# MACA[1] - A New Channel Access Method for Packet Radio

*Phil Karn, KA9Q*

## *ABSTRACT*

The existing Carrier Sense Multiple Access (CSMA) method widely used in amateur packet radio on shared simplex packet radio channels frequently suffers from the well-known "hidden terminal problem" and the less well known but related problem of the "exposed terminal. " This paper proposes a new scheme, Multiple Access with Collision Avoidance (MACA), that could greatly relieve these problems. MACA can also be easily extended to provide automatic transmitter power control. This could increase the carrying capacity of a channel substantially.

## 1. Introduction

In the classic hidden terminal situation, station Y can hear both stations X and Z, but X and Z cannot hear each other. X and Z are therefore unable to avoid colliding with each other at Y. (See figure 1.)

In the exposed terminal case (figure 2), a well-sited station X can hear far away station Y. Even though X is too fax from Y to interfere with its traffic to other nearby stations, X will defer to it unnecessarily, thus wasting an opportunity to reuse the channel locally. Sometimes there can be so much traffic in the remote area that the well-sited station seldom transmits. This is a common problem with hilltop digipeaters.

This paper suggests a new channel access algorithm for amateur packet radio use that can minimize both problems. This method, Multiple Access with Collision Avoidance (MACA), was inspired by the CSMA/CA method (used by the Apple Localtalk network for somewhat different reasons) and by the "prioritized ACK" scheme suggested by **Eric** Gustafson, N7CL, for AX.25. It is not only an elegant solution to the hidden and exposed terminal problems, but with the necessary hardware support it can be extended to do automatic per-packet transmitter power control. This could substantially increase the "carrying capacity" of a simplex packet radio channel used for local communications in a densely populated area, thus satisfying both the FCC mandate to use "the minimum power necessary to carry out the desired communications" (Part 97.313) and to "contribute to the advancement of the radio art" (Part 97.1 (b)).

## 2. How CSMA/CA Works

CSMA/CA as used by Localtalk works as follows. When a station wants to send data to another, it first sends a short Request To Send (RTS) packet to the destination. The receiver responds with a Clear to Send (CTS) packet. On receipt of the CTS, the sender sends its queued data packet(s). If the sender does not receive a CTS after a timeout, it resends its RTS and waits a little longer for a reply. This three-step process (not counting retransmissions) is called a ***dialogue***. Since a dialogue involves transmissions bv both stations, I will avoid confusion by referring to the station that sends the RTS and data packets ***as*** the ***initiator,*** and the station that sends the CTS as the ***responder.***

The RTS packet tells a responder that data follows. This gives the responder a chance to prepare, e.g., by allocating buffer space or bv entering a "spin loop" on a programmed-I/6 interface. This is the main reason Localtalk uses the CSMA/CA dialogue. The Zilog 8530 HDLC chip used in the Apple Macintosh can buffer the 3-byte Localtalk RTS packet in its FIFO, but without a DMA path to memory it needs the CPU to transfer data to memory as it arrives. The CPU responds to the arrival of an RTS packet by

[1] MACA is an acronym, not a Spanish word.

returning a CTS and entering a tight read loop, waiting for the data to arrive. [2] (A timeout prevents a system lockup if the data never arrives.)

As is standard for CSMA schemes, CSMA/CA requires stations to stay off the channel when another transmission is already in progress. CSMA/CA also requires any station that overhears an **RTS** or CTS packet directed elsewhere to inhibit **its** transmitter for a specified time. This helps reduce the probability of a collision with a subsequent CTS or data packet. This is ***the*** CA or ***Collision Avoidance*** part of CSMA/CA. However, collisions are not a major problem on Localtalk; the network is physically small, carrier sensing is fairly rapid, the data rate is relatively low, and (if the network is properly built) there are no hidden terminals. Plain CSMA would work well, but there was little extra cost to the CA feature (given that the RTS/CTS dialogue was already needed for other reasons) so it was included.

### 3. Turning CSMA/CA into MACA

Hidden and exposed terminals abound on simplex packet radio channels, and this makes them very different from Localtalk and most other types of local area networks. When hidden terminals exist, lack of carrier doesn't always mean it's OK to transmit. Conversely, when exposed terminals exist, presence of carrier doesn't always mean that it's bad to transmit. In other words, the data carrier detect line from your modem is often useless. So I'll make a radical proposal: let's ignore DCD! In other words, let's get rid of the CS in CSMA/CA. (It's too hard to build good DCD circuits anyway...)

