

Cross Layer Access (In NS2)

Till now I know 3 ways:

1. If you only need to get several values, a simple way is to use *tcl.eval(char* s)*.

1) Write a command that you will call in your sensor-agent.cc (can be others such as aadv.cc)

```
.....
if (0 == strcasecmp(argv[1], "access-sinkTable")) {
    char s[210];
    sprintf(s, "%d,%f,%d,%d,%d,%d,%c", property_.sinkTable.lID, property_.sinkTable.lIDistSum,
property_.sinkTable.lIDHopnum, property_.sinkTable.lhHopnum, property_.sinkTable.seqnum,
property_.sinkTable.lIDNext, property_.sinkTable.isValid ? 'T' : 'F');
    tcl.result(s);
    return TCL_OK;
}
.....
```

2) Then you can call it from the Mac layer and get these values

```
.....
int id = (u_int32_t)addr();
Tcl&tcl = Tcl::instance();
tcl.evalf("$sensoragent_(%d) access-sinkTable", id);
```

```
const char* sp = tcl.result();
char a[7][30];
int k = -1;
for(int i = 0; i < 7; i++) {
    k++;
    int j = 0;
    while(sp[k] != ',') {
        a[i][j] = sp[k];
        j++;
        k++;
    }
    a[i][j] = '\n';
}
```

.....

or

```

const char* sp = tcl.result();

int i=0;
char *tokenPtr, *a[7];
char stl_current[strlen(sp)+1];
strcpy(stl_current, sp);
tokenPtr = strtok(stl_current, ",");
while(tokenPtr != NULL) {
    a[i++] = tokenPtr;
    tokenPtr = strtok(NULL, ",");
}

```

If you require more features, you may need to acquire the pointer of that layer.

2. If the two layers you want to cross-access are adjacent, you can use `uptarget_/downtarget_`. The structure is shown in Chapter 16 of ns manual.
3. If they are not, it might be a little difficult to make the pointer clear by using several `uptarget_`. A method is to use tcl to get the pointer. You must have seen the usage of `TclObject::lookup()` many times in ns codes.

As an example, in my case, I need my SensorAgent to access Mac802_11, thus

in my sensor-agent.cc (can be others such as aodv.cc)

implement a command

```

if (strcmp(argv[1], "access-mac") == 0) {
    mac_pt_ = (Mac802_11*)TclObject::lookup(argv[2]);
    if (mac_pt_ == 0)
        return TCL_ERROR;
    return TCL_OK;
}

```

here `mac_pt_` is defined in sensor-agent.h

```
Mac802_11 *mac_pt_
```

you may also need to declare Mac802_11 as a friend class.

After done this, once you execute the following command in your tcl file, `mac_pt_` will get the right pointer and then you can use it.

```
$sensoragent_($i) access-mac [$node_($i) set mac_(0)]
```

vice versa

```
set mac_ [$node_($i) set mac_(0)]
```

```
$mac_ access-agent [$node_($i) agent 255]
```

here, you need to implement an "access-agent" command in the mac layer.

So the point is that you know how to get the expression of the layer you want to access.

some examples from www.baisi.net:

```
set ragent [$node_($i) agent 255]
```

```
$ragent if-queue [$node_($i) set ifq_(0)]
```

```
aodv_ tora ——> agent 255
```

```
dsr ——> dsr_agent_($i)
```

```
ll ——> ll_($i)
```

```
interface ——> netif_($i)
```

```
interface queue ——> ifq_($i)
```

```
mac ——> mac_($i)
```

```
arp ——> arptable_($i)
```

use 0 instead of \$i if single channel

[RETURN](#)