Integrate PHR source code into NS2

Awos Kh. Ali
a.k.ali@uomosul.edu.iq

This document provides a tutorial on how to use phr *source code in NS2. You need to download and install NS2 first, then integrate and compile* phr *code.* PHR *directory contains list of files, including* phr.cc,phr.h,phr_rqueue.cc*,* phr_rqueue.h *and* phr_packet.h*. In order to integrate and compile* PHR *protocol in* NS2*, the following steps should fulfilled.*

Listing 1: Packet header file changes

```
static const packet_t PT_PHR = 73;
static packet_t PT_NTYPE = 74; // This MUST be the LAST one
```

Listing 2: Make packet has high priority

```
type == PT_PHR ||
type == PT_MDART)
```

- It should install a fresh copy of *NS2.35*.
- Download and copy *PHR* directory into */ns-allinone-2.35/ns-2.35/*.
- Add *case PT_PHR* to */ns-allinone-2.35/ns-2.35/queue/priqueue.cc* from line 94.
- *PHR* packet header need to be defined, */ns-allinone-2.35/ns-2.35/common/packet.h* file should modified accordingly by adding *#define HDR_PHR(p) (hdr_phr::access(p))* after line 62.
- Modifying same file, PT_NTYPE should change to 74, and for *PHR* protocol PT_PHR = 73. If you have already installed another routing protocol. Just make sure PT_NTYPE is last, and protocol number is ordered sequentially. Code in 2 shows the changes to *packet.h*.
- Add the *type == PT_PHR* as shown in 2 at line 280 of the same *packet.h*. Then add *name_[PT_PHR]=”PHR”* in line 420.
- In order to provide a trace functionality into the simulation, it should enable *NS2* to trace all the events in the simulation, to do that, */ns-allinone-2.35/ns-2.35/trace/cmu-trace.h & cmu-trace.cc* files need to be modified.
- First, define drop reasons by adding lines in 3 into *cmu-trace.h* at line85.
- Define trace function in *cmu-trace.h* at line 165 as shown in 4.

Listing 3: Define drop reasons

```
#define DROP_PHR_PH_CLOSER          "CLOSER"//PH is closer to D
#define DROP_PHR_DKNOW              "DKNOW"// know flag set and don't
    know about D.
```

Listing 4: Define trace function

```
void   format_phr(Packet *p, int offset);
```

- The implementation of the trace function should be added in *cmu-trace.cc* at line 1182 as shown in 5.

Listing 5: Main body of PHR trace function.

```
#include <phr/phr_packet.h> //PHR protocol
// main body of the trace function.\
void
CMUTrace::format_phr(Packet * p, int offset)
   {struct hdr_phr *phr = HDR_PHR(p);
   struct hdr_phr_bc *bc = HDR_PHR_BC(p);
   switch (phr->pkt_type) {
   case PHR_BC:
      if (pt_->tagged())
         {sprintf(pt_->buffer() + offset,
            "-PHR:t %x -PHR:h %d -PHR:b %d -PHR:s %d "
            "-PHR:ts %f "
            "-PHR:c PHR ",
            bc->bc_type,
            bc->bc_hop_count,
            bc->bc_bcast_id,
            bc->bc_src,
            bc->bc_timestamp);
      } else if (newtrace_)
         {sprintf(pt_->buffer() + offset,
            "-P phr -Pt 0x%x -Ph %d -Pb %d -Ps %d -Pts %f -Pc PHR ",
            bc->bc_type,
            bc->bc_hop_count,
            bc->bc_bcast_id,
            bc->bc_src,
            bc->bc_timestamp);
      } else {sprintf(pt_->buffer() + offset,
            "[0x%x %d %d [%d] [%f]] (PHR)",
            bc->bc_type,
            bc->bc_hop_count,
            bc->bc_bcast_id,
            bc->bc_src,
            bc->bc_timestamp);
      }
      break;
   default:
#ifdef WIN32
      fprintf(stderr,
         "CMUTrace::format_phr: invalid PHR packet typen");
#else
      fprintf(stderr,
```

Listing 6: Set PHR agent

```
PHR {
  set ragent [$self create-phr-agent $node]
}
\item At line $870$ code in \~\ref{tcl} should be added.
\begin{lstlisting}[caption= Create PHR agent \label{tcl}, float]
Simulator instproc create-phr-agent {node}
    {set ragent [new Agent/PHR [$node node-addr]]
   $self at 0.0 "$ragent start"
   $node set ragent_ $ragent
   return $ragent
}
```

Listing 7: Set ports of PHR agent

```
Agent/PHR instproc init args
    {$self next $args
}
Agent/PHR set sport_ 0
Agent/PHR set dport_ 0
```

```
        "%s: invalid PHR packet typen", __FUNCTION__);
#endif
      abort();
   }
}
```

- After changing C++ files, TCL files also need to be changed to create *PHR* routing agent to be used in TCL file. This is done by modifying */ns-allinone-2.35/ns-2.35/tcl/lib/ns-packet.tcl.*
- Add *PHR* at line 172
- Set routing agent by modifying */ns-allinone-2.35/ns-2.35/tcl/lib/ns-lib.tcl* at line 639 as shown in 6.

- Set port numbers of *PHR* agent (sport is the source port, dport is destination port) by adding code in 7 to */ns-allinone-2.35/ns-2.35/tcl/lib/ns-agent.tcl* at line 201.
- Modify */ns-allinone-2.35/ns-2.35/tcl/lib/ns-mobilenode.tcl* by adding code in 8 at line 204.

Listing 8: Set ports of PHR agent

```
# Special processing for PHR
set phronly [string first "PHR" [$agent info class]]
if {$phronly!= -1}
   {$agent if-queue [$self set ifq_(0)] ;# ifq between LL and MAC
}
```

- Modify */ns-allinone-2.35/ns-2.35/Makefile* by adding *phrphr.o* after *pumapuma.o* line to the list of object files for *NS2*.

Now, *NS2* should be ready to be recompiled. To do so, run make command in */ns-allinone-2.35/ns-2.35/*. When the compilation is done, *NS2* ready to be tested with *PHR*.