Institute for the Wireless Internet of Things

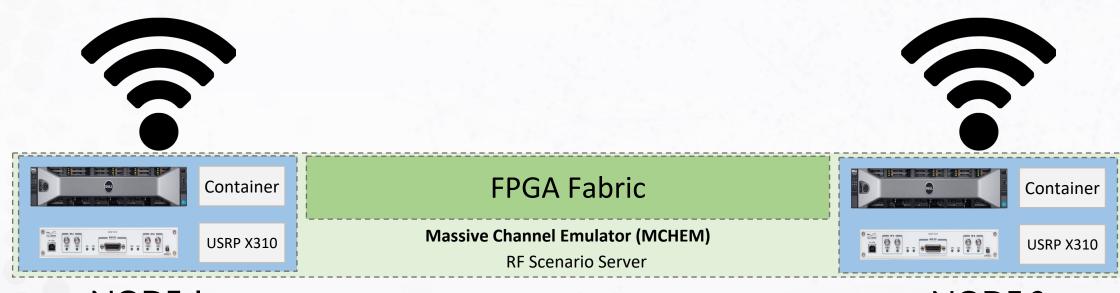
at Northeastern University

Wi-Fi on Colosseum

Ergest Beshaj and Ravis Shirkhani

Our goal

- Ad hoc Wi-Fi on Software Defined Radios in Colosseum
- Do experiments by generating traffic over the emulated channel



NODE 2



Let's start!

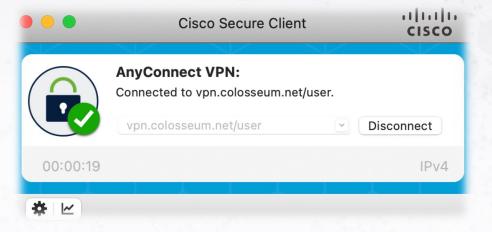
• https://github.com/colosseum-wiot/colosseum-school-2025/tree/main/wifi-assignment





Getting ready

Connecting to the VPN



• Starting a reservation with 2 nodes using webinar-interactive-v1





Getting ready

 Connect to the two SRNs when the status becomes "allocated" (root/sunflower)

ID: 145327

Status: Future

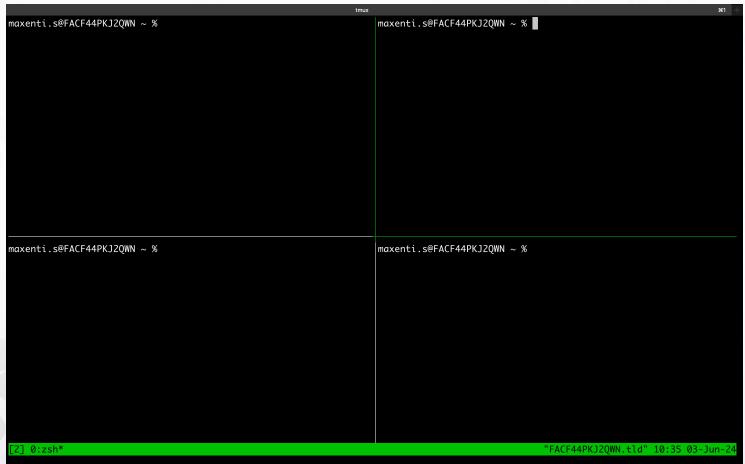
Nodes: 2

Time: 2024/06/03 10:39:00 AM - 2024/06/03 11:39:00 AM (60 minutes)

Node	Image	Hostname	Port	Status	
SRN -98	webinar-interactive-v1	wineslab-098	-	-	
SRN -99	webinar-interactive-v1	wineslab-099	-		Institute for the Wireless Internet of Things at Northeastern

Getting ready

We will need 4 terminals, use the layout that you prefer





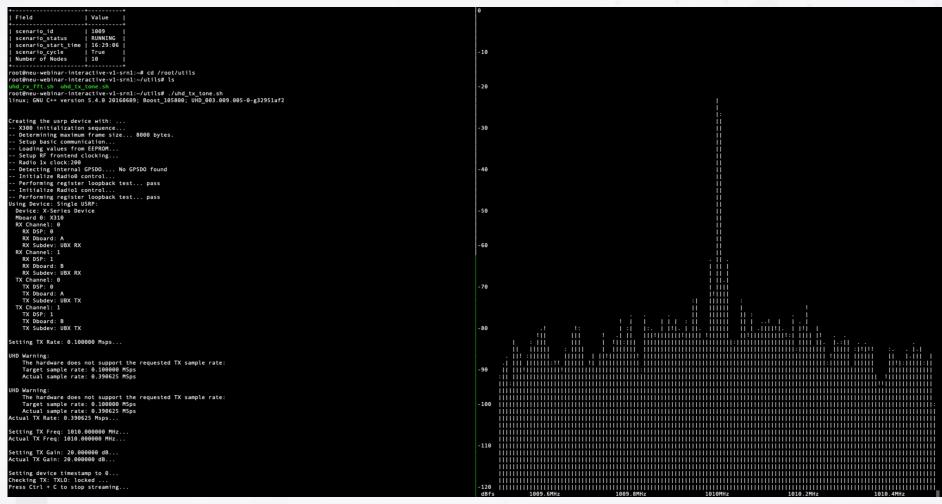
Testing if Colosseum is working

- In any of the SRNs, run:
 - colosseumcli rf start 1009 -c
 - Check if the scenario is active with colosseumcli rf info

- In one terminal, run: /root/utils/uhd_tx_tone.sh
- In the other, run: /root/utils/uhd_rx_fft.sh
- In uhd_tx_tone.sh, try to change the center frequency and see what happens



What you should see



SENDER



Getting ready: almost there

- Setting up routes to connect the two SRNs
 - /root/interactive_scripts/tap_setup.sh
 - In each SRNs, run:
 - /root/interactive_scripts/route_setup.sh <ID of the other SRN>
 - This sets up routes at L3 between the two SRNs and allows traffic routing
 - On the TAP interface, each SRN will have IP like this: 192.168.\$(ID+100).1.



