Nalluri Ravi Teja
17XJ1A0528

FAULT TOLERANT BROADCAST OF ROUTING INFORMATION

ABSTRACT

In ARPANET or other networks that follow the routing scheme as ARPANET each node sends its link information to its neighbors using Link State Packet(LSP) . These Link State Packets are propagated throughout the network by using sequence numbers, age fields that help nodes to recognize the latest information. There might be some problems in delivering these Link State Packets due to the clock failures,network partitions,node restarts,database corruption throughout the network,malfunctioning of hardware,low probability events. There are some potential problems with the ARPANET scheme and there are solutions to improve its efficiency. This paper suggests fixes to problems like Immortal packets, premature aging,Negative ages ,eliminating corruption,old packets,RESTART TIME elimination,node restarts which helps enhance efficiency. The scheme proposed has advantages over the ARPANET scheme.

Keywords: ARPANET, RESTART TIME, Link State Packet, low probability events ,sequence numbers

INTRODUCTION

This paper introduces you how Link state packets are broadcasted.Each node in the network sends the information about its neighborhood to all nodes through link State Packets(LSP) by method called flooding.So each node in the network has the complete map of the network which it stores in its database. The database describes the graph of the internetwork.Since node has complete information about the whole network route,they can determine shortest routes using algorithms like Dijkstra's. If routing is to be done correctly,all nodes must agree upon the network map.There are some modifications to the ARPANET like schemes that can improve reliability, stability and decrease overhead. Background principles behind the scheme are:

1) The algorithm should stabilize the network in reasonable time to correct routes after failure of a node due to different causes without any human involvement. It means that an algorithm must recover once the malfunction ceases.

2)Low probability events occur in the network These low probability events that can cause malfunctioning are more likely to occur.These events should not cause permanent catastrophe. The network requires some time to recover from low probability events.

3)Timers must be avoided since they may fail after some years when the network evolves.For networks which timers cannot be avoided large safe space must be given because even though the network evolves the timers won't fail.

4)Nodes should not wait due to timeout after restart i.e correct operation should not be delayed due to timeout errors.

PROBLEM STATEMENT

An algorithm is proposed that helps in reliable transfer of information throughout the network. LSPs are broadcasted through neighbors to all nodes by means of flooding. The router issues link packets now and then and floods the packet throughout the network. So the old packets must be removed from the database of the nodes. Also there may be duplicates since the packet comes from different nodes to a particular node. So the nodes compare the received packet with the previously received packet. The nodes can be compared by timestamps generated by source using clocks. All nodes must agree upon a single clock so that they are synchronized which is not possible in all networks. Local clocks can be used but just the timestamp isn't enough to compare the packets. So a sequence number scheme is introduced. So when a packet is received by a node it compares the sequence number of packets and determines whether it is an old packet or new. But there are issues with this as well due to hardware fault,data corruption or reaching the maximum value. So to overcome this a circular sequence number space scheme is introduced.This scheme is also flawed. The sequence numbers can be corrupted by source or other nodes.So, a new parameter is added to the LSP called age field to compare the packets.

This new scheme has parameters:MAX_AGE,MAX_INT,RESTART TIME for smooth and reliable broadcasting of LSP. There are some problems in this type of ARPANET scheme as well. Immortal packets, Premature Aging,Old Packets, Negative ages, RESTART TIME,parameter settings,node restarts are the potential problems,The above scheme does not ensure the aging of packets.The parameter settings should be done carefully so that there are no transmission errors. Another problem with this scheme is old Link State Packets are still kept in the databases without propagating them.This means that nodes have different information about source nodes in their databases. If this information is used, then incorrect routing may occur. Premature aging is also a problem. A node might corrupt the packet and incorrectly set its to much lesser number thus resulting in expiration of the packet before making through the entire network. This results in disagreement of contents of source node among the network. There is a restriction that the node must wait some time called RESTART TIME upon restarting.Modifications are to be made to remove this restriction. Also there is a problem with restarting a node. For a period of time after the node restarts there is a possibility that packets are still around the network which leads to delay. Another potential problem with this scheme is fields being corrupted by nodes.

