ATNP/WG2-WP/121
12 May 1995

AERONAUTICAL TELECOMMUNICATIONS NETWORK PANEL

WORKING GROUP TWO

Washington 15.5.95-19.5.95

Congestion Management Strategies

Presented By Martin Adnams

Prepared by Leen Goossens

SUMMARY

While the guidance material in the ATN Manual only gives a rough guideline on congestion management, this working paper attempts to describe different congestion management alternatives at a level of great technical detail The purpose of this document is to serve as a basis to initiate a discussion on congestion management. The congestion management alternatives presented in this paper will be evaluated through simulation and experimentation.

TABLE OF CONTENTS

1. Introduction ... 1

2. Network Layer Congestion Control ... 2
 2.1 Setting of the “congestion experienced” flag ... 2
 2.2 Sending of an Error NPDU indicating congestion ... 2

3. Transport Layer Congestion Control ... 4
 3.1 Sending Transport Layer Backoff ... 4
  3.1.1 Adjustment of the congestion window ... 4
  3.1.2 Dynamic adjustment of the retransmission timer ... 6
   3.1.2.1 Dynamic adjustment of the initial value of the retransmission timer ... 6
   3.1.2.2 Dynamic adjustment of the value of the retransmission timer each time a packet is retransmitted. ... 7
 3.2 Receiving Transport Layer Withdrawing Credit ... 8

4. Comments on the ATN Manual ... 9
  4.1.1 Network Layer Congestion Control ... 9
  4.1.2 Transport Layer Congestion Control ... 10
   4.1.2.1 Sending Transport Entity Rules ... 11
   4.1.2.2 Receiving Transport Entity Rules ... 11

5. Conclusion ... 13

6. Recommendation ... 13

1. Introduction

The ATN Manual describes three strategies for congestion management. However these strategies aren't defined in detail. This document aims at specifying the congestion management strategies in more detail and at pointing out the areas where testing is needed to choose between alternatives or in order to assign values for certain parameters. The congestion management strategies in this document are to be evaluated in simulations and experiments so that some of them may be proposed to replace the ones in the ATN Manual.

The congestion management techniques described here involve the transport layer. The network layer can assist the transport layer by informing the transport layer when it experiences congestion. Congestion management at the transport layer is exercised by limiting the amount of data that may be sent before an acknowledgement is received and by adjusting the time the transport entity will wait for acknowledgement before retransmitting a TPDU. To understand this document it is assumed that the flow control mechanism of the transport layer is known.

Two different methods are to be distinguished. There are congestion management techniques forced by the sender of the data and there are congestion management techniques forced by the receiver of the data. While the former method is obviously the best, sometimes only the receiver of the data is aware of the fact that the network is experiencing congestion.

The ATN Manual is adjusted where this seemed to be appropriate. Comments on the ATN Manual are given in section 4.

2. Network Layer Congestion Control

2.1 Setting of the “congestion experienced” flag

If an NPDU arrives at an ATN IS, the IS examines the depth of the output queue selected for that NPDU.

- If the depth of the selected output queue exceeds a certain proportion of the capacity of that queue, say α%, the ATN IS will set the *congestion experienced* flag before forwarding the NDPU to the destination. The *congestion experienced* flag is a flag in the QOS Maintenance Parameter option header.

- If the depth of the selected output queue exceeds β% of the capacity of the queue (with $\beta \geq \alpha$), the ATN IS will start to discard packets. Lower priority packets shall be discarded before higher priority packets.

The value of α and β need to be determined by investigation. Those values could be configurable as parameters for different subnetwork types and on a per priority basis.

When the *congestion experienced* flag is conveyed to a destination network entity, this entity should convey the *congestion experienced* information to the destination transport entity by local means. This can be done by using the N-REPORT primitive. The information conveyed to the transport entity should include the source and destination NSAPs of the NPDU containing the *congestion experienced* flag, so that subsequent congestion management actions can be restricted to the impacted transport connections.

This congestion management technique should be used in combination with the *Receiving Transport Layer Withdrawing Credit* technique (see section “3.2 Receiving Transport Layer Withdrawing Credit”).

