Routing and Weak/Strong QoS in IDRP

Author: UK Expert

Date: 21 January, 1993

Clause 8 of DIS 10747 only permits an NPDU to be forwarded by a BIS when a route exists such that there is an exact correspondence between the set of distinguishing path attributes of the route and the QoS Maintenance, Security and Priority attributes contained in the NPDU. For Security and Priority, although other comments exist on the matching rules employed, the basic principle is correct. For example, an NPDU should only follow a route that provides the required degree of protection. However, in the case of QoS, ISO 8473 clause 6.16 requires a "best efforts" forwarding if the required QoS cannot be met.

The ISO 8473 requirement has been termed *weak QoS*, while the DIS 10747 provided service has been termed *strong QoS*. Ultimately, both are arguably legitimate approaches. However, a situation has arisen where a draft ISO standard seriously conflicts with an existing ISO standard which it aims to support. This is not an acceptable situation.

In the long term, ISO 8473 probably ought to be revised to enable a sender to express QoS requirements in either the *weak* or the *strong* sense. However, for DIS 10747 to progress is it essential that the text is modified to at least provide support for *weak* QoS.

It is understood that the *strong* QoS approach, developed for the DIS 10747 text, was intended to ensure that routing loops could not occur, and there is concern that introducing *weak* QoS could, in consequence, allow routing loops to exist.

This paper develops a *weak* QoS approach for ISO 10747, and in order to ensure that the approach is acceptable and indeed does not result in routing loops, the paper starts by analysing inter-domain routing, and attempts to establish the rules that need to be applied if *weak* QoS is to be supported. The conclusion is that *weak* QoS can be implemented by IDRP provided that there is a clear and unambiguous rule, implemented by all BISs, for choosing between routes with different *RIB-Atts* for the purposes of NPDU forwarding under *weak* QoS, and a proof is provided that this rule is sufficient to prevent routing loops.

1. Routing in IDRP

1.2 The "Domain of a Route"

It is first useful to introduce the concept of the "Domain of a Route", with *route* meant in the IDRP sense. This Domain is not the same as a Routing Domain, but is intended to comprise all Routing Domains (RDs) which have a route to a given destination with the same distinguishing path attributes (c.f. *RIB-Att*). This concept is illustrated in figure 1.

Figure 1 illustrates an example of the Domains of routes to a given Destination. The Destination is the RD that initially advertises a route to its adjacent RDs, and in this example, routes with three different *RIB-Atts* are advertised. This results in the Domains of three such routes (A, B and C), where all RDs in Domain A, say, have a route to the Destination with the *RIB-Att* corresponding to

Domain A. A fourth Domain also exists. This is the Domain of the default route which must encompass all other Domains of routes to the same Destination.

Two individual routes are illustrated together with the RDs they pass through (illustrated by • symbols). Route δ is an example of a default route. It thus leads from the boundary of the Default Domain (note that by definition, all routes start at the boundary of the corresponding Domain) to the Destination. Route α is an example of a route in the Domain of *RIB-Att* A. Routes α and δ also illustrate the fact that even though routes in different Domains may pass through the same RD, they do not then necessarily follow the same path.

