"I can't quite decide"...

MYK DORMER IS A SENIOR RF DESIGN ENGINEER AT RADIOMETRIX LTD **WWW.RADIOMETRIX.COM**

...Or low duty-cycle p/n dithered burst transmission in low-power radio networks!

Numerous short-range data transfer applications now involve a basic architecture that's considerably more interesting than a basic one-to-one pairing. In some cases the radio network is as functionally complex as a wired-Ethernet installation, with multiple nodes exchanging large amounts of data on a peer-to-peer basis. In others it can be as simple as a single master controller, periodically broadcasting a handful of command bytes to a constellation of subordinate receivers, with each controlling a physical device.

The system type I intend to examine here is of a common variety, best identified as a "data gathering" network. In this case the system comprises a large number of sensor nodes, which could be sensing anything from complex environmental data to the state of an alarm button, all reporting to a central master station, where the data is gathered and processed.

While there are many possible uses for this network type, such as fire or intruder alarms, agricultural monitoring, perimeter intruder detection and energy efficiency monitoring, they are characterized by a handful of common characteristics:

1. Data flow is one way: from sensor nodes to master.
2. Multiple transmitters share the same channel.
3. Individual transmit duty-cycle is low, to minimise power and meet regulations.
4. Overall data throughput is low; each node sends only a few dozen bytes.
5. Response time is not critical; a ten second delay would not be disastrous.

Experienced users of low-power wireless devices will already be familiar with a number of techniques that address one or more of these requirements, such as traditional sequential polling, or beacon synchronisation, not to mention any number of more sophisticated proprietary mesh-network techniques.

Unfortunately, all these methodologies require the use of a wireless device capable of functioning as a transceiver. While such hardware (in this age of single-chip radio devices) is no longer prohibitively large or expensive, a transceiver is still more expensive than the equivalent transmitter, and all these synchronous-network methods require a finite power drain – albeit sometimes a small one – at the sensor node, to run the receiver function that maintains network synchronisation from the periodic base transmitted timing-burst.

How Else Can This Sub-Class Of Network Be Organised?

Obviously, if all transmitters are allowed to send continuously, as well as suffering from a prohibitive level of power consumption, they would all block each other and, allowing for capture effect, only the nearest node to the master would have a chance of being heard.

Time-division multiplexing is the obvious solution. Organise the data into a concise burst or packet, with the necessary framing, addressing and check-sum sequences, and then transmit it as infrequently as the overall system constraints allow; which could be anywhere from less than once per day, to several times per second. This technique obviously allows very low duty-cycles and, with some care in design of any periodic wake-up "heartbeat timers", low-to-negligible average current, all achieved with a very simple transmit-only node. Perfect?

> "The system type I intend to examine here is of a common variety, best identified as a "data gathering" network

Not quite. All "transmit blind" methods retain the risk of any given transmission colliding (occurring in the same time-slot) with another, resulting in the loss of at least one set of data. This can happen when external events simultaneously trigger two or more nodes at the same time, or when by pure bad luck the periodic transmission "beat" of one node falls into step of another. In these circumstances the only solution is for each node to re-transmit the data enough times that at least one of its messages will reach the master receiver uncorrupted. For such re-transmissions to avoid simply becoming repeats of an initial collision, it is vitally necessary that no two or more nodes transmit with the same periodicity and pattern.

A pseudo-random number sequence-generator is the key to a very effective method of achieving this: If, instead of sending a single data burst, the node sends a number of identical bursts, each separated from the previous by a (pseudo) random time-period then the likelihood of at least one getting through is greatly increased. The chance of losing an entire data-set to collisions with another transmitting node becomes insignificant if making the number of repeats large enough.

How Many Repeat Transmissions Are Necessary?

This is the critical question. The chance of a collision is related to length of burst, number of nodes transmitting on the system and the frequency of transmission. While direct analysis of the probability is possible, for practical engineering purposes I use a simple software simulation of the network.

The required coding is very simple and can be implemented in any high-level language. The technique I suggest is the following:

- Determine the length of the transmit burst. This is the basic "granularity" or "time slot" of the system timing (there is no point in a time step of 1ms if the bursts are 50ms long; a timing change of less than fifty "ticks" would not avoid collision).
- Write a good software model of the p/n code that your actual system is using (length of shift register, position of taps, initial seeding, transmission rules etc).
- Set up a separate routine for each node on the system, determining if it transmits in a given time-slot (according to the rules set up above).
- In each time slot simulated, record which nodes transmit and so determine how often each node successfully sends a burst without collision.
- Run the simulation over many thousand simulated time-slots, to yield a statistically meaningful result. I have used simple interpreted BASIC as my simulation tool.

Once a valid simulation has been written, it is then possible to vary the number of nodes, the frequency of transmission and the number of repeats, until an acceptable compromise is found.

EXAMPLE

A simple transceiver sends an 8-byte burst in 50ms – typical narrowband UHF performance with 10ms tx-on switching, then 8 (payload) + 4 (overhead) bytes at 2400 bit/s. It consumes 20mA in active transmit.

A data-gathering system is built, with 30 of these transceivers communicating with a single master node. Rather than sending short groups of bursts, each node sends a burst at an average – but p/n dithered – rate. This average period of transmission is varied in the simulation to optimise data throughput against number of collisions. In this case, the random element is a tapped shift register p/n (16-bit register with XOR feedback, tapped at bits 15 and 13). The transmit rule is "send if the lowest (n bit) word of the register is equal to zero"; obviously, far more sophisticated rules are possible.

For simulation purposes, assume a 100ms "occupied time-slot" (as the last bit of one transmission must not overlap the first bit of the next).

Results at a simulated time-period of 1000 seconds:

Comparison word length (n)	7 bits	6 bits	5 bits	4 bits	3 bits	2 bits
Average time between bursts	12s	6.2s	3.1s	1.6s	0.8s	0.4s
Average success rate	74%	63%	38%	14%	1%	<1%
Average number of valid bursts	65	97	126	103	24	<1
Usable link?	Yes	yes	yes	yes	barely	no

("Usable link" is based on sufficient bursts getting through in the time considered)

It can be seen that while (as might be expected) the greater the repetition rate, the greater the chance of a collision occurring, the actual throughput, considered as the overall number of valid bursts reaching the master, peaks in the centre of the test range and falls away to un-usability if the nodes are permitted to transmit too often.

In this example, given the burst length, the number of nodes on system and used randomisation rule, I would probably choose the second option (approximately 6s between transmissions) as offering a reasonable throughput for sub-1% duty-cycle:

50ms burst every 6.2s = 0.8% duty cycle
Assuming peak tx current of 20mA, average = 160uA

Simulations of this type are the most certain method for examining the integrity of low duty-cycle systems of this type, as they allow the randomisation methods themselves to be prototyped and examined in (simulated) operation, without needing to run real hardware in real-time over impractically long, experimental sample-periods.