THE DOUBLE-SLOT SLOTTED RING PROTOCOL (DSR)

David J Greaves.

Olivetti Research Ltd, Cambridge

Abstract

This paper describes a new slotted ring protocol which has been termed the 'double-slot' slotted ring (DSR) protocol. This protocol combines a multi-level priority and fairness guarantee mechanism over a destination-release slotted ring. The fairness mechanism ensures fair bandwidth division within each priority level and operates through a distributed reservation algorithm entirely within the media-access control layer. No additional, higher-level load regulation is required, the DSR mechanism being able to face and reject excess offered load at all priority levels. In common with other destination-release slotted rings, the DSR protocol can offer network bandwidth which exceeds the channel bandwidth. However, unlike some quota based load-balancing mechanisms, the DSR priority mechanism is effective even when supporting bursty sources at high priority.

There is no rule restricting one transmission per ring revolution. Therefore the protocol efficiency is virtually independent of the number of slots and the ring latency. The DSR protocol is therefore proposed not only for LAN applications, but for multi-media, metropolitan area applications, supporting in excess of 100 kilometres of fibre and line rates in the gigaherts region.

1 Background

Slotted multi-access techniques are rapidly gaining popularity owing to their real-time traffic capabilities and their ease of inter-operation with ATM style fast-packet switches. There are many slotted multi-access projects, including the Cambridge Backbone Ring [GREAVES 90], the Cambridge Fast Ring [HOPPER 88], Upperbus [GILOI 86], Magnet [PATIR 85], CRMA [NASSEHI 89], Metaring [OFEK 90], Orwell [ADAMS 87] and QPSX/DQDB [NEWMAN 86]. For slotted rings, there are many protocol variants, but all format the ring delay into permanently rotating slots with a full (not empty) flag at the start. One major variation between slotted protocols determines whether a station limits itself to one transmission per ring revolution, like the Cambridge Fast Ring, or not. Those which do not include the Cambridge Backbone Ring and Orwell. The second major variation between protocols is whether the source or destination station frees (or releases) the full slot after use by clearing the full flag.

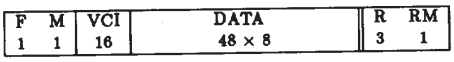
Unlike token rings, the efficiency of the slotted protocols is not inversely proportional to the number of bits stored in the ring cables; it is independent. The number of bits stored in a ring increases with ring circumference, operating frequency, number of stations and station node delay. Therefore slotted rings are also preferred for very high bandwidths or metropolitan areas. For a ring where stations limit themselves to one slot per ring revolution, the point-to-point bandwidth decreases with the number of bits stored and the access delay for bulk arrivals can tend to increase. Rings which transmit in more than one slot per revolution do not suffer from this.

Destination release of slots at the receiving station requires the station delay in bits to exceed the length of the destination address field. However, since destination-released slots can be used again immediately, the network efficiency is increased. If the average distance a full slot travels around the ring is approximately half its circumference, then the network throughput can be approximately double the ring channel bandwidth. The additional delay required for destination-release does not impact performance if stations can use multiple slots per ring revolution, and for Gigabit MAN applications, is in any case negligable.

2 Introduction

As mentioned, destination-release offers the advantage of approximately twice the throughput of source-release. However, at saturation, source-release offers fair sharing of the bandwidth, on a per-station basis, whereas destination-release possesses no intrinsic load balancing. It might be hoped that source-release also provides a performance advantage to expedited traffic below saturation. Expedited traffic is priority traffic that is transmitted by a station in preference to lower priority traffic queued within the same station. However, it has been shown that under most circumstances, below saturation, the additional bandwidth created by destination-release reduces the delays by a greater amount than the load-balancing implicit to source-release [GREAVES 89]. Another shortcoming of the source-release and expedited transfer combination is that the priority mechanism does not operate in the MAC layer. Therefore stations are unable to use a greater fraction of the bandwidth than proportional to the reciprocal of the number of stations, even though all their traffic might be of a higher priority than that transmitted by the other stations.

0238 CH 2901-7/90/0000-0238 $1.00 © 1990 IEEE



An example DSR frame. The data slot contains a full/empty flag F, a monitor passed flag M, a virtual circuit identifier for ATM mode routing VCI and a data field of 48 bytes. The reservation slot contains a 3 bit reservation field R, sufficient for seven layers of priority, and a monitor passed flag RM, for clearing permanently rotating reservations.

Figure 1: An example DSR frame format, without responses.

