University of Moratuwa

Department of Electronic and Telecommunication Engineering

EN2150

Communication Network Engineering

Routing Protocol Design

July 1, 2025

220197E Gunathilaka K.L
220200K Gunawardana I.M.P.T
220257N Jayasekara S.P.R
220405T Munavvar M.A.A

Contents

1 Abstract

This document provides details about a novel routing protocol designed to mitigate the persisting issues of the existing routing protocols. In the first part, the overview of current routing protocols RIP, OSPF, IS-IS & BGP is given along with the weaknesses of each routing protocol. Examples for those weaknesses are also provided for clear understanding. Then, the overview of the newly designed protocol **Enhanced Distance Vector Routing Protocol (EDVRP)** is given. This includes the routing table structure, message formats as well as the pseudocodes and flowcharts of the algorithms. This section also includes drawbacks of EDVRP as well as an example scenario of the protocol for better understanding. Then the simulation results of EDVRP is included along with the security and scalability analysis.

2 Abbreviations

RFC: Request for Comments

VLSM: Variable-Length Subnet Masking

ISDN: Integrated Services Digital Network

VoIP: Voice over Internet Protocol

LSDB: Link-State Database

ABRs: Area Border Routers

ASBRs: Autonomous System Boundary Routers

LSA: Link-State Advertisement

LSPs: Link-State Protocol Data Units

ISP: Internet Service Provider

CLNP: Connectionless Network Protocol

PDU: Protocol Data Unit

NBMA: Non-Broadcast Multi-Access

ATM: Asynchronous Transfer Mode

AS: Autonomous System

VPN: Virtual Private Network

EBGP: External Border Gateway Protocol

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

3 Literature Review

3.1 Routing Information Protocol

3.1.1 Overview

Routing Information Protocol (RIP) is a distance-vector routing protocol used to determine the best path for data packets in IP networks. It is one of the oldest and simplest routing protocols. It is designed for small to medium-sized networks. RIP enables routers to exchange routing information periodically, building and maintaining routing tables to forward packets efficiently.

3.1.2 Features

- Type: Distance-vector protocol, where routers share their entire routing table with neighbors.
- Metric: Uses hop count as the metric for path selection (each router in the path counts as one hop). The maximum hop count is 15, with 16 considered unreachable (infinity).
- Update Mechanism: Routers broadcast or multicast routing updates every 30 seconds, regardless of topology changes.
- Transport: Operates over UDP, using port 520.
- Versions:
  - RIPv1 (RFC 1058, 1988): Basic version, supports classful addressing, no subnet masks.
  - RIPv2 (RFC 2453, 1998): Adds support for classless addressing (VLSM), authentication, and multicast updates.
  - RIPng: An extension for IPv6 networks (RFC 2080, 1997).
- Use Case: Primarily used in small, simple networks due to its ease of configuration.

3.1.3 Weaknesses & Examples

1. 15-Hop Limit

Explanation: RIP uses a simple distance-vector routing metric: hop count. Any destination more than 15 hops away is considered unreachable. This limitation severely restricts RIP's scalability.

Example: In a campus network with 20 routers connected linearly, if a router tries to send traffic to a node 16 hops away, RIP will drop the packet. Even though a physical path exists, RIP's hop limit results in connectivity loss.

2. Counting to Infinity Problem

Explanation: RIP suffers from a distance-vector issue known as "counting to infinity". The routers increment hop counts indefinitely when a link fails, until they reach the maximum (16 = infinity). This causes very slow convergence after failures.

Example: When a router in a mesh network goes down, neighbors continue to advertise it with increasing hop counts until they stop at 16. This process delays route withdrawal and prolongs network instability, sometimes for minutes.

3. Fixed Metrics (Hop Count Only)

Explanation: RIP (Routing Information Protocol) uses a very simple routing metric. It is the number of hops, or routers a packet must pass through, to reach a destination. Each hop is treated equally — regardless of the actual quality or capacity of the links involved. This means RIP does not consider important factors such as Bandwidth, Latency, Link reliability and Congestion levels. As a result, RIP often makes suboptimal routing decisions. For instance, it may choose a single-hop path over a low-speed or heavily congested link, while ignoring a faster, more reliable multi-hop route

Example: Given two paths:

- Path A: 1 hop over a 64 kbps ISDN line
- Path B: 3 hops over 1 Gbps Ethernet

RIP would always select Path A, despite its inferior performance.

4. Slow Convergence

Explanation: IP uses a periodic update mechanism. It sends routing information every 30 seconds rather than reacting immediately to topology changes. If a route becomes unreachable, it isn't removed from the routing table until a 180-second timeout elapses. The router may continue using the invalid or outdated route during this time. This results in slow convergence, meaning the network can take minutes to adapt to changes like link failures. This is a serious issue for time-sensitive applications such as VoIP or online gaming.