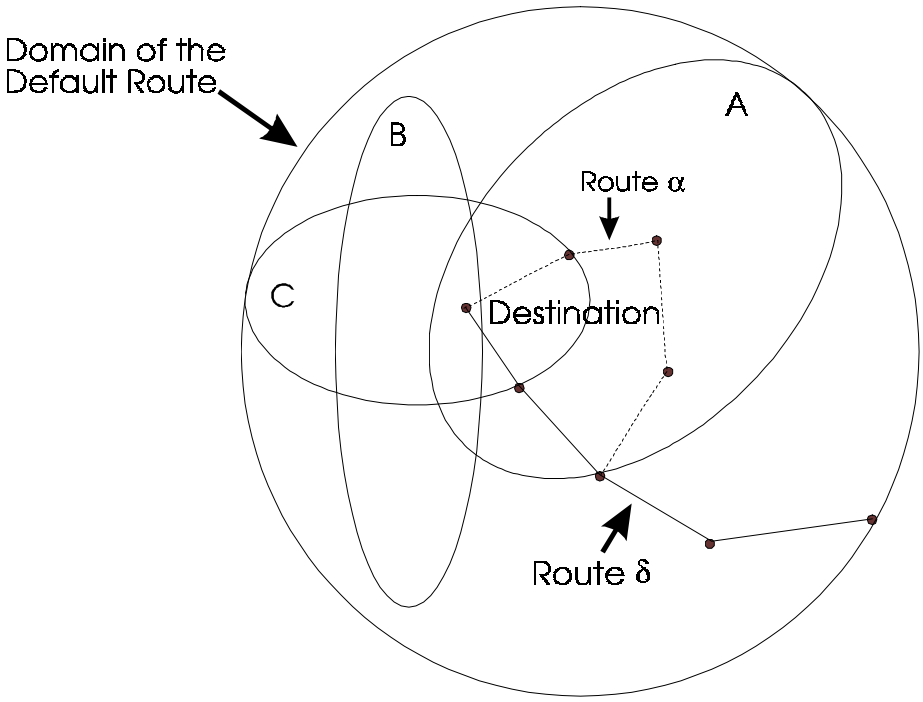

Figure 1 The Domains of Routes to a given Destination

Figure 1 implicitly assumes that routes do not loop, and it is now necessary to identify the conditions under which this is true.

1.2 The Propagation of non-looping Routes

It is relatively straightforward to establish that for a given *RIB-Att* routes do not loop.

Firstly, consider the initial propagation of a route from its Destination to the boundary of its Domain, and assume that this route will loop. At some point on the route - RD x say - the route must then be advertised back to an RD which it already passes through (this is a necessary requirement for it to loop). However, the RDI of RD x will already be present in the route's RD_PATH path attributes. It should thus never be advertised to RD x, and if it is then the receiving BIS should immediately reject it. Thus the routes within a given Domain do not loop as long as RD_PATH information is checked and acted on. Loop detection through the RD_PATH path attribute is thus both a necessary and sufficient condition to prevent an IDRP route looping.

In consequence, as long as an NPDU follows a route within a single Domain from the NPDU's source to the Destination, then the NPDU will reach that Destination without looping. This is the basis of *strong* QoS. The forwarding process attempts to identify a route which will be followed all the way to the destination, and if such a route cannot be found, the NPDU is discarded. This is true even when a route exists to the Destination, but for which it cannot be guaranteed that the NPDU will follow all the way to the Destination.

For example, in figure 1, if an NPDU originates in an RD on route δ before the route has entered Domain A, is addressed to the Destination of route δ, but has a QoS Maintenance requirement that corresponds to the *RIB-Att* for Domain A, then under a *strong* QoS forwarding regime, the NPDU is discarded rather than follow route δ. And the reason why it is not possible to guarantee that route δ will be followed all the way to the Destination is obvious. At the point of entry to Domain A, route α exists, and as this route provides an exact match of distinguishing path attributes for the NPDU's QoS Maintenance requirement, it would be chosen in preference to route δ, and the NPDU would follow that route to the Destination.

1.3 Changing Routes

However, it is also clear from figure 1 that in the above example, if the NPDU had been permitted to first follow route δ, and then change to route α it would have successfully reached its destination and would have done so within the requirements of ISO 8473 i.e. to perform "best efforts" forwarding. This is an example of *weak* QoS forwarding.

Clearly, while *strong* QoS forwarding is sufficient to ensure that an NPDU reaches its destination without looping, it is not a necessary condition. The question is: under what conditions does *weak* QoS forwarding also ensure that an NPDU reaches its destination without looping?

A routing loop cannot occur as long as an NPDU follows the same route. However, it can occur when an NPDU changes from one route to another, and then back again to the first route. Thus, for example a routing loop will occur under a *weak* QoS regime for a given NPDU when:

a. two routes (β and γ), say, exist to an NPDU's destination of which neither provides an exact match of distinguishing path attributes for the NPDU's QoS Maintenance requirement;

b. one of these routes first passes through RD y, say, and then through RD z, say, while the other passes first through RD z, and then through RD y;

c. for the NPDU, RD y chooses route β in preference to route γ, while RD z chooses route γ in preference to route β.

An NPDU that starts out from RD y will follow route β to RD z, then route γ to RD y, and so on, ad infinitum. The NPDU is caught in a routing loop. Clearly more complex examples involving multiple route changes are also possible, but the principles are the same.

As this example illustrates, *weak* QoS forwarding will break down when BISs in different RDs make incompatible "best efforts" decisions. The above case is a rather trivial example when two RDs make opposite choices. In more complex loops, RDs may be choosing between different pairs of routes.

This problem can be countered providing the standard specifies unambiguous rules for choosing a route under *weak* QoS forwarding i.e. choosing between routes with different *RIB-Atts*. Such rules must not only ensure that when two RDs are faced with a choice between the same pair of routes that they make the same decision, but that there is a strict ordering between all possible routes (i.e. ordering the *RIB-Att* combinations) so that more complex routing loops are also prevented. This ordering may be different for each possible QoS Maintenance Requirement, as an unambiguous choice is only necessary on a per NPDU basis.