SOLUTION DESCRIPTION

GLOBALLY SYNCHRONIZED CLOCKS AND LOCAL CLOCKS

An efficient way of distributing link state packets or routing information is by intelligent flooding. It means a node recognizes whether a packet is duplicate or not. If it is a duplicate the node discards it. Else it will flood the packet. This scheme uses clocks to determine whether the received packet is older or newer to the packet stored in the database. One the solution for determining whether the packet is older or newer is by using **Globally Synchronized Clocks**. The LSPs are time stamped upon generation by the source node. This way it would be easy for other nodes to check if the packet is older or newer. The nodes check the timestamp and compare it with the previously received packet thus determining if the packet is duplicate or not. The node checks its database for packets that are very old and removes them from the database before a new stamp could look old because of wrap-around. This is effective even though the timestamp is corrupted due to hardware errors or other causes. If a node timestamps a packet incorrectly indicating the packet is created in the future, other nodes detect the problem and discard the packet. But most networks do not have globally synchronized clocks.

Another scheme to recognize recent packets is by using **local clocks.** Given the time increases even though the node is down,LSPs could be time stamped by the source node with the local clock. If a node is down for a long time, the timestamp could wrap around. When the node restarts, it broadcasts packets that could look old. Moreover, timestamp alone would not be enough to determine how long ago the packet was generated as timestamp is local and doesn't mean anything to other nodes in the network. Also any hardware fault or data corruption at the source or other nodes could cause a faulty timestamp. Therefore, there might be a wrap-around that results in generation of packets that could look old. Thus, a sequence number scheme is introduced to recognize packets.

SEQUENCE NUMBERS-CIRCULAR SEQUENCE NUMBER-LOLLIPOP SEQUENCE NUMBER

Each node maintains a **Sequence number** and increments the number each time a packet is generated. When a node receives this packet, it compares the sequence number with the last received packet's sequence number in the database. Though, this field is large enough there are chances of wrap around due to hardware faults or maliciously induced packets. Thus,there is a chance of getting maximum value. When the field reaches maximum value we can reset the nodes. Reset packets must be flooded to all nodes to ensure reset of all nodes.But old reset packets might emerge,resulting in a whole new reset operation. Moreover, if a node is isolated from the network during reset, it still contains packets with higher sequence numbers., so we cannot guarantee reset of all nodes. Due to these reasons, we must consider a sequence number space as **circular sequence number space.(fig 3)** To compare sequence numbers *a* and *b* in a circular space of size n, we follow some rules. a is more recent than b if and only if:

1) a< and b-a>n/2 or

2) a>b and a-b<n/2

However,Sequence number alone is not enough to determine which of the two packets are new due to some faults in the scheme. Suppose a node goes down and restarts. It doesn't know what sequence number it is using before going down. So source node packets won't be flooded until the sequence number goes past the older sequence number issued before the crash. The packets sequence number might be corrupted by other nodes as well as the source node. Due to hardware problems,the source node might generate a packet with a random sequence number. Some random node might also corrupt the packet by incrementing the value to be greater than true value.Therefore, the packet looks like an old packet until the sequence number is incremented past the corrupted value. Network partition is another problem.Suppose two nodes A and B are partitioned with node A in as one subnet and B as another subnet. During partition there is no exchange of information between the nodes. During partition time , suppose node A generates n/2 packets,received by node B. After the network is reformed the new LSPs generated by A looks old to node B. Therefore, it is necessary to add an agefield to the Link State Packet. The age field gives sufficient information to purge the old link state packet from the database before the source node's sequence number wraps around.

AGE FIELD RULES

When a LSP is generated, MAX AGE is assigned to the packet by the source.When this packet moves to another, the age is decremented according to the holding time at that node. When the age field decrements to 0, the packet is no longer propagated. The next link state packet must be issued before MAX INT time interval but after MIN INT interval This means MAX-INT, the maximum interval between creation of Link state Packets whereas MIN-INT means minimum interval between creation of two LSPs by source node. Also, the node must wait the RESTART-TIME interval for all its old packets to expire. A node must not issue more than n/2 packets within the MAX-AGE.This scheme also has some interesting dilemmas . Some modifications to this scheme would enhance the efficiency of ARPANET.

