Batch Mode

- Radio applications and scenarios are controlled automatically by Colosseum

- Containers need to be preconfigured to use the Radio API which will allow Colosseum to control the radio applications

- Containers **do not** have access to the teams' network storage folders

- Containers **are not** accessible by SSH (except for the nodes with node_type set to "bot")

- Batch jobs are inserted in a queue and run when resources are available

Batch Mode

Set-up through configurations files:

- Batch configuration file:
  - Tells Colosseum how to run the experiment
  - Must be saved to the network storage on the File Proxy at /share/nas/teamname/batch/

- Modem configuration file(s):
  - Passes any additional parameters to the container
  - Parameters need to be handled by user code
  - Must be saved to the network storage on the File Proxy at /share/nas/teamname/config/

N Institute for the Wireless Internet of Things at Northeastern

Batch Configuration File

- Name of batch experiment

- Duration of batch experiment

- RF Scenario to run

- Traffic scenario to run

- Mapping of SRNs to nodes in the scenarios

```
1   {
2       "BatchName": "My Test Batch",
3       "Duration": 300,
4       "RFScenario": 6742,
5       "TrafficScenario": 1,
6       "NodeData": [
7           {
8               "RFNode_ID"       : 1,
9               "ImageName"       : "modem-image-v1",
10              "ModemConfig"     : "modem_config_file_1",
11              "isGateway"       : true,
12              "TrafficNode_ID"  : 1,
13              "node_type"       : "competitor"
14          },
15          {
16              "RFNode_ID"       : 2,
17              "ImageName"       : "modem-image-v1",
18              "ModemConfig"     : "modem_config_file_2",
19              "isGateway"       : false,
20              "TrafficNode_ID"  : 2,
21              "node_type"       : "competitor"
22          }
23      ]
24  }
```

Institute for the Wireless Internet of Things
at Northeastern

4

Batch Configuration File, cont'd

Mapping of SRNs to nodes in the scenarios:

- **RFNode_ID**: Node in the RF scenario the SRN should be mapped to

- **ImageName**: Container image to load on the SRN

- **ModemConfig**: The location of the modem config file to load

```
1   {
2       "BatchName": "My Test Batch",
3       "Duration": 300,
4       "RFScenario": 6742,
5       "TrafficScenario": 1,
6       "NodeData": [
7           {
8               "RFNode_ID"        : 1,
9               "ImageName"        : "modem-image-v1",
10              "ModemConfig"      : "modem_config_file_1",
11              "isGateway"        : true,
12              "TrafficNode_ID"   : 1,
13              "node_type"        : "competitor"
14          },
15          {
16              "RFNode_ID"        : 2,
17              "ImageName"        : "modem-image-v1",
18              "ModemConfig"      : "modem_config_file_2",
19              "isGateway"        : false,
20              "TrafficNode_ID"   : 2,
21              "node_type"        : "competitor"
22          }
23      ]
24  }
```

N Institute for the Wireless Internet of Things at Northeastern

Batch Configuration File, cont'd

Mapping of SRNs to nodes in the scenarios:

- **isGateway**: Determines if the collaboration gateway is connected to the node

- **TrafficNode_ID**: Node in the traffic scenario this SRN should be mapped to

```
1   {
2     "BatchName": "My Test Batch",
3     "Duration": 300,
4     "RFScenario": 6742,
5     "TrafficScenario": 1,
6     "NodeData": [
7       {
8         "RFNode_ID"        : 1,
9         "ImageName"        : "modem-image-v1",
10        "ModemConfig"      : "modem_config_file_1",
11        "isGateway"        : true,
12        "TrafficNode_ID"   : 1,
13        "node_type"        : "competitor"
14      },
15      {
16        "RFNode_ID"        : 2,
17        "ImageName"        : "modem-image-v1",
18        "ModemConfig"      : "modem_config_file_2",
19        "isGateway"        : false,
20        "TrafficNode_ID"   : 2,
21        "node_type"        : "competitor"
22      }
23    ]
24  }
```