Instead we'll extend the CA part of what we'll call MA/CA (or just plain MACA). The key to collision avoidance is the effect that RTS and CTS packets have on the other stations on the channel. When a station overhears an RTS addressed to another station, it inhibits its own transmitter long enough for the addressed station to respond with a CTS. When a station overhears a CTS addressed to another station, it inhibits its **own** transmitter **long enough for the other station to** send its **data.** The **transmitter is inhibited for** the proper **time even if nothing is heard in** response to an **RTS or CTS packet.**

Figure 3 **shows an example. Station Z** cannot **hear X's transmissions to Y,** but it ***can*** **hear** Y's CTS packets to X. If **Z** overhears **a CTS** packet from Y to X, **it will** know **not to** transmit until after Y has received its **data from X.**

But how does **Z** know **how** long **to** wait after overhearing Y's CTS? That's easy. We have X, the initiator of the dialogue, include in its RTS packet the amount of data it plans to send, and we have Y, the responder, echo that information in its CTS packet. Now everyone overhearing Y's CTS knows just how long to wait to avoid clobbering a data packet that it might not even hear.

As long as the **link** between each pair of stations in the network is reciprocal (i.e., all the stations have comparable transmitter powers and receiver noise levels), the receipt of a CTS packet by a station not party to a dialogue tells it that if it were to transmit, it would probably interfere with the reception of data by the responder (the sender of the CTS). MACA thus inhibits transmission when ordinary CSMA would permit it (and allow a collision), thus relieving the hidden terminal problem. (Collisions are not *totally* avoided; more on this point later.)

Conversely, if a station hears no response to an overheard RTS, then it may assume that the intended recipient of the RTS is either down or out of range. An example is shown in figure 4. Station X is within range of Y, but not Z. When Y sends traffic to Z, X will hear Y's RTS packets but not Z's CTS responses. X may therefore transmit on the channel without fear of interfering with Y's data transmissions to Z even though it can hear them. In this case, MACA allows a transmission to proceed when ordinary CSMA would prevent it unnecessarily, thus relieving the exposed terminal problem. (Because modems have a capture effect, hearing a CTS doesn't ***always*** mean that you'd cause a collision if you transmit, so the problem isn't yet completely solved. More on this point later.)

### 4. Metaphors for MACA

MACA is not really a novel idea; it merely formalizes a procedure many people (not just radio amateurs) instinctively use in personal conversation. A typical cocktail party has many

[2] It would be nice if we could use this feature on packet radio with our programmed-I/O HDLC interfaces (e.g., DRSI PCPA, Paccomm PC-l 00). Unfortunately, if our RTS/CTS packets carry full source and destination call signs, they would not fit into the 3-byte 8530 FIFOs. So high speed operation will still require either DMA or a dedicated I/O processor.

simultaneous conversations. The average guest seldom waits for total silence in the room before he speaks, but if someone asks him to pause because he is trying to hear someone else, he will usually do so. The MACA RTS packet is analogous to Bob saying “Hey, Tom!” and CTS packet is analogous to Tom responding with “Yeah, Bob?“. This causes most people to stop talking if they are close to Tom (except, of course, for Bob). The same thing (should) happen in manual amateur radio operation whenever a station finishes a transmission with “go only” (or "KN" on CW or RTTY).

The Prioritized ACK scheme also involves overheard packets that inhibit other stations for specified periods of time. In this case, the inhibiting packet is a data packet and the protected station is sending an acknowledgement that may not be audible at the inhibited stations. Full protection against collisions is not provided (data packets can still collide) but the performance improvement due to the lower ACK loss rate is reported to be substantial.

More formally, MACA can also be seen as a single-channel, time-multiplexed form of Busy Tone Multiple Access (BTMA). In BTMA, receivers transmit “busy tones” on secondary channels whenever their receivers are active. This warns the other stations in range that they should not transmit even if they hear nothing on the data channel. On the other hand, stations not hearing busy tones are free to transmit even if there is already a signal on the data channel. Indeed, stations need not pay any attention at all to the data channel when deciding to transmit; only the busy channel matters. As long as the propagation characteristics are identical between the main and secondary (busy tone) channels, BTMA is effective. Unfortunately, the need to use widely separated frequencies to avoid self-interference tends to make the link characteristics less symmetrical. BTMA also obviously increases the hardware complexity and spectrum requirements of each user station. On the other hand, because MACA uses the same channel for the “busy tone” and data, paths between pairs of stations are much more likely to be symmetrical.

## 5. Collisions in MACA

Unlike BTMA, however, collisions between RTS packets can still occur in MACA. These are minimized with a randomized exponential back-off strategy similar to that used in regular CSMA. Since there is no carrier sensing in MACA, each station simply adds a random amount to the minimum interval each station is required to wait after overhearing an RTS or CTS packet. As in regular CSMA, this strategy minimizes the chance that several stations will all jump on the channel at the instant it becomes free. The extra random interval would be an integral multiple of the “slot time”, and in MACA the slot time is the duration of an RTS packet. If two RTS packets collide nonetheless, each station waits a randomly chosen interval and tries again, doubling the average interval on each successive attempt. Eventually one of them will “win” (i.e., transmit first) and the CTS from its responder will inhibit the “losing” station until the winning station can complete its dialogue.