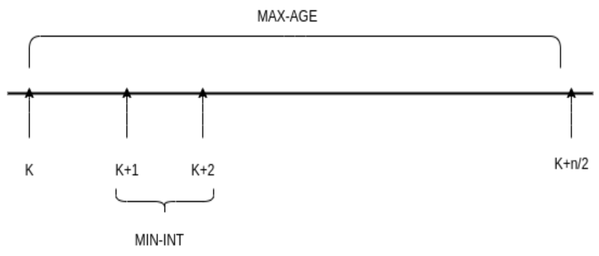


Fig 1: Representation of MAX-AGE and MIN-INT

PROBLEMS AND FIXES

The packets do not age if the packet is held less than a time period before being transmitted. This scheme doesn't ensure the aging of packets. If clock resolution is high or the packet is held less than a time period at a node, the age doesn't get decremented. This results in packets circulated indefinitely without increasing their ages. The solution is to make packets age. If the clock period is fine, then every node must decrease the age field. Otherwise, each node should decrement the age field with some random probability.But, in ARPANET this didn't follow this method. Instead, it took a lot of human effort to fix this problem. Another problem is choosing parameters. MAX-AGE must be larger than MAX-INT so that the new packet will reach all nodes before the old packet expires. Also, the propagation time might be longer due to queues or transmission errors.Further, packets can also age quickly due to distant nodes having faster-running clocks. So the difference between MAX-AGE and MAX-INT must be high. Also, if MAX-AGE is large to give scope for MAX-INT the old packets take longer to purge. So, this results in longer RESTART-TIME because a node is not allowed to restart until all its old LSPs are expired.

Therefore, MAX-AGE should be minimized so that the nodes restart in reasonable time. MAX-INT should maximize to control traffic. Maximize MAX-AGE minus MAX-INT to ensure LSPs are transmitted without disruption even though the old packet expires before the new packet reaches all parts of the network. The solution to this is elimination of RESTART-TIME. So,MAX-AGE can be set to the maximum amount considering the network is tolerable to low probability events. So this gives scope for MAX-INT to be as large as well. Thus,the amount of traffic is vastly reduced.

Another problem if a distant node corrupts the age field of a packet of source node or sets the age field to zero, the packet won't be able to make it throughout the network.The network disagrees with source's information until it issues a new packet. Any problem of this type is solved if the source generates new packets. One more solution to this is achieving synchronization. If any node reads the age of source packet as zero, it refloods the packets to its neighbours. When another node receives this packet with zero age, which has the same sequence number as the packet stored from the source, the node treats it as a new packet and floods it to its neighbours. Thus,all the nodes in the network are forced to age the packet to zero. If source node is active it recognises this and issues new LSP.

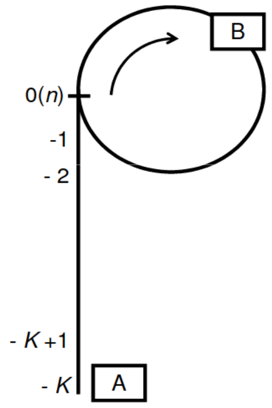
Another potential problem is that old LSPs are stored in the database but they are not propagated .Since information is not exchanged the nodes will have different information about the source,it can result in incorrect route calculation until a new Link State packet is received from the node,which may not occur because source is unreachable. Also, the information must be erased from the node because the node can issue random Link State Packets due to hardware faults. So,the packets must be removed from the database. The solution to this is to modify the synchronization policy. However, the node must discard the packet after MAX-AGE. So, to ensure reliable transmission of packets,if a packet expires while in memory, flood the packet to all neighbours with age zero.If source's Link State Packet is received with age zero, the nodes checks the sequence number with the sequence number of unexpired LSP from source. If it matches the node accepts the packet and floods it.Else the packet is ignored. If MAX-AGE is completed since the packet expired in memory,discard the packet.

If a Link State Packet is issued such that the age field is much larger than the MAX-AGE the packet doesn't not expire within the MAX-AGE. Here it is assumed that the MAX-AGE is considerably less than the maximum value that can fit in the age field. There are numerous solutions. The solution used in ARPANET is taking MAX-AGE value as large as possible that can fit in the age field.

