

Swarm logic for wireless net management

By Paul Callahan

As the volume of Wi-Fi products increases, how do we control the escalation of radio frequency (RF) interference? Manual adjustment and coordination among a large number of nodes is impossible. The most challenging situations are those with no central authority—shared public spaces, shopping malls, multitenant high-rises, airports, apartment buildings. But even enterprise environments with centralized security management will discover that RF is too unpredictable to be handled centrally. Enter self-organizing networks.

The concept of self-organizing networks originated on the battlefield. Out of the need to connect moving nodes like tanks and personnel carriers came the origins of mesh networks that could arrange themselves to communicate in hostile and fluid environments. Those networks were designed to have no wired infrastructure available and to use the airwaves as a trunking mechanism.

This generation of wireless networks held the possibility of eliminating network configuration altogether. As the technology moves into the wider consumer and enterprise markets, cost and interoperability become the overriding issue. Dominating these broad markets is low-cost, standards-based equipment based on 802.11. But their Achilles' heel is the cost and complexity of configuration.

The answer is to apply 'swarm' logic to this sea of low-cost 802.11 products and cause them to self-manage the RF domain much like mesh networks have done in the past. The technology is analogous to the way swarms of bees and hives of ants behave. Bees are not very smart, but because they communicate effectively (research has established that they have a language), they collectively exhibit a higher-order behavior. Similarly, by adding a small amount of code to 802.11 clients and access points, a distributed real-time RF control system can be created that keeps data rates high, dynamically responds to rapidly changing conditions and transparently fails over. In corporate environments, this eliminates the need for IT conducting useless site surveys and plotting out channel maps that will be obsolete as soon as they are complete.

But given the fact that networks (including Wi-Fi) will continue to be a mix of old and new equipment as they grow, how can such a distributed system be successful? Customers, for example, will reuse older access points (APs) and clients that have no ability to control their transmit power and always run at full power. Any self-organizing network must be architected assuming that it will never find itself in a homogeneous network. As each self-configuring node is added to the network, it must listen for other nodes and determine whether they are able to negotiate. Even if the neighboring APs and clients will not respond to requests to coordinate power, channel selection and load balancing, self-organizing APs must still take action to optimize performance and minimize co-channel interference by choosing noninterfering channels or reducing transmit power on APs and clients.

For example, if a loud and unresponsive neighbor AP is on channel 48, the self-configuring AP should choose another channel, but it should also choose the one that is the farthest away from all other channels in use—whether they are self-configuring or not. Self-configuring APs must negotiate and make these decisions by exchanging the known state of the network both over the air and over the wire, preferably using a low-overhead protocol. The protocol represents the control plane of the system. But to produce predictable and stable scaling conditions, these control plane conversations must occupy a very small percentage (for example, less than 1 percent) of the overall capacity of the network, regardless of how large.

To make these protocol negotiations based on real-time RF conditions instead of guesswork, self-configuring APs and clients (laptops, phones or PDAs) must constantly measure the behavior of the RF domain around them. These measurements form the basis for a feedback loop on the RF domain—a closed-loop system much like the cruise control in an automobile. As a simple example, the algorithm checks the received-signal-strength indicator (RSSI) of another AP using the same channel and negotiates a power level for both based on signal processing. If something changes in the level of RSSI, transmit power is adjusted in real-time on both ends.

An open-loop system, by contrast, has no such feedback mechanism. It can be manually tuned based on an estimate of what the optimal RF settings should be, but such adjustments are based on theoretical considerations, not on what is actually happening in the RF domain in real-time. Many such tuned systems are based on simulations or guesses about the RF domain and have been offered recently as automated site survey tools. But they are the equivalent of putting a brick on the accelerator of your car and hoping that it will cruise at 60 mph.

Clients must also play an active role in self-organizing the network. They can provide processed RF measurements that accurately detect the difference between RF anomalies and actual client movement. Since the movement of a human hand across the antenna of a typical 802.11 laptop adapter can cause a change in output that can be measured as 7 to 10 dBm, determining that this is an anomaly is a critical and required bit of signal processing. Why? Because any system that provides rapid roaming to maintain high data rates will need to distinguish actual client movement from hand waving or other anomalies. Without accurate movement detection, roaming will be unpredictable and unstable.

Clients can also engage in negotiation with APs to make load-balancing decisions. Despite claims to the contrary, WLAN load balancing cannot be accomplished without client-side participation. Having APs simply tell their clients to go away and find another AP does not work—clients just come back and try again, wasting the resources of the refusing AP. By contrast, client-assisted load balancing should use a negotiation between clients and available APs with a hysteresis to avoid network thrashing. When APs exhibit a large imbalance, they should shift load quickly, but when the imbalance is slight (eight clients on one AP and 10 on another), load balancing should take longer.

From a performance point of view, client transmit power control is even more important than load balancing. An 802.11 network without coordinated client transmit power control will operate at less than 50 percent of its potential throughput because of the massive co-channel interference created by clients running at full power. Instead of creating tightly controlled microcells, client laptops running at full transmit power will each operate RF domains in the range of ~200 feet in diameter. This kind of client RF range inevitably results in large collision domains and dramatically lower performance. Transmit power on clients should reduce the size of the client's RF domain to the minimum required to reach the AP at the highest data rate possible.

Software integration

How can self-configuring software be integrated? On APs, it can be placed into the existing code base. As firmware, self-configuring code must have small code footprint (10s of kilobits)—a significant cost issue in most low-cost APs. In addition, this firmware must also be designed to be conservative in its use of compute resources. Most low-cost chip sets offer embedded MIPS-core processors with relatively low horsepower, and since ODMs and OEMs often choose to build low-cost APs with-

out any external CPU, continuously running self-configuration processes must avoid significant CPU impact.

On clients, ODM and OEMs can integrate client code as a standard miniport driver, downloadable and installable by users as a replacement for current drivers, or shipped with adapter cards by ODM/OEMs. As in the case of APs, the code footprint should be small and CPU utilization low to accommodate integration in phones and PDAs as well as laptops.

Paul Callahan is co-founder and vice president of business development for Propagate Networks (Acton, Mass.).