Example: In a small office network with VoIP phones, if the main internet link goes down, RIP may take several minutes to update all routers. During this delay, calls drop or become garbled due to black-holed or bouncing routes.

3.2 Open Shortest Path First (OSPF) Protocol

3.2.1 Overview

OSPF is a link-state routing protocol that enables routers within an Autonomous System (AS) to exchange topology information and compute the shortest paths to destinations. It uses Dijkstra's Shortest Path First algorithm. Unlike RIP, which uses hop count, OSPF employs a cost metric based on link bandwidth or administrator-assigned values. This allows more flexible and efficient routing. OSPF is designed for scalability, supporting large networks with fast convergence and hierarchical routing through areas. It is used as an Interior Gateway Protocol (IGP) in IP networks.

3.2.2 Features

- Link-State Database: Each router maintains a Link-State Database (LSDB) containing information about the network topology, which is represented as Link-State Advertisements (LSAs). LSAs describe routers, links, and their states (e.g., cost, status).

- Hierarchical Routing: OSPF organizes networks into areas to reduce routing overhead. The backbone area (Area 0) connects all other areas, ensuring inter-area communication. Areas reduce the scope of LSA flooding, improving scalability.
- Cost-Based Metric: The metric is the sum of link costs, typically inversely proportional to bandwidth (e.g., a 100 Mbps link has a lower cost than a 10 Mbps link). Administrators can customize costs.
- Fast Convergence: OSPF detects topology changes quickly (via Hello packets) and floods updates, enabling rapid recalculation of routes using the SPF algorithm.
- Support for VLSM and CIDR: OSPF includes subnet masks in LSAs, supporting Variable-Length Subnet Masking (VLSM) and Classless Inter-Domain Routing (CIDR).

3.2.3 Weaknesses & Examples

1. High Resource Usage (CPU and Memory)

Explanation: OSPF's link-state approach requires maintaining a complete LSDB and running Dijkstra's algorithm for each area. This is computationally intensive. It consumes significant CPU and memory resources in large networks with many routers or frequent topology changes. The LSDB grows with the number of routers and links, and SPF calculations are triggered by any topology change.

Example: In a data center with 100+ routers and frequent link oscillations (due to hardware issues), OSPF's repeated SPF calculations overload router CPUs, leading to delayed convergence and packet loss for critical applications like database transactions.

2.Configuration Complexity

Explanation: OSPF's hierarchical structure (areas, ABRs, ASBRs) and parameters (e.g., area types, authentication, timers) require careful configuration. Misconfigurations, such as mismatched area IDs or incorrect stub area settings, can disrupt routing. Errors in configuration can lead to routing failures, blackholes, or suboptimal paths, requiring skilled administrators.

Example: In an enterprise network, an administrator configures a stub area with an incorrect area ID, causing routers to reject LSAs and block external routes. This results in loss of internet connectivity for a branch office.

3. Limited Scalability in Very Large Networks

Explanation: Although OSPF is more scalable than RIP, its LSDB size grows with the number of routers, links, and external routes. In very large networks, the LSDB can become unmanageable, and flooding LSAs across areas consumes bandwidth. Area partitioning is essential for scalability, but managing many areas increases complexity. In massive networks (e.g., global ISPs), OSPF requires careful design to avoid performance degradation.

Example: A global enterprise with 500+ routers in a single area experiences slow convergence and high bandwidth usage due to a large LSDB. This requires reconfiguration into multiple areas to restore performance.

4. Dependency on Area 0 (Backbone Area

Explanation: OSPF requires all areas to connect to the backbone area (Area 0), creating a single point of failure. If Area 0 becomes unstable or partitioned, inter-area routing fails. The

RFC states that all inter-area traffic passes through Area 0.Network designs must ensure Area 0's reliability, limiting flexibility in some topologies.

Example: In a multi-site enterprise, a failure in the backbone area (e.g., due to a core router crash) isolates branch office areas, preventing communication between sites until Area 0 is restored.

3.3 Integrated Intermediate System to Intermediate System (IS-IS) Protocol

3.3.1 Overview

IS-IS is a link-state routing protocol similar to OSPF, designed for robust and scalable routing in large networks. It uses the Dijkstra Shortest Path First algorithm to compute optimal paths based on a cost metric. IS-IS operates within a single AS. It exchanges topology information via Link-State Protocol Data Units (LSPs) to build a Link-State Database (LSDB). It supports hierarchical routing with Level 1 (intra-area) and Level 2 (inter-area) routing, which makes it suitable for enterprise networks, ISPs, and service provider backbones.

