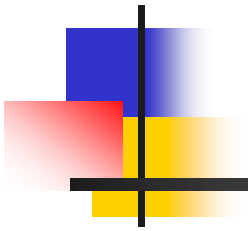
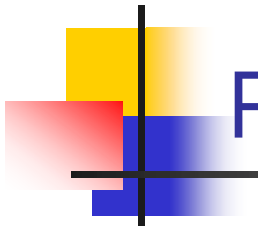


ENSC 835: OPNET Tutorial

Modupe Omueti and Renju Narayanan

Communication Networks Laboratory
http://www.ensc.sfu.ca/research/cnl
School of Engineering Science
Simon Fraser University



Roadmap

- OPNET Modeler
- Settings
- Creating projects
- Creating links
- Node models
- Packet format
- ICI format
- Process model
- Kernel procedures
- Compiling and debugging
- Collecting results

OPNET modeler

- Editors:
  - Project Editor
  - Node Editor
  - Process Editor
  - Link Editor
  - Packet Editor

Settings

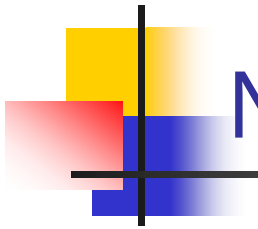
- Model directories
- Edit-> Preferences:
  - bind_shobj_prog: bind_so_gcc
  - bind_static_prog: bind_gcc
  - comp_prog: comp_gcc
  - repositories: ()

Creating projects

- Network models: scenarios
- Choosing the size of the network
  - world
  - campus
  - office
  - logical
- Nodes in the network
- Creating object palette
- Trajectories
- Managing scenarios

Creating links

- Links
  - create links using link editor
  - example: gprs_llc_link
- Type of link:
  - point-to-point:
    - simplex – ptsimp
    - duplex- ptdup
  - bus
- Packet formats supported
- Transmission delay model (txdel):
  - point-to-point link: dpt_txdel
  - bus: dbu_txdel
- Propagation model
- Error model

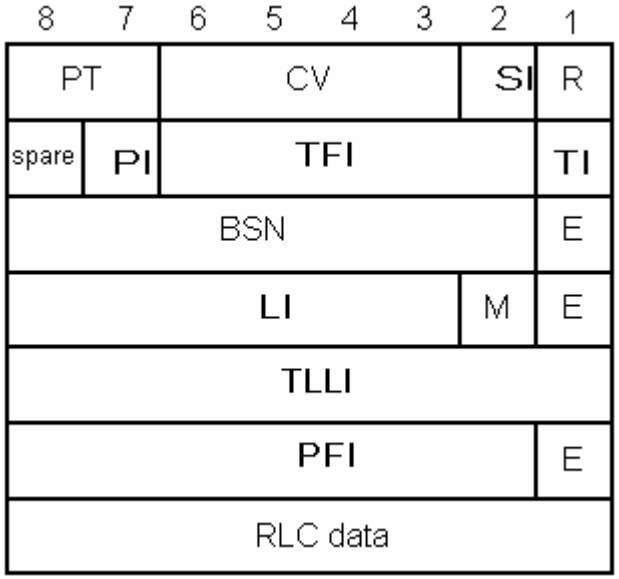


Node models

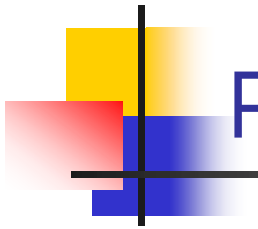
- Create your own: e.g., GPRS
- Modify an existing model: e.g., MTCP
- Various modules:
  - processors
  - queues: active, passive
    - first-in-first-out
    - priority
    - last-in-first-out
  - transmitters, receivers, antenna
  - packet stream
  - statistic wires

Packet format

- Packet editor
- KP: op_pk_create_fmt()
- Fields: length could be zero
- Set and unset fields inside code



Uplink RLC data block



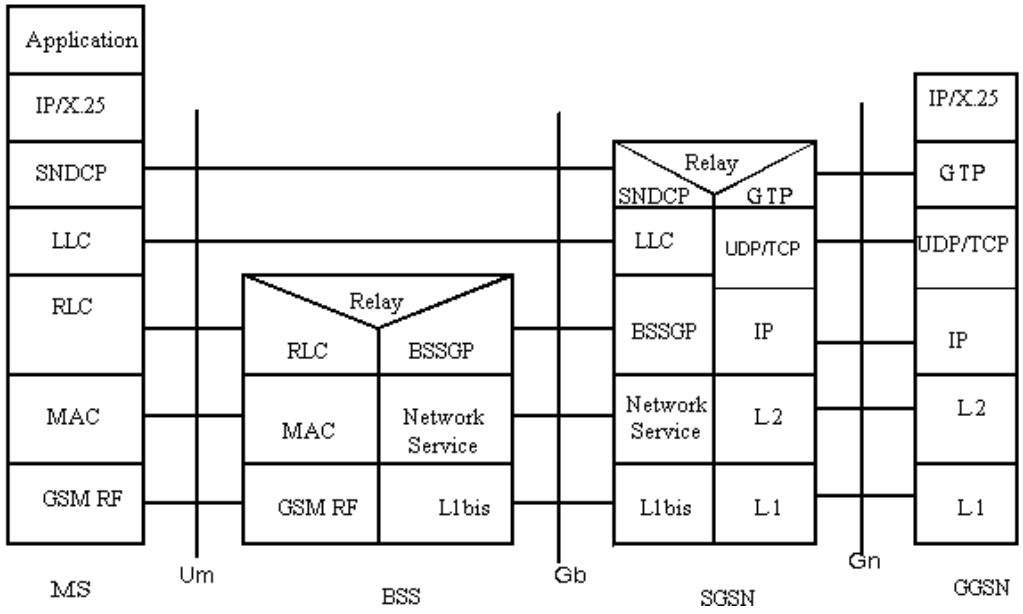
Process model

- States
- Forced and unforced states
- Transitions
- Enter and exit executives
- State variables
- Temporary variables
- Header block
- Function block
- Include files (.h)

General Packet Radio Service (GPRS)

- Packet switched service
- GPRS uses a combination of Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) schemes
- Direction of data transfer:
  - Mobile Station (MS) to Base Station Subsystem (BSS): uplink
  - BSS to MS: downlink

GPRS: transmission plane



SNDCP: Sub Network Dependent Convergence Protocol
LLC: Logical Link Control layer
RLC: Radio Link Control
MAC: Medium Access Control
BSSGP: Base Station Subsystem GPRS Protocol
GTP: GPRS Tunneling Protocol

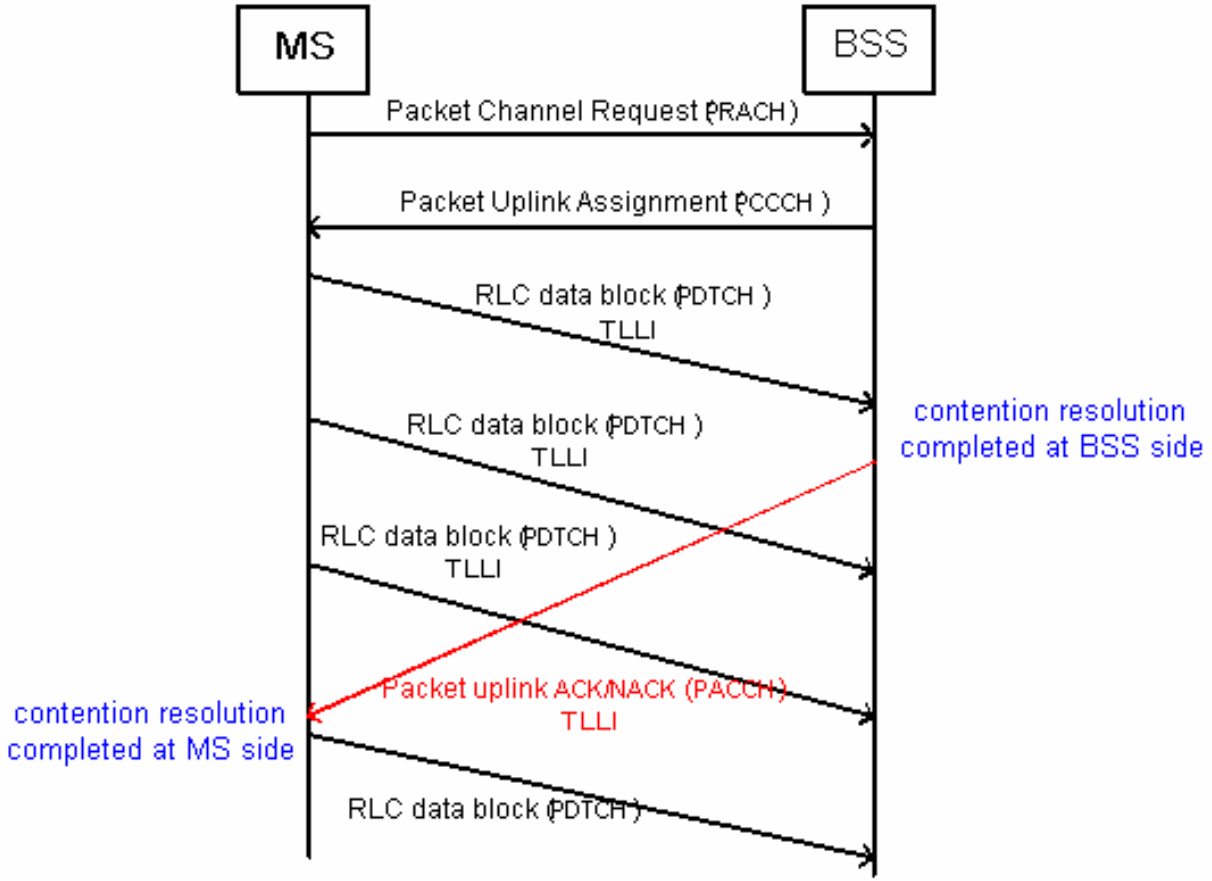
GPRS: RLC/MAC layer

- Radio Link Control layer:
  - segments and reassembles LLC PDUs into RLC/MAC blocks
  - acknowledged operation
  - unacknowledged operation
- Medium Access Control layer:
  - controls the allocation of channels and timeslots
  - multiplexes data and control signals
  - provides contention resolution

