DESIGN OF DISTRIBUTED CUBESAT NETWORKS

Alexis Zamora
Department of Electrical Engineering
University of Hawai‘i at Mānoa
Honolulu, HI 96822

ABSTRACT

Small satellites are more attractive if they can be networked. Toward this end, two different network topologies with different on-board radios were investigated. The MHX 2400 radio, which uses a master-centered networking architecture, was considered first but was abandoned in favor of the more flexible mesh networking architecture of the Maxstream Xtend. This paper presents the network design for both hardware selections, and explains the interface between the radios and the onboard processor as well as the driving software to configure the transmit/receive modules.

INTRODUCTION

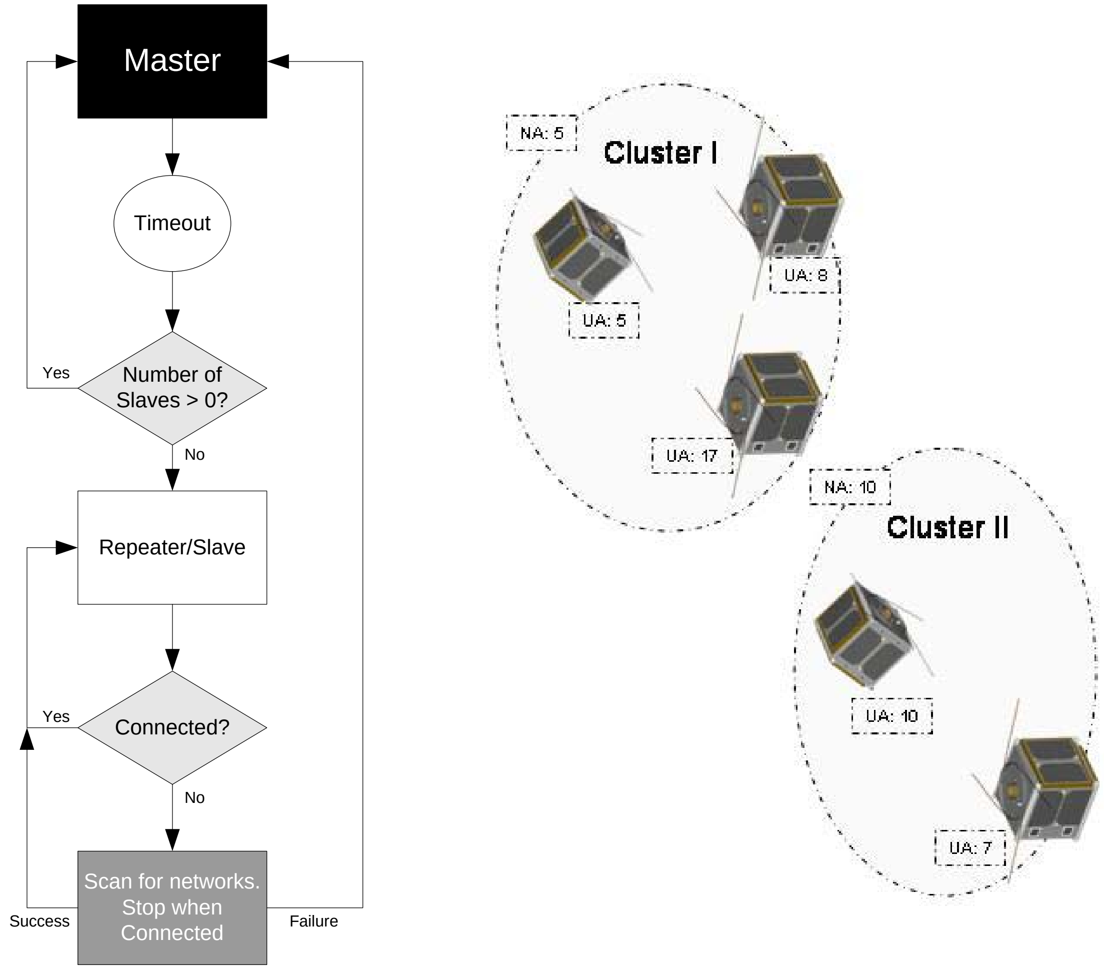
Since they were first conceptualized in 1999 by Stanford University, the mission of CubeSats has been clear: to minimize satellites' size, development time, and cost [1]. CubeSats are standardized satellite packages that measure 10x10x10 cm, having volumes of exactly one liter and whose mass may not exceed 1 kg [2]. To achieve both standardization and low cost, it is customary to choose commercial off-the-shelf (COTS) components when designing these small satellites. The potential for these satellites is accentuated when several of these units are distributed in space and may cooperate on a particular mission that requires multitasking or large area coverage. Since networking capability would enhance the area coverage for low earth orbit (LEO) imaging missions, this paper focuses on designing CubeSats to possess this networking capability.

