Media Access Methods

Table of Contents

Media Access Methods -1 ............................................................................................................... 2
Media Access Methods -2 ............................................................................................................... 6
Notices .......................................................................................................................................... 10
Prevents collisions, or multiple hosts talking at once

CSMA – Carrier Sense, Multiple Access

- CSMA with Collision Avoidance (CSMA/CA) – hosts sense if a transmission is taking place, and if so, wait a random period of time; used by 802.11 wireless
- CSMA with Collision Detection (CSMA/CD) – hosts stop and wait a random period of time before retransmitting; used by Ethernet

So, these nodes get onto this network, and how do they know when it’s their turn to talk? Well, what we do with most of our networking today is we do what’s called contention-based. We fight for the wire.

Now, those collisions occur. We want to limit those collisions the best way we possibly can. So multiple hosts will jump onto the wire in a contention-based network, and then they’ll go, “Oh, I’m sorry, I see that there’s a collision there. I'll let that one finish.”
So they have two ways of doing this: They can listen to see if it's all clear 
and transmit, or what they can do is 
they can just go ahead and jump out 
onto the wire. When we talk about 
collision avoidance-- CSMA/CA. 
That's carrier-sensing multiple 
access. I sense the carrier, and then 
what I will do is I will do a collision 
avoidance technique. That collision 
avoidance technique, when we talk 
about wireless, is to listen for a 
beacon or to beacon. "I'm about to 
send. Here, I'll drop this in, and now 
I can send." Because when I drop 
that in, everybody else is listening. I 
listen for the beacon. "Somebody's 
already dropped one in. Okay, I 
won't transmit."

Now, the problem with collision 
avoidance in wireless networks is the 
further and further that we spread 
the nodes out, theoretically two 
devices could beacon at the same 
time and jump onto that, and we 
would create the collision all over 
again.

So I really like the standard-- follow 
what’s in the standard and use what's 
there for the standard-- but then 
there’s an unfortunate problem that 
we run into, and that is not all 
countries play the same game. You 
can go online today and get a 
wireless-- we'll call it a wireless node; 
basically it’s a USB plugin to your 
laptop-- that will transmit at 500, 
1000, and even 2000 millamps. You 
say, "Well, what's the big deal in 
that?" Well, now what happens is, is 
when I transmit my beacon, because
I can transmit it further away, then I will cause a contention on the wire that's inappropriate and confuse conversations. Evildoers do this all the time. They basically blot out the sun of others.

But there is one time when this happens that we have to be careful, and it's not because of people that we work with in our different offices and how we've got things set up; it's when we deal with ham radio operators, because they have the ability to transmit a lot larger signal over a longer period of time, and that could screw up our signals, because they would overlap.

So when we're talking about carrier-sensing multiple access, we should plan by using all the same devices following the same set of rules. The FCC in the U.S. actually monitors and mentioned and says, "You can't sell those devices." But I can go on eBay in Germany and buy a German wireless access point that goes way beyond that normal scope, and then that could cause problems.

So that's collision avoidance--beaconing--putting it out on the wire before you actually transmit. CSMA/CD, collision detection, there's no beaconing whatsoever. It says, "I'm going to just floor it and reverse back into the street and see what happens. And when I floor it and back up, if I run into somebody, I go, "Oh, that's what my bumper is for." And then I pull forward and I let that
person pass, and I just keep on doing that.

The problem is, is that on this particular piece of our network where there is a single collision domain, what happens is we tend to collide with a lot more-- other nodes or packets on the network-- we tend to collide a lot more when we have more and more devices on that network. So when it comes to CSMA/CD and the original use, where we would have a hub, at a certain point in time when we plug too many hosts into that hub, what we would do is we would get so much collision that no work would actually get done. And then we'll talk about switches and bridges and how they actually helped us with this, separating collision domains, in a little bit.