Even though collisions can occur between RTS packets, MACA still has the advantage over CSMA as long as the RTS packets are significantly smaller than the data packets. As long as this is true, collisions between RTS packets are much less “costly” than the collisions that would otherwise occur between data packets. The savings in collision time also pays for the overhead of the RTS and CTS packets.

As mentioned earlier, the basic MACA protocol only reduces the chances of collisions involving data packets; it does not guarantee that they will never occur. This is because a CTS packet requires a certain minimum signal-to-noise ratio at a station for it to be understood and obeyed. Even if the station powers are well matched, a pair of stations might have just enough of a path between them to allow them to interfere with each other’s reception of weak signals, but not enough of a path to allow them to hear each other’s CTS packets. Although the seriousness of this problem is unknown, it does appear that the power-controlled version of MACA discussed later would greatly reduce it.

## 6. Bypassing the MACA Dialogue

If the data packets are of comparable size to the RTS packets, the overhead of the RTS/CTS dialogue may be excessive. In this case, a station may choose to bypass the normal dialogue by simply sending its data without the dialogue. It must, of course, still defer to any RTS or CTS packets it may overhear.

Of course, the bypass mechanism carries with it the risk of a collision. However, for some types of data packets this may be an acceptable tradeoff. An example might be the acknowledgements in a sliding-window TCP transfer.[3] TCP ACKs are cumulative, so the loss of a single ACK causes no harm as long as another one gets through before the sending TCP fills its window.

### 7. Automatic Power Control

MACA lends itself well to automatic transmitter power control. To support this we need some extra hardware: a D/A converter that controls transmitter power level, and an A/D converter that gives received signal strengths. By including calibrated "S-meter" readings[4] in CTS packets, responders could help initiators to adjust their power levels accordingly.

Each RTS/CTS exchange updates the initiator's estimate of the power needed to reach a particular responder so that future packets (including the data packet in the current dialogue) can be sent with only the necessary power. Even RTS packets could be sent at reduced power, since their main purpose is to elicit a CTS from the responder. This reduces the probability of collision between RTS packets.

By changing the MACA rule to "inhibit a transmitter when a CTS packet is overheard" to "temporarily limit power output when a CTS packet is overheard," geographic reuse of the channel can be significantly improved. For example, if station X has recently sent traffic to station Y, it knows how much power is required to reach Y. If X overhears station Y responding with a CTS to a third station Z, then X need not remain completely silent for the required interval; it need only limit its transmitter power to, say, 20 dB [5] below the level needed to reach Y. During this time it would be free to transmit to any station that it could reach with that reduced power level, because its signal at Y would be overridden by Z's signal. (This is analogous to the people at the cocktail party continuing their conversations in whispers instead of stopping completely when Tom tells Bob to go ahead.)

The CTS packets, however, pose a problem. In addition to telling the initiator to send its data, the CTS must inhibit all potential interferers from transmitting. It may therefore need more power than that needed just to reach the initiator to ensure that everyone "gets the message." (A CTS packet might therefore be more like Tom shouting "Hey, everyone, shut up! I'm trying to hear Bob speak!" at the cocktail party mentioned earlier.)

All this shouting potentially limits the geographic channel reuse ability we've worked so hard to get. But all is not lost. A station responding to an RTS with a CTS can always expect data to follow. If it doesn't arrive within a reasonable period, or if a retransmitted RTS arrives instead, then either the CTS was stepped on, or the CTS wasn't heard widely enough to prevent the data transmission that follows from being stepped on. It should then respond to the next RTS from the same station (which will likely be a repeated attempt to send the same data) with a CTS at higher power. On the other hand, if a responder has had good luck in getting data in response to its CTS packets, it might try lowering the power it uses to transmit them in order to help limit channel loading. Of course, it would never lower its CTS power below the level it knows is necessary to reach the initiator.

In sum, MACA with power control automatically determines the exact amount of power required for each RTS and data transmission, and learns by experience (i.e., trial and error) the power required for CTS transmissions. It also appears to avoid the runaway power escalation that can occur when power control is done on a conventional CSMA channel when stations naively "turn up the wick" each time they fail to get through. About the only time power escalation seems possible in MACA is when an initiator's receiver fails so it is not able to hear CTS responses to its RTS packets no matter how much power the responder uses. This possibility should be handled by back-offs and/or retry limits in the dialogue code.

[3] The use of sliding windows in TCP might seem to contradict the advice I gave several years ago to always operate in stop-and-wan mode (MAXFRAME 1) on half duplex channels. However, that conclusion applied only to link level protocols; TCP is an end-to-end transport protocol. Sliding windows are usually appropriate in a transport protocol even when the individual hops in the network path are half duplex.

[4] Only one point in the S-meter scale really needs to be calibrated. This is the signal level just high enough to achieve an acceptable bit error rate. A more completely calibrated scale makes it easier for the transmitter to zero in on the correct power setting, but even a simple "too strong/too weak/OK" indication is enough for a transmitter to determine the correct power level by Newtonian iteration.