2.2 Sending of an Error NPDU indicating congestion

If a Data NPDU arrives at an ATN, the IS examines the depth of the output queue selected for that NPDU.

- If the depth of the selected output queue exceeds α% of the capacity of the queue, the ATN IS will send an error NPDU indicating congestion to the sender of the Data NPDU.

- If the depth of the selected output queue exceeds β% of the capacity of the queue, the ATN IS will start to discard packets. An Error NPDU indicating congestion shall be sent to the sender of the NPDU, for each discarded packet that is a Data NPDU and that has the ER flag set to allow Error Reports.[1]

The value of α and β need to be determined by investigation. The value of α can but needs not be the same value as the one used in the previous section. The value of β needs to be the same. Those values could be configurable as parameters for different subnetwork types and on a per priority basis.

[1] Returning an Error NPDU for every discarded packet is not a good idea, because that may result in further congestion. However ISO/IEC 8473 and the ATN Manual mandate that error reports are sent in certain circumstances, so that the number of Error Reports sent cannot be limited because congestion is experienced.

When the Error NPDU is returned to the originating network entity, this entity should convey the Error information to the originating transport entity by local means, for example by using the N-REPORT primitive. The Error information that is passed on to the transport entity should include the source and destination NSAPs of the NPDU that caused the Error NPDU to be generated. These addresses can be extracted from the Data NPDU header contained in the Error NPDU. In this way subsequent congestion management actions can be restricted to the impacted transport connections.

After receiving the ER information indicating congestion, the sending transport layer should activate the *Sending Transport Layer Backoff* congestion management technique which is explained in section “3.1 Sending Transport Layer Backoff”.

3. Transport Layer Congestion Control

3.1 Sending Transport Layer Backoff

This section discusses the congestion management techniques that can be applied when a packet needs to be retransmitted because the retransmission timer expired or when the network layer signalled congestion (see section 2.2 Sending of an Error NPDU indicating congestion). Congestion management is then exercised by the sender of the data. A sliding window flow control strategy is used in combination with the dynamic adjustment of the retransmission time.

The sliding window flow control strategy is based on the following: The transport layer keeps track of two different flow control windows:

- the *advertised* window: this is the window advertised by the receiver of the data. It indicates the amount of data that the receiver is willing to accept.

- the *congestion* window: this window is chosen by the sender of the data. It indicates the amount of data the sender will send before an acknowledgement is received. This window should always be a subwindow of the advertised window and the lower window edge of the congestion window should always equal the lower window edge of the advertised window.

Congestion management is exercised by controlling the size of the congestion window. When a packet needs to be retransmitted, it is assumed that the packet got discarded due to congestion. The size of the congestion window is then shrinked, i.e. the credit of the congestion window becomes 1. The credit is increased afterwards after packets are successfully transmitted, i.e. after acknowledgements are received.

This congestion management strategy is based on the assumption, that packet loss is mostly due to congestion and not due to fact that the packet is damaged. A second assumption is that a packet is lost when no acknowledgement is received for it before the retransmission timer expires. This is not always the case. If the retransmission time is too small, the timer will expire before the acknowledgement is received. Therefore it is very important that a good value is chosen for the retransmission time. This time can't be too small, but it may not be too large either, otherwise performance will suffer under it.

The retransmission time should be based on the round trip time of a packet. The round trip time is the time passed between the moment a packet is sent and the moment the acknowledgement for the packet is received. An approximate value for the maximum round trip time is given by

$$E_{LR}+E_{RL}+A_R+x$$

where E_{LR} is the expected maximum transit delay local-to-remote,
E_{RL} is the expected maximum transit delay remote-to-local
A_R is the remote acknowledgement time,
x is a small quantity to allow for additional internal delays, ...

Because the transport layer makes use of the CLNS, there can be great variances in the transit delay experienced by packets. Therefore it is important that the sliding window flow control strategy is coupled with the dynamic adjustment of the retransmission time.

3.1.1 Adjustment of the congestion window

The sliding window flow control strategy works as follows:

The lifetime of a transport connection is divided into phases.

