On the Convergence of Distance Vector Routing Protocols

Dejan Spasov, Marjan Gushev

Faculty of Computer Science and Engineering
Skopje, Macedonia
{dejan.spasov, marjan.gushev}@finki.ukim.mk

Abstract. In this paper we give an overview of distance vector routing protocols. We focus on the convergence mechanisms in two widely known distance vector routing protocols: EIGRP and RIP. With the aim to provide open source protocol, we propose a solution that inherits the simplicity of the RIP protocol and the fast convergence of the EIGRP protocol. We believe that our proposal will provide faster convergence and better scalability in large networks.

1 Introduction

In a computer network, it is vital to know the shortest paths between each pair of nodes (routers), because shortest pats are preferred choice for directing the flow of end-user traffic. In the early networking days, network administrators were manually configuring routes that were under their administrative domain. However it became obvious that this approach did not scale well and it was prone to errors. As the number of nodes in computer networks grew linearly, the number of links among the nodes grew with quadratic speed. Hence it became impossible for administrators to catch up on such a growth, i.e. to maintain best routes, to keep second the best routes as back-ups, and so on.

In the early ‘80s, routing protocols started to emerge on the commercial routers. A routing protocol is a network protocol that implements graph-based algorithm for finding shortest paths to distant networks. In addition, routing protocols specify message format and communication procedures that will allow them to share information about the remote networks. Routing protocols determine the best path to each network which is then added to the routing table. Most often, it is considered that routing protocols operate at layer three of the OSI model, with the exception of the IS-IS protocol which operates at layer two.

Internet can be seen as interconnection of separate routing domains or autonomous systems. This formulation divides routing protocols in two categories:

- *Interior gateway routing protocols* (IGPs) – protocols used for intra-domain routing

adfa, p. 1, 2012.

- *Exterior gateway routing protocols* (EGPs) –routing protocols used for routing between autonomous systems ([1]-[3]).

Interior Gateway Protocols exchange routing information within a single routing domain. Prominent members of IGP family are: OSPF, EIGRP, and RIP routing protocols. Considering the type of the shortest-path algorithm they use, these protocols are further subdivided in two categories:

- *Distance vector* routing protocols (EIGRP and RIP)
- *Link-state* routing protocols (OSPF)

In distance vector routing protocols routes to distant networks are advertised as vectors (objects with distance and direction). A *metric* must be defined within these protocols and the distance is measured according to this metric. The direction represents the neighbor router along the path to the advertised distant network. Well-known example of the algorithm for finding best routes in distance vector protocols is the Bellman-Ford algorithm. Early distance vector routing protocols were designed to periodically send their complete routing table to all neighbors. This approach guaranteed consistent routing information among all routers in a network, but did not scale well for large networks [1]-[3].

Link-state routing protocols need to have a complete view of the topology before applying the Dijkstra's shortest path algorithm. Thus the first step for link-state routers is to exchange information about the topology. In contrast to early distance-vector protocols, link-state protocols offered faster convergence with almost zero control traffic taxing the network. However, with the advent of the EIGRP it has been shown that distance vector protocols can maintain fast convergence with the same amount of control traffic as link-state protocols [1]-[3].

A disadvantage of link-state routing protocols is that if a link goes down then entire network will be down for the time the re-computation of shortest paths takes place. This can be alleviated with dividing the entire routing domain in sub-domains – but this step requires a knowledgeable administrator and more configuration commands on routers. In distance vector networks if a link goes down, only the routes that were going through that link will be unavailable for the time the re-computation takes place. Another disadvantage of link-state routing protocols is that they require more processor time than distance vector protocols.

In this paper we analyze the metric and convergence mechanisms of distance vector routing protocols. We say that a network has *converged* if all routers have complete and accurate knowledge about the network. Our goal is to propose a new routing protocol that is based on the RIP protocol.

2 Distance vector routing protocols

First we will illustrate the differences between distance vector and link-state routing protocols. Imagine a road infrastructure of a country, but without accompanying guide lines or information signs. How would a driver know to drive from city A to city B? An obvious solution is the government to install guide and information signs. This solution represents distance-vector routing protocols. Another solution is in each car the government to install GPS navigating device. This solution represents link-state protocols. The question that arises is which approach is better? It is obvious that GPS solution is more expensive, but doable; though twenty years ago this would have been impossible task.

The following example demonstrates disadvantages of link-state protocols: Assume that a distant road, (not on the route from A to B) is under construction. Then imagine that all GPS devices will be updating the topology change for 24 hours. In addition, imagine that each 60 days all GPS devices will not be working for 24 hours due to maintenance reasons.