The node must wait RESTART-TIME upon restarting in the ARPANET scheme. Some modifications to the scheme would eliminate this restriction. One of them is sending a stored LSP to the neighbour in response to receiving an older Link State Packet from the neighbour. If a node restarts and it chooses an older sequence number than the number it had before it stopped, then other nodes think of it as an old packet.So, the node must wait for all the old packets to time out. Even if the ARPANET design is fixed in a way that the age increases at each hop,it would still be intolerable for networks with large diameter and long delay lines because RESTART-TIME>MAX-INT+Propagation time+aging error. If MAX-INT,propagation time,aging error are large then RESTART-TIME must be large. A procedure to eliminate the RESTART-TIME is instead of discarding the older packets received at a node, the node should send back the stored packet across the link which the older packet was received. In this way,the source switches to a new sequence number that the network agrees. And again here the source node should not issue sequence numbers differing by n/2 or more within the MAX-TIME because they would be immortal packets. Thus the source node may make at most n/2 minus the number of LSPs it can issue within MAX-AGE jumps within the MAX-AGE. THis parameter is defined as J. J<n/2-(MAX-AGE/MIN-INT). So,in this way the old LSPs will die within MAX-AGE. If the last untime out packet has a sequence number s which is closer than J then the source can switch to s+1 and have its packets believed.

One more way to eliminate the RESTART-TIME is using lollipop-shaped sequence numbers(fig 2) instead of circular sequence numbers.Suppose,a node A(source node) restarts, it would be nice to begin with a number that is less than all other numbers. Neighbors will recognize this number for what it is, and if they have a pre-restart number b in their databases from node A, they can send that number to node A and node A will jump to that

sequence number. Node A might be able to send more than one Link State Packet before hearing about the sequence number it had been using before restarting. Therefore, it is important to have enough restart numbers so that A cannot use them all before neighbors either inform it of a previously used number or the previously used number ages out of all databases.Solution to this is lollipop sequence numbers starting from -k and increment to 0 and proceed to circular space. Here k must be larger than the no.of packets A can issue within MAX-AGE. As said,sequence number start at -k and proceed to circular space.Given two numbers *a* and *b* and a sequence number space *n*, *b* is more recent than *a* if and only if:
1)$a < 0$ and $a < b$, or
2)$a > 0$, $a < b$, and $(b - a) < n/2$, or
3)$a > 0$, $b > 0$, $a > b$, and $(a - b) > n/2$.



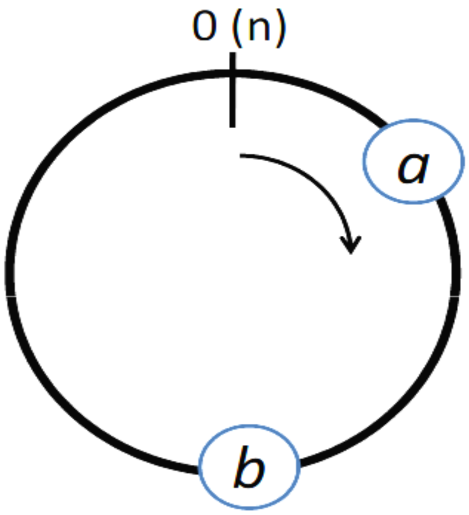


Fig 2:Lollipop sequence number Fig3: Circular sequence number

When the linear restart numbers have been used up, or after a neighbor has provided a sequence number to which A can jump, A enters a circular number space. So, the lollipop sequence ensures that the node need not wait MAX-AGE for all its old packets to expire. However,there are a couple of low probability events where the nodes have to wait MAX-AGE. When some node issues LSPs with the source ID A and all sequence numbers are in circular space the node has to wait MAX-AGE. Second one is a sequence of carefully timed restarts by A and network partitions.Node A is in the East and it issues a packet with sequence number 0. Then the network partitions. Meanwhile, A issues #b packets before MAX-AGE. So, East has sequence number b and west has sequence number 0 for A. A restart and network partition ends at same time. A is informed of West sequence number #0. So it issues a sequence number #1. The network partitions again. Now A is informed about East's sequence number #b.So,it issues a packet #b+1. And again it issues c packets before elapsing MAX-AGE. If this event repeats the gap between two unexpired packets increases.