3.3.2 Features

- Link-State Database: Each router maintains an LSDB containing LSPs that describe the network topology, including routers, links, and their states (e.g., cost, neighbors).
- Hierarchical Routing:
  - Level 1 (L1): Routing within an area, where routers maintain an LSDB for their area.
  - Level 2 (L2): Moves traffic between areas; conceptually like OSPF's backbone, but not restricted to Area 0.
  - Level 1-2 (L1/L2): Routers that participate in both levels, facilitating inter-area communication.
- Cost-Based Metric: The default metric is 10 for all links, but administrators can assign custom costs (up to 63 in RFC 1195, extended to $2^{24} - 1$ in wide metrics per RFC 5308). Unlike OSPF, IS-IS does not link costs to bandwidth by default.
- Protocol Flexibility: IS-IS supports multiple protocols (CLNP, IPv4, IPv6) using Type-Length-Value (TLV) fields, which allow adaptabiliy.
- Neighbor Relationships: Routers form adjacencies using Hello PDUs (sent every 10 seconds by default), synchronizing LSDBs via Complete Sequence Number PDUs (CSNPs) and Partial Sequence Number PDUs (PSNPs).
- IPv6 Support: RFC 5308 extends IS-IS to support IPv6 routing by introducing new TLVs (e.g., IPv6 Reachability TLV) and multi-topology routing to handle IPv4 and IPv6 separately.

3.3.3 Weaknesses & Examples

1. Configuration Complexity

Explanation: IS-IS requires careful configuration of parameters like System IDs, Area Addresses, and Level types (L1, L2, L1/L2). Misconfigurations, such as mismatched Area Addresses or incorrect Level settings, can prevent adjacencies or cause routing failures. RFC 1195 emphasizes the need for consistent area configurations, and RFC 5308 adds complexity with multi-topology routing for IPv6 .

Example: In an ISP network, an administrator configures a router as L1 instead of L1/L2, preventing it from advertising inter-area routes. This isolates a regional office, blocking access to central servers until the configuration is corrected.

2. Limited Support for Non-Broadcast Networks

Explanation: IS-IS has limited support for non-broadcast multi-access (NBMA) networks (Frame Relay, ATM) compared to OSPF. RFC 1195 specifies point-to-point and broadcast networks but requires manual configuration for NBMA, lacking OSPF's robust NBMA support. This complicates deployment in legacy environments.

Example: A legacy enterprise network using Frame Relay configures IS-IS on an NBMA network, requiring manual neighbor definitions. A missed neighbor configuration prevents adjacency formation, disrupting connectivity to a branch office.

3. Security Vulnerabilities

Explanation: IS-IS supports clear-text and HMAC-MD5 authentication. But clear-text is insecure, and MD5 is vulnerable to modern attacks (e.g., brute-forcing). RFC 1195 specifies authentication but notes it is optional and RFC 5308 references RFC 5310 for stronger cryptography, which may not be universally implemented. Without authentication, IS-IS is susceptible to spoofed LSPs. Malicious LSPs can create routing loops or redirect traffic, compromising network integrity.

Example: In a financial institution's network lacking MD5 authentication, an attacker exploits the vulnerability of IS-IS by injecting false Link-State Protocol Data Units (LSPs), falsely advertising optimal routes to a malicious router. This misdirection causes sensitive financial data, such as customer transactions, to be redirected to the attacker's router, which compromises confidentiality. The absence of cryptographic authentication, as noted in RFC 1195, enables this data breach. This could potentially lead to significant financial and reputation damage.

4. Dependence on Level 2 Backbone

Explanation: Similar to OSPF's Area 0, IS-IS requires a contiguous Level 2 backbone for inter-area routing. If the L2 backbone is partitioned (e.g., due to a router failure), inter-area communication fails. RFC 1195 requires L2 connectivity for inter-area routing. Network designs must ensure L2 backbone reliability, limiting flexibility in some topologies.

Example: In a multi-site enterprise, a failure in a core Level 2 (L2) IS-IS router partitions the backbone. This disrupts inter-area routing and isolates regional areas from each other, as specified in RFC 1195. This disconnection prevents access to centralized Customer Relationship Management (CRM) systems, halting critical business operations until the L2 backbone is restored.

3.4 Border Gateway Protocol (BGP)

3.4.1 Overview

BGP is a path-vector routing protocol designed to exchange Network Layer Reachability Information (NLRI) between ASes, enabling inter-domain routing across the Internet. Unlike Interior Gateway Protocols (IGPs) like OSPF or IS-IS, which operate within a single AS, BGP facilitates routing between different ASes. These can be managed by ISPs or large enterprises. BGP maintains a table of paths (AS paths) to destinations, allowing routers to construct a graph of AS connectivity, prune routing loops, and enforce policy-based routing decisions. It supports both External BGP (EBGP) for inter-AS routing and Internal BGP (IBGP) for intra-AS routing.