The following example demonstrates disadvantages of distance vector routing protocols: let A' and B' are two neighboring cities on the road between A and B. Let assume that the road between A' and B' is closed for repair. Then a driver will be driving a car in a loop around city A and its neighboring cities.

When we speak about computer networks, we use the term *autonomous system* to refer to a collection of routers interconnected with links and operated by single administrative authority. End-users, or hosts, usually are not considered in the network model. *Routers* are special-built computers with the ability to find the shortest path to each network in the entire domain. Shortest paths to all networks in the autonomous system are kept in the fast memory of the router as a special data structure known as *routing table*. For each network in the autonomous system there is only one entry in the routing table usually composed of: the IP network address, metric, next-hop router, exit interface and expiration timer.

The existence of shortest paths implies that there must be a metric by which routes will be measured and compared. Simple metrics are based on hop count, or the number of transiting routers, while more complex metrics include bandwidth and delay in their calculations, even the waiting times in router's ques. Usually, a concrete metric value is referred as *cost* – which is a term from weighted graphs. Let $d(i,j)$ represents the cost between edges i and j. We will assume

$$d(i,j)=\begin{cases}\infty & i \text{ and } j \text{ are not adjacent}\\ \in \mathrm{N} & i \text{ and } j \quad \text{are} \quad \text{adjacent}\end{cases} \tag{1}$$

Assuming that costs are additive, the best metric between any two nodes $D(i,j)$ can be found by finding the minimum

$$D(i,j)=\min_{k\in N}\{d(i,k)+D(k,j)\} \tag{2}$$

where N represents all routers and $D(i,i)=\infty$. It has been proved that procedure (2) will lead to shortest paths and several algorithms have been designed according to this procedure [4].

The theory of shortest paths on graphs, though useful in finding the shortest routes, does not solve all problems that may show up in reality. For example, networks have frequent changes in topology due to router failure or network maintenance. The mechanism that distance vector protocols use against router crashes is route timing out. For example, in RIP routing protocol the time out mechanism is set to 180 seconds. If a router does not get an update message that a certain route is alive for 180 seconds, it declares that route unreachable. If a network becomes unreachable, the nearest router upon noticing this will advertise that network as unreachable. Each distance vector protocol has reserved a special *infinity-metric* value for unreachable destinations [4].

The above procedure equipped with route time out and infinity metric will always converge to appropriate shortest paths for each router. However, we did not mention the time needed for routers in a network to converge to shortest paths list. For example, consider a simple network of four routers (Fig. 1) and assume that all routers are in a state of consistency, i.e. routers A and B know that to get to the server S their packets must pass through C [4].

Now assume that connection between C and D fails. With the help of timeout timers, C will notice that D is unreachable, but meanwhile A and B will falsely advertise a route to D through themselves. C will accept this false advertised route with bigger metric and it will advertise back to A and B a slower and unreachable route. This process of mutual deception, known as “counting to infinity”, will continue until infinity metric has been reached. In RIP, for example, hop count is used as metric and the infinity metric is represented by the number 16. Route time-out timers are set to 180 seconds. This means that the convergence process of the network on Fig. 1 will last unacceptable 48 minutes.

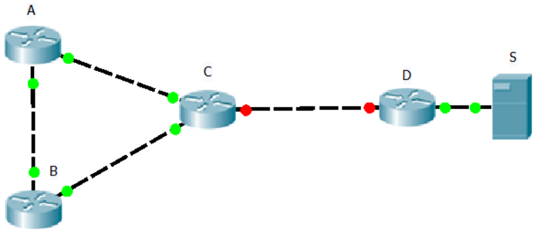

Figure 1. Example of routing instability.

Several mechanisms have been proposed to speed up the convergence of the distance vector protocols. *Split horizon* rule forbids sending back routes to the neighbor from which these routes have been learned. *Split horizon with poisoned reverse* mechanism will advertise back these routes, but with infinite metric, thus improving the convergence. *Triggered updates* is a rule that requires routers to send update messages immediately when they notice a change of metric in their routing tables. The receiving routers will change their metric if their routes were through the sending router and will trigger update message to their neighbors [4].

3 Routing Information Protocol

Routing Information Protocol (RIP) is one of the oldest and still alive routing protocols. Its development began in the late '70s from the Xerox's XNS protocol. The first document that describes RIP was published in 1988 [5], however recent RFC extensions that were proposed to support IPv6 [6] and cryptographic authentications [7] secured its future existence.