NETWORK DESIGN USING MHX 2400 RADIO

The design in this section was based on the Microhard MHX-2400 spread spectrum transceiver, which operates in the 2.4000 – 2.4835 GHz ISM band. The emphasis on network redundancy and reconfigurability in the design results in an autonomous and self-healing network, ensuring small satellites will perform optimally in space.

Due to the specifications of the MHX 2400 transceiver, the following assumptions are made before developing the networking algorithm. Firstly, every node in the network is assigned a unique unit address to ensure that there are no duplicate addresses on the network. Secondly, each unique node may function either as a master or as a repeater. Thirdly, every unit must have the same network address to belong to that network. Fourthly, to ensure a single master per network, and for simplicity, each master will assume a network address equal to its unit address. Lastly, it is important to note that only one master may exist per cluster containing a specific network address.

Each node wishes to belong to a network, but is limited to communication only with a master possessing the same network address as itself. With this limitation in mind the algorithm described by the flow chart was developed. To follow the discussion regarding network formation, refer to the flowchart (Fig. 1a).



(a) (b)

Fig. 1: (a): Flow chart depicting the configurations that must take place to create small networks or clusters of satellites; (b): End result of the flow chart logic. Instead of one large network forming, small clusters form. NA stands for network address. UA stands for unit address. The UA that matches the NA is the master of the given cluster.

Each node begins knowing only the number of units within the system. Every unit is initialized as a master and waits for slaves to connect to it. Given the lack of presence of slaves in the network at initialization, masters must be able to reconfigure themselves to become slave/repeaters. A random timer determines how long each master spends in the master state before converting. In the event that a master has formed a connection with at least one slave/repeater after the timeout period, it will remain a master; otherwise, it will convert. As a repeater/slave, each unit wishes to connect to a master. Each node may only connect to a master possessing the same network address as it itself possesses; hence a repeater/slave will iteratively change its network address with delays in between each change in the expectation of joining a network. After each network address change, it will determine if it is connected to a network or not. In the event that the node has joined a master, it will remain in the slave/repeater state. If on the other hand the node has iterated through all available network addresses without successfully connecting to a network, it will revert back to being a master and reinitialize the process of network formation. Following this protocol, each node in the network will continue to search for a network until it belongs to one. The most likely outcome, using this approach, is for there to exist a series of small cluster networks rather than one large one (Fig. 1b). Hence several masters, with different network addresses, will be hosting several smaller networks. It should be noted that stray nodes may exist within the system, and will continue to search for a network until belonging to one.

The protocol illustrated thus far successfully defines how nodes are to make a connection with other units, resulting in a series of small cluster networks. This is analogous to a series of tribes in a primitive civilization. As with tribes, satellites may wish to create larger networks to have access to more information. The MHX 2400 does not permit master-to-master communication as it limits communication to happen only between a repeater/slave and a master that form part of the same network. To deal with this limitation, each master assigns a single node at a time to act as a liaison between two clusters. This designated repeater/slave is effectively a "scout" searching for nearby networks (Fig. 2). Each scout must remember the address of the network it originally belongs to so that it may successfully reconnect. Each scout is also provided with the network details of its original cluster by the master. The designated node then begins to search for other clusters. In the event that it does connect, it requests network information from the new master and provides it with the details of the original cluster. The scout then returns to its original cluster and provides this master with the network details of the cluster just visited. The smaller master's network instructs each node in its cluster to change their respective network addresses to match the address of the larger cluster's master. In this fashion two smaller clusters join to form a larger network. By each cluster in the system sending out scouts to search for other networks to merge with, the largest physically possible network may be formed.

