Problems with WiRES-X 2.X

Category: WiRES-X
August 9, 2026
Note: Yaesu has now released version 2.01. Our recommendation remains unchanged. This version mostly addresses those few people who need the software to work with the FT-1.

Yaesu recently released version 2.x of the WiRES-X software. These are the primary things to know:

- It is the only version of WiRES-X that will work with the FT-1. (But hopefully you're using your FT-1 to do more than make a WiRES-X node!)
- IPv6 is now supported. Good news for people stuck with ISPs that only do IPv6.
- There is an IPv4 compatibility mode that can be used.
- Version 2.x IS NOT backward compatible. Version 2.x CANNOT connect to rooms running 1.5.x.
- Most Room operators are sticking with version 1.5.x because of bugs and other issues. This means that if you update to 2.x, you will NO LONGER be able to connect to almost all of the popular Rooms.
- Version 2.0 is buggy. It is a complex update to the software and it may take several releases before Yaesu gets it stable.

Note: I've been taking one for the team and have been running FusionTech on 2.01 for more than a month. These are the reasons why I don't like running a room on 2.01.

- This really is a problem! If you're running a room and need to reboot WiRES-X, you want that reboot to go as fast as possible. With a 1.5.7 reboot, nodes would rejoin the room immediately. This is because the room still exists in

the Yaesu list server. Nodes will automatically try to rejoin every 30 seconds. Thus within 30 seconds the nodes have rejoined. This is NOT TRUE with 2.0.1!!! If a 2.0.1 room is rebooted, it sends a command to the list server to DELETE THE ROOM'S ENTRY IN THE LIST SERVER. This means that a node has to propagate through the system. The result is that it can take more than 10 MINUTES for nodes to rejoin. Very bad if this happens during a net! (It has happened to me a number of times.)

- But it gets worse. I use the repeater that hosts the room to bridge to our YSF reflector. However the repeater can't rejoin the room until the Yaesu list server updates the local list of rooms. Depending upon where your room is in the list, this can take 10 to 15 minutes. So in the middle of the net, the YSF users are disconnected from WiRES-X for a very long time as is true for all users on the local repeater – which probably includes net control.
- What???? The node running on the same WiRES-X instance as the room cannot connect to that room right away? Nope. Even though both are running in the same software!
- Here's what's happening. 1.5.7 used a distributed model for providing rooms. When connecting to a room the traffic between your node and the room goes from you WiRES-X to the WiRES-X software used to host the room. Not so with 2.0.1. With 2.0.1 your WiRES-X connects to a Yaesu server in Japan. All of the rooms are hosted in Japan, not on the rooms WiRES-X software. This has advantages and disadvantages.
- Advantages: Port forwarding is not required since your WiRES-X software connects directly to a public server – like pulling up a web page. No port forwarding is needed. The efficiency of hosting on a server is much better than hosting on a consumer IP service. (Performance is mostly constant and not degraded by Netflix.) This also

integrates well when you don’t host a room and directly connect your computer to your PC via PDN. In this case Yaesu uses TCP to connect directly to the server in Japan. Japan then converts the TCP to UDP and sends the traffic to the host WiRES-X. Now both methods do the same thing.

- Disadvantages: Yaesu chose to use TCP to send audio. Nobody does this. When you listen to a podcast, music online, or watch Netflix, this is all done over UDP. Why? TCP requires a hand-shake between the server and the client. It also retries transmissions if one is lost. It guarantees that all packets are received in the correct order? Nice, eh? Until a packet or two get lost and a retry occurs. This can cause a delay of up to a second based on the ping times between the two computers. Thus the audio will have a noticeable stutter even though only 100 ms of data is lost. TCP is much more compute intensive than UDP. UDP does not have a hand-shake. Once the connection is made it just starts streaming packets to the client. The packets may arrive out of order if the Internet path between the two computers changes from one packet to the next. One would think this is terrible, but it isn’t. It means that UDP keeps on sending even if random packets get dropped. You’ll hardly notice a dropped packet (100 ms, or part of a word) versus loosing the time it takes for a retry – a second or more. You won’t notice a dropped packet on opening a web page, but you will certainly notice it when watching video or listening to audio.
- Why is Yaesu using TCP? Nobody else uses it for this purpose? I honestly don’t know. We know that UDP works fine because with 1.5.7 that’s what we’re using. Maybe they’re using TCP so that a port is open to connect to a web server? Or something? This makes no sense as they could just as easily have the UDP port open. That works.

That’s what we do with 1.5.7. TCP requires more server overhead, while UDP is very, very simple.

- Yaesu never needed users to open ports!!!! Here’s how that works. Let’s say A and B what to have a VoIP conversation using UDP. But A can’t talk to B because B doesn’t have any open ports. That where a public server, C, comes in. Both A and B connect to C and tell C who they are and what their IP address is. When A want’s to call B, A contacts C with his intentions. C responds with B’s IP address. Meanwhile the server lets B know that A is trying to contact him so B then sends a UDP packet to A. Meanwhile A is sending a UDP packet to B. These packet probably never get received but what they do is important. Sending a UDP packet causes your NAT router/firewall to open an incoming port on the same port you send the UDP packet from. Thus A and B have both opened a UDP port known only to A and B, and they can start sending packets directly to each other without involving C. This is how the older version of WiRES-X should have worked! This is what everybody else does. It’s not a secret.
- YSF works and an open port is not required. In many ways YSF is a lot nicer than WiRES-X. It only uses UDP. When a YSF user wants to connect to a reflector, they look in a Host file for the IP address and port number of the reflector. They send a UDP packet to that address, opening up an incoming port. The reflector then sends traffic to that port. YSF only requires an open UDP port if a reflector is being hosted and most people are not doing this. There are other advantages to YSF over WiRES-X which I will point out is a subsequent post.
- Single point of failure. Since ALL WiRES-X traffic is now going to through one service, failure of that service shuts down the ENTIRE WiRES-X network. This can happen when Yaesu does a reboot. No matter when they do a reboot

it will be prime time for someone on this earth. That wouldn't happen with 1.5.7. Only people that have just booted WiRES-X will notice that the list server is down. That's because everyone who has WiRES-X running already have a cache of all the nodes and rooms – they're just not getting any updates. But with 2.0.1 everything goes down. No longer can we recommend WiRES-X networking for anything related to emergency services. WiRES-X IS NOT RECOMMENDED FOR ANY EMERGENCY SERVICES ORGANIZATION. YSF is a much better choice as you have total flexibility in how the networking is done and where the server is.

- One last disadvantage is that it's impossible to connect to 1.5.7 rooms. A number of rooms that I normally connect to have been inaccessible since running 2.0.1. It is inconceivable to me why Yaesu chose to make 2.0.1 incompatible with prior versions of WiRES-X!

It is for these reasons that FusionTech recommends AGAINST updating to WiRES-X 2.x. Stay tuned to this website or the FusionTech net for any updates.