So that's, first, our contention-based method.
Media Access Methods -2

Token Passing

- A host can only transmit when it has the token; token is passed to next host in ring when done or after a certain time
- Used by Token Ring

Polling

- A host can only transmit when it has permission of the master controller; controller queries hosts to see who needs to transmit
- Used by Bluetooth devices

**014 Then there are two other methodologies that are out there: token passing, which is used on Token Ring; and polling. In token passing, we don't contend for the media; we wait our turn for the token, or the talking stick, as it were. So we keep on passing that around the ring. Now, the way the token works is I say, "Okay, here's my token, right here." This clicker right here. And what I'm going to do, is the token comes to me and I look inside of it, and there's no destination on this token. So what I do is say, "I want to talk to that student right there." And so what I do is I put my request to that student, and also
from me, into this token, and then I pass it, from student to student to student.

As the students pick it up, they say, "Well, this isn't destined for me. I'll just retransmit it." "This isn't destined for me. I'll retransmit it." And it passes all the way around until it gets to that student right there. They pick it up and they go, "What do you want?" And then they transmit it back to me. Now what they do is they replace it and they put their node number in there and say, "I'm the destination node number," and it passes around to me.

All of the nodes that look at this as it's passing it around, this is really effective and really efficient, and they're not allowed to say anything at this moment in time, according to the rules of Token Ring. Only these two people are allowed to have this communication. So now what I'll do is I'll say, "I request this," and I pass it on around the ring. He fulfills that, and sends it back to me. And I just keep on doing that over and over and over again until our conversation has finished.

What's really elegant about this also is, is they're very low packet overhead. So when we transmit between each other, since we don't have to contend with others on the ring, the standard setup is that the data portion of this particular token that's being passed with these send-and-receive nodes identified on it can
be much larger. When we look at a contention-based media like TCP/IP-- I’m sorry, Ethernet and TCP/IP-- we’re maxed out at about 1400 bytes. But in Token Ring, when we pass this around the ring, we’re up to 4096 bytes. That’s a lot more bang for the buck in a single communication because, remember, none of the other nodes can talk on this. So we keep on passing that around back and forth. We want to get that conversation over with quickly enough so that we can then release it and say, "Okay, well, you two want to talk. And then you two want to talk, and you two want to talk to each other."

So when we talk about token passing, it’s like the talking stick, if you will, being passed between these different nodes on the network. It is used in Token Ring. It is a very elegant way to do things. Unfortunately, Token Ring is a little bit slower as far as transmission is concerned as compared with Ethernet today when we’re running our individual ports, when we’re running at gigabit speeds. So it tends to cause us some problems. We don’t choose token passing that much, unless we’ve got a FID1 ring. They happen every once in a while in a metropolitan area network. Again, we’ll talk about that later.

The last way to do this is through polling. And polling is usually associated with-- today is associated with mainframes, and then Bluetooth devices. And in polling, what I do is
there's a master that says, "Do you want to talk?" And I can skip all of those nodes right there and say, "Do you want to talk?" I can change my priority. You're the most important person, so therefore what I'm going to do is I'm going to ask you twice before I ask everybody else once. So I'll go ahead and say-- I'll just go around the room and I'll go, "Once, once, once, once." To you, "Do you want to talk?"

Well, what I could do is I say, "Once, once, once, once, once. Do you want to talk? Okay." Check these people. "Do you want to talk?" And I could increase your priority in polling. And polling was primarily done from a mainframe standpoint. Most of the time, what was beautiful about the polling concept when we were dealing with mainframes, is we skipped from layer two-- polling-- all the way up to layer five, which we'll talk about in TCP. The beauty of that is it simplifies things. The problem is, is now we've got one central control here and there might be, theoretically, political infighting to get more polling done for a more important workstation.

Now, this is done with Bluetooth devices, but this is only done in what we call a personal area network, when we've only got like three or four devices that are communicating. It works really well in limited-number-of-nodes setups, and mainframes had that condition in place, so polling worked very well.