**Notes on AODVv2 Draft dated December 30th 2014**

3 – PktSource should be the source address of the packet for which the destination address is unreachable (triggering a RERR)

5 – Could SMF interfere with the RREQ process by not forwarding along the best route, or in the worst case, not even forwarding along the only route?

6.4 – Why does each router need a unique sequence number?

6.4 – If the AODVv2 router’s SeqNum is lost, and it sets all its routes to Broken, does it send RERR messages? Router Clients would be among the Unreachable Addresses. Would it have SeqNums for those to include in RERR?

A RERR would force other routers using this next hop to immediately set the route as broken and stop using it. Otherwise if they tried to use it and reset the LastUsed time, does the MAX\_SEQNUM\_LIFETIME restart for that route? If so, the router with lost SeqNum will wait MAX\_SEQNUM\_LIFETIME before re-starting normal activity but the route could still exist on other nodes. If we send RERR and other routers set the route as Broken, is there a maximum time a route can stay broken before deletion, or will they still wait for MAX\_SEQNUM\_LIFETIME before deleting? This is irrelevant though, since the router with lost SeqNum would have to wait the MAX\_SEQNUM\_LIFETIME to be certain that all other routers would have deleted the route(s).

6.4 – Could the special sequence number of zero be used as a flag to reset any previous information, or delete any previous information, flooding this RERR through the network, prompting deletion at all nodes and prompting new requests to the router with lost SeqNum? May cause issues if certain nodes are currently out of reach and don’t receive that “reset” type message and still have Broken routes with newer SeqNum in their tables when they re-enter the network.

6.4 – If the AODVv2 router’s SeqNum is lost, why would it not forward RREQ or RREP? These don’t renew any information relating to the router’s own SeqNum. It should only refrain from sending anything with its own SeqNum, so that any routes back to its associated destinations from other routers end up timing out and being deleted. *The router can still forward data for other routers for destinations other than its own Router Clients, in which case the RERR should be sent as described.* This results in other AODVv2 routers updating the last used time on routes for other destinations even if this router is the next hop, but routes to destinations attached to this router will time out.

6.4 – If the AODVv2 router’s SeqNum is lost, why would it mark the routes it has learned so far as Broken? The SeqNum only relates to the information this router originates. It can safely forward to other destinations.

6.4 – If a router loses the SeqNum for its route to network A, then by all means set that route to Broken, but others are unaffected.

6.4 – If during the wait period, RREQs and RREPs are received, do they take the info from the RteMsg and adjust the pertinent route state from Broken, and is the router then allowed to use those routes and forward data on those routes? See also 7.2.

6.4 – After the wait period of MAX\_SEQNUM\_LIFETIME, does the router proactively do anything to re-advertise its routes? Does it store any RREQs it would have responded to if it wasn’t in the waiting period, and respond to them after the timeout? Otherwise, other routers could have sent RREQs which were ignored, and reached the maximum number of retries and be waiting the RREQ\_HOLDDOWN\_TIME before retrying, delaying the restart in connectivity.

6.5 – It isn’t clear if the protocol can run with multiple metric types in use simultaneously. In this case, when there are two routes to a destination, with different metric types, which route is preferred when forwarding data?

6.5.2 – The LoopFree paragraph should define clearly whether Cost(AdvRte) includes the cost of the link to the advertising router. It should also have clear notation. e.g.

“…comparing the cost of route update information, Cost(AdvRte), to the cost of an existing stored route, Cost(Route)…”

“…LoopFree(AdvRte, Route)…” replacing R1 with AdvRte and R2 with Route

“..for routes AdvRte and Route which use the same metric *type* to the same destination”

It could also give the LoopFree function which applies to the default metric type (HopCount).

6.5.2 – suggest that the loopfree function says advertised cost < route cost + link cost. Will work on examples….

6.5.4 – Move the bit from 16.4 about non-additive metrics to here. Maybe hint at the places where non-additive metrics are assumed. For example, the LoopFree function, the Cost calculation, the comparison in 7.1 “it is not stale and is less costly” …

6.6 – Simplify the first sentence to say

“Two incoming RREQ messages are considered to be “comparable” if they have the same OrigAddr, TargAddr, and Metric Type.”

This is also repeated in the middle of section 6.6 which could be removed.

