DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

800 ANNI 1222·2022

UNIVERSITÀ DEGLI STUDI DI PADOVA

W&M

SIGNET

signet.dei.unipd.it

Multimodal Networks - How to

UNWiS - Padova (Italy)

30th of January – 3rd of February 2023

Filippo Campagnaro, Roberto Francescon, Angela Soldà, Antonio Montanari, Michele Zorzi

filippo.campagnaro@unipd.it

Overview

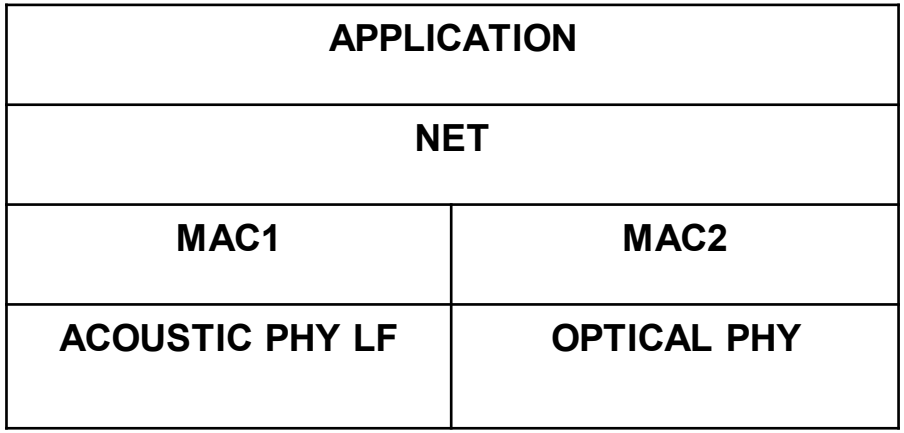
Install multimodal layers
sendUp and sendDown APIs
Tcl example

Install multimodal layers

- Multimodal layers are DESERT addons
- With the DESERT installation wizard, you need to write ALL when it asks if you need any addons
- Afterwards, all addons will be available

sendUp and sendDown

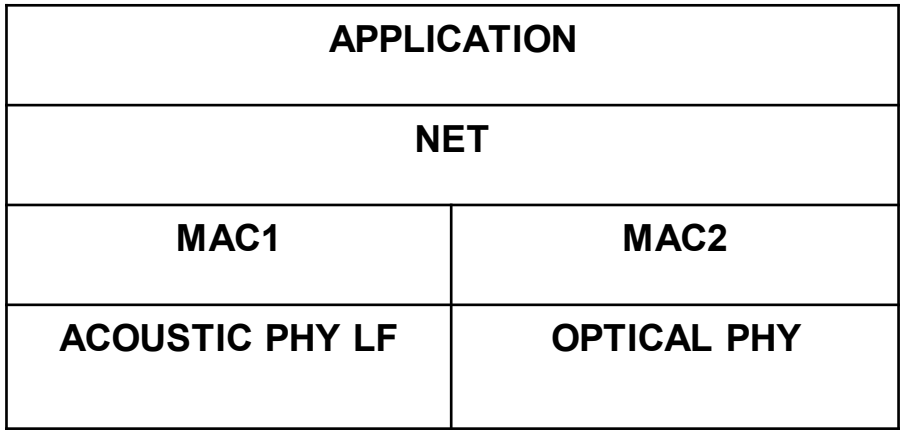
- **sendUp(Packet* p)** and **sendDown(Packet* p)** by default send the packet to either all adjacent upper layers or all adjacent lower layers, respectively, and so does the transmission of clmsgs. e.g.:



- NET sendsDown to both MAC1 and MAC3

sendUp and sendDown

- **sendUp(int id, Packet* p)**
- **sendDown(int id, Packet* p)**
- Send in unicast only to the upper or lower layer identified with id



Get the module id

There are two possibilities to get the module ID:

- Method Id_ from tcl

```
set id [$phy2 Id_]
```

- Get it from c++ discovering the other layers with crosslayer messages and looking for tags

```
$node($id) addModule 3 $phy($id)  1 "PHY1"
```

