About Dynamic Reconfiguration feature in IRFlexSim

1. Format of a reconfiguration file
2. Handling topology changes

1. Format of a reconfiguration file

The reconfiguration file name is specified with parameter IRRCFGFILE=<name>.

The first line of a reconfiguration file is the number of reconfiguration events: N. Afterwards, there are N lines describing each event.

Each line starts with a number t, denoting the simulation time of the event. A topology change can be addition or removal (failure) of a node or a link. So the rest part of every line is composed of three parts: a sign (+/-) denoting addition or removal, a character (N/L) denoting node or link, and node number(s), separated by spaces.

For addition of a node, the node numbers following `+ N' specifies the existing nodes to which the newly added node is connected to, for example, a line like:

+ N 3 7

Means that a new node is added and it is connected to node 3 and node 7.

For removal of nodes, the node number following `- N' is the node to be removed. Therefore, a line:

- N 1

Removes node 1 and all links connected to the removed node.

For addition or removal of a link, the two node numbers specifies the pair of end nodes of the link. Links are assumed to be bi-directional. If more than one link is connecting the two nodes specified, then one link is removed. (Since links are considered to be equivalent, it does not matter which one is removed.)

So lines:

+ L 8 9
adds a link between node 8 and 9.
- L 7 8
removes a link between node 7 and node 8.

2. Handling topology changes

When a link fails:
1) Transmission from out_buffer to in_buffer cannot be carried out;
2) Mark the link is as fault at the source node;
3) Further routing to this link is not allowed;
4) Kill:
a) Messages occupying the out_buffers of the failed link with a backward killing flit;
b) Messages whose next_channel is the failed link with a backward killing flit;
c) Messages in the in_buffer of failed channel without tail flits already there with a forward killing flit;
d) Unroutable messages
5) Trigger reconfiguration

When a node fails:

1) For out-going channels, if the tail flit is not in in_buffer (including next_in), kill the message forward;

2) Kill messages whose next channel connects to the failed node with backward killing flit;

3) Kill messages destinated for the failed node;

Ignore failed links in inter-node-move;

Ignore failed nodes in node processing and intra-node-move;

About general routing table in IRFlexSim

* Purpose:

Originally the routing function in IRFlexSim is in the form of N x N -> P(C), in which N is set of nodes and C is the set of virtual channels. In order to define more sophisticated routing functions, such as Duato's scheme, in a more general form, routing function has to be defined as C x N -> P(C), i.e. the output channel set is determined by input channel and destination node.

** Implementation:

The general_routing_table is an array of type TABLEENTRY. The array is first divided into N parts, each corresponds to a node in which the part is stored, that is, the current node at which message is being routed. Each part is further separated into VC_PER_NODE segments, each corresponds to an input virtual channel leads to current node. Each segment contains N entries, corresponding to N destination nodes. Each entry is a cyclic linked list. Every element in the list contains the output virtual channel number.

*** Up/Down Duato

Computation of up/down/Duato routing function at one node is divided into four steps:

1. Calculate the shortest distances between nodes and generate shortest path routing matrix -- next;

2. Select the root node(s), (there might be multiple nodes if the network is partitioned) and determine up/down directio
ns on links;

3. Calculate up/down routing matrix -- ud_next;

4. Generate the routing table according to next and ud_next.

Routing matrix is a N x N matrix of bits, bit at (i, j) represents whether node i is one of the next nodes to go in order to reach node j.

README File

Usage: irsim

Options:

D=<dimensions> (default=2)

HALF=n (0=full duplex, 1=half duplex, default=1)

X=n (0=X-Windows off 1 = on, default=0)

DEMAND=<demand> Indicates whether to use demand-driven or time-slice multiplexing. (default=1)

