Reliability:
Introduction to how addressing and retries work

Contents

Abstract	2
Point-to-point Broadcast Communications	2
Point-to-point with Retries	3
MaxStream's Method of Retries	3
MaxStream's Point-to-multipoint solution (remote initiated)	4
Point-to-multipoint (base or remote initiated)	6

MaxStream®

Technical Support

Live Chat: www.maxstream.net

E-mail: rf-xperts@maxstream.net

Phone: (801) 765-9885

XST-AN020a - December 2005

Abstract

In most communication systems, it is desirable for data not only to reach the destination, but also for the data to arrive in a reliable and consistent manner. In general, wireless communications tend to be subject to more interference and lost data than a wired system. Because radio frequency transmissions take more errors, usually some sort of protocol and packetization of data is done to help ensure data integrity and make sure the data safely arrives at the destination. This application note discusses various modes of data transmission and built-in features available in some MaxStream radios that aid with reliable data transmission.

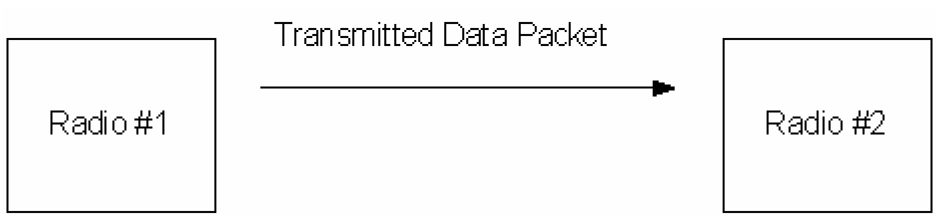
The following scenarios will be discussed:

- Basic transmission
- Point-to-point with Retries
- MaxStream Method of Retries
- MaxStream Point-to-multipoint solution (remote initiated)
- Point-to-multipoint (base or remote initiated)

Basic (Broadcast) Transmission

The basic form of transmission is a data broadcast. In a broadcast architecture, the transmitter sends out a message and does not need or expect an acknowledgement response. Point-to-point communications consist of two radios. In a point-to-point system, if one radio is broadcasting and the other receiving, the broadcasting radio sends out a message (refer to Figure. 1). If the receiving radio does not receive the data packet, the transmitting radio has no way to know that the data was not accepted. If the receiver were turned off or if some interference corrupted the data packet, the information would be lost. If data integrity is required, something more than a simple broadcast must be used. The transmitting radio would need to know the packet did not arrive and a method to retransmit lost packets has to be used.

Figure 1. Point-to-point Data Transmission

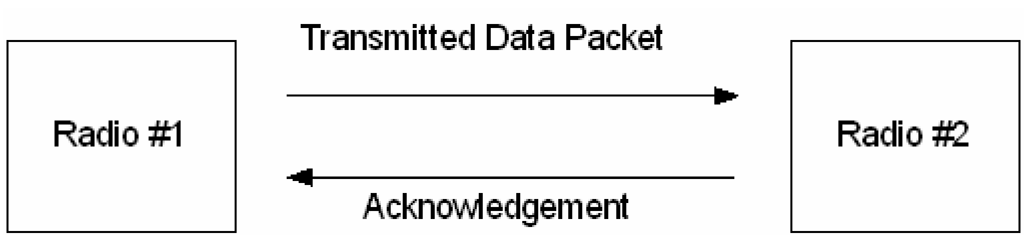


Point-to-point with Retries

Whatever the reason for the lost information, in order for the receiver to get good information, the transmitter must rebroadcast the message. If the message is small then re-broadcasting the entire message may not pose any problems. If the message is large then re-broadcasting the message may take a significant amount of time. Also, with a large message there is a good probability that at least a portion of the message will get corrupted. For these reason large messages are usually broken down into smaller chunks of data called packets. If a portion of the data gets corrupted, then only the corrupted packets have to be resent instead of the entire message.

In a retry and acknowledgment scheme, for every data packet that gets sent out an acknowledgement must be sent back to the transmitting radio to let the radio know that a data packet has arrived (refer to Figure. 2).

Figure 2. Point-to-point Data Transmission



If no acknowledgement is received by the transmitting radio then the packet is resent. The packet is usually sent a finite number of times before the system times out.

MaxStream Method of Retries

In the MaxStream XTend and XStream radios the RR parameter controls the number of retries that the radio sends. The RR parameter must be set to a non-zero value on the receiving radio to let it know that an acknowledgement has to be sent for each incoming packet. The RR values do not have to be the same on both radios, but if communication is bi-directional, then usually the values are set to match. Turning on the Retry parameter can slow down the radio's throughput, so depending on circumstances flow control may need to be enabled to keep from overflowing the radio's buffer.

EXAMPLE

Suppose we have a system with two radios and we wish to use the retries and acknowledgements.

Radio #1	Radio #2
RR=8	RR=8

Both radios would need to have matching DT & HP parameters.