It is actually straightforward to prove that this is a sufficient condition to avoid looping, using a proof by contradiction, assuming that the actual IDRP advertised routes do not themselves loop.

First assume that under a *weak* QoS regime, an NPDU has looped i.e. it has returned to an RD which it has already visited. For this to be possible, there must be a set of at least two IDRP advertised routes to the NPDU's destination, each with a different *RIB-Att.* This set of routes must form a complete path, starting at RD x, passing through the RDs that the NPDU has passed through

during its loop, and back again to RD x. A looping NPDU must have followed such a set of routes for the looping condition to occur. These routes may be referred to as:

$r_1, r_2, r_3.....r_n$

Furthermore, as these routes have been chosen under a *weak* QoS Regime with the ordering condition identified above, they must have an order of preference such that:

$$r_1 < r_2 < r_3.<.....< r_n \quad (1)$$

Where $r_i < r_j$ indicates that route r_j has a *RIB-Att* which is preferable to that of r_i for the NPDU's QoS requirements. Otherwise, the NPDU would not have changed routes.

However, for RD x to have originally chosen r_1, in preference to r_n, when the NPDU first passed through RD x (r_n must have been advertised by RD x and hence known to RD x), then r_1 must either be more preferable than r_n i.e.

$$r_1 > r_n \quad (2)$$

or identical to r_n i.e.

$$r_1 = r_n \quad (3)$$

Both (2) and (3) are contradictions of (1). Hence , such a set of routes is impossible, and a routing loop cannot occur under a *weak* QoS regime, as long as a strict ordering is maintained regarding the order of preference for selecting between routes with different *RIB-Atts,* for the purposes of choosing a route to meet an NPDU's QoS Maintenance requirements. As this contradiction is due to the application of the ordering condition, it must be both a necessary and a sufficient condition to prevent routing loops under a *weak* QoS regime, provided that none of $r_1, r_2, r_3.....r_n$ themselves do not loop. It is a necessary condition because without (1), the example of looping given earlier is not prevented. It is a sufficient condition because there is no need to assume any further constraints in order to demonstrate the contradiction.

1.4 Other Issues

1.4.1 Routing Stability

If a route is subject to rapid dynamic change, then even with a *strong* QoS regime problems can arise. Although the RD_PATH information ensures that a propagated route never loops on itself, the path that an NPDU follows may not be the same as that described by the route on which the NPDU started out, if new routing information is propagated while the NPDU is in transit. In the extreme condition, this may result in an NPDU visiting the same RD more than once. This is not a problem if this is a transient condition. However, if rapid change in the underlying routes does occur then it is possible for an NPDU to never reach its destination i.e. its lifetime expires before it gets there.

This situation can be avoided as long as the route update interval is much longer than the transit time of an NPDU. This is already a requirement of the DIS text.

Under a *weak* QoS regime, the same problem can be observed. Even with the ordering condition, if the underlying routes are subject to rapid dynamic change, then the contradiction observed above may not arise (i.e. (1) cannot be assumed), and hence an NPDU may be prevented from reaching its destination.

However, the existing requirement in the DIS text for a minimum route update interval also prevents problems in this case.

1.4.2 Route Aggregation

Routes can only be aggregated if they have the same distinguishing path attributes. Therefore even if a route r_i in (1) above was an aggregated route, or for part of the path traversed by the NPDU was an aggregated route, this would not affect the eventual contradiction, as the ordering requirement applies to *RIB-Atts* and not the values of individual path attributes.

1.5 Conclusion

The DIS text has been very concerned with the problems of choosing between routes with the same distinguishing path attributes, which is necessary for route advertisement, but has given relatively little attention to choosing between routes with different distinguishing path attributes, which is necessary to meet the ISO 8473 requirements for forwarding.

In order to make a consistent choice between routes with different distinguishing path attributes in support of *weak* QoS, it is necessary to specify a strict ordering between all legitimate combinations of QoS related distinguishing path attributes that may be present in a route. In fact, the DIS text does make a very limited statement of ordering, identifying one *RIB-Att* combination which is the most preferable for a given QoS Maintenance request, and the rest. This enables a *strong* QoS regime, but does not permit a *weak* QoS regime as the other combinations are un-ordered.

Additional ordering conditions need to be provided if *weak* QoS is to be implemented and hence for ISO 10747 to meet the requirements of ISO 8473.