RIP metric is an integer between 1 and 15, with 16 being reserved for infinity. The way the costs for traversing networks are associated is not specified in the standard, but due to the limit of 15, the cost is usually 1. This is the well known *hop-count* metric used by RIP.

RIP packets are encapsulated in UDP segments before being sent over IP network. RIP configured routers send and receive RIP packets on port 520. RIP packet format is given on figure 2 [4]:

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  command (1)  |  version (1)  |       must be zero (2)        |
+---------------+---------------+-------------------------------+
|                                                               |
|                         RIP Entry (20)                        |
|                                                               |
+---------------+---------------+---------------+---------------+
```

Figure 2. RIP packet format.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Address Family Identifier (2) |        Route Tag (2)          |
+-------------------------------+-------------------------------+
|                         IP Address (4)                        |
+---------------------------------------------------------------+
|                         Subnet Mask (4)                       |
+---------------------------------------------------------------+
|                         Next Hop (4)                          |
+---------------------------------------------------------------+
|                         Metric (4)                            |
+---------------------------------------------------------------+
```

Figure 3. RIP entry.

We can notice that RIP packets are aligned on 32 bit boundaries. Version field (fig. 2) helps to distinguish between RIP version 1 and RIP version 2 packets. The command field defines two types of messages:

1. *Request* – from a neighbor router to send all or part of the routing table
2. *Response* – from the neighbor router with all or part of the routing table.

Each RIP packet (fig. 2) can carry information for up to 25 routes. Parameters requested or sent back for one route are carried with one RIP entry (fig.3). Response packets can be generated for three reasons: response to request packet, regular update or triggered update. Every 30 seconds each router will send its routing table to every neighbor with response packets. In order to avoid synchronization and unnecessary collisions over broadcast networks, each 30 second interval is jittered with a small random time less than 5 seconds. Triggered updates can over flood the network. Thus after a triggered update is sent, a timer is set for a random time less than 5 seconds. If other trigger events occur before the timer expiration, a single update is sent after the timer expires. The timer is then reset to another random value between 1 and 5 seconds [4].

With each route, RIP process on a router associates two timers: the *time out* timer (as described in previous section) and *garbage collector* timer. Once a route enters the routing table, the timeout timer is set to 180 seconds and reset each time if an update for the route is received. If the timer expires, the garbage collector timer is set to 120 seconds and the route is considered unreachable. The route will remain in the routing table for the duration of the garbage collector, but it will be advertised as unreachable. After the garbage collector expires, the route is removed from routing table [4].

It is obvious that if we increase infinity in the RIP protocol, we will create more space for manipulating route costs. However, this will create backward compatibility problem and will confuse older versions of RIP. The best thing we can hope is that older versions will ignore routes with costs greater than infinity. Thus, the committee responsible for maintaining the RIP standard remained adamant to demands for increasing the infinity value.

4 Enhanced Interior Gateway Protocol

Enhanced Interior Gateway Routing Protocol (EIGRP) is a CISCO proprietary distance vector routing protocol that was developed to address shortcomings of the RIP protocol, like the hop-count as a metric, maximum network diameter of 15, and the periodic broadcasts of the entire routing table [1]-[3].

The proprietary part of EIGRP is protected with *Protocol Dependent Modules* (PDM) and *Reliable Transport Protocol* (RTP). Protocol Dependent Modules gave to EIGRP capability to operate over various Layer 3 network protocols: IPv4, IPX, and AppleTalk, while RTP provided connectionless and connection oriented services over these networks. In other words, RTP offers TCP-like and UDP-like services to EIGRP that do not depend on the protocol stack.

EIGRP defines four packet types needed for its communication: Hello, Update, Acknowledgement, Query, and Reply [1]-[3].

The first thing an EIGRP router must do is to establish adjacency with its neighbors. This is done with the help of *Hello* packets and this is lifelong adjacency. Hello packets are usually exchanged over 5 second intervals.

The next step for neighbor routers is to exchange routing information. This is done with *Update* and *Acknowledgement* messages. This communication is connection oriented thus eliminating the need for periodic route refreshment and route timeout timers. A new Update message for a particular route is sent only if the metric for that route changes. EIGRP uses the term partial and bounded to describe Update messages. Partial refers to the fact that only routes with changed metric are included in the update, and the term bounded refers to the fact that updates are sent only to those routers affected by the change.

If a route becomes unavailable, *Query*, and *Reply* messages are used in the search for alternative routes. Again, these two types of messages are sent in connection-oriented manner accompanied with Acknowledgement message.