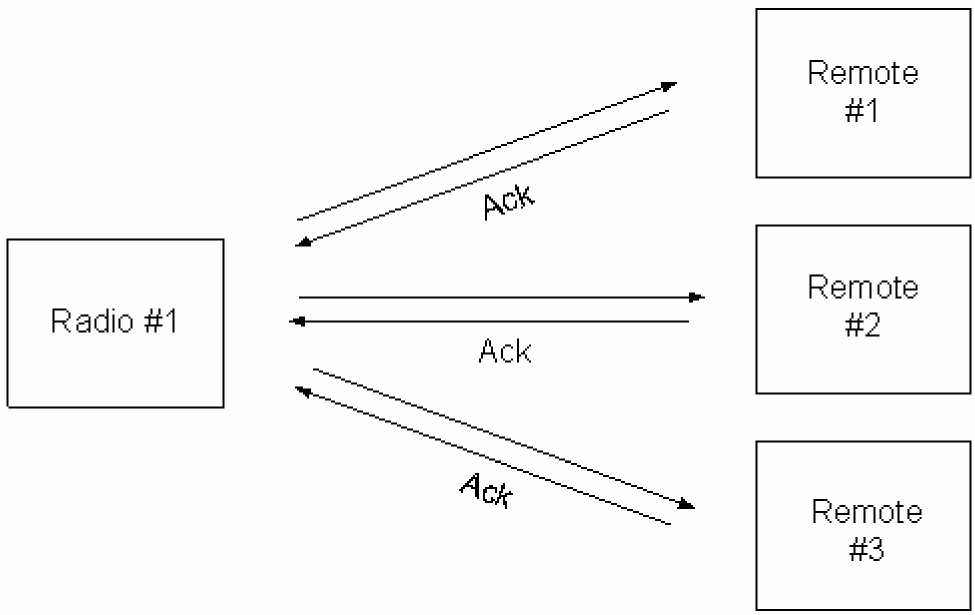
Point-to-multipoint problem

Suppose now that we have a point-to-multipoint scenario where multiple remotes can communicate with a single base.

Communications could be either initiated by the base or be initiated by the remotes. When only the RR parameter is enabled, the acknowledgement packet that is issued by a receiving radio is not unique in the sense that there is no information within the acknowledgement to know from which radio it is coming. In the case where the base initiates the communication, multiple acknowledgements could be received for every packet sent out from the base.

The acknowledgements may end up colliding causing the base to resend unnecessarily, or if only a single acknowledgement is received the base will not resend to a remote that did not successfully receive the data.

Figure 3. Point-to-point Data Transmission



In a peer-to-peer architecture like MaxStream uses, there is no limit to the number of remote nodes that can be deployed. For every packet that goes out from the base, any number of acknowledgements can come back from all of the remote radios. If the acknowledgements collide, then the base would send an additional packet even if the additional packet is not needed. If only one of the units got the data packet then the base would not send an additional packet as a valid acknowledgement has already been received.

MaxStream Point-to-multipoint solution (remote initiated)

In addition to the standard destination address (DT parameter), the XStream and XTend radios also have a source address (MY parameter) so that a single collection point can differentiate packets from multiple remote radios. This also allows the radio at the collection point to be able to send unique acknowledgements to whatever remote sends a data packet. In situations where all communication is initiated by the remote radios, each remote radio must have a unique source address so that the packets can be distinguished from different radios at the collection point.

EXAMPLE

Assuming the setup is the same as illustrated in Figure. 3 an example of a typical setup for XStream radios is given below.

Radio #1

RR=8

MY=0

DT=FFFF

Remote #1

DT=0

MY=1

RR=8

RN=4

Remote #2

DT=0

MY=2

RR=8

RN=4

Remote #3

DT=0

MY=3

RR=8

RN=4

The RN parameter automatically inserts some random delay in case all remote radios transmitted at the same time. One radio would get acknowledged and the other radios would do a random back off and retry their data packet. Because the base radio cannot accept multiple acknowledgements, its address is set to the global address.

Point-to-multipoint (base or remote initiated)

If communication can be initiated at either end, then the situation becomes more complex and there are multiple ways that the communication can be handled. The simplest method from a protocol standpoint is to have the base radio change its DT address to match the MY address of the remote unit of interest. This works well if the base always initiates communication. If the remote or the base can initiate the communication, then the base cannot use the standard way of retrying and acknowledging. Instead, the base adds redundancy by sending out multiple copies of the same packet on multiple frequencies. Even if the remote modules get several copies of the packet, they will only pass a single copy of the packet onto the host. The XTend has a special parameter (MT) to enable multiple transmits of the same packet from the base radio. The XStream radios do not have a MT parameter, but can send multiple packets with a special configuration. Please see example 4 of the application note "Addresses and Masks" for more details about the XStream.

EXAMPLE

Situation: communication can be either initiated by base or remote radios.

Product Family: XTend

Radio #1
RR=8
MY=0
DT=FFFF
MT=5

Remote #1
DT=0
MY=1
RR=8
RN=4

Remote #2
DT=0
MY=2
RR=8
RN=4

Remote #3
DT=0
MY=3
RR=8
RN=4

In the above situation, the Base will send out a total of six radio packets for every packet coming over the serial port.