This paper presents a new slotted ring protocol which attempts to combine the load-balancing intrinsic to source-release with the performance of destination release. In addition the protocol includes a fully effective priority mechanism where all traffic above the saturated priority level is transmitted, fairness is ensured within the saturated level and traffic at lower priorities is cut off. The granularity of load-balancing within a priority level is the same as that of a source-released ring.

The protocol has been termed the 'double-slot' slotted ring (DSR) protocol. It operates through a distributed reservation algorithm entirely within the media-access control layer and without higher level load regulation being required to face and reject excess offered load at any priority level. There is no rule restricting one transmission per ring revolution. Therefore the protocol efficiency is virtually independent of the number of slots and the ring latency. The DSR protocol therefore looks very attractive for multi-service networks at large geometries.

This paper first describes the frame format required by the DSR protocol and then the protocol itself. As an aid to understanding, the protocol is first described without its MAC layer priority mechanism in section 4, then the full protocol is presented in section 5. Some brief simulation results are presented in section 6, examining the DSR protocol under multi-media workloads. The DSR protocol is also compared with a destination-release slotted ring without DSR load-balancing under particularly difficult traffic conditions. The final section suggests a method for providing a low-level, hardware response mechanism that can be added to any destination-release ring.

3 DSR Frame Format

As is usual for a slotted ring, the ring physical layer is formatted into frames of fixed length which continuously rotate. An example is shown in figure 1. For the DSR protocol without low-level responses, each frame contains two different sized slots. There is no logical association between the two slots, they are grouped into a frame to ensure that there are equal numbers of each type of slot and for ease of a hardware implementation. Low-level responses can be added using a third type of slot in each frame, as described in section 7

One of the slots in the frame is termed the data slot. This forms the data carrying payload part of the frame. It dominates the other type(s) of slot in size, resulting in good channel efficiency. A payload of 48 bytes is appropriate for ATM compatibility. The second type of slot is the reservation slot, which can be encoded in two bits for the simplified version of the protocol, and in less than half a byte to support seven distinct priority levels.

The data slot contains a minimum of a full flag, a monitor-passed flag, either a routing tag VCI or address fields and a cell data field. The data slot is used in the usual way for a destination-release slotted ring, being filled and marked full by the transmitter, and copied and marked empty by the receiver. Broadcasts require source-release of course. The DSR protocol does not prohibit a station from immediately refilling a data slot it has just cleared with new data, provided it is authorised to transmit by the load-balancing mechanism. Whether a station immediately re-uses the slot is an implementation detail, but higher throughput will result if it does.

The monitor-passed mechanism frees permanently rotating full slots and corrects for 'livelock' as described in [WHEELER 89]. Both the data and reservation slots independently require such protection.

4 DSR Protocol Without Priority

Under this simplified DSR protocol, the transmit side of each station maintains a specific counter register whose value is termed the station's current authority. This counter is non-negative, and contains a value which represents the number of cells the station is authorised to transmit. For each transmission into an empty, passing data slot, the station decrements the authority counter and it cannot transmit when the counter is zero. Although the description in this section is for the DSR without priority, it should be noted that a fairly effective priority mechanism can be supported by maintaining expedited and non-expedited queues within each station, the expedited queue always being served first.

The authority counter is increased in value by successful reservations. A reservation is a successful transmission into a reservation slot. The transmission simply consists of marking full an empty reservation slot, and then, one revolution later, freeing it, *and passing it on free.* The authority counter may be incremented as the full bit of the reservation slot is written. Therefore no delay penalty need be incurred when making a reservation and an optimum implementation would be able to transmit into the data slot of the frame which contained the reservation slot. The amount that the authority counter is increased by a reservation is two. Two is an approximation to the ratio of data to reservation slot transmissions that occur at saturation. (This is because, in the absence of multi-cast traffic, on average, a full slot carries its cell half way around the ring before the cell is received and the slot freed.) For example, under homogeneous traffic with 18 stations which do not send to themselves and pass on used data slots free, the actual ratio is given by $2N(N-1)/(N(N+1)-2) = 1.8$ for $N = 18$ stations. This expression is obtained from some simple algebra with arithmetic progressions. The effect of this approximation is for further study. A station is only permitted to make reservation transmissions if its queue length exceeds its authority.

In order to free reservation slots used in the previous rotation, stations are obliged to keep state about each slot on the ring. This is termed 'profile' information and is kept in a profile RAM, indexed by a counter which is incremented for each frame that passes and overflows once per ring revolution. A profile RAM is required for all protocols which utilise any sort of source-release and can transmit in more than one slot per ring revolution. The profile entry for each slot needs be only one bit which records whether a reservation slot is outstanding and will need to be cleared. More state will need to be kept in the profile RAM as we increase the protocol complexity.