SIZE=c1c2...cn size of network in powers of 2. 33 gives an 8x8 network, default = 44. (NOTE: number of dimensions must be specified before this parameter if different than the default.)

```
   DEBUG=n set debug level (default=0)

   BUFFERS=n set buffer depth of each input virtual channel
       (must be > 1, default=8)

   INJECT=n set inject rate per inject epoch (default=1)

   PER=n set inject epoch.  The simulation will inject on
             average INJECT messages per node every PER
             network cycles (default=1000)

   SECURITY_LEVELS=n set number of security level

   RDELAY= routing header intra-node delay in network
              cycles (default=1)

   ADELAY= acknowledgement flit intra-node delay in network
              cycles (default=1)

   DDELAY= data flit intra-node delay in network
              cycles (default=1)

   HSPOTS=n  set the number of network hot spots (default=0).

   HSPLACE=c1c2..cn  set the location of a hotspot.
              A=10, Z=36, a=37, z=73
              (may be used repeatedly.  Otherwise hot spots
              are randomly placed.)

   M=n  set the number misroutes allowed by MB-m routing
              protocol. (default=3)

   NO_CTS=n  0=CTS lookahead used, 1=CTS lookahead not used.
              (default=0)

   MAP=n  0=No congestion map generated, 1=Generate congestion
              map file. (default=0)

   SLOW=n  1=Place one second delays between flit movement stages
              (Helps in debugging. default=0)

   FAULTS=n  Number of static virtual channel faults to place

   UPDATE=n  0=No dynamic updation (default mode). 1=Enables it.

   UPDATE_DELAY=n ; n > default value;
                  defalut value (UPDATE_DELAY) is #define in dynamic_update.h
                  randomly in the network (default=0)

   PFAULTS=n Number of static physical link faults to place

   TRANS=n   1=display transient statistics (default=1)

   COMM=<communication mechanism>
           W = wormhole routing (WR)
           P = pipelined circuit-switching (PCS)
           AP = acknowledged PCS
           Ax = acknowledged WR
           R = recon routing
           This is set correctly by PROTO. WR is not compatible
           with some routing protocols.

   SCOUT=n  number of acks the routing header sends back to the
            dest before the data flits are sent.  Recon routing.

   SRCQ=n  number of injection queues (default=8)

   TRACE=trace_fn  Turns on trace driven simulation.  Specifies
            trace file.

PROTO=<routing protocol>
```

```
                E = Ecube (defaults to WR communication mechanism)
                D = Duato (psuedo-min congestion selection function) (WR)
                M = Duato with m misroutes (AWR)
                O = Duato (Dimension Order selection function) (WR)
                C = Duato (Minimum congestion selection function) (WR)
                N = Negative First (WR)
                R = Dimension Reversal (WR)
                L = Misrouting Backtracking-m (PCS)
                P = MB-m + (PCS)
                A = A1 (PCS)
                T = Two-phase Backtracking (PCS)
                B = Exhaustive Profitable Backtracking (PCS)
                F = MB-m SW (PCS)
                I = Directional Ordered Routing (WR)

        SELECT=<selection function>
                N = Normal (psuedo-minimum congestion)
                M = Minimum congestion
                O = Dimension Order

        ORDER=specify order of traversal for dor and mesh or toroid
                0 = DOR using y+x+y-x- (toroid)
                1 = DOR using y+x+x-y- (toroid)
                2 = DOR using y+x+y-x- (mesh)
                3 = DOR using y+x+x-y- (mesh)
                0 = Ecube (toroid)
                1 = Ecube (mesh)

        VARY=0/1 Turn off/on halving of virtual channels in different
                 dimensions.  0 = off, 1 = on

        MSGL=<message length in flits>

        DUR=<duration of simulation in network clock cycles>

        VIRTS=<number of virtual channels per physical link direction>

        RADIUS=Size of exhaustive search radius for MB-m SW

        DIST=<distance/communication pattern>
                DIST=0 Random destinations selected from entire network
                DIST=-100 Bit reversal message pattern
                DIST=-900 Transpose message pattern
                DIST=-999 Perfect shuffle message pattern
                DIST=-1000 Flip bits message pattern
                DIST=-1001 Hot spot message pattern
                DIST>0 Random destinations selected from box DISTx...xDIST
                    centered on source node.
                DIST<0 Random destinations selected such that the distance
                    from source to destination is exactly |DIST|.
                (default = 4)

        DIST=<distance/communication pattern>

        DYN=n  Number of dynamic virtual channel faults

        PDYN=n Number of dynamic physical channel faults

(Options for irregular networks:)

        IRTEST=n ?

        IRAUTOGEN=n ?

        IRTOPOFILE=<file name> If not IRAUTOGEN, then read topology from file $IRTOPOFILE

        IRAUTOGENTABLE=n ?

        IRNUMOFNODES=n Number of nodes in irregular networks

        IRNUMOFLINKS=n Number of links in irregular networks
```

IRMINDEGREE=n Minimal degree of nodes in irregular networks

IRMAXDEGREE=n Maximal degree of nodes in irregular networks

IRDUPLINKS=n ?

IRDUATORESTR=n ?