EIGRP uses the most complex metric of all routing protocols. It can be made of four parameters: bandwidth, delay, reliability, and load. Reliability and load are dynamic parameters measured at each interface, but they are seldom used in calculations. Thus most often the well-known *bandwidth + delay* formula is used in metric calculations. Let $L(R,D)$ be a route from the router R to a destination D. Let $L(R,D)$ be made of links l_i with bandwidth w_i measured in bps and delay d_i. Then the EIGRP's bandwidth+delay metric for $L(R,D)$ is computed according to the formula

$$metric\{L(R,D)\} = \frac{256 \cdot 10^7}{\min\{w_i\}} + \frac{256}{10} \cdot \sum d_i \tag{3}$$

The bandwidth is usually specified on the interface by the producer. Cisco's defaults are 100 Mbps for LAN interfaces and 1.544 Mbps for WAN interfaces. Default delays on Cisco's routers are given on the following table:

Media	Delay
100 M ATM, Fast Ethernet, FDDI,	100 µs
T1, 512K, DSO, 56K, 1HSSI	20 000 µs

Table 1. Default delays on Cisco routers.

EIGRP uses *Diffusing Update Algorithm* (DUAL) to perform the shortest path computation. Although, it is still a distance vector protocol, it is advanced version that is supposed to be better than the Bellman-Ford algorithm that is used by RIP.

In order to explain how it works, we have to explain the terms used by DUAL (all terms refer to one destination):

1. *Successor* – this is the next-hop neighbor on the route to a destination network;
2. *Feasible Distance* (FD) – is the best (lowest) metric to the destination network;
3. *Feasible Successor* (FS) – is a neighbor who has a loop-free backup route, should any router on the best route fails;
4. *Reported Distance* (RD) – is the feasible distance to the destination network of the neighboring routers

Feasibility Condition (FC) – is a criterion based on which backup loop-free routes to destination network are found. EIGRP's DUAL algorithm maintains a *topology table* separate from the routing table. The topology table includes the best path to a destination network and backup path (via the Feasible Successor) that DUAL has found to be loop-free. In order a neighbour to qualify for Feasible Successor, it has to pass the Feasibility Condition:

$$FD > RD \tag{4}$$

In [8] it has been proved that if (4) holds true, that neighbor has loop-free path to the destination network (a path that does not pass through the router that performs the feasibility test.

The EIGRP protocol explained so far, though better than rip, will not scale well on large networks. Thus several patches have been proposed that improve the convergence time in large networks. Stuck-in-active, stub router, graceful shutdown, graceful restart, multiple AS support, and so on.

5 Analysis of the EIGRP Protocol

EIGRP's superiority has been attributed to the use of the DUAL algorithm, while other distance vector protocols use inferior Bellman-Ford or Ford-Fulkerson algorithms. We believe that main advantage of EIGRP over RIP is EIGRP's metric. Using (3) EIGRP is capable of finding faster routes than RIP. However, we believe that this metric does not always find shortest paths. Consider the network on figure 4.

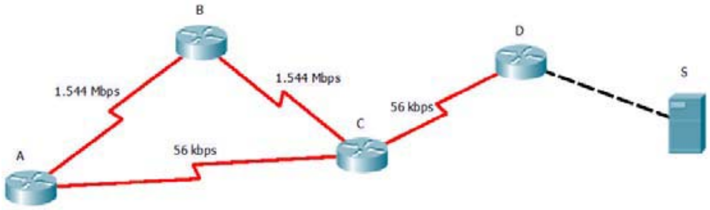

Figure 4. Shortest path demonstration on EIGRP network.

The first thing an administrator should do to ensure proper operation of the EIGRP protocol is to set appropriate bandwidth values on each router's interface. However, if default values for the delay are used, then the shortest path from the router A to the server S would be

`A->C->D->S`

On this example intuitively it is clear that the shortest path is `A->B->C->D->S`. The reason for error in EIGRP's computations is the default value for delay over serial links. The default delay value is same for T1 and 56k links (table 1). Some scholarly papers [10] suggest that we use propagation delay in (3). We believe that this approach is too expensive and again will lead the protocol to wrong conclusions. The best option is to use the serialization delay in (3), thus (3) will become

$$metric\{L(R,D)\} \sim \sum \frac{1}{w_i} . \tag{5}$$

This result is proportional to the metric of the OSPF link-state routing protocol ([1] ch. 11).

It is frequently advertised that one of the main advantages of EIGRP over RIP is that EIGRP maintains a topology table in which it stores backup routes. These backup routes are used if the best route fails, thus reducing the convergence time. We believe that this mechanism of EIGRP is not very useful. Consider the example on figure 5.