[5] This figure depends on the capture ratio of the modems in use.

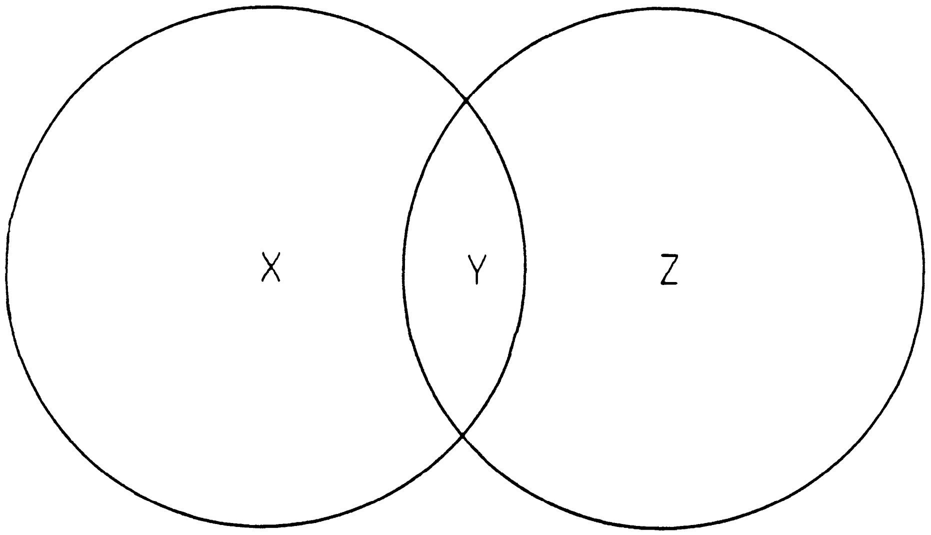
## 8. Applications for MACA

If MACA proves effective, it may *finally* make single-frequency amateur packet radio networks practical. Although it would still be preferable for fixed backbones to use separate, dedicated channels or point-to-point links whenever possible. the ability to create usable, ad-hoc, single frequency networks could be very useful in certain situations. These include user access channels (such as 145.01 MHz in many areas) and in temporary portable and mobile operations where it is often infeasible to coordinate a multi-frequency network in advance. This would be especially useful for emergency situations in remote areas without dedicated packet facilities.

An ideal emergency packet radio network would consist of identical stations operating on a common frequency (to maximize interchangeability) placed in arbitrary locations. These stations would automatically discover their neighbors and build routing and power control tables that maximize the total amount of traffic that can be carried in the coverage area. To do this, routing algorithms would use a different metric than usual. Instead of simply minimizing the number of hops needed to reach a given destination, the routing algorithm would instead minimize the *total transmitter energy* required by all the stations along a path to the destination. Because of the laws of RF propagation (doubling the range of a signal in free space requires four times as much transmitter power, and on the ground it may take much more). this approach would often *increase* the number of hops required to reach a given destination. However, overall network throughput would increase because the lower transmitter power levels would permit more simultaneous transmissions to occur in different parts of the network without interference. This would also minimize the power consumed at the stations, and this could be important when operating from batteries. The direct, minimum-hop path could still be provided as an option for special applications requiring minimum delay.

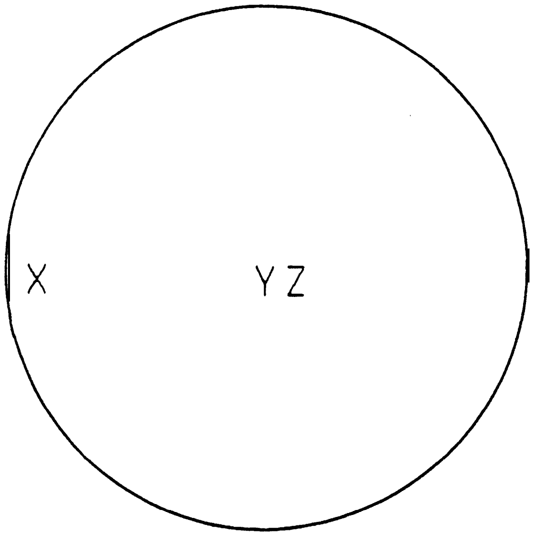
## 9. Conclusion and Open Questions

At the moment, MACA is just an idea. Much simulation and experimental work needs to be done to answer many questions about how well it will really work. Here are just some of the questions that can be asked. How much of the savings from avoided collisions in MACA is spent on RTS/CTS overhead given typical modem turnaround times and data packet sizes? How much better does power-controlled MACA perform than the basic MACA scheme? How about a partial implementation of power control, e.g., one that relies on trial-and-error instead of explicit S-meter feedback? How do the various forms of MACA behave as modem capture ratios change? How serious is the problem of interference from stations just below threshold? And how does MACA compare in overall spectral efficiency with other improved multiple access methods, such as conventional CSMA or CSMA/CD operation through full duplex repeaters? I invite anyone interested in pursuing these topics to contact me.