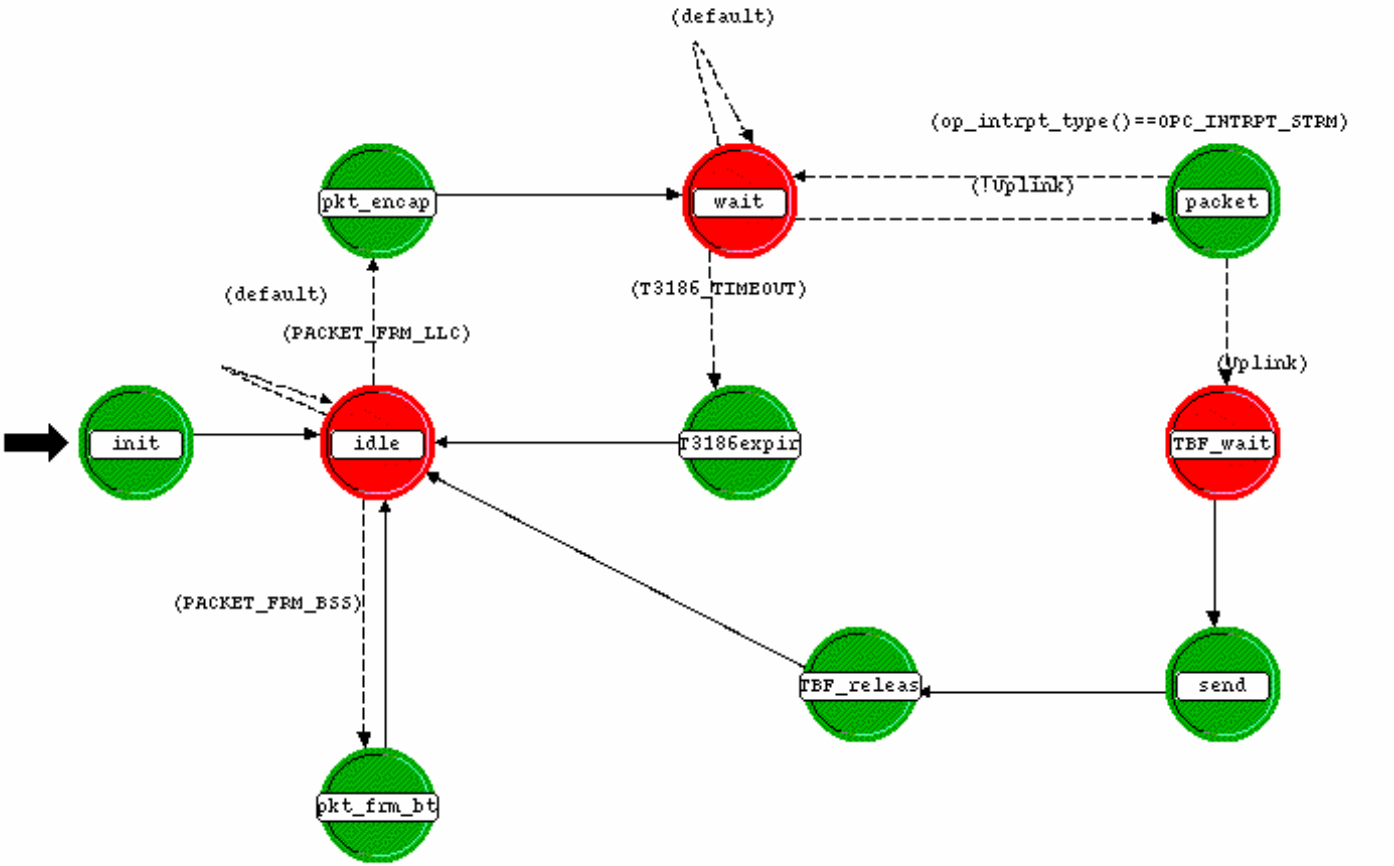
RLC/MAC parameters

- Temporary Block Flow (TBF): physical connection used by two radio resource entities to support unidirectional data transfer on physical channels
  - downlink and uplink TBF
  - temporary
  - maintained for the duration of data transfer only
- Network assigns a Temporary Flow Identity (TFI) to each TBF
  - TFI is unique among TBFs in the same direction

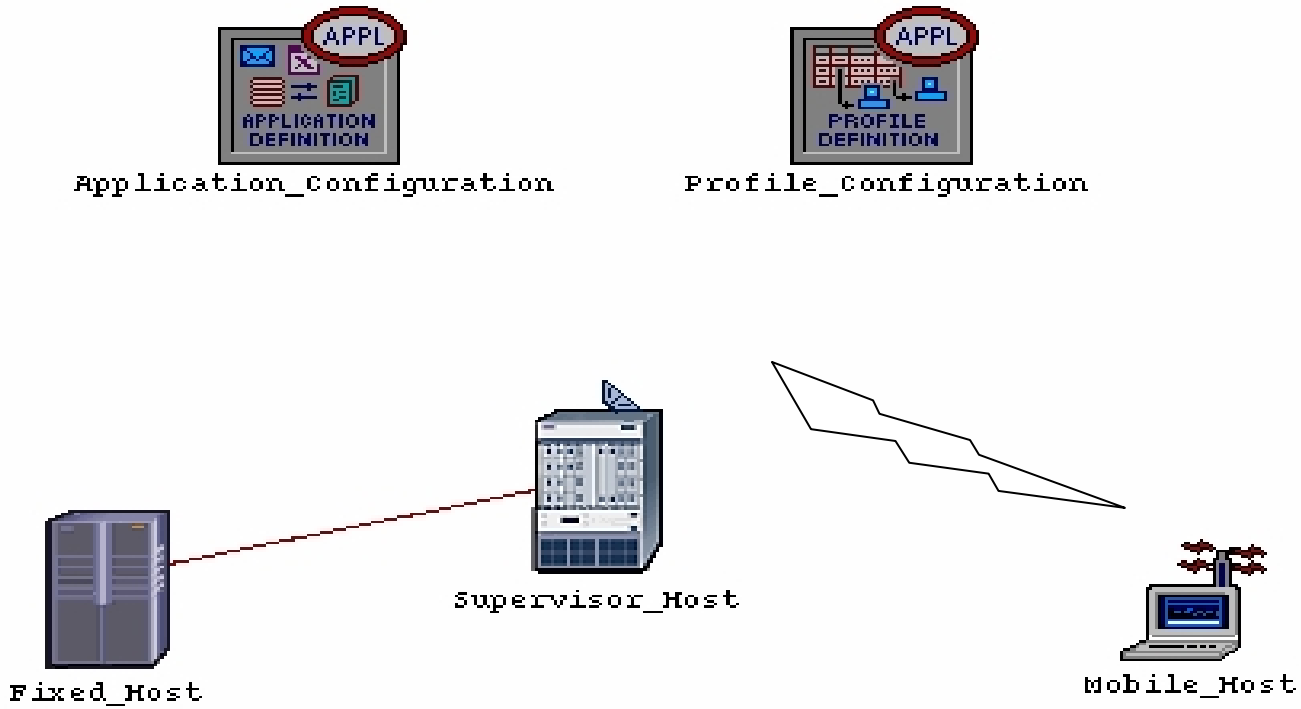
One phase access and contention resolution

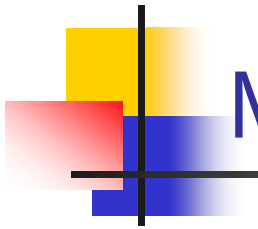


Process model: RLC/MAC (MS)



M-TCP


APPL
APPLICATION DEFINITION
Application_Configuration
APPL
PROFILE DEFINITION
Profile_Configuration
Supervisor_Host
Fixed_Host
Mobile_Host

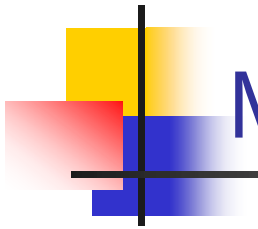


M-TCP design considerations

- Dynamic change of bandwidth in cells
- Frequent periods of disconnection
- Scarce power resources