When a node goes down and restarts the node needs to receive all the LSPs it has received before it goes down. In ARPANET the node waits 90 seconds assuming the other nodes revert with the LSPs. In this scheme the nodes need to issue LSPs frequently. Some networks find it excessive. A simple solution is to have a special packet to a neighbour to send the all the Link State Packets. Then a node will not wait upon restart. An alternative is to have a node A mark the node B as not having acknowledged any packets when the line B went down.When the line comes back it will be clogged for a CLOG TIME with routing messages while A send packet to every node in network to B.CLOG TIME isn't large when compared to RESTART TIME. It is much less than the time taken to propagate LSP through the network.
So CLOG TIME isn't much of a period as compared to RESTART TIME.

There is still a problem with restarting a node. When a node,there might be packets in the network with age less than MAX-AGE issued by the node before the restart.There are three cases here. First if the sequence number of

LSP generated by the node is greater than the sequence number before restart then new packets replace the old packets. Secondly,if sequence number is less than sequence number before restart then the node receives information from the old packet about sequence number and switches the packet number accordingly.If the sequence number is the same as sequence number before restart,there are two cases. First, node had time to issue only one packet before going down and upon restart it will start with the same number. Second one is if node issued two LSPs with the later packet not yet propagated through the net. So upon restart it is informed the former sequence number since later is not yet propagated to the other node. If a node uses the same sequence number that is already in the network the new packets are ignored . The solution to this is if the node's new LSP has the same sequence number to the one in the database then compare the data and inform the source node. But informing is not an easy task since routing to source node will not work as net disagrees with the LSP.The solution includes a confusion bit in the LSP. If a packet arrives at a distant node with sequence number n and the node already has a packet with the same sequence number and different data,confusion is set and floods the packet with newer sequence numbers. If a node receives a packet with the sequence number n which already has a packet with number n and confusion bit or if a node receives packet with number n+1 and confusion bit then the packet is treated as newer and flooded. To check if the packet is duplicate or not,compare the checksum of two packets instead of checking whole data.

One more potential problem is corruption of LSP fields by distant nodes. A data checksum must be added to the LSP which should be computed by source node. Corruption of the age field isn't a serious event. If it is considered a serious event,then the LSP must contain two copies of age fields with one being the 1's complement of the other. Corruption of confusion bit isn't considered a serious event unless it prevents source from receiving its own copy.If a node corrupts the confusion bit,the packet would propagate throughout the net eventually reaching the source which will issue a new Link State Packet. The corrupted packet won't reach the source if the source restarts with a sequence number still alive in the network or a node corrupts only confusion bit so that the checksum catches the problem. However, this is a low probability event that can wait MAX-INT. Also the source node must check the source ID in the packet it is about to issue against its ID copy stored elsewhere so that any hardware problems wouldn't change the source ID of the packet.

CONCLUSION

The scheme presented when compared to the ARPANET scheme has some advantages for distributing the routing information.It proposes a sequencing technique to handle updated flooding of link state packets. Unlike in ARPANET,nodes need not wait upon restart to issue packets. As said above, this scheme introduced the concept of sequence numbers, agefields, confusion bits which resulted in lesser reliance on timers or clocks. However, timers that exist are used for backup after low probability events. If any error conditions occur, which are less likely, are fixed by the network itself without any human involvement as soon as the offending software/hardware is repaired. This scheme enhances the robustness and efficiency of the network. It also ensures fast transmission of the Link State Packets.

FURTHER RESEARCH

Further research can be done on other alternatives for the lollipop sequence number system to eliminate RESTART-TIME. Also, research can be done on removing conflicting goals for choosing parameter values. Security is an area that can be looked into.

BIBLIOGRAPHY

https://book.systemsapproach.org/internetworking/routing.html
https://www.ccexpert.us/routing-tcp-ip/lollipopshaped-sequence-number-spaces.html
https://www.ciscopress.com/articles/article.asp?p=24090&seqNum=4
https://www.geeksforgeeks.org/unicast-routing-link-state-routing/