The paragraph about multicast RREP should be moved to 14.4. Similarly the paragraph about RERR messages should be moved to 9.4.

7 – Just for complete clarity, rearrange “…including the Metric measured to the IP address in the route update from the node transmitting the AODVv2 message.”

7.1 – Leave out the “3” and just say HopCount – the IANA section will show that it is 3.

7.1 – The first bit of code should be moved to the end of section 7.1. The text which currently follows this looks like the start of the sentence might be missing.

7.1 – Make the definition of AdvRte.Cost more prominent and declare that this is true for additive metrics. Non-additive metrics are unsupported (as it says in 16.4, and relevant to 6.5.4)

7.1 – The paragraph and the pseudo-code and the bullet points make it unclear what should happen and in what order. Suggest:

Is it redundant? Redundant RREQ described in 6.6. Redundant RRE P (if using Multicast RREP) is described in 14.4. If redundant, ignore.

Is there a matching route entry, with address and metric type matching those in the route message? If not, create one, otherwise do the following comparison:

Is the information stale? If so, ignore.

*Is the information newer? If so – should it be used even if metric is lower? Without testing LoopFree?*

Is the advertised route guaranteed to be loop free? If not, ignore.

Is the advertised route better, i.e. is the Cost(AdvRte) less than the current route’s Metric? If not, ignore, unless the current route is broken, in which case the advertised route can be used to repair the current route. If the advertised cost is less, the advertised route offers improvement and should be used to update the current route.

Reword the pseudo-code: “LoopFree(AdvRte, Route) == FALSE” to make it easier to read.

The final bullet point should read “it is not stale and can safely repair a broken route (i.e. it is LoopFree).” It then matches the pseudo-code.

7.2 – The paragraph after the bullet list should mention that the route can be used, but only if the router is not in the lost sequence number wait period, or the boot period.

7.2 – Mentions validity time being given in RteMsg – doesn’t explain this anywhere

7.2 – Add further explanation to the statement “An updated route entry also fulfils any outstanding route discovery (RREQ) attempts for Route.Address.” i.e. Any retry timers for the RREQ can be cancelled.

7.3 – How does this relate to the ACTIVE\_INTERVAL, MAX\_IDLETIME, and the transitions from Active->Idle->Expired. If a route has an explicit lifetime set and transitions from Active to Expired, would a RERR be sent?

7.3 – Looks like a cut and paste error with the sentence about precursor lists.

7.4 – In the second paragraph, clarify that the forwarding route should be in Active or Idle state.

7.4 – The RREQ contains information to enable recipients to route packets back TOWARDS the OrigAddr, and similarly for RREP, to route TOWARDS the TargAddr. It only gives next hop information.

8 – This section could go before section 7.

8.3 – Is order of message contents important? Is this structure valid since RFC5444 will order it differently?

8.3 – If we find longest prefix match for a route to TargAddr which is no longer valid, would using its SeqNum affect any HandlingRtr’s route using the same prefix/overlapping/subnet/supernet of it? I.e. Does TargAddr information in an RREQ, besides the address itself, ever get used? Is there any point including more than the TargAddr itself?

8.4 – How does RREP\_Gen get a route about TargAddr from the RREQ?

8.4 – Point 1 relates to receiving the RREQ and should go in 8.5.

8.4 – In point 4, does setting the hop limit to be equal to the hop count of the RREQ force the return route to be the same?

8.4 – The RREP is unicast to OrigAddr – does this mean IP destination address is OrigAddr, surely it gets sent to next hop? See also 8.5.2. This also implies AODVv2 routers process packets that aren’t sent to their IP address? What is the destination MAC and destination IP?

8.5 – Does point 1 mean AODVv2 routers with bidirectional links? Do we need to keep a neighbour list? Or is it “if not in blacklist”…? Section 6.2 is Blacklists – not sure what the text is referring to in Section 5.

8.5 – Point 5 would be easier to read if it said “RteMsg.Metric + Cost(L) ≥ MAX\_METRIC[MetricType]”. Similar for point 3 in the second list.

8.5.1 – Move the sentence “Unless the router is prepared…” to the start of this section, and put a sentence like “Processing is as follows:” before the second bullet list. Remove “Otherwise” from the second point. On the fourth point, write “If TargNode is a Router Client…” and “…processing continues to forward the RREQ to other AODVv2 routers.”