- "PHY1" is the tag, the crosslayer looks for this tag and gets the id

Get the module id from the TAG

```
int getLayerIdFromTag(const std::string& tag) {
   ClMsgDiscovery m;
   m.addSenderData((const PlugIn*) this, getLayer(), getId(),
            getStackId(), name() , getTag());
   sendSyncClMsgDown(&m);
   DiscoveryStorage low_layer_storage = m.findTag(tag.c_str());
   if (low_layer_storage.getSize() == 1) {
      DiscoveryData low_layer = (*low_layer_storage.begin()).second;
      return low_layer.getId();
   }
  return -1; // or whatever to say that it's not found
}
```

Multistack and multi traffic controller

APPLICATION		
MAC		
MULTI – STACK – CONTROLLER		
ACOUSTIC PHY LF	ACOUSTIC PHY HF	OPTICAL PHY

CTR	IMAGE	HEALTH	SOS
MULTI-TRAFFIC-CONTROLLER			
TDMA	CSMA	ALOHA	
ACOUSTIC PHY LF	ACOUSTIC PHY HF	OPTICAL PHY	

- Multistack controller between one MAC and multiple PHYs → uses Id_

Filippo Campagnaro, Federico Favaro, Federico Guerra, et al, "Simulation of Multimodal Optical and Acoustic Communications in Underwater Networks", Oceans 2015 Genova

- Multitraffic controller between multiple NET and MLL → uses TAGS

Filippo Campagnaro, Federico Guerra, Paolo Casari, et al, "Implementation of a Multi-modal Acoustic-Optical Underwater Network Protocol Stack", Oceans 2016 Shanghai