A phase starts:

- when the connection is established,
- when a packet needs to be retransmitted because the retransmission timer of the packet expired (it is assumed then that the packet is lost),
- when the network layer signalled congestion (see section 2.2 Sending of an Error NPDU indicating congestion).

In the beginning of a phase the size of the congestion window equals one packet. Each time an acknowledgement arrives, the size of the congestion window is increased with the number of packets that are acknowledged. In this way the size of the congestion window exponentially increases until an upper bound is reached. The upper bound consist of the minimum of the size of the advertised window and a certain threshold. The threshold equals half of the size of the congestion window at the end of the previous phase. The size of the congestion window can grow past the threshold but much more slowly. The size of the congestion window is then increased with one each time “size(congestion window)” acknowledgements are received.

So schematically this becomes:

```
size(cong_win) = 2 * α;
while ( transport connection exists )
{
    /* Beginning of a phase */
    threshold = max( size(cong_win ) / 2, 1 );
    size(cong_win ) = 1;
    ackrcvd = 0;
    while ( no retransmission timer expires )
    {
        if ( (acknowl. arrives ) && ( size(cong_win) < size(advertised_win) ) )
        then if ( size(cong_win) < threshold )
            then size(cong_win) = min(  size(cong_win) + #packets_ackn,
                                        threshold,
                                        size(advertised_win) );
            else
            {
                ackrcvd = ackrcvd + #packet_ackn;
                if ( ackrcvd > size(cong_win) )
                then
                {
                    ackrcvd = ackrcvd - size(cong_win);
                    size(cong_win) = min(size(cong_win)+1,
                                         size(advertised_win)  );
                }
            }
    }
    /* End of a phase */
}
```

α is the initial value for the threshold. A proper value for α should be determined through investigation.

3.1.2 Dynamic adjustment of the retransmission timer

To maximise performance the initial value of the retransmission timer for a packet (i.e. the value of the retransmission timer the first time a packet is sent) should be chosen as small as possible. To account for great variances in the round trip time of packets, the value of the retransmission timer should be increased each time a packet is retransmitted, so that when there is a significant increase in the round trip time, the transport connection is not closed because the round trip time exceeds "retransmission time x max_number_of_transmissions".

The discussion of the dynamic adjustment of the retransmission timer is divided into two parts:

- the dynamic adjustment of the initial value of the retransmission timer,
- the dynamic adjustment of the value of the retransmission timer each time a packet is retransmitted.

3.1.2.1 Dynamic adjustment of the initial value of the retransmission timer

The value of the retransmission timer should be based on the round trip time of packets. The round trip time can be estimated by measuring the delay between the sending of a packet and the receiving of the corresponding acknowledgement. This delay should be measured frequently. The delay may not be measured for retransmitted packets because there is no way of telling that the received acknowledgement corresponds to the first transmission of a packet or to one of the retransmissions of the packet.

The algorithm used to compute the value of the retransmission time is then:

```
estimated_round_trip_time =
                ( 1 - $\alpha$ )  * estimated_round_trip_time + $\alpha$ * measured_round_trip_time;
retransmission time = $\beta$ * estimated_round_trip_time;

with $0 < \alpha < 1$ and $1 < \beta$
```

((1 - α) * 100)% of the new estimated round trip time is taken from the previous estimated round trip time and (α * 100)% is taken from the last measured round trip time. The recommended value for α is 0.1, so that a temporary change in the round trip time doesn't influence the retransmission time too much.

β should be a value greater than 1. approximate_round_trip_time is multiplied by β to take sudden variances in round trip time into account.

It should be studied what good values for α and β are. It should also be investigated when the round trip time should be measured, and that it should be measured for each packet (that is not retransmitted) or not.[2]

[2] It is probably not necessary to measure the round trip delay for each packet and then one timer could suffice to do the measurements. If the timer is already in use when a packet is transmitted, the round trip delay for that packet will not be measured.

Under TP4 every TPDU has to be retained after transmission until it is acknowledged. So, an alternative way to measure the round trip time is to timestamp each retained TPDU against a local clock. Then, when an acknowledgement arrives, the current local time less the timestamp on the least recently transmitted and acknowledged TPDU provides the measured round trip delay without needing a single timer. Only a clock is necessary.