8.5.1 – If the RREQ was unicast for some reason such as those in Section 5, and the receiving router has no route to the TargAddr, how can it send it to the next hop? Also on the final point, remove the “>” and say “SHOULD be multicast *to* the IP Address…”

8.5.1 – On receiving an RREQ, we update our route to the originator but we don’t do a sanity check that if the originator of the RREQ is the next hop we use to get to TargAddr, then the route is no longer available. Not likely to happen but it is a simple check that could be done? If not, sending packets on that route will result in a RERR coming back anyway.

8.6 – Consider rewording – rather than “we avoid sending redundant information by checking if we received redundant information” – its more that we notice incoming information is redundant and don’t process it, therefore avoiding sending redundant information.

8.6 – Could move this section to before 8.5.1?

9 – When a packet arrives and there is no route to the destination, is this strictly IP destination addresses, or does this include RREP messages where there is no route to OrigAddr? And what if a unicast RREQ is received but there is no route to TargAddr, would a RERR be sent backwards?

9.1 – Extra words “is marked has been marked” in first bullet point. Also if Expired it can’t be used for forwarding. In point two, this route should be expunged. In point three, the state should be changed to Expired. Also should mention that if it’s in Idle and we use the route to forward, the state should change to Active.

9.2 - This section should go with Section 6.2 about bi-directionality and blacklists. When the word “unreachable” is used, does this mean the link is no longer bidirectional? Or lost connectivity in one direction only?

9.3 – PktSource should be the source address of the packet which triggered the RERR.

9.3.1 – Again, does this apply to unicast RREQ if there’s no route to TargAddr?

9.3.1 – “…(or TargAddr in case the undeliverable…” should be “…(which is TargAddr if the undeliverable…” and a couple of words missing in “included in a SeqNum List data element in the RERR”.

9.3.1 – I don’t really understand the reasoning behind including the SeqNum List. Because if we don’t include that info, all nodes handling the RERR will assume their route through RRER\_Gen to the Unreachable Address is no longer valid – well it isn’t valid, else we wouldn’t be sending a RERR...

9.3.1 – Move the last sentence of the first paragraph, the one about discarding the packet that triggered the RERR, to a new paragraph at the end of this section. Insert this at the beginning of the paragraph about ICMP “Another indicator of an undeliverable packet is an ICMP Destination Unreachable packet”. Is this a function an AODVv2 router needs to handle though, or is it done automatically, considering the ICMP packet as any normal data packet? In which case, it should be enough to say “ICMP Destination Unreachable packets are forwarded in the same way as any network traffic and do not trigger an RRER message.”

9.3.2 – Does a link breaking also cover it becoming unidirectional? Move the sentence about marking routes that use the link as broken after the first paragraph (i.e. explain finding affected routes and putting them in AddressList first before marking them as broken.

9.3.2 – Route.Dest should be Route.Address.

9.3.2 – If there are affected routes with different MetricType, should a different RERR message be sent for each MetricType?

9.4 – This mentions “If HandlingRtr has neighbours that are using the affected routes…” without saying this is the optional precursor feature.

9.4 – If no route is found to an Unreachable Address in a received RERR, does that Unreachable Address get removed from any regenerated version of this RERR? What if a broken or expired route is found?

9.4 – A note on the SeqNum bit –the person who sent the RERR has lost the route, why does SeqNum matter? Also, since we only process if the message came from our next hop, how can we ever have a higher sequence number than our next hop?

9.4 – Note that if it doesn’t match ALL of these conditions, the processing for that Unreachable Address is done, and it shouldn’t be included in any regenerated RERR.

9.4 – Put the sentence about msg\_hop\_limit being zero at the end of the next paragraph.

10 – TLV definition looks like it should be in Section 2.

10 - Exactly how much could an RFC5444 parser adjust an AODVv2 message? Clearly the address list, prefix list and sequence number list need to remain present and intact. Would any changes made by the parser be reliably undone before passing the message to an AODVv2 router process?

11 – How do you configure an AODVv2 router as an IAR and what is classed as “the internet” – all networks except the configured one on the inside interface, or all public IP addresses (assuming a network within this range has not been used on the inside of this network?)

