```
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
#include<math.h>    // have to add -lm to gcc

typedef double Time ;

#define SIMTIME 5000    // Far too small
#define BSIZE 24000
#define tau 1.0

#define PD  1e-1    // 100 ns in microseconds
#define R   1e-2    // 10 ns
#define EQSize 500
```

```
#define MU 5   //  time is microseconds


#define ARRIVAL 1
#define STARTSEND 2
#define ENDSEND 3
```

```
struct IP {
    int x ; // coordinates x.y
    int y ;
} ;

struct Router {
    struct IP ip ;
    char OB[5][BSIZE] ; // OB[4] is Ethernet
    int OBptr[5] ;
    Time LFree[4] ;
    int lost ;
    int arrived ;
} Router77 ;
```

```
// Every event deals with a packet, so I encoded
// events inside packet descriptions.
struct Packet {
    struct IP source , dest ;
    int length ;
// The following are event decriptors
    struct IP where ;
    int NextEvent ;
    Time when ;
    int link ;
    Time Sstart ;
    Time Send ;
} ;
```

```
while( (P = Delete()) != NULL ) {
    Now = P->when ;
// Use in this place:
    if( Now > SIMTIME ) break ;
// as one way of ending simulation
    switch( P->NextEvent ) {
    case ARRIVAL:
        P->where.x = MyRouter->ip.x ;
        P->where.y = MyRouter->ip.y ;
// Use in this place:
//     if( Now < SIMTIME )
// as another way of ending simulation
        Insert( Create( MyRouter , Now ) , ARRIVAL , Now + IA() ) ;
    }
}
```

```
P→link = PickLink( MyRouter , P ) ;
if( P→link == 4 ) {
    MyRouter→arrived++ ;
    free( P ) ;
} else if( MyRouter→OBptr[P→link] + P→length > BSIZE ) {
    MyRouter→lost++ ;
    free( P ) ;
} else { ... ... ...
```

```
case ENDSEND:
    MyRouter→OBptr[P→link] −= P→length ;
    free( P ) ;
    break ;
```