M-TCP protocol characteristics

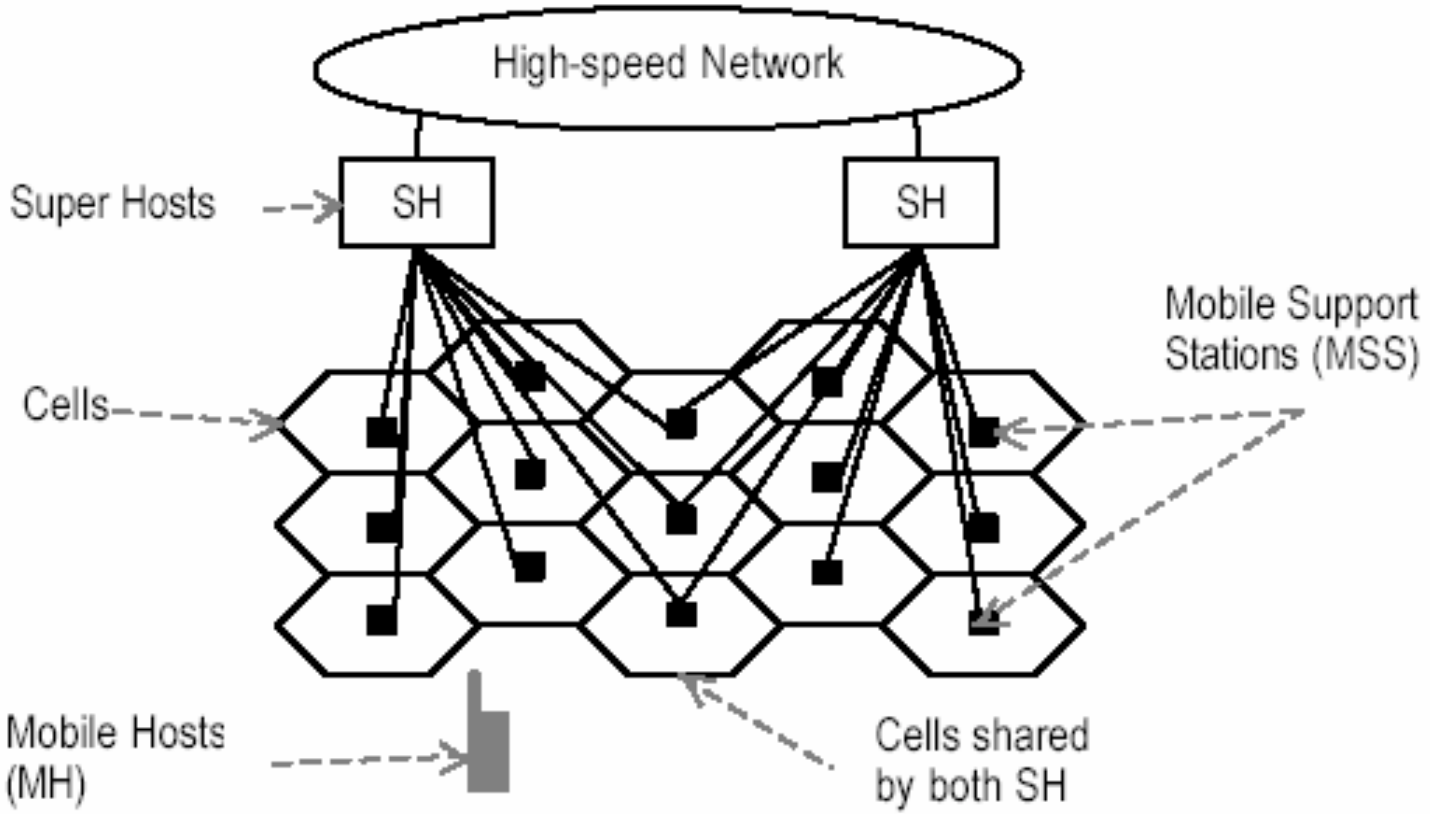
- Maintain end-to-end semantics of TCP
- Adapt to dynamic bandwidth changes
- Deal with disconnections
- Ensure efficient handoffs



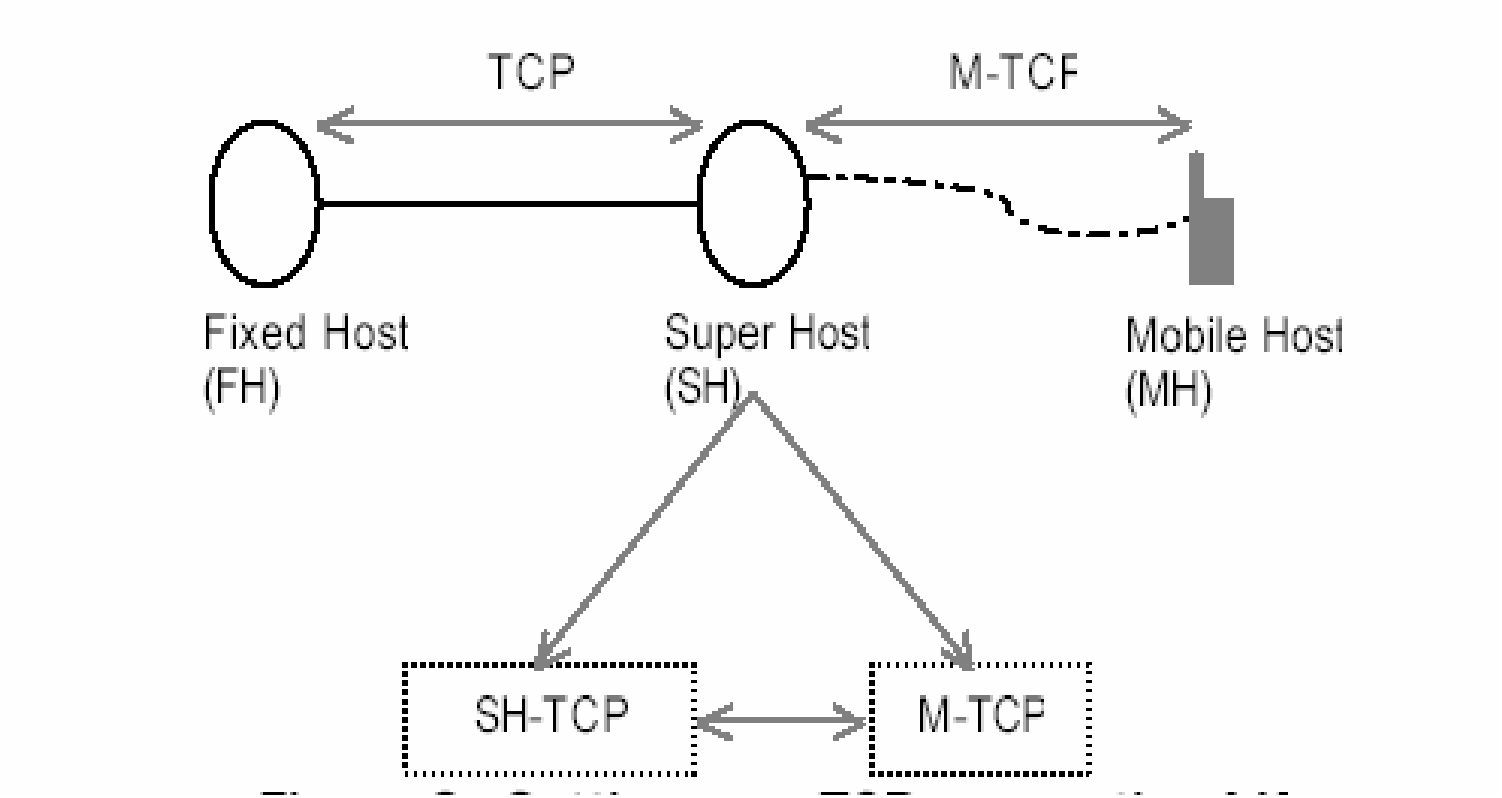
M-TCP mode of operation

- Freezes all timers when disconnections occur
- Monitors wireless link connectivity
- Puts sender (fixed host) into persist mode
- Sets receiver's (mobile host) *cwnd* to zero
- Resumes with previous sender's *cwnd*