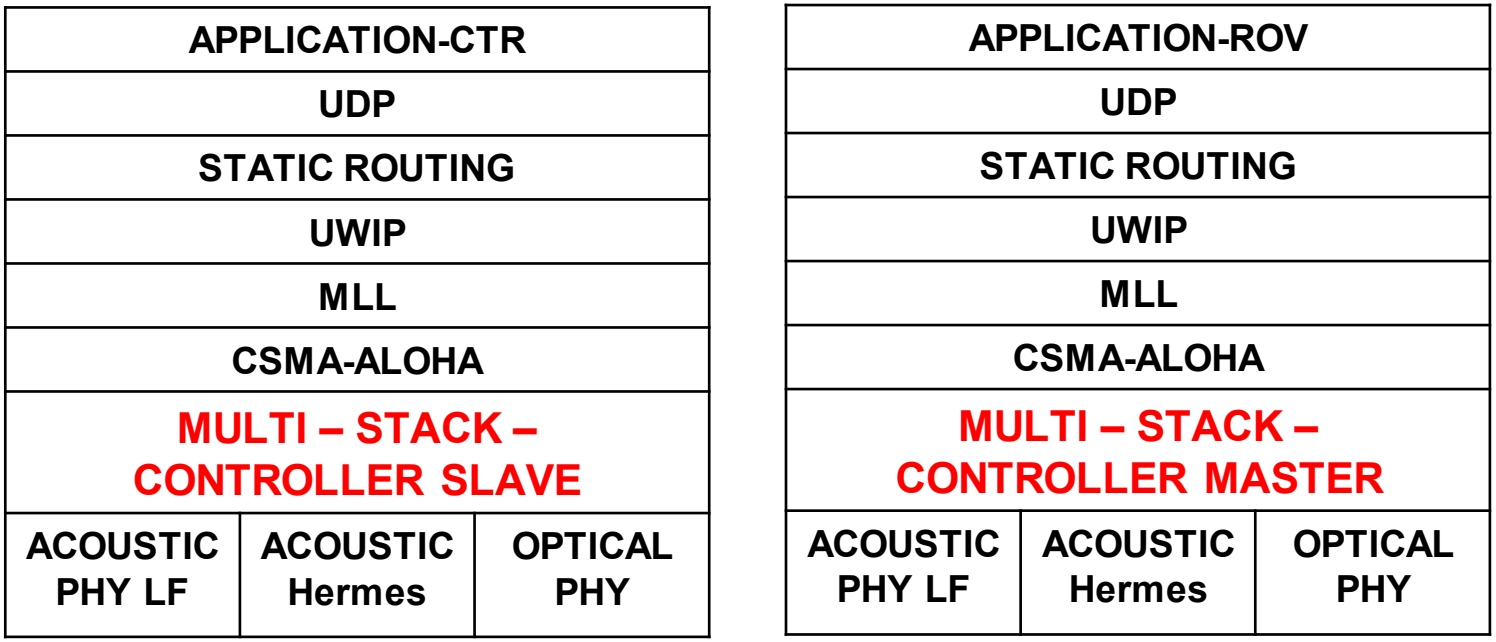
test_uwmultiphy_ctr.tcl

APPLICATION		
UDP		
STATIC ROUTING		
UWIP		
MLL		
CSMA-ALOHA		
MULTI – STACK – CONTROLLER SLAVE		
ACOUSTIC PHY LF	ACOUSTIC PHY HF	ACOUSTIC PHY HF

APPLICATION		
UDP		
STATIC ROUTING		
UWIP		
MLL		
CSMA-ALOHA		
MULTI – STACK – CONTROLLER MASTER		
ACOUSTIC PHY LF	ACOUSTIC PHY HF	ACOUSTIC PHY HF

- Master decides when to switch, based on thresholds of received power
- we set a precomputed threshold with hysteresis to avoid continuous switching

test_uwmultiphy_ctr_rov.tcl



- Uses optical as well
- Hermes is an HF acoustic modem (PER from LUT)
- Switch also applies signaling
- The application controls the position of an ROV

test_uwmultiphy_*.tcl

```
set ctr($id)  [new Module/UW/MULTI_STACK_CONTROLLER_PHY_MASTER]
…
$node($id) setConnection $mac($id)   $ctr($id)   1
$node($id) setConnection $ctr($id)   $phy($id)   1
$node($id) setConnection $ctr($id)   $phy2($id)  1
$node($id) setConnection $ctr($id)   $phy3($id)  1
…
$ctr($id) addLayer [$phy($id) Id_] 1
ctr($id) addLayer [$phy2($id) Id_] 2
$ctr($id) addLayer [$phy3($id) Id_] 3
```

addLayer <ID> <order>
order: 1 ←→ 2 ←→ 3

```
$ctr($id) addThreshold [$phy($id) Id_] [$phy2($id) Id_] 1.72434e11; #1498m
$ctr($id) addThreshold [$phy2($id) Id_] [$phy($id) Id_] 2.35975e10; #1499 m
$ctr($id) addThreshold [$phy2($id) Id_] [$phy3($id) Id_] 3.93457e11;#998 m
$ctr($id) addThreshold [$phy3($id) Id_] [$phy2($id) Id_] 1.3956e+11; #997m
```

Threshold with hysteresis computed with Urick/Thorp formulas

Exercises

1) Check the examples inside Channel/Optical and get familiar with the optical model
2) Check the examples inside test_uwmultiphy, focusing on test_uwmultiphy_ctr.tcl and test_uwmultiphy_ctr_rov.tcl
3) Create your multimodal tcl example with 2 nodes (one master and one slave), uwcsma-aloha, uwcbr, and at least 2 physical layers
   - You can decide if two acoustic or one acoustic and one optical layers
   - Find the correct threshold. Suggestion: either use the matlab scripts with beer law and Urick formulas or just create a tcl with two nodes, one tx and one rx and find the threshold.

Find the threshold with DESERT

- Create two nodes, one transmitting and one receiving
- Set proper acoustic parameters (freq, bw, tx power, …)
- Set the nodes position (x,y,z)
- Find the position of the out of range
- Enable debug
- Check the log

UnderwaterPhysical(0)::startRx() snr_dB = 54.1674; AcquisitionThreshold_dB_ = 4 **pr 118.193** pn 64.0254 end 99957.6 src 3 dest 1 size 129

- pr is the received power, but in dB (we need linear scale)
- In uwphysical this printout is computed as 10 * log10(ph->Pr), please, revert

Find the threshold with DESERT

- In addition, a ns2miracle message prints the received power in linear scale

Pt=1.00000e+18 SMG=1.00000 TAG=1.00000 RAG=1.00000 PG=9.42268e-06 CG=1.00000 Pr=9.42268e+12 duration=0.0955556

- Works also for the optical phy (the vales will be significantly smaller as we are talking about Watt vs microPascal)