X and Z collide at Y

Fig 1. Hidden Terminal



X defers unnecessarily to Y's transmissions to Z

Fig 2. Exposed Terminal

returning a CTS and entering a tight read loop, waiting for the data to arrive. [2] (A timeout prevents a system lockup if the data never arrives.)

As is standard for CSMA schemes, CSMA/CA requires stations to stay off the channel when another transmission is already in progress. CSMA/CA also requires any station that overhears an **RTS** or CTS packet directed elsewhere to inhibit **its** transmitter for a specified time. This helps reduce the probability of a collision with a subsequent CTS or data packet. This is ***the*** CA or ***Collision Avoidance*** part of CSMA/CA. However, collisions are not a major problem on Localtalk; the network is physically small, carrier sensing is fairly rapid, the data rate is relatively low, and (if the network is properly built) there are no hidden terminals. Plain CSMA would work well, but there was little extra cost to the CA feature (given that the RTS/CTS dialogue was already needed for other reasons) so it was included.

### 3. Turning CSMA/CA into MACA

Hidden and exposed terminals abound on simplex packet radio channels, and this makes them very different from Localtalk and most other types of local area networks. When hidden terminals exist, lack of carrier doesn't always mean it's OK to transmit. Conversely, when exposed terminals exist, presence of carrier doesn't always mean that it's bad to transmit. In other words, the data carrier detect line from your modem is often useless. So I'll make a radical proposal: let's ignore DCD! In other words, let's get rid of the CS in CSMA/CA. (It's too hard to build good DCD circuits anyway...)

Instead we'll extend the CA part of what we'll call MA/CA (or just plain MACA). The key to collision avoidance is the effect that RTS and CTS packets have on the other stations on the channel. When a station overhears an RTS addressed to another station, it inhibits its own transmitter long enough for the addressed station to respond with a CTS. When a station overhears a CTS addressed to another station, it inhibits its **own** transmitter **long enough for the other station to** send its data. The **transmitter is inhibited for** the proper **time even if nothing is heard in** response to an **RTS or CTS packet.**

Figure 3 **shows an example. Station Z** cannot **hear X's transmissions to Y,** but it ***can* hear** Y's CTS packets to X. If Z overhears **a CTS** packet from Y to X, **it will** know **not to** transmit until after Y has received its **data from** X.

But how does Z know **how** long **to** wait after overhearing Y's CTS? That's easy. We have X, the initiator of the dialogue, include in its RTS packet the amount of data it plans to send, and we have Y, the responder, echo that information in its CTS packet. Now everyone overhearing Y's CTS knows just how long to wait to avoid clobbering a data packet that it might not even hear.

As long as the **link** between each pair of stations in the network is reciprocal (i.e., all the stations have comparable transmitter powers and receiver noise levels), the receipt of a CTS packet by a station not party to a dialogue tells it that if it were to transmit, it would probably interfere with the reception of data by the responder (the sender of the CTS). MACA thus inhibits transmission when ordinary CSMA would permit it (and allow a collision), thus relieving the hidden terminal problem. (Collisions are not *totally* avoided; more on this point later.)

Conversely, if a station hears no response to an overheard RTS, then it may assume that the intended recipient of the RTS is either down or out of range. An example is shown in figure 4. Station X is within range of Y, but not Z. When Y sends traffic to Z, X will hear Y's RTS packets but not Z's CTS responses. X may therefore transmit on the channel without fear of interfering with Y's data transmissions to Z even though it can hear them. In this case, MACA allows a transmission to proceed when ordinary CSMA would prevent it unnecessarily, thus relieving the exposed terminal problem. (Because modems have a capture effect, hearing a CTS doesn't ***always*** mean that you'd cause a collision if you transmit, so the problem isn't yet completely solved. More on this point later.)

### 4. Metaphors for MACA

MACA is not really a novel idea; it merely formalizes a procedure many people (not just radio amateurs) instinctively use in personal conversation. A typical cocktail party has many

[2] It would be nice if we could use this feature on packet radio with our programmed-I/O HDLC interfaces (e.g., DRSI PCPA, Paccomm PC-l 00). Unfortunately, if our RTS/CTS packets carry full source and destination call signs, they would not fit into the 3-byte 8530 FIFOs. So high speed operation will still require either DMA or a dedicated I/O processor.

simultaneous conversations. The average guest seldom waits for total silence in the room before he speaks, but if someone asks him to pause because he is trying to hear someone else, he will usually do so. The MACA RTS packet is analogous to Bob saying "Hey, Tom!" and CTS packet is analogous to Tom responding with "Yeah, Bob?". This causes most people to stop talking if they are close to Tom (except, of course, for Bob). The same thing (should) happen in manual amateur radio operation whenever a station finishes a transmission with "go only" (or "KN" on CW or RTTY).

