*i*Journals: International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN-2347-4890
Volume 9 Issue 3 March 2021

# *The Solution to the Problem of Count-to-Infinity in Network Routing*

**Author: Shin-Jye Lee[1]; Ching-Hsun Tseng[2]; Hsueh-Cheng Liu[3]**

*Affiliation: Institute of Management of Technology, National Chiao Tung University, Hinschu[1,3], Taiwan[1]; Department of Computer Science, The University of Manchester, Manchester, UK[2]*

*E-mail: camhero@gamil.com[1]; hank131415go61@gmail.com[2]; stdm11528@gmail.com[3]*

**DOI: 10.26821/IJSHRE.9.3.2021.9318**

## ABSTRACT

*Routing algorithms are worked between routers to determine paths and maintain routing tables. Once the path is determined, routers can route routing protocols with other routers in the whole topology. As a whole, a proper routing protocol cannot only reduce the loading of the traffic in the Network Layer, but also can improve the performance of the entire network. In this work, to fulfil the aforementioned goal, we propose a topology which combines algorithms and protocols into a one integrated solution to fix count-to-infinity problem in network routing.*

**Keywords: Count-to-Infinity Problem, Network Routing**

## 1. INTRODUCTION

Distance Vector Routing Algorithm, one of dynamical routing algorithms, is prevalent in the network routing. The features of Distance Vector Routing Algorithm [1,2] do not bring convenient and fast network routing dynamically, but also carry the drawback of inconsistent routing unfortunately, likes the Count-to-Infinity Problem [1,2]. Also, the primary focus of this proposal is to work out the Count-to-Infinity Problem.

## 2. Distance Vector Routing Algorithm

Distance Vector Routing algorithms operate by having each router maintains a table (routing table) giving the best-known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbors. Totally, the features of Distance Vector Routing Algorithms are under below:

- Each router maintains a routing table, which contains the preferred route, together with the corresponding distance, to each of other routers in the subnet.
- Each router knows the "distance" to each of its neighbors.
- Each router periodically updates the routing table by using the information from neighbors.
- Fast response to good news, but leisurely to bad news.

**Fast response to good news:** It will add a length of hop per exchange. As a whole subnet, the longest path is the length of N hops, it means that all routers will receive the newest routes via N exchanges only.

**Leisurely to bad news:**No router ever has a value more than one higher than the minimum of all its neighbors. Gradually, all the routers work their way up to infinity. However, the number of exchanges required depends on the definition of the value used about infinity.

## 3. The Count-to-Infinity Problem

In the situation of inconsistent routing, the routing protocol continually increases its metric or distance, and route packets back and forth between the devices, despite the fundamental fact that the destination network is down. Most important of all, whilst the routing protocol counts to infinity, the invalid information allows a routing loop to exist.

For example, a topology comprises three routers, and they are Router A, Router B and Router C, as illustrated in Figure 5-1. Router A is connected with Router B by Router A's interface S0 and Router B's interface S0; Router B is connected with Router C by Router B's interface S1 and

Router C's interface S0. Router C's interface E0 is connected with network 10.4.0.0. When network 10.4.0.0 is down, Router C detects the failure and stops routing packets out its E0 interface. However, Router A and Router B have not yet received notification of the failure. Router A still believes it can access 10.4.0.0 through Router B. Router A's routing table still reflects a path to network 10.4.0.0 with a distance of 2. Because Router B's routing table indicates a path to network 10.4.0.0. Router C believes it has a viable path to network 10.4.0.0 through Router B. Meanwhile, Router C updates its routing table to reflect a path to network 10.4.0.0 with a hop count of 2. Router B receives a new update from Router C. Router A receives the new routing table from Router B and detects the modified distance vector to network 10.4.0.0 and recalculates its own distance vector to 10.4.0.0. Because Routers A, B and C conclude that the best path to network 10.4.0.0 is through each other, packets destined to network 10.4.0.0 continue to bounce between these three routers. Final, the invalid updates about network 10.4.0.0 continue to loop, which results from the routers update with each other inappropriately.

## 4. The Solution to the Count-to-Infinity Problem

### 4.1 Maximum Metric Settings

A router reduces the Time-To-Live (TTL) value [2] by at least 1 each time it gets the packet. The router discards that packet, if the Time-To-Live value becomes 0. Meanwhile, the network is considered unreachable. Distance Vector Routing protocols define infinity as some maximum number, and this number refers to a routing metric [2]. The metrics that routers most commonly use are as follows:

- Hop count – Numbers of routers through which a packet will pass.
- Cost – Arbitrary value, usually based on bandwidth, money expense, or another measurement, that can be assigned by a network administrator.
- Bandwidth – Data capacity of a link.
- Delay – Length of time required to move a packet from source to destination.
- Load – Amount of activity on a network resource.
- Reliability – The bit-error rate of each network link.
- Maximum Transfer Unit (MTU) – The maximum frame length in octets. The upper bound on segment size.

In Distance Vector Routing Protocols, there are two protocols are commonly used, including RIP-1 and IGRP. RIP-1 computes the metric by hop count, and IGRP computes the metric by bandwidth, delay, load, reliability, and MTU.

### 4.2 Split Horizon Algorithm

In Split Horizon Algorithm [1,2], sending information about a route back in the direction from which the original information comes is never useful.