5 DSR Protocol With MAC Layer Priority

In order to support priority, a station maintains separate queues for each level of priority. The levels are numbered 1 to P with P being the highest level. Instead of a full bit, the reservation slot of the full DSR protocol contains an integer, R, which represents a reservation level. If R is zero, then there is no reservation and the reservation slot is termed 'empty'. Slots are initially written with R equal to zero by a ring monitor and a lost slot collection system implemented by the monitor must ensure that they return to zero under error conditions. The M and RM flags shown in figure 1 will serve for this purpose, using the standard monitor-passed algorithm.

The single authority counter at each station of the non-priority DSR turns into an array under the full DSR protocol. The array contains one entry for each priority level, 1 to P.

The full DSR protocol operates as follows. To make a reservation at priority level P, a station must overwrite the R field in a reservation slot with the value P and this new value P must have been higher than the value which was previously in the reservation slot. Upon making such a reservation, the station double-increments the authority counter at level P and is then able to transmit in two data slots as before. When the reservation slot comes around again during the next revolution, if it still contains the same R value, as determined by comparing with the profile entry, then it is cleared to zero and passed on free. On the other hand, if the reservation slot has been updated, implying that it will contain a higher value of R, then the new value of R is left intact, but the station's authority for the previous value must be double decremented. The station may already have spent these authorities, so the authority counter is not decremented if it would go negative.

Additional elements of the protocol apply to lower priority authorities being stolen by higher priority traffic within a station. It would be silly if a station were barred from transmitting a cell when it was in possession of lower priority authorities. A station attempts to make a reservation at the highest priority level for which its queue length exceeds its authority. When it transmits, it always sends its highest priority cell and always decrements the lowest authority counter that is non-zero. If there is no non-zero authority, then it cannot transmit. It is not allowed to use higher priority authorities for lower priority traffic. This stealing of authorities from lower class traffic requires an additional rule: if the authority level at any priority should exceed the queue length at that level (as can occur since traffic has borrowed lower authorities) then the authority level must be clipped to the queue length. This is all summarised in the code of figure 2. This code has been included in a simulator. The routine 'Protocol' is called once per station, then all of the slots are moved on by incrementing the 'pos' variable at each station, mod the number of slots on the ring.

6 Three Brief Simulation Results

Table 1 gives some point simulation results for the DSR protocol at HSLAN or small MAN dimensions. This table shows that as the amount of a foreground[1] non-bursty source, such as voice or constant rate synchronous video source, is increased, it suffers little interference from a fixed background loading of Poisson cell arrivals.

The next result, table 2 shows the same information when the background source consists of bulk arrivals of cells with Poisson inter-bulk times and fixed size of 300. The delay column for the bulk arrivals reports the time for the last cell of a message. The bulk arrival background source has raised the 99^{th} percentile of delay for the foreground load to nearly the bulk size.[2] This is a consequence of certain slot occupancy patterns which can occur on a slotted ring, which, in the worst case, can reduce the slotted ring delay performance to that of a token ring. Although this limit is not nearly approached in practice, it suggests that for optimum performance, stations should be banned from transmitting in multiple consecutive slots at low loads. Again, in practice, host interfaces may not be able to deliver cells fast enough to keep a bulk in one piece. This effect may be sufficient to improve the 99^{th} percentile of delay for the synchronous traffic.

Table 2 also shows that the synchronous traffic experiences lower delays as its quantity is increased. This again may be attributed to the breaking up of the bulk background packets on the rings.

Table 3 presents a comparison between the DSR protocol and a destination-release ring without a load-balancing mechanism and also a source-release ring. The same network parameters were used as before. Unlike the first two tables, in this table the applied traffic was not homogeneous in its choice of destination station. The applied load consisted of 22 Mbit/second of priority 3 synchronous load, evenly applied and destined, a bursty source of priority 2 applied at one station only, and destined for the station on the other side of the ring, and a background, saturated source, applied with priority 1 to all stations, destined to all other stations in an even manner. The bursty source queues one cell each slot time for 250 slot times, then rests for 250 before starting again. This is a vicious example of the type of traffic which might be presented by variable rate video sources. The average bandwidth of this burst source is half the channel rate, namely 128 Mbit/second.

[1] 'Foreground' refers to the traffic of primary interest. 'Background' refers to other traffic on the same network.