Mobile network architecture for M-TCP



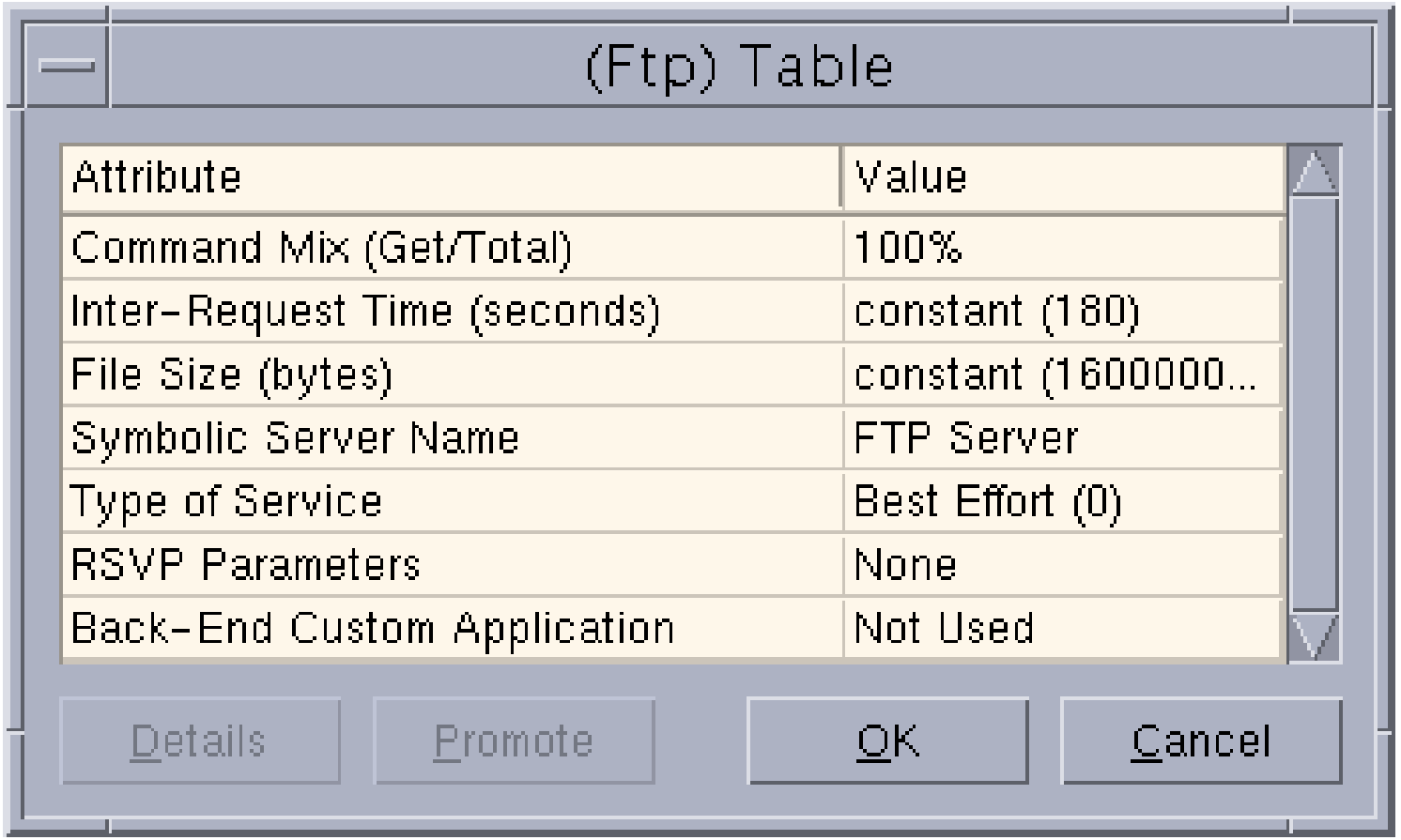
M-TCP split connection



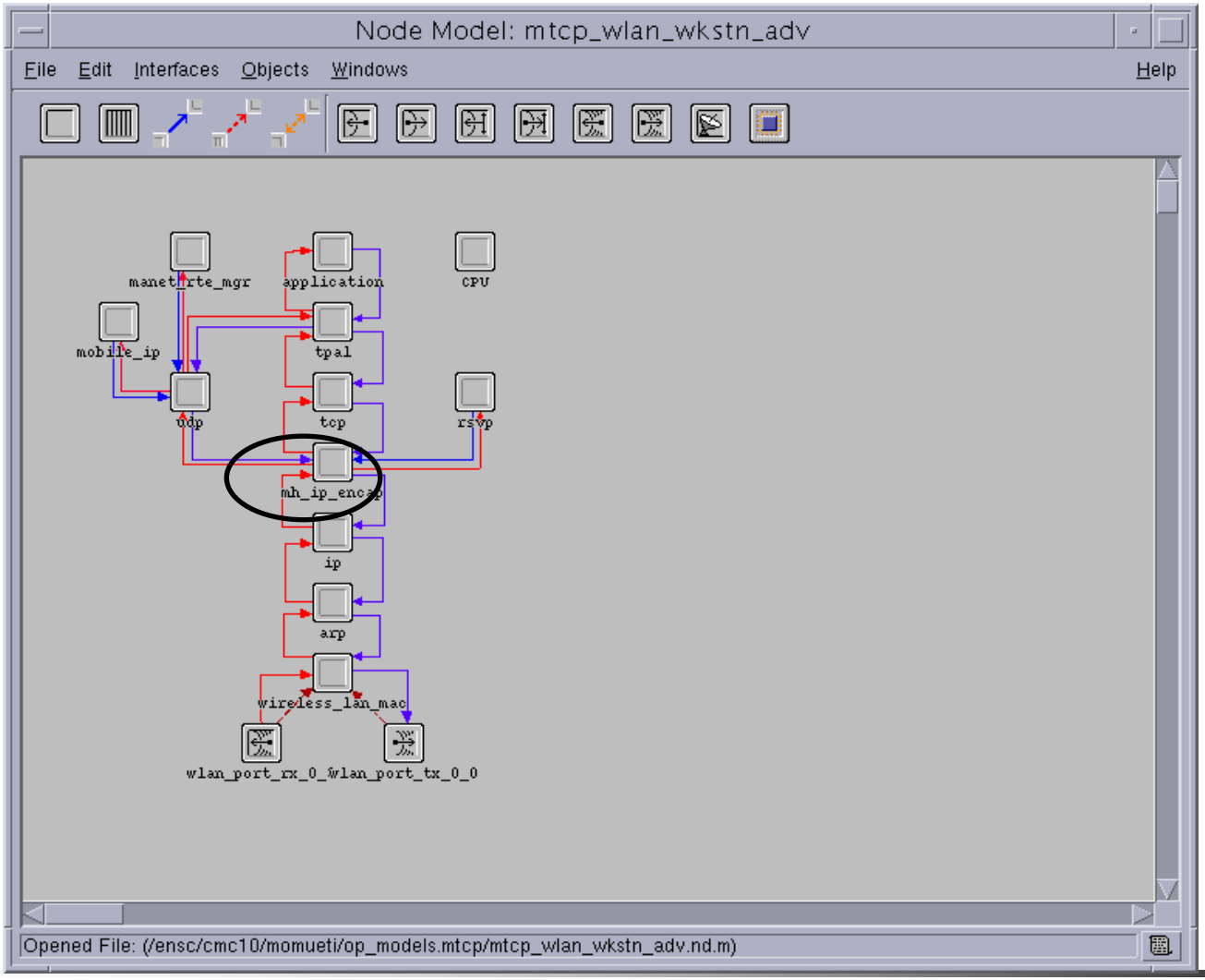
M-TCP parameters

- File size
- Inter-request time and distribution
- Sender Maximum Segment Size (SMSS)
- Note:
  - If the file size is small and auto assigned is selected for the SMSS, then the congestion window would be constant.
  - The explanation for this behavior is that the file is downloaded in so short a time that the congestion window did not need to increase in size.

