**Implements:**

ProbabilisticBroadcast (pb).

**Uses:**

FairLossPointToPointLinks(flp2p);

UnreliableBroadcast(un).

**upon event** <*Init*> **do**

**forall** Pi П **do** delivered[Pi] := 0;

lsn := 0; pending := stored := 0;

**procedure** deliver-pending (s) **is**

**while exists** [Data, s, x, SNx] pending **such that**

SNx = delivered[s]+1 **do**

delivered[s] := delivered[s]+1;

pending := pending \ {[Data, s, x, SNx]};

**trigger** <pbDeliver | s, x>;

**procedure** gossip (msg) **is**

**forall** t pick-targets (fanout) **do**

**trigger** <flp2pSend | t, msg>;

**upon event** <pbBroadcast | m> **do**

lsn := lsn+1; **trigger** <unBroadcast | [Data, self, m, lsn]>;

**upon event** < unDeliver | Pi, [DATA, Sm, m, SNm]> **do**

**if** (random() > store-thresold) **then**

stored := stored  { [DATA, Sm, m, SNm] };

**if** (SNm = delivered[Sm] + 1)  **then**

delivered[Sm] := delivered[Sm] + 1;

**trigger** < pbDeliver | Sm, m >;

**else if** (SNm > delivered[Sm] + 1) **then**

pending := pending  { [DATA, Sm, m, SNm] };

**forall** seqnb  [SNm - 1, delivered[Sm] + 1] **do**

gossip ([REQUEST, self, Sm, seqnb, maxrounds - 1]);

startTimer (TimeDelay, Pi, SNm);

**upon event** < flp2pDeliver | Pj, [REQUEST, Pi, Sm, SNm, r] > 

**if** ([DATA, Sm, m, SNm]  stored) **then**

**trigger** < flp2pSend | Pi, [DATA, Sm, m, SNm] >;

**else if** (r > 0) **then**

gossip ([REQUEST, Pi, Sm, SNm, r - 1]);

**upon event** < flp2pDeliver | Pj, [DATA, Sm, m, SNm]>

**if** (SNm = delivered[Sm] + 1) **then**

delivered[Sm] = delivered[Sm] + 1;

**trigger** < pbDeliver | Sm, m >;

deliver-pending (Sm);

**else**

pending := pending  { [DATA, Sm, m, SNm] };

**upon event** < Timeout | s, Sn >

**if** sn = delivered[s] + 1 **then**

delivered[s] := delivered[s] + 1;