[2] That is, the delay for uninterrupted transmission of a single bulk arrival.

```
TYPE Q = (0..P);          (* The priority level type *)
TYPE STATION = RECORD
    pos        : INTEGER; (* pointer to slot number at station *)
    Buf :ARRAY Q OF POINTER TO ARRAY bufferindex OF slotPTR;
    In, Out    : ARRAY Q OF bufferindex;  (* Circular buffer pointers *)
    auth       : ARRAY Q OF INTEGER;      (* Authority array*)
    profile    : POINTER TO ARRAY[0..maxframes-1] OF Q;
    END;

VAR stationarray   : ARRAY [ 0..maxstations-1 ] OF STATION;

PROCEDURE Protocol(station :INTEGER);
VAR oldp, pri, prim : Q;
    sp  : slotPTR;
    qlen : INTEGER;
BEGIN
  WITH stationarray[station]^ DO
    sp := ring[pos];                (* Pointer to current slot *)
    oldp := profile^[pos];          (* Old value from profile  *)
    profile^[pos] := 0;

    IF oldp .GT. 0 THEN             (* If slot in use before   *)
      IF oldp .EQ. sp^.R THEN       (* and if still ours       *)
        sp^.R := 0;                 (* then reservation slot clear and pass on free *)
      ELSE
        IF auth[oldp] .GT. 2 THEN INC(auth[oldp], -2) END
      END                           (* else lose that authority *)
    ELSE
      (* Attempt transmit into free reservation slot, or one owned by another station *)
      pri := P;                     (* Search down from highest priority level *)
      LOOP
        qlen :=INTEGER(In[pri])-INTEGER(Out[pri]);       (* Measure queue length *)
        IF qlen .LT. 0 THEN qlen := qlen + buffersize END;  (* Correct for circular buffer *)
        IF qlen .LT. auth[pri]  THEN auth[pri] := qlen END; (* Clip authorities *)
        IF (qlen .GT. auth[pri]) AND (pri .GT. sp^.R) THEN     (* If allowed to reserve *)
          INC(auth[pri], 2);            (* then increment authority *)
          sp^.R := pri;                 (* write to reservation slot *)
          profile^[pos] := pri;         (* and record in profile. *)
          EXIT
        END;

        IF pri .EQ. 1 THEN              (* Else try next lower priority *)
          EXIT
        ELSE
          pri := pri - 1
        END
      END
    END;

    CASE sp^.flag OF                    (* Data field update, examine full bit *)
      empty:
        prim := 1;                      (* Empty, so find minimum transmit authority *)
        WHILE (auth[prim]=0) AND (prim .LT. P) DO INC(prim) END;

        IF auth[prim] .GT. 0 THEN       (* Attempt to send at this level or higher. *)
          pri := P;                     (* Try highest first. *)
          LOOP
            IF TrySend(station, pri) THEN
              INC(auth[prim], -1);      (* Make and record transmission *)
              EXIT                      (* stealing authority from lqwest. *)
            ELSE
              IF pri .GT. prim THEN     (* else try next lower priority *)
                pri := pri - 1
              ELSE
                EXIT
              END
            END
          END
        END
(* If full, examine destination field *)
    | full:
        IF station .EQ. sp^.dest THEN
          ReceiveFromSlot(sp);
          sp^.flag := empty             (* and pass on free to next station *)
        END
    END
  END
END Protocol;
```

Figure 2: Modula 2 encoding of the DSR protocol

Protocol	Synchronous priority 3			Bursty priority 2			Background saturated	ρ
	MBit/s	Mean	99tile	MBit/s	Mean	99tile	priority 1. MBit/s	
DSR protocol	22	12	100	128	140	239	206	1.39
No protocol	22	10	45	23	-	-	417	1.8
Source-release	22	48	100	12.25	-	-	208	0.95

Table 3: DSR performance compared with an unbalanced destination-release slotted ring.

Synchronous priority 2			Poisson priority 1.			ρ
MBit/s	Mean	99tile	MBit/s	Mean	99tile	
26	1	5	129	2	10	0.6
51	1	5	127	3	15	0.7
77	1	5	129	4	20	0.8
102	2	10	128	5	25	0.9
128	3	15	128	7	25	1
154	2	10	129	8	30	1.1
179	5	20	127	12	45	1.2
205	7	25	129	26	105	1.3
230	8	30	128	42	205	1.4
256	9	35	124	879	-	1.49

Simulated DSR delay of priority 2 cells from a synchronous source for various offered loads, against a fixed background offered load of Poisson arrivals. This table, and the others, are for a ring with 18 stations and 20 kilometres of cable with an effective data rate of 256 Mbit/second after all control fields have been accounted for. This ring has 256 bit cells and therefore 100 slots. The column designated ρ is the sum of the two throughputs divided by the channel bandwidth. The delays are measured from cell arrival in a station's queue to their transmission into a data slot (the last cell of a bulk is used where applicable). The times are in units of one slot and the throughputs are in megabit per second. The accuracy of the mean delays is about 5 percent and for the 99^{th} percentiles it is better than 20 percent.