Finally, let's transmit!

- On each SRN, run
 - /root/interactive_scripts/modem_start.sh
- This starts the transmission between the two nodes
- You should see some small traffic flowing between the nodes
- Connectivity can be checked with the ping command, or with iperf3



Let's create a lot of traffic

- We will be using the other two terminals you created
- We use TGEN
 - colosseumcli tg start 10090
 - colosseumcli tg info to check if it works
 - It takes around ~5min to start
- In the meanwhile, start on each node a traffic sniffer:
 - tcpdump —i tr0



What you should see

. W. [q. X. z. o.A. K. s. Je BCAT PP 9.T. Y. R. jG.3u. X. yGh L.) . * 2& NE.U *tt8	. @. JC.u. \$.0; 8. 8. xR]q. e. h. g.5. 3. %mi. p) .u. KP F. — \ 17b\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
new mac frame (length 1460)	new mac frame (length 1460)
duration: 00 00 frame control: 00 08 (DATA) Subtype: Data seq nr: 854 mac 1: asibb:cc:dd:ee:66 mac 2: asibb:cc:dd:ee:65 mac 3: 42:42:42:42:42:42:42 instantaneous fer: 0.978261 . E. G. W. D. e. f. S. C. E. E. Q. C. E. W. F. L. W. S. L. Y. V. De. V. De	N. ^ 1, R.O. [-7 m.8, v.G. C; Em.Q? -(*E "#.(uA*S])s. 2, y.F1. Tyag. a.b e.M
16:51:05. 815570 IP 192.168.101.2.4001 > 192.168.102.2.5001: UDP. Length 1400 16:51:05.827487 IP 192.168.101.2.4001 > 192.168.102.2.5001: UDP. Length 1400 16:51:05.80995 IP 192.168.101.2.4001 > 192.168.102.2.5001: UDP. Length 1400 16:51:05.80995 IP 192.168.101.2.4001 > 192.168.102.2.5001: UDP. Length 1400 16:51:05.80995 IP 192.168.101.2.4001 > 192.168.102.3.5001: UDP. Length 1400 16:51:05.80913 IP 192.168.101.2.4001 > 192.168.102.3.5001: UDP. Length 1400 16:51:05.80913 IP 192.168.101.2.4001 > 192.168.102.3.5001: UDP. Length 1400 16:51:05.80913 IP 192.168.101.2.4001 > 192.168.102.3.5001: UDP. Length 1400 16:51:05.80703 IP 192.168.101.2.4001 > 192.168.102.3.5001: UDP. Length 1400 16:51:05.90926 IP 192.168.101.2.4001 > 192.168.101.3.5010: UDP. Length 1400 16:51:05.90926 IP 192.168.101.2.4001 > 192.168.101.3.5010: UDP. Length 1400 16:51:05.90930 IP 192.168.101.2.4001 > 192.168.101.3.5010: UDP. Length 1400 16:51:06.80070 IP 192.168.101.2.4001 > 192.168.101.3.5010: UDP. Length 1400 16:51:06.80070 IP 192.168.101.2.4001 > 192.168.101.3.5010: UDP. Len	16:51:05.826721 IP 192.168.101.2.4001 > 192.168.102.3.5001 UDP, length 1400 16:51:05.837474 IP 192.168.103.4.010 > 192.168.102.3.5010 UDP, length 1400 16:51:05.837376 IP 192.168.103.4.010 > 192.168.103.3.5010 UDP, length 1400 16:51:05.844210 IP 192.168.103.4.001 > 192.168.102.3.5010 UDP, length 1400 16:51:05.844210 IP 192.168.103.4.001 > 192.168.102.3.5010 UDP, length 1400 16:51:05.861448 IP 192.168.103.4.001 > 192.168.102.3.5010 UDP, length 1400 16:51:05.86222 IP 192.168.103.4.001 > 192.168.102.3.5010 UDP, length 1400 16:51:05.86222 IP 192.168.103.4.001 > 192.168.103.3.5010 UDP, length 1400 16:51:05.875237 IP 192.168.103.4.001 > 192.168.103.3.5010 UDP, length 1400 16:51:05.875237 IP 192.168.103.4.001 > 192.168.103.3.5010 UDP, length 1400 16:51:05.887195 IP 192.168.103.4.001 > 192.168.103.3.5010 UDP, length 1400 16:51:05.897195 IP 192.168.103.4.001 > 192.168.103.3.5010 UDP, length 1400 16:51:05.899144 IP 192.168.103.4.001 > 192.168.103.3.5010 UDP, length 1400 16:51:05.899144 IP 192.168.103.4.001 > 192.168.103.3.5010 UDP, length 1400 16:51:05.99144 IP 192.168.103.4.001 > 192.168.103.3.5010 UDP, length 1400 16:51:05.99144 IP 192.168.103.3.4010 > 192.168.103.3.5010 UDP, length 1400 16:51:06.000774 IP 192.



Closing the experiment and ending the reservation

- ctrl-c on all terminals
- colosseumcli tg stop # to close the traffic generator
- colosseumcli rf stop # to close the channel emulator
- exit on all terminals
- DONE! You have transmitted over Wi-Fi on Colosseum



Thank you, any questions?