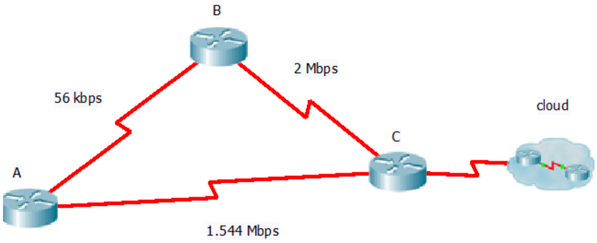

Figure 5. Backup route demonstration on EIGRP network.

In this example, only router A will have a backup route through router B to cloud networks. If the link A-C fails, router A will use its backup route through the link A-B. If the link B-C fails, then B engages in regular search for new route and eventually will find that it can use the link A-B. If any link in the cloud fails, then backup link may contribute to longer convergence, by installing wrong paths. Eventually, this problem will be solved with split horizon and triggered updates mechanisms.

To summarize, with Feasibility Condition (4) only a fraction of the routers in a network will have backup routes in their topology tables. Failure of even smaller fraction of links in a network will trigger proper usage of the EIGRP's backup route mechanism.

Finally, we would like to emphasize that EIGRP does not use periodic updates for disseminating routing updates, but a sophisticated system of queries and replies. This mechanism provides faster convergence in case a route becomes unavailable.

6 New RIP-like Routing Protocol

The purpose of this paper was to analyze the convergence mechanisms of the EIGRP protocol and to propose new RIP-like routing protocol. With this proposal we are proposing a new protocol that is not backward compatible with previous RIP protocols. Our work is based on the implementation of the RIP protocol in the open-source Qugga routing software [11] .

The first upgrade that was done was to improve RIP's metric. Our solution is to introduce additional field in RIP entries (fig. 6). From previous chapter we concluded that it would be simpler and more efficient if we use OSPF's metric instead of EIGRP's.

The second step in building a better routing protocol is to eliminate the dependence on periodic broadcast updates and route aging timers. These mechanisms provide reliable but slow convergence. In order to achieve faster and reliable convergence, we have to use the TCP protocol for the RIP packets (fig.2). If a route becomes unreachable neighbor routers will be queried for alternative routes. This implies that stuck in active timers must be implemented. In TCP usage requires periodic UDP broadcast to router's neighbors to establish associativity. We propose this messages to include the number of entries in the routing table. If two neighbors disagree on this number will prompt routers to exchange their routing tables.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Address Family Identifier     |        Route Tag              |
+-------------------------------+-------------------------------+
|                         IP Address                            |
+---------------------------------------------------------------+
|                         Subnet Mask                           |
+---------------------------------------------------------------+
|                         Next Hop                              |
+---------------------------------------------------------------+
|                         Hop Count                             |
+---------------------------------------------------------------+
|                         OSPF Metric                           |
+---------------------------------------------------------------+
```

Figure 6. Modified RIP entry.

The final step will be to implement a system for finding and keeping backup routes. Here we propose more powerful feasibility condition

$$FD \geq RD \tag{6}$$

This condition is better because we require the feasible distance (FD) to be greater than or equal to the reported distance (RD). In addition we apply (6) over two metrics: hop count and OSPF metric. With our proposal more backup routes will be found. For example, with (6) router B (fig. 5) will have a backup route through router A, and vice versa. Thus we increase the number of backup routes in a computer network.

References

1. Cisco Networking Academy: CCNA Exploration 4.0 - Routing Protocols and Concepts.
2. Cisco Networking Academy: CCNP Advanced Routing Protocols.
3. Graziani, R.: Lecture notes on CCNA exploration and CCNP courses. Available at: http://www.cabrillo.edu/~rgraziani.
4. Malkin G.: RIP version 2. RFC 2453, IETF standard, November 1998.
5. Hedrick C.: Routing Information Protocol. RFC 1088, IETF standard, June 1988.
6. Malkin, G., Minnear, R.: RIPng for IPv6. RFC 2080, IETF standard, January, 1997.
7. Atkinson, R., Fanto, M.: RIPv2 Cryptographic Authentication. RFC 4822, IETF standard, February, 2007.
8. Garcia-Lunes-Aceves, J. J.: Loop-Free Routing Using Diffusing Computations. *IEEE/ACM Transactions on Networking*, vol. 1, no. 1, February 1993.
9. Albrightson, B., Garcia-Lunes-Aceves, J. J., Boyle, J.: EIGRP – A Fast Routing Protocol Based on Distance Vectors.
10. Yee, J. R.: On the Internet Routing Protocol EIGRP: Is it Optimal? In: IFORS, 2005.
11. http://www.nongnu.org/quagga