The Prioritized ACK scheme also involves overheard packets that inhibit other stations for specified periods of time. In this case, the inhibiting packet is a data packet and the protected station is sending an acknowledgement that may not be audible at the inhibited stations. Full protection against collisions is not provided (data packets can still collide) but the performance improvement due to the lower ACK loss rate is reported to be substantial.

More formally, MACA can also be seen as a single-channel, time-multiplexed form of Busy Tone Multiple Access (BTMA). In BTMA, receivers transmit "busy tones" on secondary channels whenever their receivers are active. This warns the other stations in range that they should not transmit even if they hear nothing on the data channel. On the other hand, stations not hearing busy tones are free to transmit even if there is already a signal on the data channel. Indeed, stations need not pay any attention at all to the data channel when deciding to transmit; only the busy channel matters. As long as the propagation characteristics are identical between the main and secondary (busy tone) channels, BTMA is effective. Unfortunately, the need to use widely separated frequencies to avoid self-interference tends to make the link characteristics less symmetrical. BTMA also obviously increases the hardware complexity and spectrum requirements of each user station. On the other hand, because MACA uses the same channel for the "busy tone" and data, paths between pairs of stations are much more likely to be symmetrical.

## 5. Collisions in MACA

Unlike BTMA, however, collisions between RTS packets can still occur in MACA. These are minimized with a randomized exponential back-off strategy similar to that used in regular CSMA. Since there is no carrier sensing in MACA, each station simply adds a random amount to the minimum interval each station is required to wait after overhearing an RTS or CTS packet. As in regular CSMA, this strategy minimizes the chance that several stations will all jump on the channel at the instant it becomes free. The extra random interval would be an integral multiple of the "slot time", and in MACA the slot time is the duration of an RTS packet. If two RTS packets collide nonetheless, each station waits a randomly chosen interval and tries again, doubling the average interval on each successive attempt. Eventually one of them will "win" (i.e., transmit first) and the CTS from its responder will inhibit the "losing" station until the winning station can complete its dialogue.

Even though collisions can occur between RTS packets, MACA still has the advantage over CSMA as long as the RTS packets are significantly smaller than the data packets. As long as this is true, collisions between RTS packets are much less "costly" than the collisions that would otherwise occur between data packets. The savings in collision time also pays for the overhead of the RTS and CTS packets.

As mentioned earlier, the basic MACA protocol only reduces the chances of collisions involving data packets; it does not guarantee that they will never occur. This is because a CTS packet requires a certain minimum signal-to-noise ratio at a station for it to be understood and obeyed. Even if the station powers are well matched, a pair of stations might have just enough of a path between them to allow them to interfere with each other's reception of weak signals, but not enough of a path to allow them to hear each other's CTS packets. Although the seriousness of this problem is unknown, it does appear that the power-controlled version of MACA discussed later would greatly reduce it.

## 6. Bypassing the MACA Dialogue

If the data packets are of comparable size to the RTS packets, the overhead of the RTS/CTS dialogue may be excessive. In this case, a station may choose to bypass the normal dialogue by simply sending its data without the dialogue. It must, of course, still defer to any RTS or CTS packets it may overhear.

Of course, the bypass mechanism carries with it the risk of a collision. However, for some types of data packets this may be an acceptable tradeoff. An example might be the acknowledgements in a sliding-window TCP transfer.[3] TCP ACKs are cumulative, so the loss of a single ACK causes no harm as long as another one gets through before the sending TCP fills its window.

## 7. Automatic Power Control

MACA lends itself well to automatic transmitter power control. To support this we need some extra hardware: a D/A converter that controls transmitter power level, and an A/D converter that gives received signal strengths. By including calibrated "S-meter" readings[4] in CTS packets, responders could help initiators to adjust their power levels accordingly.

Each RTS/CTS exchange updates the initiator's estimate of the power needed to reach a particular responder so that future packets (including the data packet in the current dialogue) can be sent with only the necessary power. Even RTS packets could be sent at reduced power, since their main purpose is to elicit a CTS from the responder. This reduces the probability of collision between RTS packets.

By changing the MACA rule to "inhibit a transmitter when a CTS packet is overheard" to "temporarily limit power output when a CTS packet is overheard," geographic reuse of the channel can be significantly improved. For example, if station X has recently sent traffic to station Y, it knows how much power is required to reach Y. If X overhears station Y responding with a CTS to a third station Z, then X need not remain completely silent for the required interval; it need only limit its transmitter power to, say, 20 dB [5] below the level needed to reach Y. During this time it would be free to transmit to any station that it could reach with that reduced power level, because its signal at Y would be overridden by Z's signal. (This is analogous to the people at the cocktail party continuing their conversations in whispers instead of stopping completely when Tom tells Bob to go ahead.)

The CTS packets, however, pose a problem. In addition to telling the initiator to send its data, the CTS must inhibit all potential interferers from transmitting. It may therefore need more power than that needed just to reach the initiator to ensure that everyone "gets the message." (A CTS packet might therefore be more like Tom shouting "Hey, everyone, shut up! I'm trying to hear Bob speak!" at the cocktail party mentioned earlier.)