For example, the topology as illustrated in Figure 5-1, Router B tells Router C the route about the distance to network 10.4.0.0. It doesn't make sense for Router B to announce to Router C that Router B has access to network 10.4.0.0, because Router B has to access to network 10.4.0.0 through Router C. Therefore, in Split Horizon algorithm, Router B cannot send information to Router C, and Router C cannot receive information from Router B. On the other hand, it's never useful to send information about a route back in the direction from which the original information came.

### 4.3 Route Poisoning

Route Poisoning [2] attempts to improve convergence time and eliminate routing loops caused by inconsistent updates. When a router loses a link, the router advertises the distance of routes that have gone down to infinity.

For example, the topology as illustrated in Figure 5-1, when network 10.4.0.0 goes down, Router C poisons its link to network 10.4.0.0 by entering a table entry for that link as having infinite metrics. Poisoning its route to network 10.4.0.0 makes Router C not be able to being influenced by other incorrect updates about network 10.4.0.0.

### 4.4 Route Poison Reverse

After Route Poisoning [2] works, Route Poisoning allows the receiving router to advertise a route back toward the source with a metric higher than the



 

maximum. It seems to violate Split Horizon Algorithm, but it lets the router know that the update about the down network was received.

For example, the topology as illustrated in Figure 5-1, at the beginning, Router C poisons its link to network 10.4.0.0 and sends an update (usually triggered update [2]) to Router B. Then Router B starts the Holddown Timer [2]. After the Holddown Timer expires, a new route with a better metric doesn't still arrive. Router B states network 10.4.0.0 is inaccessible and sends information back to Router C to inform Router C that it gets network 10.4.0.0 is unreachable. This is a specific circumstance overriding split horizon, occurring to make sure that all routers on that segment have received information about the poisoned route.

## 4.5 Holddown Timers

Holddown Timers prevent regular update messages from inappropriately reinstating a route that might have gone bad. Holddown Timers tell routers to hold any changes that might affect routers for some period of time. The holddown period is usually calculated to be just greater than the period of time necessary to update the entire network with a routing change. Meanwhile the router keeps an entry for the network's possible down state, allowing time for other routers to recompute for this topology change. By the way, Holddown Timers work as follows:

1. A previously accessible network is down, the router marks the route as inaccessible and starts a holddown timer.

2. During the period of Holddown Timer, if an update with a better metric than the original record arrives, the router marks the network as accessible and ends the holddown timer. Otherwise, any update with a poor metric than the original record is supposed to be ignored.

3. If an update with a better metric still not arrives after the holddown timer expires, there are two ways: one is the original route would be removed at the situation of no update arriving during the period of Holddown Timer; the other is chosen the better update arriving during the period of Holddown Timer.

During the period of Holddown Timer, routes appear in the routing table as "possibly down".

## 4.6 Triggered Updates

The Triggered Update is a new routing table that is sent immediately, in response to a change. The detecting router immediately sends an update message to adjacent routers, which, in turn, generate triggered updates notifying their adjacent neighbors of the change.

## 4.7 The Solution to work out the Count-to-Infinity problem

First, we structured a topology, as illustrated in the following:


**Fig. 1**

We integrate these algorithms and protocols into an integral concept.

1. In routers, Split Horizon Algorithm is a default setting and supreme than other protocols except Poison Reverse.

2. When the metrics is grater than Maximum Metric Settings (RIP is 16, and IGRP is 255), the Link is being considered unreachable. For example, the metrics to network 10.4.0.0 exceeds the maximum number, and then network 10.4.0.0 is considered unreachable.

3. In the Split Horizon rule, sending information about a route back in the direction from which the original update came is never useful. It means that the router cannot receive information from the adjacent router with a lower metrics than itself. For example, Router C cannot receive information form Router B, and Router B cannot send information to Router C, too.

4. Then, in the Route Poisoning rules, the router poisons its link to the unreachable network. For example, Router C poisons its link to network 10.4.0.0 by entering a table entry for that link as infinity.

5. Meanwhile, Router C send a trigger update to Router B, which informs Router B that network 10.4.0.0 is unreachable.



 

6. In Holddown Timers, after Router B receives the triggered update form Router C, Router B starts holddown timers.

7. In Poison Reverse, in case the update with a better metric is not still available before holddown timers expired. Poison Reverse works, the router will tell the router C that it gets the network 10.4.0.0 is unreachable and poisons its link to network 10.4.0.0. Then, Router B informs Router A by sending Triggered Updates that network 10.4.0.0 is unreachable, and Router A starts Holddown Timers.

8. When the 10.4.0.0 network comes back up, Router C sends a Triggered Update to Router B, notifying it that the link is up. After Holddown Timer expires, Router B adds route 10.4.4.0 back to the routing table as accessible.

9. Then, Router B sends a Triggered Update to Router A, notifying it that the link is up. After Holddown Timer expires, Router A adds route 10.4.4.0 back to the routing table as accessible.

10. All networks link up.

## 5. CONCLUSION

By proposed solution, it integrated every possible aspect toward count-to-infinity problem. From our observation, the solution can reduce the loading of the traffic in the Network Layer, but also can improve the performance of the entire network.

## 6. REFERENCES

[1] Andrew S. Tanenbaum, "Computer Networks", third edition, Prentice Hall, 1966.

[2] Steve McQuery, "CCNA Self-Study: Interconnecting Cisco Network Devices (ICND), Second Edition, Cisco Press, 2003.



 