Institute for the Wireless Internet of Things at Northeastern

Batch Configuration File, cont'd

Mapping of SRNs to nodes in the scenarios:

- **node_type**: The type of this node
  - **competitor**: used for standard batch jobs
  - **bot**: allows users to SSH into the node during a batch job, e.g., for debugging purposes.

```
1   {
2       "BatchName": "My Test Batch",
3       "Duration": 300,
4       "RFScenario": 6742,
5       "TrafficScenario": 1,
6       "NodeData": [
7           {
8               "RFNode_ID"        : 1,
9               "ImageName"        : "modem-image-v1",
10              "ModemConfig"      : "modem_config_file_1",
11              "isGateway"        : true,
12              "TrafficNode_ID"   : 1,
13              "node_type"        : "competitor"
14          },
15          {
16              "RFNode_ID"        : 2,
17              "ImageName"        : "modem-image-v1",
18              "ModemConfig"      : "modem_config_file_2",
19              "isGateway"        : false,
20              "TrafficNode_ID"   : 2,
21              "node_type"        : "competitor"
22          }
23      ]
24  }
```

Institute for the Wireless Internet of Things
at Northeastern

Batch Job Workflow

After users queue batch jobs and start executing:

- Containers are loaded on the SRNs

- The colosseum.ini file is copied into the container → contains info in the batch job

- If specified, the radio configuration is pre-loaded into the container

Institute for the Wireless Internet of Things
at Northeastern

Batch Job Workflow, cont'd

- Startup user scripts are executed (e.g., upstart of sysvinit scripts)

- After boot, the SRN controller periodically calls the status.sh Radio API script to check that the status of each container is **READY**

- 5 minutes after container boot, the SRN controller calls the start.sh Radio API script regardless of whether or not the container radio is in the **READY** state

**Institute for the Wireless
Internet of Things**
at Northeastern

Batch Job Workflow, cont'd

- Once start.sh has been called, RF and traffic scenarios start.

- The radios are now able to communicate through MCHEM.

- The containers needs to ensure that the status.sh script now returns **ACTIVE**

N **Institute for the Wireless**
Internet of Things
at Northeastern

Batch Job Workflow, cont'd

- When the scenario is completed, the stop.sh Radio API script is called

- The containers begin preparing for teardown

- Users may use this time to copy files to the /logs directory (saved on the NAS after the batch job ends)

- The container needs to ensure that the status.sh script now returns **STOPPING**

Batch Job Workflow, cont'd

- When experiment has ended, the radio container needs to ensure that the status.sh script now returns **FINISHED**

- Two minutes after stop.sh is called, the container deallocates, regardless of whether or not the container radio is in the **FINISHED** state

N **Institute for the Wireless Internet of Things**
at Northeastern

Batch Mode Timeline Example

Batch mode timeline example of 600 seconds:

- Minute 00:00 - 13:00 → batch job starts
  - The USRP if flashed
  - Containers are allocated/instantiated
  - Initial startup scripts are called

- Minute 13:00 → check containers for readiness
  - Did all the containers status.sh report a **READY** state?
  - Did the RF subsystem report **READY**? (Colosseum internal readiness check)
  - Did the traffic subsystem report **READY**? (Colosseum internal readiness check)

Batch Mode Timeline Example, cont'd

- Minute 13:00 - 16:00 → scenario preparation
  - 3 minutes for Colosseum scenario preparation

- Minute 16:00 → scenario starts
  - All containers call to start.sh

- Minute 16:00 - 26:00 → user experiment runs

**N Institute for the Wireless
Internet of Things**
at Northeastern

Batch Mode Timeline Example, cont'd

- Minute 26:00 → scenario stops
  - All containers call to stop.sh

- Minute 26:00 - 28:00 → container teardown
  - 2 minutes for user radio application cleanup (e.g., copying any data to the /logs directory)
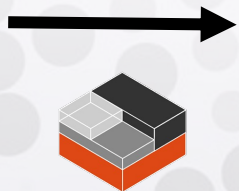
More info on batch mode jobs:

- https://tinyurl.com/5y4cens8

Institute for the Wireless Internet of Things at Northeastern

Validation Pipeline Example

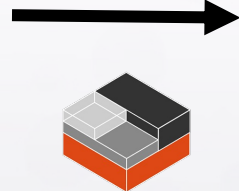**Port containers to different testbeds**

- **Prototype** on Colosseum

- **Validate in real environment**

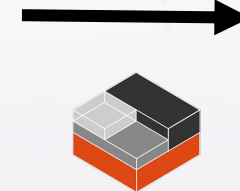- **Test** large-scale capabilities **on city-scale platforms**
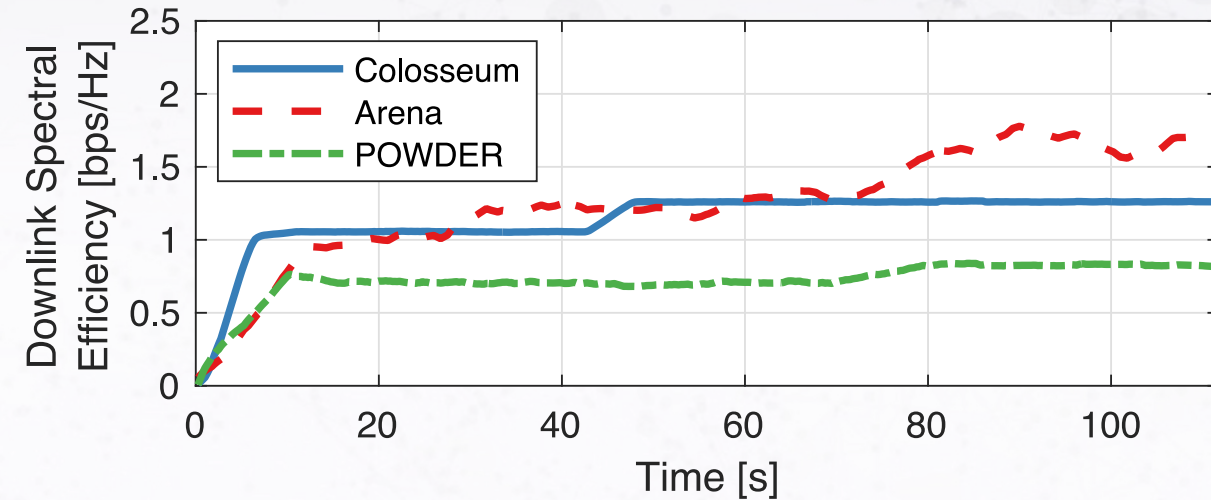
Test at-a-scale on emulated scenarios → COLOSSEUM at Northeastern University → Validate in real wireless environment → Test large-scale capabilities → Platforms for Advanced Wireless Research
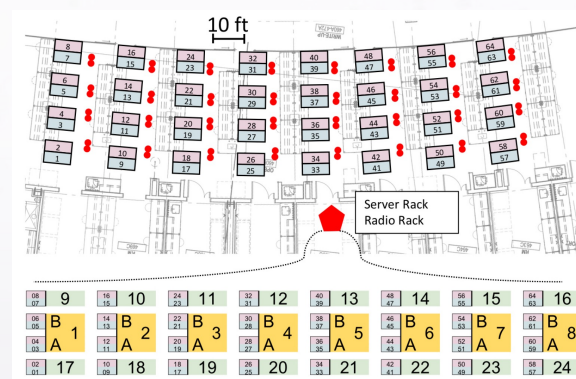
Institute for the Wireless Internet of Things
at Northeastern University

Thank You! (Questions?)