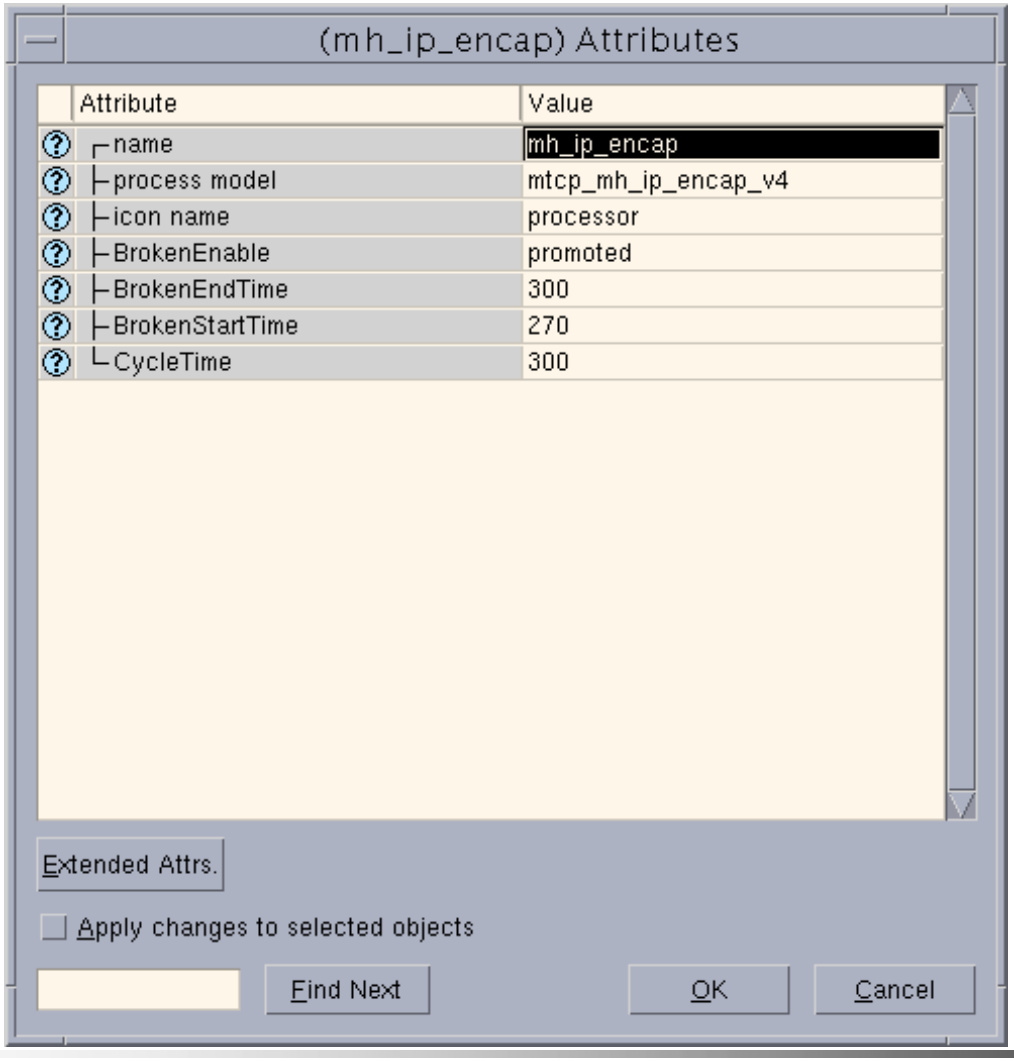
Application definition



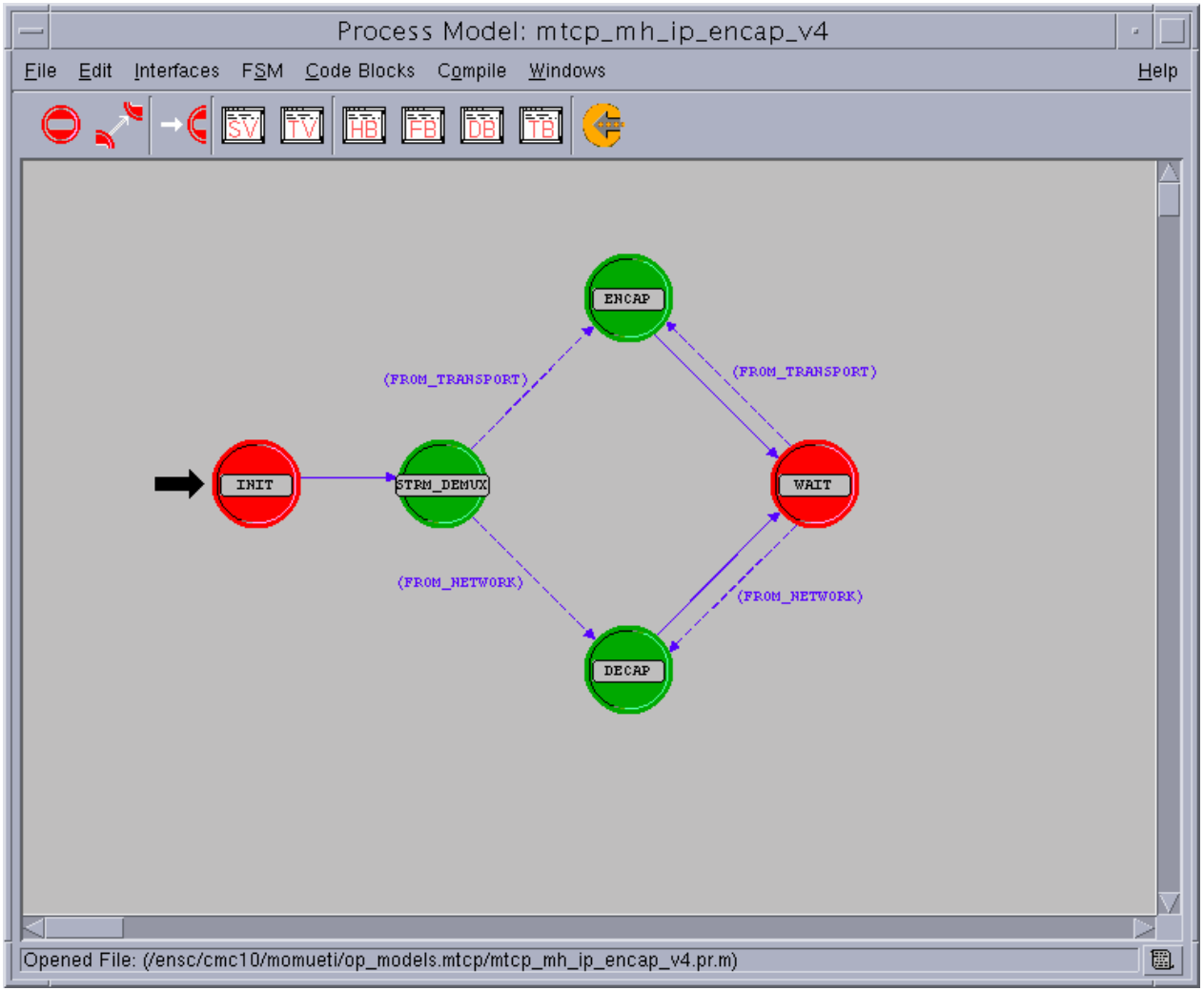
Mobile host node model



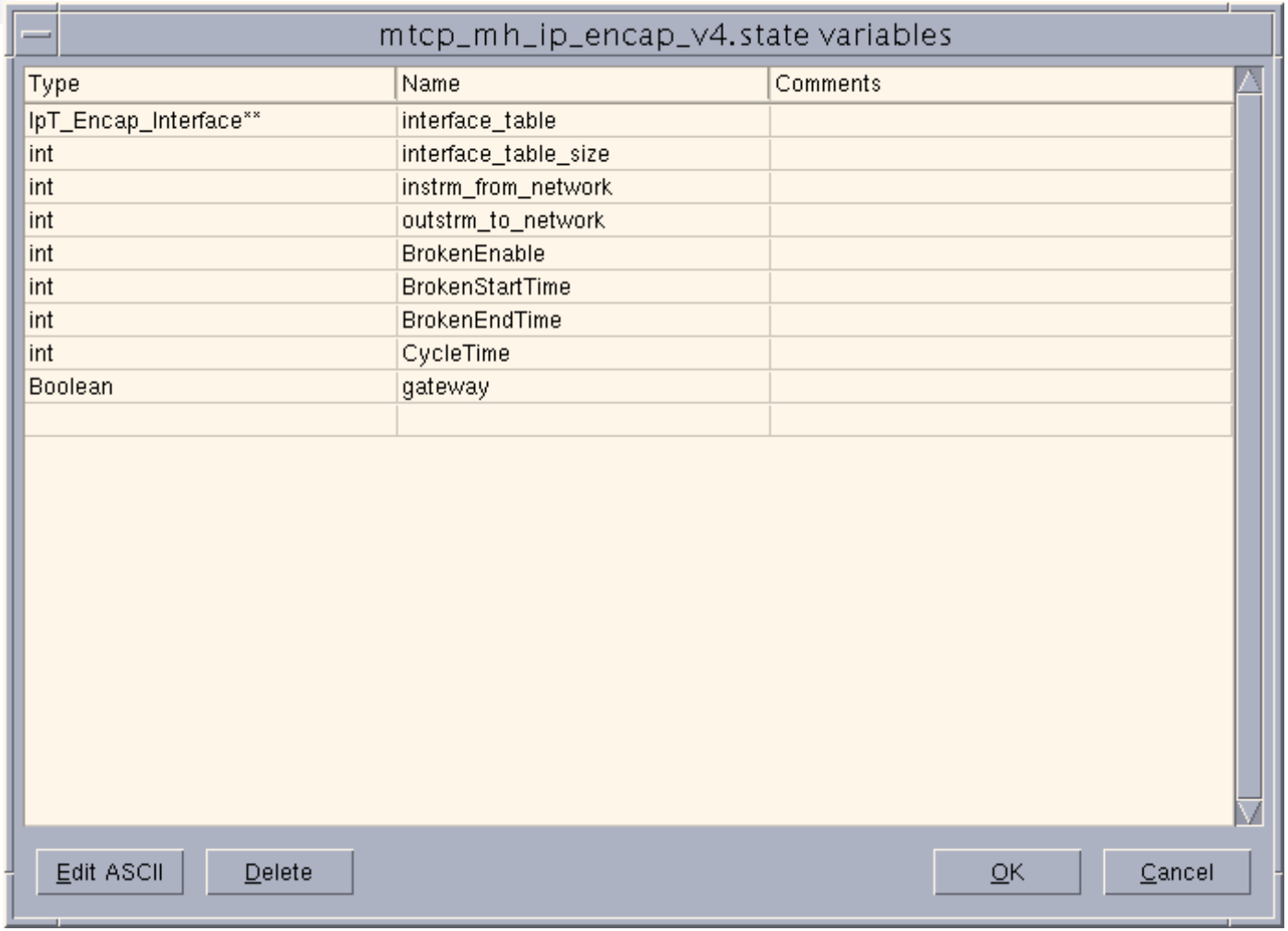
Modified ip_encap process attributes



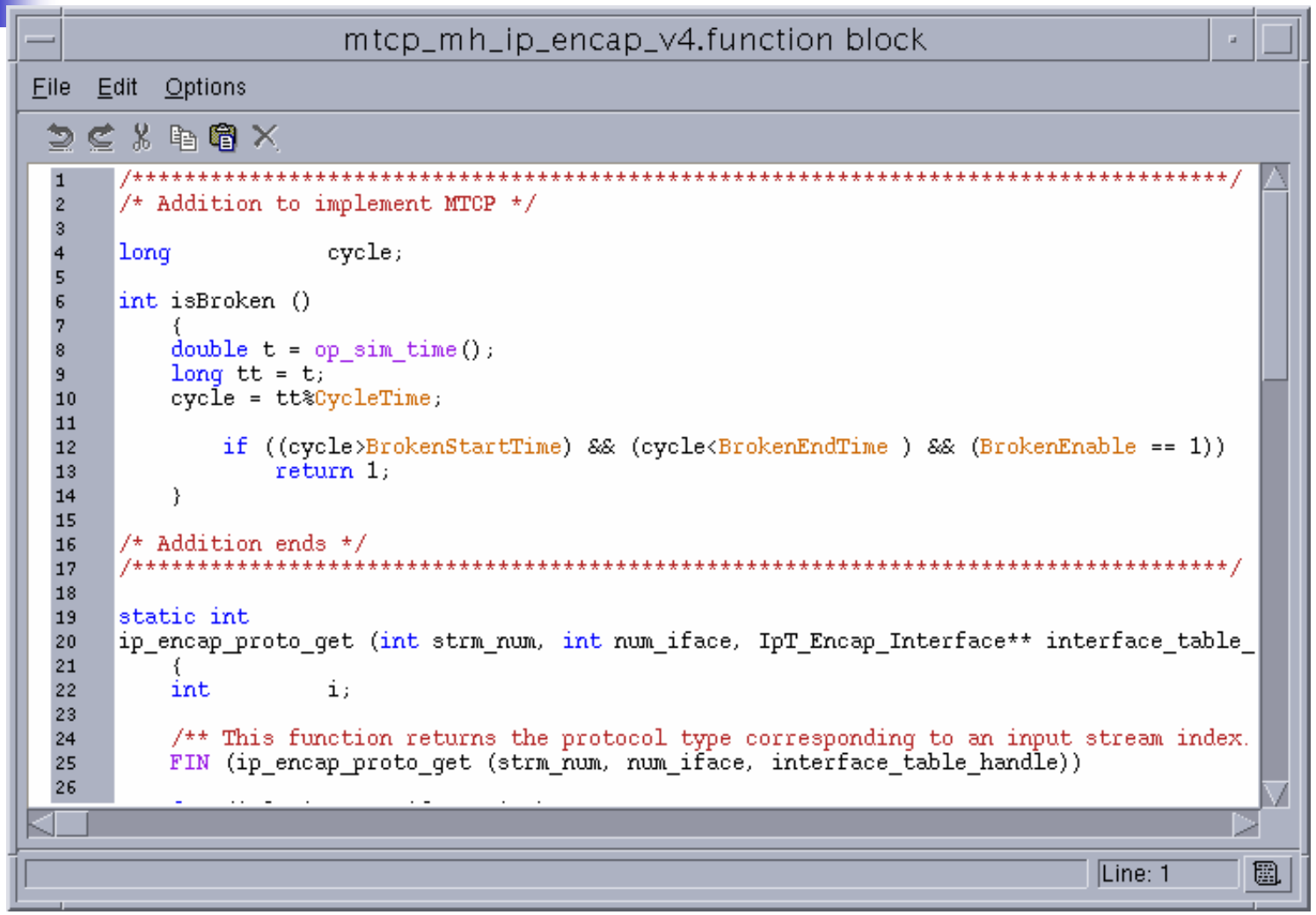
Modified ip_encap process model



Modified ip_encap state variables

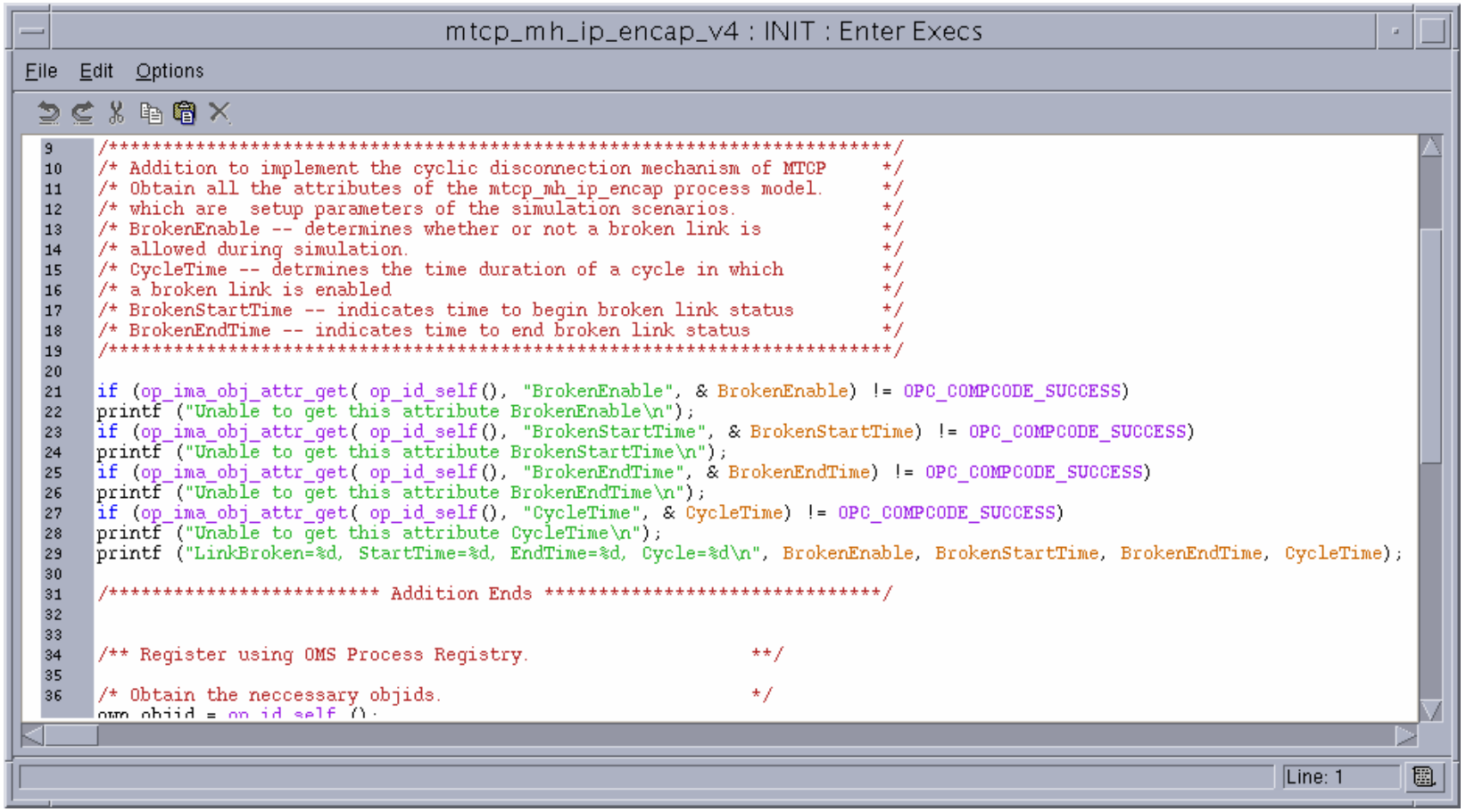


Modified ip_encap function block

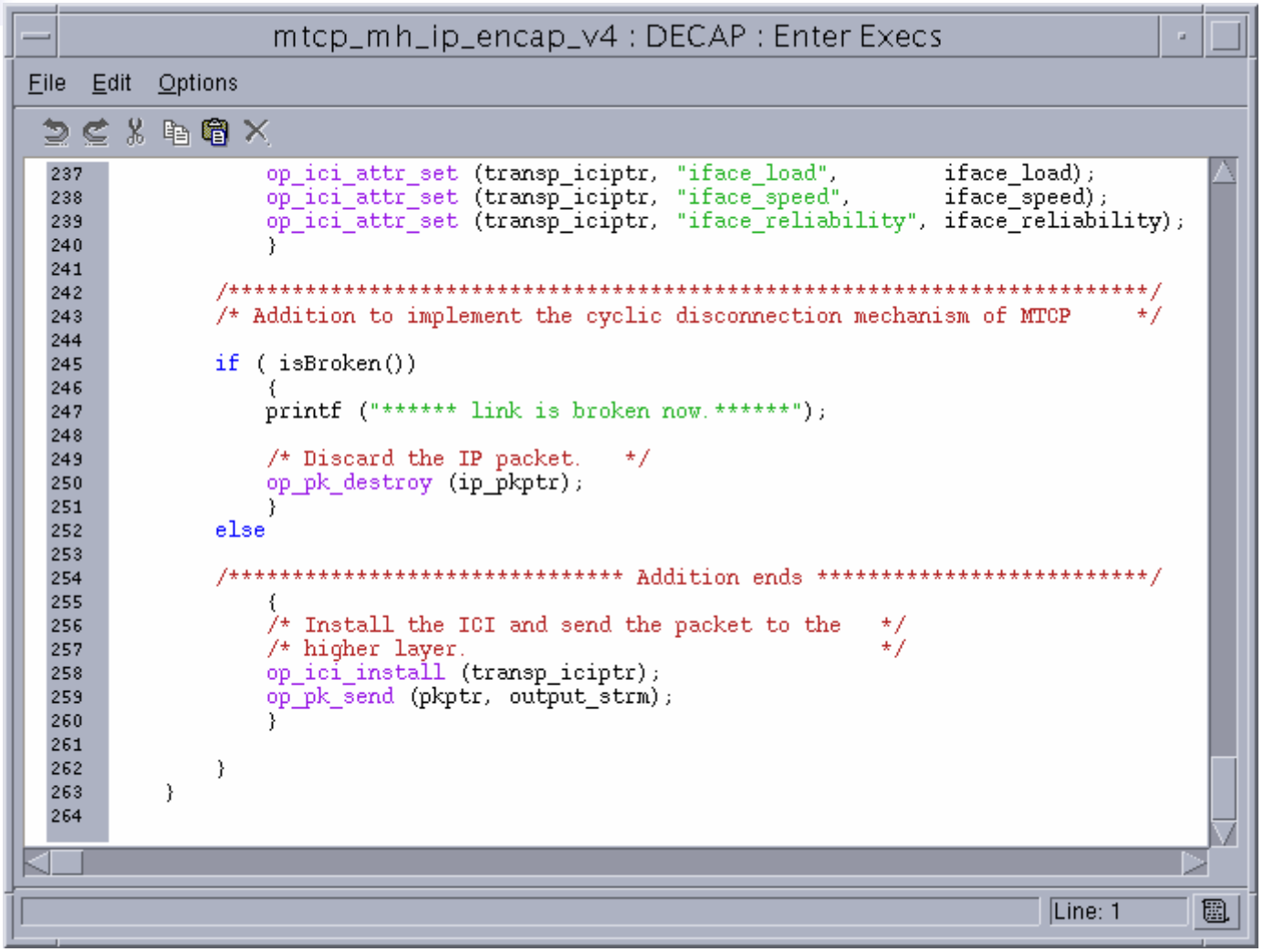



```
1  /**************************************************************************************/
2  /* Addition to implement MTCP */
3
4  long            cycle;
5
6  int isBroken ()
7      {
8      double t = op_sim_time();
9      long tt = t;
10     cycle = tt%CycleTime;
11
12         if ((cycle>BrokenStartTime) && (cycle<BrokenEndTime ) && (BrokenEnable == 1))
13             return 1;
14     }
15
16 /* Addition ends */
17 /**************************************************************************************/
18
19 static int
20 ip_encap_proto_get (int strm_num, int num_iface, IpT_Encap_Interface** interface_table_
21     {
22     int         i;
23
24     /** This function returns the protocol type corresponding to an input stream index.
25     FIN (ip_encap_proto_get (strm_num, num_iface, interface_table_handle))
26
```