Table 1: DSR delay performance in slot times against a Poisson background.

It may be remarked that all three protocols acceptably handled the highest priority, synchronous sources. However, the DSR protocol was effective in reserving bandwidth for the bursty source whereas with the others, not all of the bursty traffic was transmitted. The source-release ring simply divided the available bandwidth between the eighteen stations. Each received 13.5 Mbit/second giving an overall utilisation of $N/(N+1) = 0.9473$. For the destination-release ring without the DSR protocol, a greater amount of traffic was transmitted overall, but not of the desired type. The network simply saturated at a throughput of 1.8 times the channel bandwidth (eighteen stations again). This reduction of throughput is intrinsic to the deliberately extreme requirements of the applied load; the DSR protocol does not result in unnecessary throughput reduction, as can be seen from the previous tables.

The delays shown in table 3 for the priority 3 destination-released traffic are lower without the DSR protocol. This shows that, although a reservation slot can quickly be obtained at high priority, the corresponding data slot can be delayed. The delay is greater with the DSR protocol than without because the DSR has correctly provided access to the exceedingly bursty priority 2 traffic and this has again resulted in a slight 'token like' behaviour.

8 Summary

This article has described the double-slot slotted ring protocol (DSR) and shown that it is successful at handling bursty, high priority loads. The throughput of the DSR protocol exceeds that of a source-release slotted ring, while offering similar granularity of load-balancing. It also offers an effective, MAC layer priority mechanism.

Synchronous priority 2			Bulk arrivals priority 1.			ρ
MBit/s	Mean	99tile	MBit/s	Mean	99tile	
26	18	270	120	499	1170	0.57
51	14	240	120	613	2105	0.67
77	11	165	132	702	1895	0.82
102	10	145	126	892	2910	0.89
128	8	115	141	1167	3525	1.05
154	9	100	138	1681	-	1.14
179	10	60	129	1840	-	1.2
205	8	40	132	3077	-	1.32
230	10	40	132	4075	-	1.42
256	11	45	129	4992	-	1.5

Table 2: DSR performance against a background of Poisson bulk arrivals of size 300.

It is recognised that further work is required in order to make a more complete investigation of the protocol's behaviour. In particular, as is common with priority access protocols (such as FDDI's timed-token protocol) low-priority stations immediately downstream of very active high-priority stations experience favoured access delays.

I am indebted to John Porter for his criticisms.

References

[**GREAVES 89**] "Multi-access metropolitan area networks." University of Cambridge PhD dissertation. DJ Greaves, September 1989.

[**WHEELER 89**] "The livelock-free protocol of the Cambridge Ring." Prof DJ Wheeler. The Computer Journal, Vol 32 No 1, page 95. 1989.

[**NASSEHI 89**] 'CRMA: An access scheme for high-speed LANs and MANs.' M M Nassehi, IBM Zurich. IEEE Supercom/ICC 1990. April 16-19 Atlanta, GA.

[**ADAMS 87**] 'The Orwell Torus communications switch.' JL Adams. BTRL. Proceedings CEPT seminar on broadband switching. Albufeira, Portugal. January 1987.

[**GILOI 86**] 'Upperbus: A high speed backbone for metropolitan area networks.' WK Giloi, P Behr and G Zuber (BERCOM). Proceedings of EFOC/LAN, Amsterdam, June 1986.

[**PATIR 85**] 'An optical fibre based LAN for MAGNET's testbed environment.' A Patir, T Takahashi, Y Tamura and ME Zarki. Columbia University. IEEE JSAC Vol SAC-3 number 6. November 1985.

[**NEWMANN 86**] 'Distributed Queueing: A fast and efficient packet access protocol for DQDB.' RM Newmann and JL Hullet. University of Western Australia. Proceedings of ICCC 1986, pp 294-299. North-Holland.

[**GREAVES 90**] 'The Cambridge Backbone Ring Network.' DJ Greaves and A Hopper and D Lioupis. Proceedings IEEE Infocom 90, San Fransisco June 1990.

[**OFEK 90**] 'Metaring - A full-duplex ring with fairness and spatial reuse.' I Cidon and Y Ofek, IBM TJ Watson Research Centre. Proceedings IEEE Infocom 90, San Fransisco June 1990.