3.4.2 Features

- Path-Vector Protocol: BGP maintains a table of AS paths (sequences of AS numbers) to reach destinations, included in UPDATE messages. This allows loop prevention by rejecting paths containing the local AS number.
- NLRI: BGP advertises reachability as IP prefixes (e.g., 192.168.1.0/24), supporting Classless Inter-Domain Routing (CIDR) to enable efficient address aggregation.
- Policy-Based Routing: BGP allows administrators to enforce policies using attributes like AS_PATH, LOCAL_PREF, MED, and COMMUNITY to influence route selection. : BGP uses TCP (port 179) for reliable communication, leveraging TCP's flow control to handle large routing tables.
- Graceful Restart: Allows a BGP speaker to maintain forwarding during restarts, configurable as disabled, helper, or full mode.
- Scalability Features:
  - Route Reflection: Reduces IBGP full-mesh requirements by allowing a router to reflect routes to other IBGP peers.
  - Confederations: Divides an AS into sub-ASes to improve scalability.
  - Route Target Filtering: Limits VPN route distribution (Juniper Documentation, Section on Route Target Filtering).

3.4.3 Weaknesses & Examples

1. Security Vulnerabilities

Explanation: BGP lacks inherent security mechanisms, relying on TCP for transport without mandatory authentication. RFC 4271 supports optional MD5 authentication, but it is vulnerable to modern attacks (e.g., brute-forcing) and not universally implemented. Malicious UPDATE messages can advertise false routes, leading to traffic interception or blackholing. Attackers can hijack prefixes, redirect traffic, or cause outages, compromising network integrity.

Example: In a financial institution's multi-homed network, an attacker injects a false BGP UPDATE advertising a shorter AS path for the bank's prefix (e.g., 192.168.1.0/24). This redirects customer transactions to a malicious AS, enabling data theft until the ISP filters the invalid route.

2. Complex Configuration and Management

Explanation: BGP's flexibility (e.g., policy-based routing, route reflection) requires complex configuration of parameters like AS numbers, peer groups, and policies. Misconfigurations can lead to route leaks, loops, or session failures. RFC 4271 emphasizes the need for careful policy design. Errors cause routing disruptions or suboptimal paths, requiring skilled administrators.
Example: An enterprise configures an incorrect AS number in an EBGP session, causing the session to fail. This isolates a branch office from the Internet, disrupting remote access to cloud services until the configuration is corrected.

3. Slow Convergence

Explanation: BGP's convergence can be slow. Especially in large networks, due to the time required to propagate and process UPDATE messages across ASes. The Minimum Route Advertisement Interval (MRAI) (30 seconds default for EBGP) delays updates, and route flapping can trigger persistent oscillations. Delays in convergence disrupt real-time applications like VoIP or streaming.

Example: In an ISP network, a link failure triggers a cascade of BGP UPDATE messages across 10 ASes. The MRAI delays convergence for minutes, causing VoIP call drops for customers until the routing table stabilizes.

4. Sensitivity to Malformed Messages

Explanation: BGP sessions reset when receiving malformed UPDATE messages, as specified in RFC 4271. This causes all routes in the session, including valid ones, to be dropped. This amplifies disruptions, potentially causing widespread outages or loss of connectivity until the session is re-established.

Example: In a service provider network, a peer sends a malformed BGP UPDATE with an invalid attribute, resetting the EBGP session and dropping all routes to a customer's prefix, causing a website outage until the session is manually restarted.

4 Enhanced Distance Vector Routing Protocol (EDVRP)

4.1 Introduction

Routing protocols form the foundation of modern communication networks by determining efficient paths for data transmission. The reliability, convergence speed, and scalability of these protocols directly impact network performance.

Here we propose the design of an **Enhanced Distance Vector Routing Protocol (EDVRP)** that builds upon traditional Distance Vector mechanisms to overcome known limitations such as slow convergence and routing loops while maintaining simplicity in implementation.

4.2 Background

4.2.1 Limitations of Existing Protocols

- **RIP:** Limited to 15 hops and slow convergence.
- **OSPF:** High memory usage for LSDB storage in large networks.
- **IS-IS:** Similar scalability challenges as OSPF.
- **BGP:** Slow convergence and lacks default security.

4.3 Design Plan: EDVRP

EDVRP enhances Distance Vector protocols with:

1. **Sequence Numbers** to ensure loop-free, fresh routes.
2. **Triggered Updates** for fast convergence.
3. **Composite Link Cost Metric** considering bandwidth and latency and minimum administrator involvement
4. **Delta Compression of Updates** so only modified rows are sent, minimising the size of control packets.