All this shouting potentially limits the geographic channel reuse ability we've worked so hard to get. But all is not lost. A station responding to an RTS with a CTS can always expect data to follow. If it doesn't arrive within a reasonable period, or if a retransmitted RTS arrives instead, then either the CTS was stepped on, or the CTS wasn't heard widely enough to prevent the data transmission that follows from being stepped on. It should then respond to the next RTS from the same station (which will likely be a repeated attempt to send the same data) with a CTS at higher power. On the other hand, if a responder has had good luck in getting data in response to its CTS packets, it might try lowering the power it uses to transmit them in order to help limit channel loading. Of course, it would never lower its CTS power below the level it knows is necessary to reach the initiator.

In sum, MACA with power control automatically determines the exact amount of power required for each RTS and data transmission, and learns by experience (i.e., trial and error) the power required for CTS transmissions. It also appears to avoid the runaway power escalation that can occur when power control is done on a conventional CSMA channel when stations naively "turn up the wick" each time they fail to get through. About the only time power escalation seems possible in MACA is when an initiator's receiver fails so it is not able to hear CTS responses to its RTS packets no matter how much power the responder uses. This possibility should be handled by back-offs and/or retry limits in the dialogue code.

[3] The use of sliding windows in TCP might seem to contradict the advice I gave several years ago to always operate in stop-and-wan mode (MAXFRAME 1) on half duplex channels. However, that conclusion applied only to link level protocols; TCP is an end-to-end transport protocol. Sliding windows are usually appropriate in a transport protocol even when the individual hops in the network path are half duplex.

[4] Only one point in the S-meter scale really needs to be calibrated. This is the signal level just high enough to achieve an acceptable bit error rate. A more completely calibrated scale makes it easier for the transmitter to zero in on the correct power setting, but even a simple "too strong/too weak/OK" indication is enough for a transmitter to determine the correct power level by Newtonian iteration.

[5] This figure depends on the capture ratio of the modems in use.

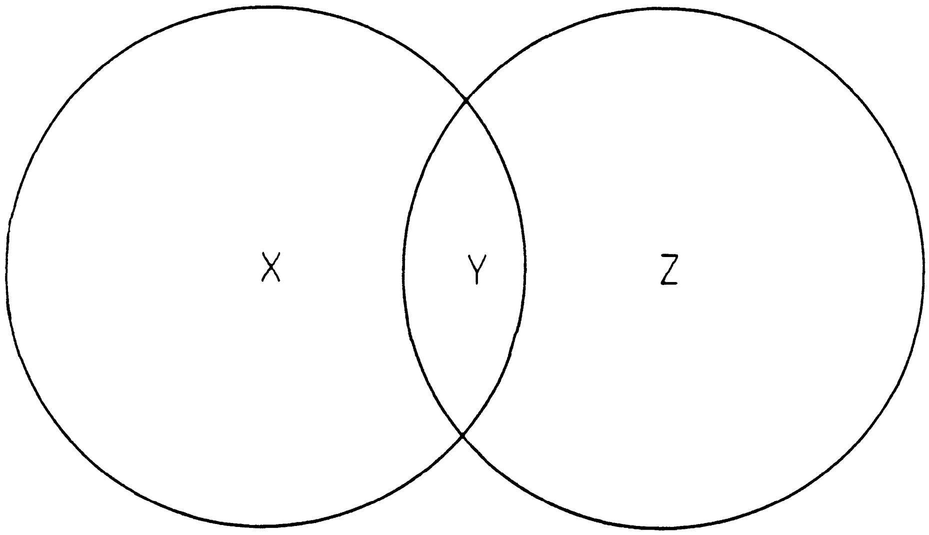
## 8. Applications for MACA

If MACA proves effective, it may *finally* make single-frequency amateur packet radio networks practical. Although it would still be preferable for fixed backbones to use separate, dedicated channels or point-to-point links whenever possible. the ability to create usable, ad-hoc, single frequency networks could be very useful in certain situations. These include user access channels (such as 145.01 MHz in many areas) and in temporary portable and mobile operations where it is often infeasible to coordinate a multi-frequency network in advance. This would be especially useful for emergency situations in remote areas without dedicated packet facilities.

An ideal emergency packet radio network would consist of identical stations operating on a common frequency (to maximize interchangeability) placed in arbitrary locations. These stations would automatically discover their neighbors and build routing and power control tables that maximize the total amount of traffic that can be carried in the coverage area. To do this, routing algorithms would use a different metric than usual. Instead of simply minimizing the number of hops needed to reach a given destination, the routing algorithm would instead minimize the *total transmitter energy* required by all the stations along a path to the destination. Because of the laws of RF propagation (doubling the range of a signal in free space requires four times as much transmitter power, and on the ground it may take much more). this approach would often *increase* the number of hops required to reach a given destination. However, overall network throughput would increase because the lower transmitter power levels would permit more simultaneous transmissions to occur in different parts of the network without interference. This would also minimize the power consumed at the stations, and this could be important when operating from batteries. The direct, minimum-hop path could still be provided as an option for special applications requiring minimum delay.