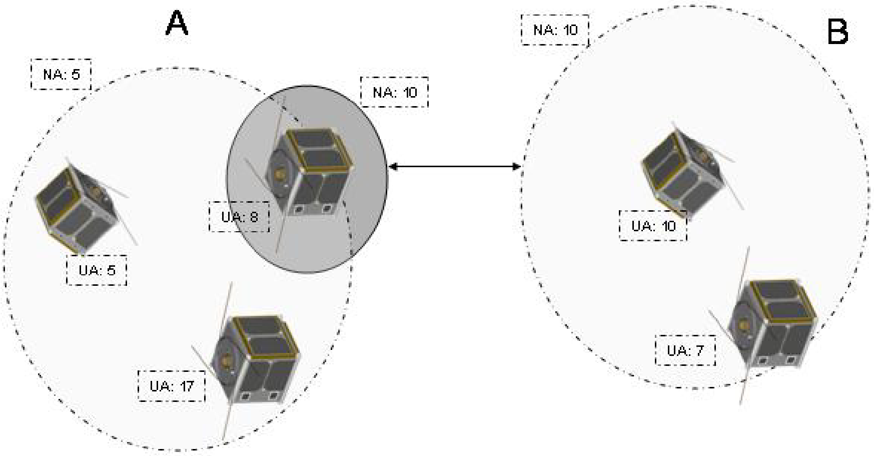


Fig. 2: Depiction of the scout method for merging clusters and thus establishing enlarged unified networks. Cluster A assigns satellite with UA of 8 as a scout. This scout changes its network address until coinciding with the network address of cluster B. UA 8 informs cluster B master of NA 5's existence. UA 8 returns to cluster A to inform master of NA 10's existence.

With a network in place, data may be readily shared between nodes permitting any multi-tasking required by the mission.

This method for developing networks is believed to be novel when using a master-centered architecture. As described in the next section, however, the mesh architecture of the Xtend radio allows the user to forget about forming a network as the radio takes care of this autonomously.

NETWORK DESIGN USING MAXSTREAM XTEND RADIO

The Xtend is a frequency hopping spread spectrum radio that operates on the 902-928 MHz band. The network topology is peer-to-peer (mesh) which means each node in the network is connected to every other node (Fig. 3). Once powered, a network is formed provided the radios are in range of one another. In this sense, network formation is performed automatically and about the remaining task is the transmission of data between satellites. A series of Application Program Interfaces (APIs) were programmed in C to send certain data packets, namely images. An API is a convention for invoking certain built-in features in the radio. For instance, the radio has built-in capabilities for handling file transmission. To transmit a file, however, a HEX sequence must be employed to instruct the radio to send certain data types.

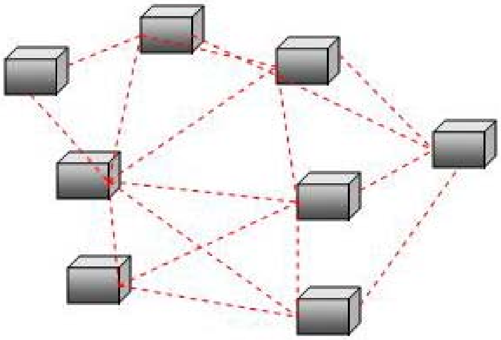


Fig. 3: Mesh or peer-to-peer network. Unlike master headed architectures, a central node does not exist in this type of network. All nodes are interlinked and all nodes are equals.

Each satellite, having its own radio, needs to interface its radio with the on-board processor which controls the radio when data is transmitted. The computing environment chosen for our satellites was the Gumstix basix 400 which uses the Intel Xscale PXA255 to do its processing. The advantage of this microcomputer is that it comes preinstalled with a Linux operating system which makes development easy and with wide access to Linux's free-of-charge development world. Another advantage of using Linux is that all software may readily be developed in C which is a familiar language.

Networking and file transmission was demonstrated using the Xtend radio. A program on the Gumstix autonomously configured the radio to have a peer-to-peer networking topology. With this configuration in place the radios were able to autonomously form a network within the laboratory setting. To ensure that each of the nodes was indeed talking to one another, communication was verified using the terminal program minicom which was installed on the gumstix. Minicom is a serial terminal program like Hyperterminal for Windows which allows a user to manually send characters through the serial port of the computer. Since the serial port of the Gumstix was output to the radios, characters could be sent between all nodes in the intralab network. Autonomous file transmission was demonstrated by executing a program on each of the nodes that sent a file to each of the other nodes. In this way every node on the network had a carbon copy of three different files.

FUTURE RESEARCH

For future research it would be interesting to develop a radio specific for the needs of picosatellites. In this effort the footprint of the radio could be better controlled to suit a specific mission. The frequency spectrum could also be made more specific. The purpose of developing a unique radio would be to overcome the limitations imposed by a using these particular COTS radios.

CONCLUSION

A radio with peer-to-peer networking capabilities was found to have more advantages for ad hoc CubeSat networks than a master-headed architecture. It was discovered that the MaxStream Xtend provided a very appealing COTS radio that met the demands of the project. It was demonstrated that these radios could be networked to demonstrate effectively that CubeSats may be networked to accomplish a given mission.

ACKNOWLEDGEMENTS

I would like to thank Dr. Wayne Shiroma, my mentor, for both guiding me in this project as well as opening the door for me to do this research. I would also like to thank NASA as well as the Hawaii Space Grant College for providing me with the avenue to do research that develops NASA’s goals. Dr. Ed Scott, thank you for coordinating this effort.

REFERENCES

[1] CubeSat. http://en.wikipedia.org/wiki/CubeSat.
[2] Cube Sat. http://www.cubesat.org.