How the round trip time is measured is a local matter.

If great variances in round trip time are to be expected, the following algorithm should be used:

```
diff = measured_round_trip_time - estimated_round_trip_time;
estimated_round_trip_time =
                ( 1 - $\alpha$ )  * estimated_round_trip_time + $\alpha$ * measured_round_trip_time;
estimated_variance = ( 1 - $\beta$ )estimated_variance + $\beta$ * |diff|;
retransmission time = estimated_round_trip_time + $\chi$ * estimated_variance;

with   $0 < \alpha < 1$,
       $0 < \beta < 1$,
       $1 < \chi$
```

This algorithm takes great variances in round trip time into account. The recommended value for α is 1/8, the recommended value for β is 1/4 and the recommended value for χ is 4. It should be examined that these values are appropriate or that different values should be taken. It is recommended that α, β and χ are powers of 2 so that shifts can be used to do the operations instead of multiplications and divisions.

If the remote acknowledgement time is added to the round trip time measured for the connection request packet or for the connection confirm packet, then this sum can be taken as a start value for the estimated round trip delay. If the acknowledgement times weren't exchanged during connection establishment, one should take an estimated value for the remote acknowledgement time.

It should be investigated what a good initial value is for the retransmission time for the connection request and the connection confirm packet. This could be a random value (for example 250-500 milliseconds), or this value could be based on previous establishment attempts. Further study on this subject is needed. Once an initial value is determined for the CR and the CC packet, the procedure explained in the next section can be executed.

3.1.2.2 Dynamic adjustment of the value of the retransmission timer each time a packet is retransmitted.

If the initial value of the retransmission timer is chosen much too small, either the transport connection is released because the round trip time exceeds "retransmission time x max_number_of_transmissions" or a lot of unnecessary retransmissions will be sent before an acknowledgement arrives. Therefore it is advised that the retransmission timer is increased for each retransmission of the same packet if great variances in the round trip time are expected.

The following algorithm is proposed:

```
retransmission time =  the initial value for the retransmission time which is
                       dynamically determined as explained in section "3.1.2.1
                       Dynamic adjustment of the initial value of the retransmission
                       timer";
while ( packet needs to be retransmitted )
    retransmission time  = retransmission time + $\alpha$;
```

α is a fixed value. It should be investigated what a good value is for α.

With this algorithm the initial value for the retransmission timer can be chosen as small as possible to achieve good performance.

This algorithm is especially useful to adjust the retransmission timers for a connection request and a connection confirm packets. Because these are the first packets to be sent over a transport connection, no measurements to estimate the round trip time for a packet could have been taken.

3.2 Receiving Transport Layer Withdrawing Credit

Congestion management is exercised by the receiver of the data. A sliding window flow control strategy is used for the transmit credit window (i.e. for the window advertised by the receiver of the data to the sender of the data).

Initially the size of the advertised window is based on the local buffer management policy. There is no need to limit this size to one to check that the network isn't experiencing congestion because the sending transport entity already verifies this (the sending transport entity starts with a congestion window that equals one and only increases the size of this window as no congestion is experienced).

When the receiving transport layer is informed that congestion is experienced at the network layer, he gradually has to close the advertised window. He does that by adjusting the lower window edge of the advertised window as packets are received, but the upper window edge of the advertised window can't be changed until all packets that are in the current advertised window are accepted. After all packets are accepted that were in the advertised window, those packets are acknowledged and a window size of one is advertised. Afterwards the window size may be increased by one each time n packets successfully arrive where n is a multiple (say α) of the size of the advertised window and "successfully arrive" includes that the *congestion experienced* flag may not be set for those packets. The advertised window will increase until a maximum is reached. This maximum depends on the local buffer management policy.

A proper value for α should be determined through investigation.

4. Comments on the ATN Manual

4.1.1 Network Layer Congestion Control

- The ATN manual states the following (9.4.7 Congestion Notification Function):

  When the depth of a particular output queue exceeds a certain proportion of the depth of that queue, an ATN IS will start to discard NPDUs; at this time, the ATN IS sets the *Congestion Experienced* flag in the next NPDU to be forwarded toward one or more source Network entities and continues to do so until the congestion condition is alleviated.