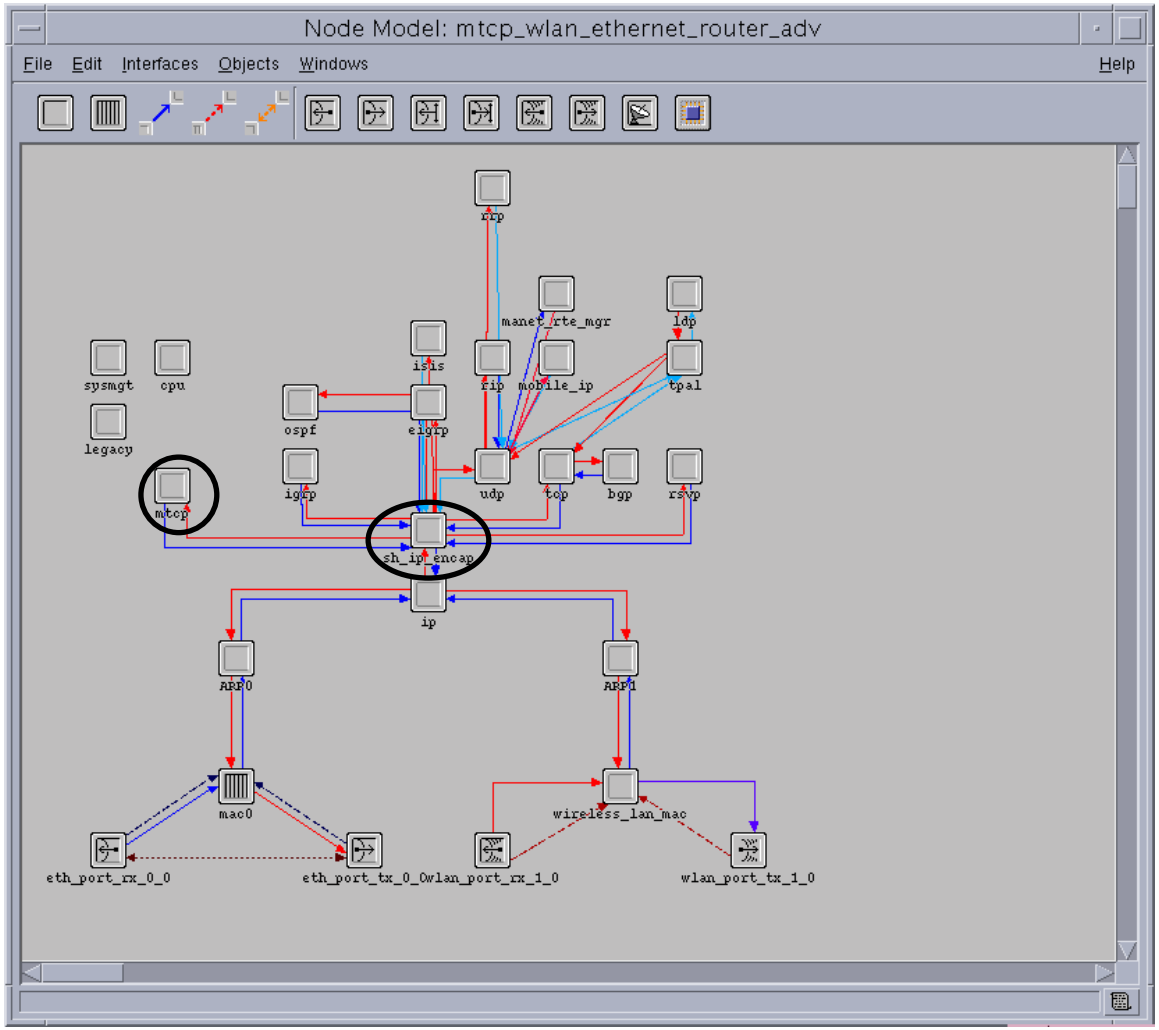
Modified ip_encap init state



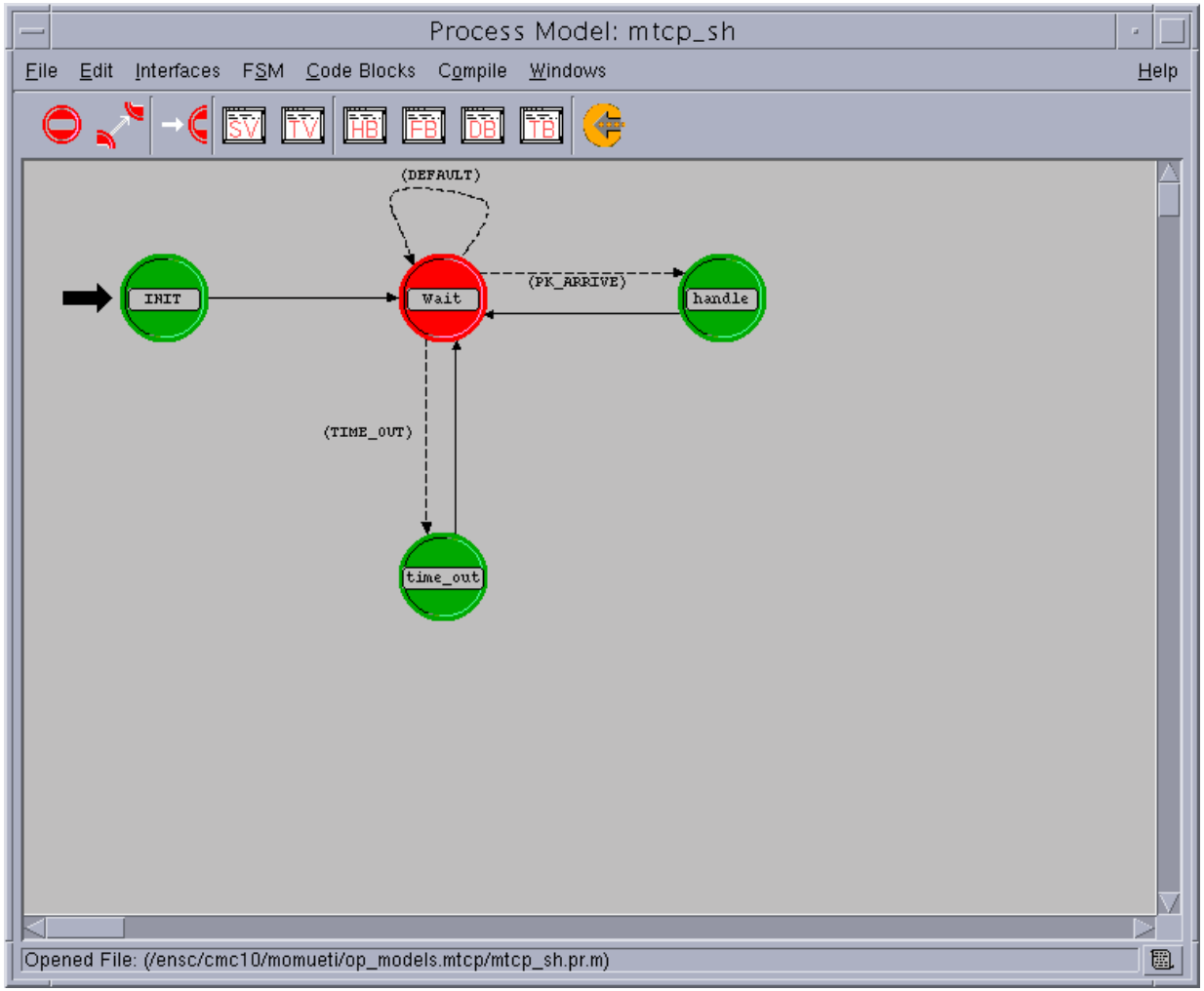
Modified ip_encap decap state



Supervisor host node model



M-TCP process model



M-TCP process header block

mtcp_sh.header block

File Edit Options

```
/* Header Block */
#include <opnet.h>

#include <ip_addr_v4.h>
#include <oms_dt.h>
#include <tcp_api_v3.h>
#include <tcp_v3.h>
#include <tcp_support.h>
#include <oms_pr.h>
#include <oms_qm.h>
#include <oms_tan.h>
#include <tcp_seg_sup.h>
#include <ip_notif_log_support.h>
#include <ip_rte_v4.h>
#include <ip_higher_layer_proto_reg_sup.h>
#include <ip_qos_support.h>

static Ici * iciptr;
static Packet * pkptr;

/* Define transition macros */

#define PK_ARRIVE ( op_intrpt_type() == OPC_INTRPT_STRM )
#define TIME_OUT ( (op_intrpt_type() == OPC_INTRPT_REMOTE )|| (op_intrpt_type() ==
OPC_INTRPT_SELF ))
#define DEFAULT ( (op_intrpt_type() != OPC_INTRPT_STRM) && (op_intrpt_type() !=
OPC_INTRPT_REMOTE ) && (op_intrpt_type() != OPC_INTRPT_SELF))

#define FROM_ENCAP_STRM 0
#define TO_ENCAP_STRM 0

#define FIXED_HOST_ADDR     0xc0000009
#define MOBILE_HOST_ADDR    0xc0000101

static int isInit =0;

static int fh_data_len;

static unsigned fixed_host_port ;
static unsigned mobile_host_port;

unsigned fh_last_seq_num;
unsigned fh_last_ack_num;

// the ack from Mobile host, seen by router, which is
// ack seen by Fixed host plus 1
unsigned mh_last_ack_num;
unsigned mh_last_seq_num;

// shared variables for incoming ip pkt
static IpT_Address      org_addr;
static IpT_Address      dest_addr;
static int              type_of_service;
static int              protocol_type ;
```