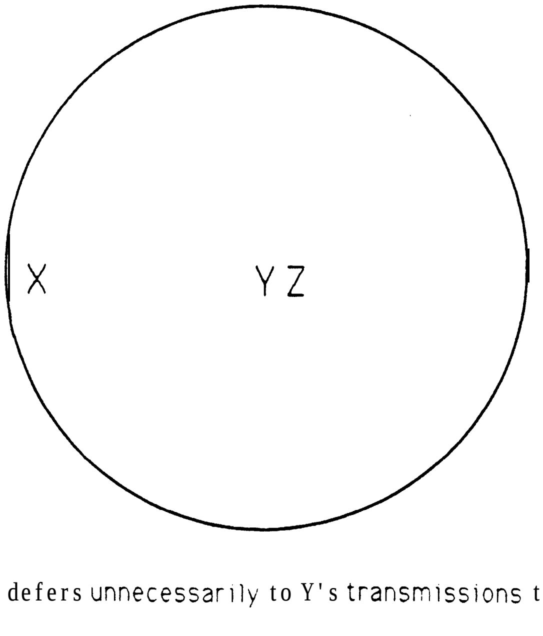
## 9. Conclusion and Open Questions

At the moment, MACA is just an idea. Much simulation and experimental work needs to be done to answer many questions about how well it will really work. Here are just some of the questions that can be asked. How much of the savings from avoided collisions in MACA is spent on RTS/CTS overhead given typical modem turnaround times and data packet sizes? How much better does power-controlled MACA perform than the basic MACA scheme? How about a partial implementation of power control, e.g., one that relies on trial-and-error instead of explicit S-meter feedback? How do the various forms of **MACA** behave as modem capture ratios change? How serious is the problem of interference from stations just below threshold? And how does MACA compare in overall spectral **efficiency** with other improved multiple access methods, such as conventional CSMA or CSMA/CD operation through full duplex repeaters? I invite anyone interested in pursuing these topics to contact me.



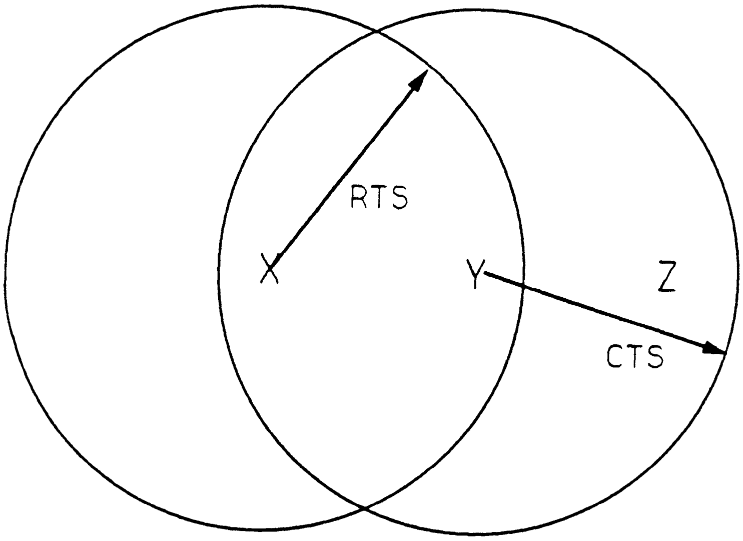
X and Z collide at Y

Fig 1. Hidden Terminal



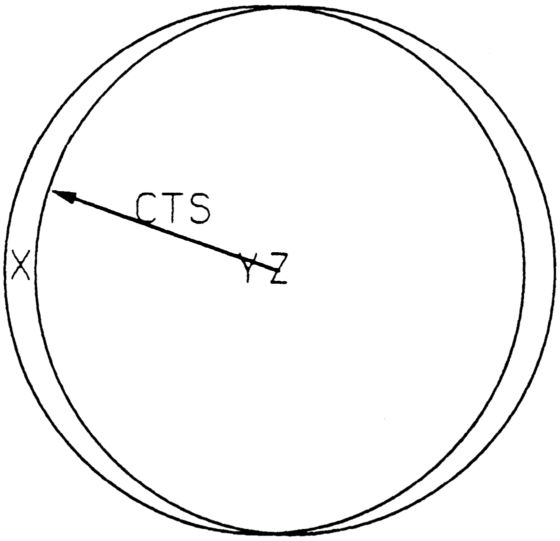
X defers unnecessarily to Y's transmissions to Z

Fig 2. Exposed Terminal



Y's CT-S holds off Z

Fig 3. MACA with hidden terminal



X doesn't hear Z's CTS, so it doesn't need to defer to Y

Fig **4.** MACA with exposed terminal