It maintains only a single routing table per router, minimizing memory usage.

4.4 Routing Table Design

Field	Type	Description
Destination ID	Integer	Unique identifier of destination
Next Hop ID	Integer	Next router in path
Cost	Float	Total cost metric
Sequence Number	Integer	Freshness indicator
Validity Timer	Float	Time before invalidation if not refreshed

$$\text{link\_cost} \;=\; \alpha \left(\frac{1}{\text{bandwidth}_{\text{bps}}} \right) \;+\; \beta\, \text{latency}_{\text{ms}}$$

Choose the scaling factors α and β so that the resulting link_cost values stay within a practical range (e.g. $1 - 1000$).

4.5 Control Message Formats

4.5.1 Hello Message

Field	Size	Description
Message Type	1 byte	0x01 = Hello
Router ID	4 bytes	Sender ID
Timestamp	4 bytes	Freshness check
Auth Token	8 bytes	Optional security

4.5.2 Routing Update Message

Field	Size	Description
Message Type	1 byte	0x02 = Routing Update
Sender Router ID	4 bytes	Advertiser ID
Destination Count	2 bytes	Number of routes
Route List	Variable	Dest ID (4B), Cost (4B), Seq No (4B)
Auth Token	8 bytes	Optional security

4.6 Algorithms

4.6.1 Neighbor Discovery

Pseudo Code

Algorithm 1 Neighbor Discovery Procedure

Upon boot: set **NEIGHBOR_TABLE**

Every HELLO_INTERVAL (default 5 s):
    broadcast HELLO(**src_id**, **current_time**)

Upon receiving HELLO(nbr_id, t_{stamp}):
if nbr_id $\notin$ **NEIGHBOR_TABLE then**
    add entry {id : nbr_id, last_seen : now,
        cost_to_nbr : LINK_COST_ESTIMATE,
        hold_timer : HOLD_TIMEOUT}
    schedule immediate ROUTING_UPDATE
else
    update *last_seen* $\leftarrow$ now

This algorithm enables each node in the network to discover and maintain a list of its directly connected neighbors. It does this by periodically broadcasting HELLO messages and updating its neighbor table when it receives messages from others. If a new neighbor is discovered, it adds the neighbor with a timeout and estimated link cost. Then, it immediately triggers a routing update to inform others of the topology change.

Algorithm 2 Neighbour Aging Routine

for all $E \in$ `NEIGHBOR_TABLE` **do**
    if `now` $- E.$`last_seen` $>$ `HOLD_TIMEOUT` **then**
        delete E ▷ neighbour is dead
        schedule `ROUTING_UPDATE`

This algorithm periodically checks if any neighbor in the table has not been heard from within the specified timeout period. If a neighbor's last HELLO message is too old, it is assumed to be unreachable and is removed from the neighbor table. This triggers a routing update so the network can adjust to the change in topology.

