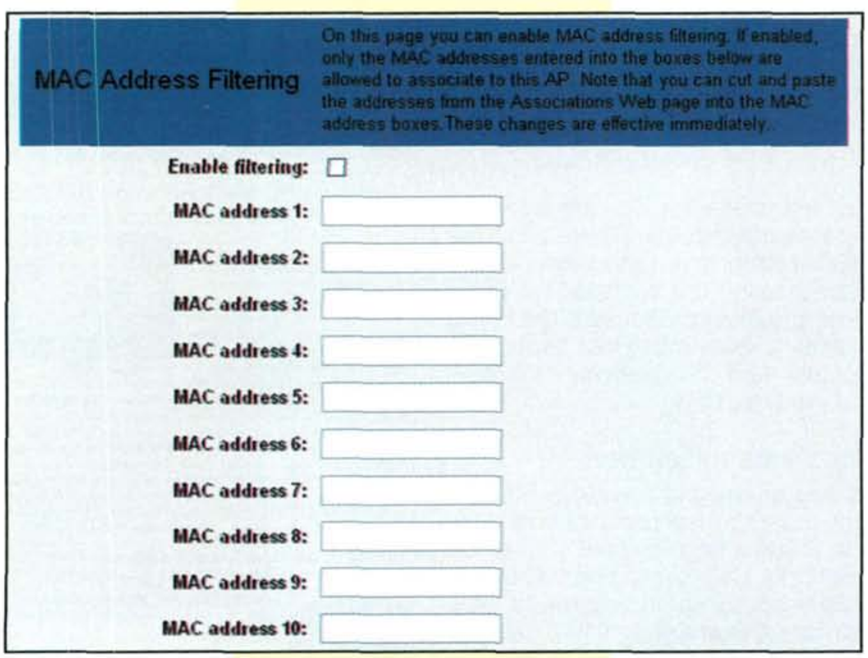


*Fig. 3– On this screen you can explicitly set which network interface cards will be recognized by the Access Point. If the MAC address isn't on this screen, you will be unable to access the network. While hams will have to register with you for access, knowing who they are may help drum up support for network improvements.*

That brings up another point: Intelligently derived backbone speeds. If a multi-port network hub services three user ports and concentrates that data load onto one major backbone, the backbone must be able to handle the load without slowing down anyone. This seems obvious, but some people don't figure it into their plans. The data loading on each backbone link should be measured or estimated, and link speeds adjusted accordingly.

Once you have your network's performance optimized (given the resources available), you should start thinking about redundancy. Draw the network on a piece of paper and look for links that, if they fail, cause a part of the network to become unreachable. Then look for a place to install another backbone link, even a lower speed one, to provide a backup. It is reasonable to set up bunch of backup links on a shared channel, as long as it is really a backup and not a primary link. This saves money while preserving performance, since only two stations should be affected by a broken link. Only those two stations will be using the shared channel, with the others just listening, effectively a DPPL.

It should go without saying that tenuous or weak RF paths are unacceptable. Add power, or better yet, antenna gain to the link until it functions well; perform the basic link calculations and determine the actual fade margin. Antenna gain is usually cheaper than RF power, narrows the beam width to allow more reuse of channels, and helps your receiver, too. Breaking up a long path into two segments can also make a big difference, or you can move down in frequency for better propagation.

Avoid confusing radio networks with wire-based networks. There are far more wire-based network designers out there, but their knowledge and experience often doesn't apply to the RF world. Be wary of "experts" who don't have any experience with radios.

## Security

One commonly encountered issue when deploying a network using commercially available gear (such as 802.11) is the need to prevent non-amateurs from passing data over links operating under Part 97. Note that encryption such as WEP does nothing to prevent access to your network. It only prevents viewing of the data being carried (see fig. 2). Instead what you need is access control.

Virtually all Access Points allow for access control based on the MAC address (a 12-character unique "serial number" assigned to every network interface card) of the wireless Network Interface Cards (NICs) trying to access the network. For user ports, the users will have to register their wireless LAN cards with you so they can be put into the access table. Backbone links should be set to recognize only one another, of course (see fig. 3). The downside is that any hams in the area will have to contact the network sysop to get into the network. This has advantages, however: Once you know who is using the network, it is easier to get them to support the network with time, talent, or treasure. You can also plan the network better, considering loading and HTS (Hidden Transmitter Syndrome). It definitely keeps casual intruders out. Although MAC addresses can be spoofed, you'd need to know what's on the list first. It isn't perfect security, but it should be more than adequate for our purposes.

## Conclusion

There are many resources on the internet that will help you design a great network. Much of it is common sense, such as performing "link RF budget" calculations, and some of it is a bit more subtle. There will always be compromises based on limited resources, but there are some things on which you can compromise, and some things on which you should not.

Don't compromise, for example, on network topology. Running backbone data on user channels will bring performance to its knees. Don't wait for high-tech solutions when low-tech solutions will work sooner. Network design is a little bit logic, a little bit skill, a little bit politics, and a little bit luck. A well-designed network will always outperform a poorly designed one, often at lower cost. There are people out there who have spent their entire careers doing network design, so don't expect to become an expert overnight, but a little effort will pay off handsomely. There aren't many books out there that deal with radio-based networking design, but the engineering library at a local college should have something useful, even if it's only a single chapter.

Well, that's all the time we have this month. I hope you found this column interesting and useful, especially now as we find more and more uses for a non-internet network solution. As our RF-network performance improves to meet and exceed T1 and cable modem levels, we'll be seeing more and more hams—especially those with dial-up connections—preferring to get their ham radio information over radio instead of wires. Packet radio is almost dead, not because of slow data but because there was limited useful content. HSMM will have the content, something I am looking forward to.

Until next time . . . 73, Don, N2IRZ

### Why Digipeating Won't Work

Before networks there were digipeaters. By hopping down a chain of other hams' TNCs (terminal node controllers, for the newbies among us), you could go farther than your radio's range. It sounds okay, until reality is considered. The problem is that digis are dumb; they just take what they hear and repeat it. The digi doesn't acknowledge your packets; that is done only by the destination station. Thus, each digi just passes your packet down the line, and passes the distant station's acknowledgement packet back up the line to you. In the real world, links between digis are not 100% efficient; not all packets make it through without getting lost or corrupted. Note that an 802.11 network extender is nothing but a digipeater.

If you use only one digi, you have to get across four links before you can send the next packet-two to get there, and two for the acknowledgement to get back. If we assume that nine out of ten packets make it on each hop (actually quite good), then that one-digi path has an efficiency of only $(0.9 \times 0.9 \times 0.9 \times 0.9) = 0.65$, or 65%. With two digis, it drops to 53%.

With only half of your packets making it through on any given try, it is very difficult, if not impossible, to pass any reasonable amount of data. If other users are on frequency, the hops get worse than 90%, easily down to the range of 50% per hop. Do the math and you'll see why for longer-distance communications digipeating just won't work.