In summary, the following conditions are necessary for inter-domain routing to function successfully without routing loops:

1. Suppression of looping routes using RD_PATH information

2. Route stability through minimum update intervals that are long compared with NPDU transit times

3. an unambiguous ordering of preference for selection of routes used for forwarding under *weak* QoS

4. Route Aggregation restricted to routes with the same *RIB-Att*.

1, 2, and 4 are already specified by the current text. 3 needs to be added.

2. Proposed Changes to DIS 10747

Two approaches to meeting the requirements of ISO 8473 have been identified:

1. In addition to the most preferable *RIB-Att* combination for a given QoS Maintenance request, the default route is specified as the second preference, and is hence used whenever the most preferable route is not available.

2. A complete ordering of *RIB-Att* combinations for each QoS Maintenance request is developed.

The first approach is attractive in that it enables the ISO 8473 requirement to be met with relatively little effort. However, it is difficult to claim that it is in the spirit of the user requirement. A user that states a QoS requirement of transit delay, say, in preference to expense, which in turn is in preference to residual error rate, does not have their requirement met, if a route is not available for a transit delay computation, but a route is available for an expense computation, while hop count (i.e. the default route) is used instead.

In practice, it is not that difficult to write down the an unambiguous ordering, as there are only a limited number of QoS related distinguishing path attributes that need to be considered.

Furthermore, the Globally Unique format of the QoS Maintenance parameter consists of only five active bits of information. The fifth bit is used to state a preference for sequence preservation versus transit delay, and the fourth to report Congestion Experienced. These can be ignored by the forwarding decision, as 10747 currently neither provides support for sequence preservation nor routing around points of congestion.

The remaining three bits are used to indicate various combinations of the following preferences:

- bit 3:transit delay vs cost
- bit 2: residual error probability vs transit delay
- bit 1: residual error probability vs cost

It is for each combination of the above, that an ordering needs to be established. There are thus only a few cases to consider and Table 1 provides a proposed ordering.

The derivation of table 1 is straightforward. The first preference column is equivalent to table 4 in the DIS text, and corresponds the *strong* QoS case. The lower preferences are then derived from the ordering present in the QoS request, followed by CAPACITY and finally the default (hop count). In two cases, the QoS request gives a circular statement of preferences, and in these case CAPACITY is thus taken as the first preference, on the grounds that the others are all equal.

3. Proposed Editorial Instructions

The changes to clause 8 of the DIS text to bring in weak QoS are reasonably straightforward. However, there needs to be a change of emphasis. The current text is based on the forwarding process first deriving a *RIB-Att* from the NPDU header information, and then finding a route with a corresponding *RIB-Att*. This approach needs to be changed to one of choosing the most appropriate route (c.f. *RIB-Att*) out of those available.

The proposed changes are as follows.

QoS Parameter	First Preference	Second Preference	Third Preference	Fourth Preference	Fifth Preference
11xxx111	RESIDUAL ERROR	TRANSIT DELAY	EXPENSE	CAPACITY	DEFAULT
11xxx110	CAPACITY	DEFAULT	-	-	-
11xxx101	TRANSIT DELAY	RESIDUAL ERROR	EXPENSE	CAPACITY	DEFAULT
11xxx100	TRANSIT DELAY	EXPENSE	RESIDUAL ERROR	CAPACITY	DEFAULT
11xxx011	RESIDUAL ERROR	EXPENSE	TRANSIT DELAY	CAPACITY	DEFAULT
11xxx010	EXPENSE	RESIDUAL ERROR	TRANSIT DELAY	CAPACITY	DEFAULT
11xxx001	CAPACITY	DEFAULT	-	-	-
11xxx000	EXPENSE	TRANSIT DELAY	RESIDUAL ERROR	CAPACITY	DEFAULT
No QoS Maintenance Parameter	DEFAULT	-	-	-	-
01000000	SOURCE SPECIFIC QOS	DEFAULT	-	-	-
10000000	DESTINATION SPECIFIC QOS	DEFAULT	-	-	-

Table 1 DEFAULT Order of Preference of QoS Path Attributes for use route selection

3.1 Table 4

Replace with table 1 in this paper.

3.2 Clause 8.2

Delete last bullet and replace with:

- the remaining NPDU-Derived Distinguishing Attribute is derived by decoding the first octet of the QoS Maintenance parameter as shown in table 4, using the first preference column.

3.3 Clause 8.3

Append the following new paragraph to this clause:

If no match is found, then the procedures of clause 8.2 are again used to determine the NPDU-derived Distinguishing Attributes except that when table 4 is applied, the second preference column is used instead, and the matching rule specified above are repeated. If again no match is found then the procedures of clause 8.2, are again repeated using the third preference column, and so until either a match is found, or until table 4 is exhausted, which indicates no possibility of a match being found.