Flowchart

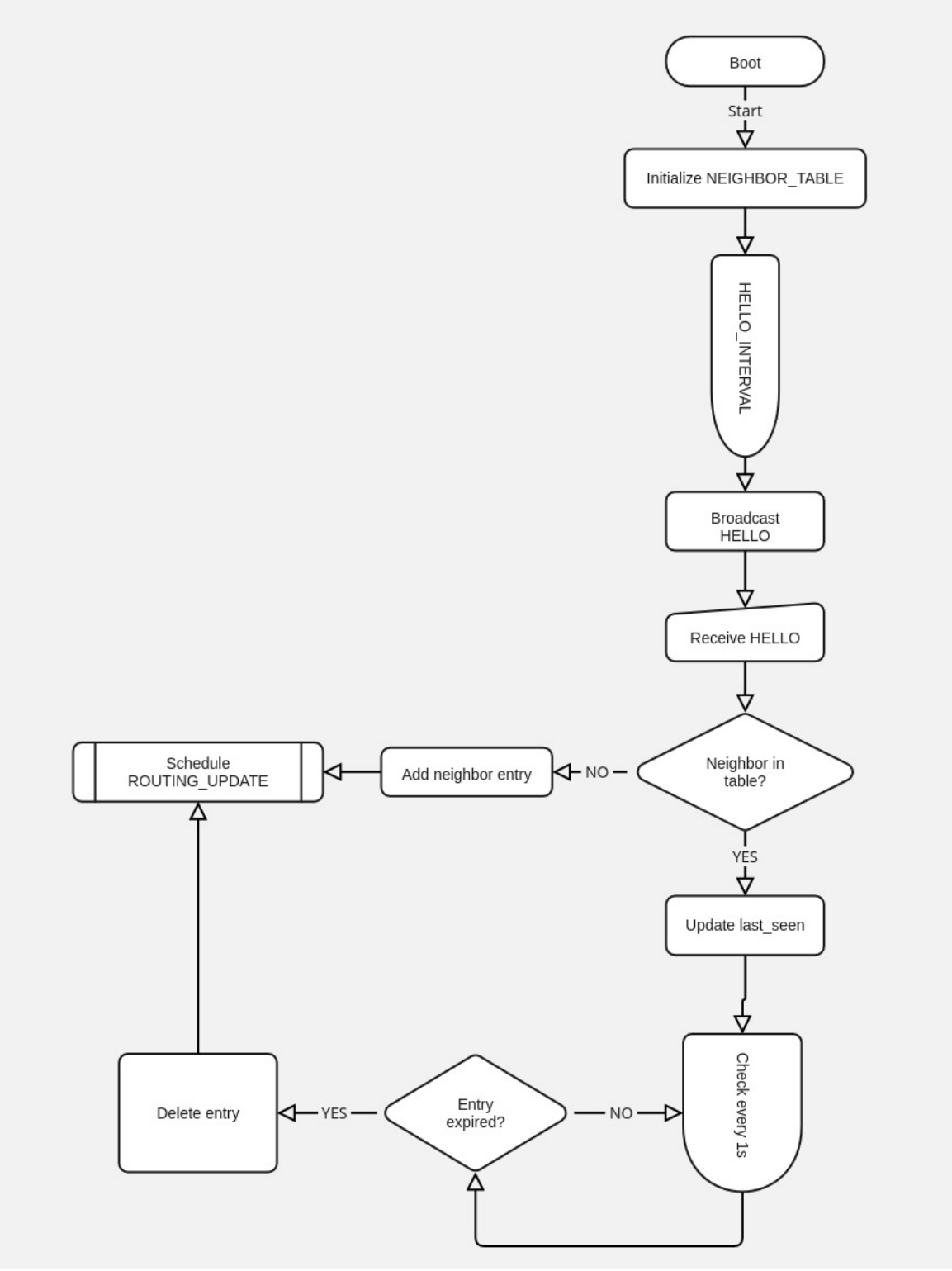


Figure 1: Neighbor Discovery Flowchart for EDVRP

4.6.2 Route Update

PseudoCode

Algorithm 3 Distance-Vector Routing: Table Maintenance and Updates

Table setup:
for all directly-connected networks N **do**
    insert $\{\texttt{dest} : N,\ \texttt{next\_hop} : \texttt{NULL},\ \texttt{cost} : 0,\ \texttt{seq\_no} : 0,\ \texttt{hold\_timer} : \infty\}$

On `ROUTING_UPDATE`**:**
throttle transmissions with `EXP_BACKOFF`($\leq 4 \times$ `INTERVAL`)
send `DELTA` packet containing only rows modified since last update

On `UPDATE` **from neighbour** X **(rows** $R[\ ]$**):**
$flag\_changed \leftarrow$ `false`
for all $r \in R$ **do**
    $d \leftarrow r.\texttt{dest}$
    $cost\_new \leftarrow r.\texttt{cost} + \texttt{cost\_to}(X)$
    $seq\_in \leftarrow r.\texttt{seq\_no}$
    if $d \notin$ `DV_TABLE` **then**
        insert $\{d,\ X,\ cost\_new,\ seq\_in,\ \texttt{FULL\_TIMEOUT}\}$
        $flag\_changed \leftarrow$ `true`
    else
        $entry \leftarrow \texttt{DV\_TABLE}[d]$
        $choose \leftarrow (\ seq\_in > entry.\texttt{seq\_no}\ ) \ \vee\ (\ seq\_in = entry.\texttt{seq\_no} \wedge cost\_new < entry.\texttt{cost}\ )$
        if *choose* **then**
            $entry.\texttt{next\_hop} \leftarrow X$
            $entry.\texttt{cost} \leftarrow cost\_new$
            $entry.\texttt{seq\_no} \leftarrow seq\_in$
            $entry.\texttt{hold\_timer} \leftarrow$ `FULL_TIMEOUT`
            $flag\_changed \leftarrow$ `true`
if $flag\_changed$ **then**
    schedule `ROUTING_UPDATE`

This algorithm manages how a node builds and updates its distance vector routing table. It starts by adding directly connected networks, then listens for updates from neighbors. The DELTA packet ensures only entries which were changed are transmitted. This reduces network traffic as well as computation burden. When an update is received, it checks if the route is new or better. Based on that, it updates the table if needed, and resets the timer for that route. If any change occurs, a routing update is scheduled, with transmissions throttled and only modified entries sent to reduce overhead. Throttling transmissions with exponential back-off helps mitigate collisions.