Line: 1

M-TCP process function block

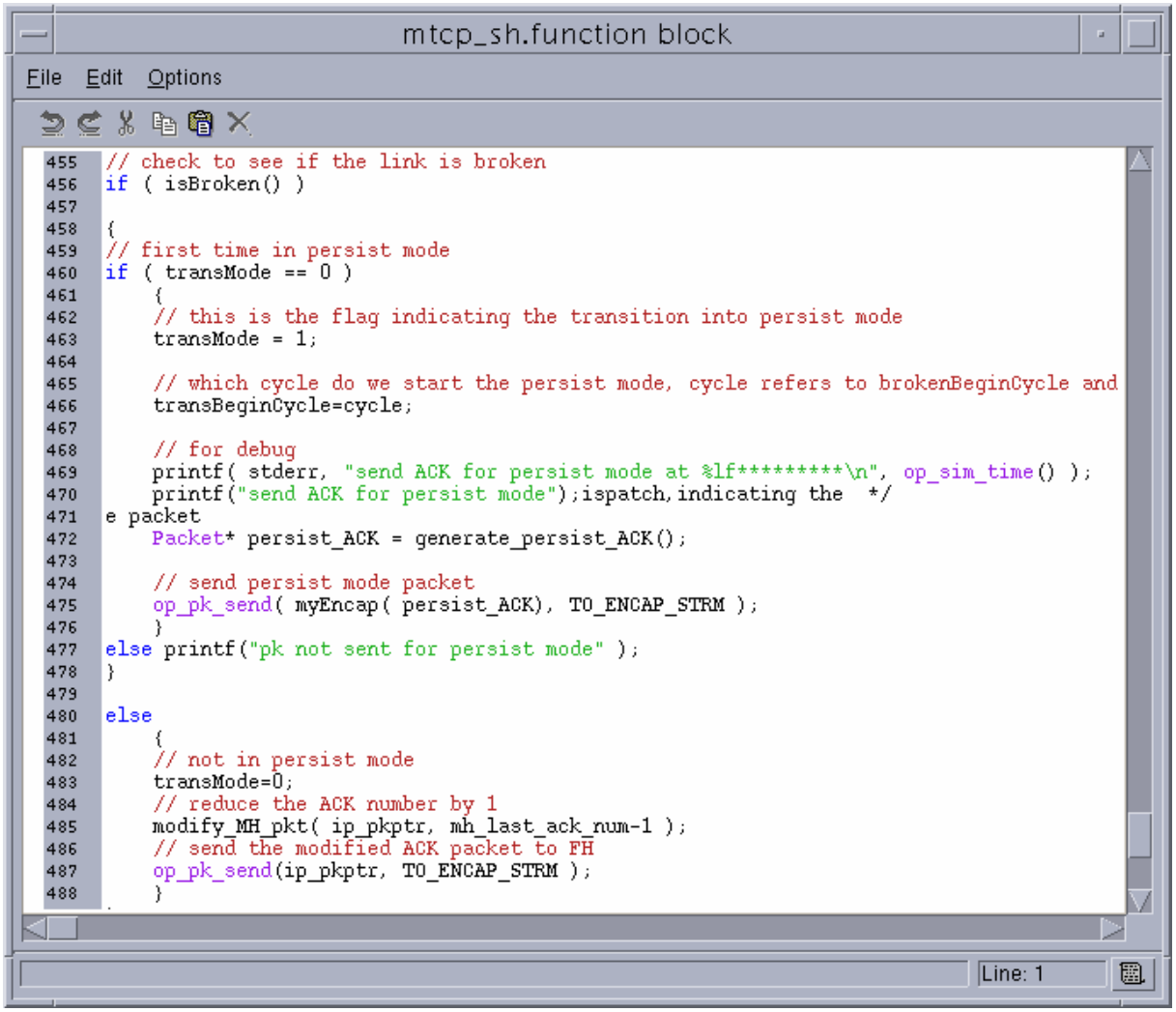
mtcp_sh.function block

File Edit Options

```
212 if (op_pk_nfd_access (original_pkt, "fields", &tcp_seg_fd_ptr) == OPC_COMPCODE_FAILURE
213 {
214 printf ("modify MH: Unable to get data from TCP Segment to modify tcp pkt.\n");
215 }
216 tcp_seg_fd_ptr->ack_num = ack_num;
217 }
218
219
220 ////////////////////////////////////////////////////////////////////////////////
221 //                                                                            //
222 // Generate an ACK packet with rcv_wnd=0, this packet is used to force        //
223 // the sender into persist mode                                               //
224 //                                                                            //
225 ////////////////////////////////////////////////////////////////////////////////
226 static Packet * generate_persist_ACK( )
227 {
228 Packet * seg_ptr;
229 TcpT_Seg_Fields * tcp_seg_fd_ptr;
230
231 // duplicate the sample ACK packet
232 seg_ptr = op_pk_copy ( sample_MH_pkt );
233
234 // get the tcp control fields
235 if (op_pk_nfd_access (seg_ptr, "fields", &tcp_seg_fd_ptr) == OPC_COMPCODE_FAILURE)
236 {
237 printf ("Unable to get data from TCP Segment to get persist tcp pkt.\n");
238 }
239
    // set the tcp control fields and make sure rcv_win=0
```

Line: 1

M-TCP process function block




```
455 // check to see if the link is broken
456 if ( isBroken() )
457
458 {
459 // first time in persist mode
460 if ( transMode == 0 )
461     {
462     // this is the flag indicating the transition into persist mode
463     transMode = 1;
464
465     // which cycle do we start the persist mode, cycle refers to brokenBeginCycle and
466     transBeginCycle=cycle;
467
468     // for debug
469     printf( stderr, "send ACK for persist mode at %lf*********\n", op_sim_time() );
470     printf("send ACK for persist mode");ispatch,indicating the  */
471 e packet
472     Packet* persist_ACK = generate_persist_ACK();
473
474     // send persist mode packet
475     op_pk_send( myEncap( persist_ACK), TO_ENCAP_STRM );
476     }
477 else printf("pk not sent for persist mode" );
478 }
479
480 else
481     {
482     // not in persist mode
483     transMode=0;
484     // reduce the ACK number by 1
485     modify_MH_pkt( ip_pkptr, mh_last_ack_num-1 );
486     // send the modified ACK packet to FH
487     op_pk_send(ip_pkptr, TO_ENCAP_STRM );
488     }
```

Some kernel procedures (KPs) used

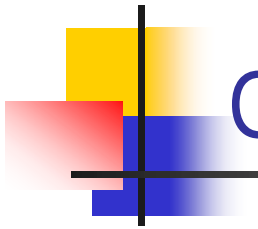
- Packet processing:
  - Op_pk_get()
  - Op_pk_nfd_set()
  - Op_pk_nfd_get()
  - Op_pk_send()
- Interrupt processing:
  - Op_pk_intrpt_type()
  - Op_pk_intrpt_strm()
  - Op_pk_intrpt_schedule_self()

Some kernel procedures (KPs) used

- Segmentation and reassembly
  - Op_sar_segbuf_pk_insert()
- Queues
  - Op_subq_pk_remove()

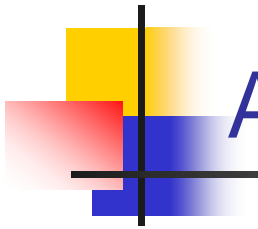
Compiling and debugging

- OPNET debugger
- Print statements
- Error file



Collecting statistics

- Global statistics
- Local statistics

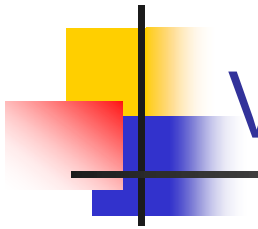


Animation

- Selecting animation

Running scenarios

- Creating a simulation set



Viewing results and animation

- Comparing scenarios
- Playing animation

Cleaning up

- Files that could be deleted to get more space:
  - .ah
  - .ov
  - temporary files
  - backup files
  - Error files

References

- E. Seurre, P. Savelli, and P. Pietri, *GPRS for Mobile Internet*. Norwood, MA: Artech House, 2003 .
- 3rd Generation Partnership Project, TS 04.60 version 7.9.0 General Packet Radio Service (GPRS) Radio Link Control/Medium Access Control (RLC/MAC) layer specification.
- K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 5, pp. 19-42, Oct. 1997.
- W. G. Zeng, M. Zhan, Z. Li, and Lj. Trajkovic, ``Performance evaluation of M-TCP over wireless links with periodic disconnections,'' *OPNETWORK 2003*, Washington, DC, Aug. 2003.
- OPNET documentation V.11.0.A, OPNET Technologies, Inc., Bethesda, MD, 2004.