11 – When a packet arrives from the internet destined for a node in the AODVv2 MANET, and the IAR doesn’t have a route to it…a) how can it know if its inside the MANET if it doesn’t have a route, also b) how could it have advertised this to the internet so that the internet knew to direct the packet to the IAR? If it advertised the subnet (191.0.2.0/24), does it then need to RREQ when a packet arrives from the internet, in order to reach a single router in the network? Or if the dest IP packet is of a network behind a router in the MANET, does the IAR have to be configured with all these networks to advertise them to the internet too? What if one of these networks wants to communicate with something on one of the other networks, how does the IAR know not to try to respond if it thinks that address might be on “the internet”?

14 – Move description next to iRREP into section 14.2 Reporting multiple unreachable addresses – I thought this was already done?

14.3 – Says “more economical notifications” – suggest “more economical RERR notifications”

14.3.1 – Precursors are defined as any device that ever sent anything via this route, not just next hops, then you say the precursor list can hold next hops, and maybe more? If it holds more than just next hops, it will cause much more redundant traffic. Third paragraph can be simplified to avoid repeating section 9.3.

14.3.2 – Not clear that the precursor list exists for each route in the table. “…maintains a precursor table *for each route* and updates the table…”

14.4 – This section is not mentioned in the list in section 14. Should also note that this technique can be used to find the best return path rather than follow the same path as the RREQ took (section 6.6).

14.4 – suggest to remove the part of the sentence starting “that is, RREQ\_Gen…” up to and including “section 6.4, but” and replace it with “by”. Also remove the final sentence on the first paragraph – “Afterwards…” This implies that all implementations should be able to handle multiple RREPs arriving.

14.5 – If RREP\_Gen requires an ack, what happens if RREP receiver doesn’t support it? What happens if the receiver did support it but it got lost somehow? Would we decide the link was unidirectional and maybe delete the route to OrigAddr?

14.5 – When it says “Since the RREP\_ACK is simply echoed back …” implies it is a duplicate of something, maybe the RREP itself, but the packet format is different – could be displayed here instead of appendix B4.

14.5 – If the ACK is sent back from intermediate nodes, what does RREP\_Gen do? Does it expect an ack from the next hop only, the destination only, or all hops?

14.6 – When it says “Implementations MAY”, does this mean RFC5444 parser implementations?

15.2 – MAX\_HOPCOUNT is a constant? But it should depend on the network diameter so should it be configurable? In the table, write MAX\_METRIC[MetricType]. MAXTIME should reference 9.1. Maybe have a Reference column and use the Description column to briefly describe each item? e.g. For MAXTIME the description could be “Maximum expressible clock time, used for Route.ExpirationTime.” Use consistent approach in 15.1, 15.3, 15.4

17 – Seems like half a sentence “if the mobile nodes in the ad hoc network…” at the start of the final paragraph, the actual sentence is already in the paragraph above.

A – Missing algorithms: Fetch\_Route\_Table\_Entry(), Update\_Route\_Table\_Entry(), Create\_Route\_Table\_Entry(), LoopFree(), Consume\_RREP(), Receive\_RREP\_ACK(), Timeout\_RREQ().

A1 – Make a subheading A.1.1 for Process\_Routing\_Info(). The code here doesn’t mention LoopFree.

A1 – Maybe create more subheadings for LoopFree, fetch/update/create route table entry algorithms.

A.2.2 – A couple of steps occur before checking if the RREQ is redundant. Is this correct? Also doesn’t mention adding to RREQ table.

A.3.2 – Although multicast RREP is optional, this section could mention handling of a multicast RREP, checking if redundant.

A.3.2 – This section makes it look like all intermediate nodes will send an RREP\_ACK?

A.3.3 – Should mention checking if we have a route to OrigAddr, and maybe hint at the multicast option?

B – Suggest reordering. e.g. If first four paragraphs are labelled 1-4, figure 4 is labelled 5, and the next two paragraphs are 6 and 7, reorder to be 1 (3?) 6 7 5 4 2. 3 could go into the list in section B1 to explain why there’s no MetricType TLV. Same in B2

H – Replace ROUTER\_SEQNUM\_AGE\_MAX\_TIMEOUT with MAX\_SEQNUM\_LIFETIME. Doesn’t explain how the negotiation might occur for one router to know to stop advertising because another will start.