Correctness of Failures:-

Assumption 1: There is atleast one correct replica
Assumption 2: In all the failure cases we assume that at most t-1 servers are faulty

Hist(objId) : update request sequence
Pending(objId) : request set

Case 1: Failure of Head

1) Master removes the head H and makes H+, the successor of H, the new head.
This new head exists as per the above assumption 2.

2) Consistency with T1, T2 and/or T3 (Specifications in [vRS2004chain])

As H gets removed from the chain, the requests received by H and not yet forwarded to H+ are removed
from the Pending(objId). Removing a request from Pending(objID) is consistent with T2, so deleting H from the
chain is consistent with specification mentioned.

Case 2: Failure of Tail

1) Master removes the tail T from the chain and makes T-, the predecessor of T as the new Tail.
This new tail exists as per the above assumption 2.

2) Consistency with T1, T2 and/or T3 (Specifications in [vRS2004chain])

This change alters the value of both Pending(objId) and Hist(objId).
a) Pending(objId) decreases in size because
b) Hist(T, objId) <= Hist(T+, objId) (Update Propogation invariant as T- < T holds)
so changing the tail from T to T- increases the no of requests completed by tail
and thus reduce the number of requests pending in Pending(objId)
c) As per T3, the update requests completed by T- and not by T do not appear in Hist(objId)
because with T- as now the tail, Hist(objId) = Hist(T-, objId)

Thus the above transitions alter the chain in a manner consistent with repeated T3 transitions.

Case 3: Failure of Internal Servers

1) Faulty server S, gets removed from the chain and the master informs S+, successor of S and then S-, predecessor
of S about the new chain configuration.

2) Update Propogation invariant is preserved as S- which now connects to S+, first sends to S+, all those requests which are there in
Hist(S-, objId) and have not reached S+.

3) Each server i also maintains a list Sent(i) of update requests that i has forwarded to some successor but have not been
processed by the tail.

a) When server S forwards an update r for o : sent(S, o).add(r)
b) When tail receives update r : send ack(r) to T- (if any) after processing the update r
c) When S receives ack(r) for o: Sent(s, o).remove(r)
Send ack(r) to S- (if any)

Thus a request received by the tail must have been received by all of its predecessors so the Inprocess Requests Invariant is also satisfied.

Extending a chain:-

1) A new server T+ is added to the end of the chain.
2) Original Tail T initially forwards its history so that
Hist(T+, objId) = Hist(T, objId) and simultaneously processes requests from its predec
essors and updates Send(T) accordingly.
Therefore as Hist(T+, objId) <= Hist(T, objId) holds throughout this forwarding, the U
pdate Propogation Invariant is preserved.

As the inprocess invariant Hist(T, objId) = Hist(T+, objId) ^ Sent(T) is established T+ starts
serving a s the new tail.