  Two remarks can be made:

  - Section “9.4.7 Congestion Notification Function” states that the ATN ISs will start to discard packets when the depth of a particular output queue exceeds a certain proportion of the depth of that queue. This is in contradiction with section “9.3.9 Discard PDU Function” which states that a packet needs to be discarded when it cannot be processed due to local congestion.

    An alternative is proposed in this paper (see section “2 Network Layer Congestion Control”).

  - Section “9.4.7 Congestion Notification Function” states that the *congestion experienced* flag is set in the next NPDU to be forwarded toward one or more source Network entities.

    This is in contradiction with ISO/IEC 8473 which says that, when an IS is experiencing congestion, it can set the *Congestion Experienced* flag in a data NPDU received before forwarding that NPDU to its destination.

    This is also in contradiction with section “8.2.6 Congestion control” of the ATN Manual

- The ATN manual also states the following (9.4.7 Congestion Notification Function):

  The method of initiating Congestion Notification is a local matter. Two different techniques are given as follows:

  1. Congestion notification is initiated when outgoing NPDU queues reach 75% of their capacity, based on a moving average analysis of the outgoing queue lengths. ATN Network entities receiving congestion notification cease forwarding NPDUs toward the congested ATN Network entity until:

     - receipt of five successive NPDUs from that Network entity indicating a lack of congestion, or

     - a time period of $t_{congested}$ elapses.

  2. Congestion notification is initiated when the outgoing NPDU queue has an average length greater than one. A queue length averaging algorithm computes the average queue length over two cycles, where two cycles are:

     - the “previous cycle”, which is the interval from when the IS becomes busy, until it becomes idle and the idle ends (indicated by the instant the first packet arrives to the idle IS);

- the “current cycle”, which is the interval from the end of the idle interval to the current time instant when the average queue length is computed.

The following remarks can be made:

- These two alternatives are quite different when the capacity of the outgoing queue is much larger than one packet. The efficiency and the performance of these two alternatives should be investigated.

- The following problem can arise with the first technique:

  If a transport connection exists between two transport entities, and one of the two network entities supporting the transport entities stops forwarding NPDUs because he has received congestion notification, a problem can arise if the other side hasn’t got any data to be sent. The only packets received from the other side will be acknowledgements containing up-to-date window information and these acknowledgements will only be sent each time the window timer expires. If the network entity that has received congestion notification, ceases forwarding until the receipt of five successive NPDUs indicating a lack of congestion, the inactivity timer at the other side can expire and the release procedure will be initiated.

  When a network entity cease forwarding NPDUs to the congestion network, what does it do with those NPDUs ? Are they discarded ?

- “ATN Network entities receiving congestion notification cease forwarding NPDUs toward the congested ATN Network ...” How can a Network entity know that a network is congested ? Take the following scenario:

  A packet travelling from host A to host B, passes three networks: netw1, netw2 and netw3. If congestion is experienced in netw1, the CE flag is set in the packet. When the packet arrives at netw3, netw3 seeing that the CE flag is set, can suppose that netw2 is congested (netw3 has no way of knowing that is was netw1 instead of netw2). If netw3 ceases forwarding packets to netw2, no packets can travel from netw3 through netw2 towards a fourth network, although there is no congestion experienced in those networks.

  So, if a network entity can’t determine the congested network, a congestion management technique that ceases forwarding NPDUs toward the congested network, can’t be effective.

- The second technique isn’t easy to understand.

4.1.2 Transport Layer Congestion Control

- The ATN Manual talks about local receive credit window and local sending credit window. It is not clear what is meant by these terms. In this document the terms “advertised window” and “congestion window” were used.

- The ATN manual states the following (A 8.2.6 Transport Layer Congestion Avoidance):

  3. An ATN implementation shall use a retransmission timer per transport connection. If, upon expiration of the retransmission timer, an implementation allows more than one TPDU to be transmitted, then a means to locally adjust the maximum number shall be provided.

The following remark is made:

It is not clear what this sentence means. What is meant by “the maximum number” ? Is it the size of the local sending credit window ?

4.1.2.1 Sending Transport Entity Rules

The Sending Transport Entity Rules specified here (see “3.1 Sending Transport Layer Backoff”) and those specified in the ATN Manual differ slightly. The rules specified in section 3.1 consist of two parts: the adjustment of the local sending credit window (the congestion window) and the adjustment of the retransmission timer. The details of the adjustment of the local sending credit window specified in the ATN manual differ from the details specified in this paper. Both should be evaluated. The ATN Manual doesn’t specify details about the adjustment of the retransmission timer. Details are proposed in this paper.

4.1.2.2 Receiving Transport Entity Rules

The ATN manual states the following (A.8.2.6Transport Layer Congestion Avoidance):

Rule 1: Initialisation of Window

The initial value of the dynamic value of the receive window size, WR (WR_0) should have a locally configurable upper bound. This window is sent to the sending TE (STE) in the next CDT field transmitted.

Rule 2: Recommended Sampling Period

All receiving TEs (RTEs) should maintain a fixed value for WR until the next 2*WR DT TPDUs arrive since the last CDT field was transmitted by the RTE.

Rule 3: Recommended Counting of Received TPDUs in a Sampling Period

All RTEs should maintain a count, N, equal to the total number of TPDUs received, and a count, NC, equal to the total number of TPDUs received which had the CE flag set in the associated N-REPORT primitive. All types of TPDUs are included in the counts for N and NC, not just DT TPDUs.

Rule 4: Recommended Action upon the end of a Sampling Period

All RTEs should take the following action at the end of each sampling period:

1. If the count NC is less than 50% of the count N, the RTE should increase WR by adding 1 up to a maximum, WR_1, based on the local buffer management policy. Otherwise, it should decrease WR by multiplying by 0.875 (WR can go to a minimum of 1).

2. Reset N and NC to zero.

3. Transmit the new window WR in the next CDT field sent to the sending transport entity.

The following remarks can be made:

- This strategy can only work if the network layer reacts very quickly to congestion, i.e. if the *congestion experienced* flag is set from the moment the traffic load increases a little.

- Why should one wait to the end of a sampling period before reacting to congestion ? If all systems keep on sending packets when congestion is experienced, the situation will get worse and eventually packets will be discarded. A good congestion management strategy should react immediately to congestion, i.e. before the network discards packets.

- If more than 50% of the received packets have the *congestion experienced* flag set, one should decrease the WR significantly. Multiplying WR by 0.875 is probably not enough.

  The WR should already be decreased if some of the packets have the *congestion experienced* flag set. One shouldn't wait to do this until more than 50% of the received packets have the *congestion experienced* flag set.

- The rules for the receiving transport entity specified in this paper differ significantly from those specified in the ATN Manual. Both should be evaluated.

5. Conclusion

The different congestion management techniques should be examined and tested. Proper values for the parameters should be searched for through examination. Combination of some of the techniques should also be considered. These studies should result in the rejection or adaptation of the techniques.

In a first stage simulation exercises will be performed to evaluate the strategies. The impact that one congestion management strategy has on another will also be investigated. The first results of the simulation exercises are expected by mid-July. In a later stage experiments could be performed.

Defect reports on the ATN Manual will be written. As a result of the simulation exercises and the experiments, propositions for the ATN Manual regarding congestion management will be made.

In order for a congestion management technique to be effective, it is important that every ES and IS co-operate together. It has no sense that one system reacts to congestion while other systems keep on sending packets into the congested network. Therefore the ATN Manual should oblige ESs and ISs to implement a congestion management strategy.

6. Recommendation

The intention of this document is to serve as an initial basis for a discussion on congestion management. The ATNP WG2 is recommended to:

- review this working paper and provide detailed written comments,
- communicate any results from existing work in the field of congestion management,
- to decide if it is necessary to include congestion management in Package 1,
- to consider the results of the simulation exercises to determine what SARPs or guidance material should be developed for Package 1.