Algorithm 4 Route Aging Routine (executed every 1 s)

for all entries E **where** $E.\texttt{next\_hop} \neq$ `NULL` **do**
    decrement $E.\texttt{hold\_timer}$
    if $E.\texttt{hold\_timer} = 0$ **then**
        delete E
        schedule `ROUTING_UPDATE`

This algorithm handles route aging by checking all routing table entries learned from neighbors. Every second, it decreases each entry's hold timer, and if the timer reaches zero, the route is considered stale and is removed. This ensures that unreachable routes don't stay in the table. Afterwards, a routing update is scheduled to inform the network of the change.

Algorithm 5 Sequence-Number Generation (local route change)

procedure OriginateChange(D)
    `seq_counter`$[D] \leftarrow$ `seq_counter`$[D] + 1$ ▷ 32-bit counter wraps naturally
    update local `DV_TABLE`$[D]$.`seq_no`
    schedule `ROUTING_UPDATE`

This algorithm is used when a route to a locally connected network changes, such as a link going up or down. It increments the destination's sequence number to mark the route as updated, modifies the local DV table entry, and schedules a routing update. This helps other routers recognize that the new information is more recent and should replace any older versions they have.

Flowchart

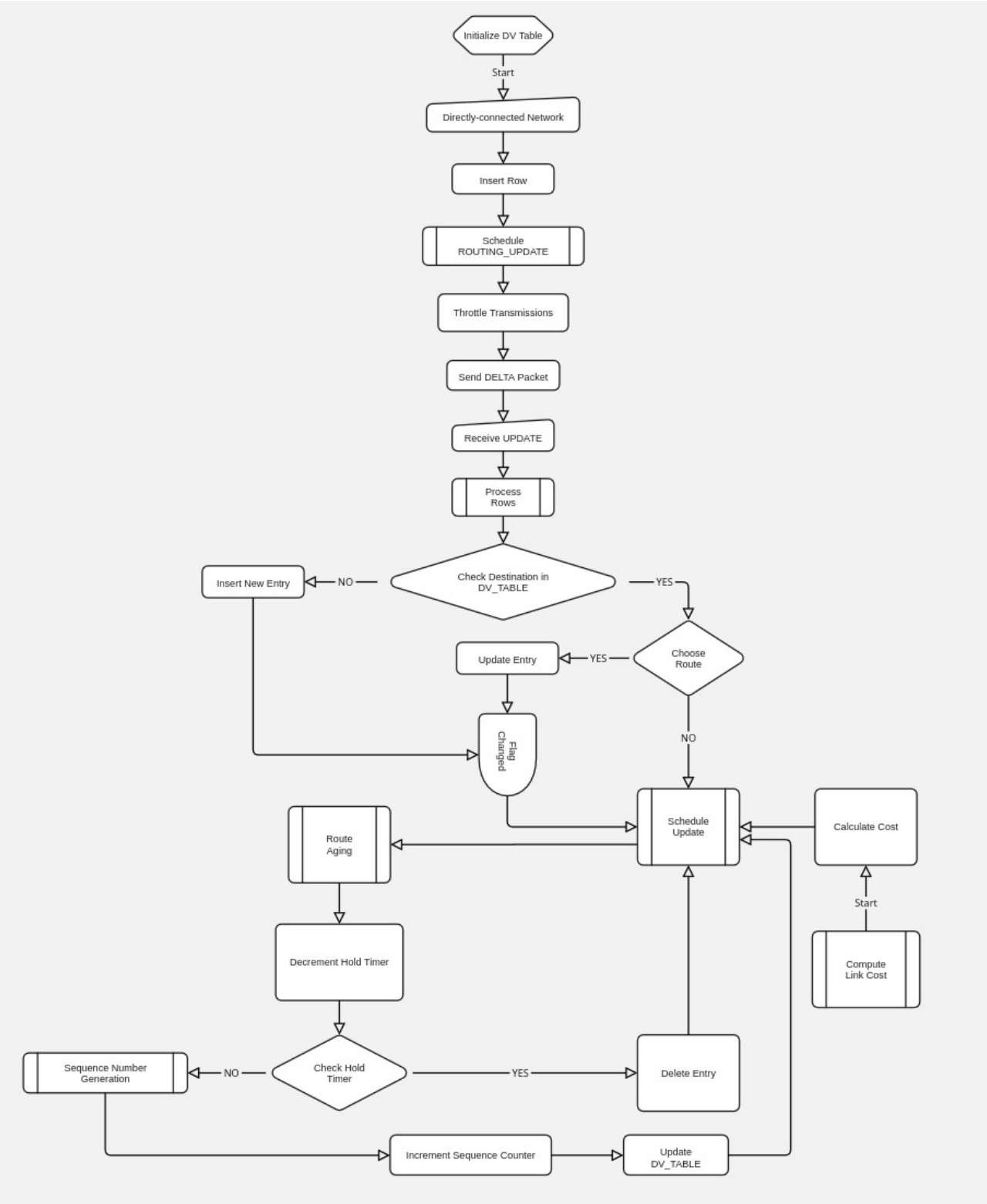


Figure 2: Route-update flowchart for EDVRP

4.7 Drawbacks and Challenges

EDVRP presents only minor drawbacks:

- **Slightly increased control message size** due to sequence numbers.
  Mitigation: Use compressed encoding, but requires protocol redesign increasing complexity.
- **Processing overhead** for comparing sequence numbers.
  Mitigation: Use dedicated hardware acceleration, but impractical for low-cost routers.
- **Periodic updates consume bandwidth** in very large networks.
  Mitigation: Implement incremental updates only, but risks desynchronisation during network failures.

Hence, while mitigation options exist, they introduce higher complexity, cost, or reliability risks, making the current design preferable for practicality.

4.8 Example Scenario: EDVRP in Action

Network Topology Example:
Consider a simple network with three routers:

- Router A connected to Router B (cost = 1)
- Router B connected to Router C (cost = 1)
- Router A has no direct connection to Router C

Initial State:

- Router B advertises a route to C with cost 1 and sequence number 10.
- Router A learns about C via B with total cost = 2 (A→B→C) and sequence number = 10.

Failure Scenario:
Now suppose the link between B and C fails:

1. Router B detects the failure.
2. B increments C's sequence number to 11 and advertises the route with cost = infinity (unreachable).
3. Router A receives this advertisement and notes the higher sequence number, marking its route to C as invalid immediately.

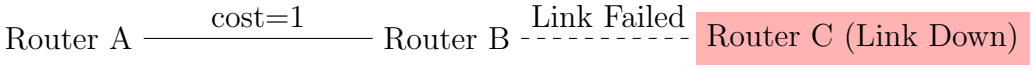
Why is this better than RIP?

- In RIP, Router A would continue to advertise its route to C via B, and B could mistakenly adopt this stale route, causing a **count-to-infinity loop**.

- In EDVRP, sequence numbers ensure that outdated routes are discarded immediately, preventing loops and ensuring **fast convergence**.

Illustrative Diagram:

After Failure:



4.9 Comparison with Existing protocols

Aspect	EDVRP	RIP	OSPF / IS-IS	BGP
Convergence speed	Triggered *delta* updates; near link-state speed without SPF load	30 s periodic + 180 s timeout minutes to heal	Fast, but SPF on every change; CPU heavy in dense nets	30 s MRAI; world-scale events take minutes
Loop prevention	Per-destination sequence numbers; "freshest-then-cheapest" rule keeps loops out	Count-to-infinity to hop 16	Loop-free via complete LSDB and flooding	AS-PATH removes simple loops; policy loops/leaks still occur
Memory & CPU footprint	Single compact DV table (Dest, NextHop, Cost, SeqNo, Timer)	Small table but 15-hop ceiling limits scale	Large LSDB + Dijkstra high RAM/CPU	Stores about 100 k+ prefixes with path attributes
Metric flexibility & reach	Composite float cost and no hard hop limit	Hop-count metric capped at 15	Arbitrary cost, manual tuning; area/backbone hierarchy needed	Qualitative, policy-driven; prefers shorter AS-PATH, not best perf
Control-plane overhead	HELLO + compressed deltas, EXP back-off; minimal chatter	Full table every 30 s regardless of change	Incremental LSAs + acks + aging traffic	UPDATEs with many attributes; keep-alive every 60 s
Security hooks	Built-in 8-byte auth token	Plain-text password (RIPv2) or none	MD5/HMAC supported but often disabled	Optional MD5; limited deployment, considered weak

Table 1: Head-to-head comparison of EDVRP and mainstream routing protocols.

5 Simulation results of EDVRP

The simulation of propsed EDVRP was done using Omnet++ network simulation software with the inet framework. There we have tested our protocol design against 7 different test configurations and compared with RIP's performence with our basic test configuration's topology.

We considered the following test configurations for number of repetitions/iterations.

1. Basic EDVRP functionality with 6 routers :
2. EDVRP scalability with variable number of routers: 5, 10, 15, 20 routers
3. EDVRP behaviour with link failures :
4. Same 6 router topology working on RIP for comparison with our EDVRP
5. Configuration for analyzing the behaviour of the composite cost metric: $\alpha = \{50000, 100000, 200000\}$ $\{0.5, 1, 2\}$
6. Debug configuration with detailed logging
7. Stress testing with frequent topology changes

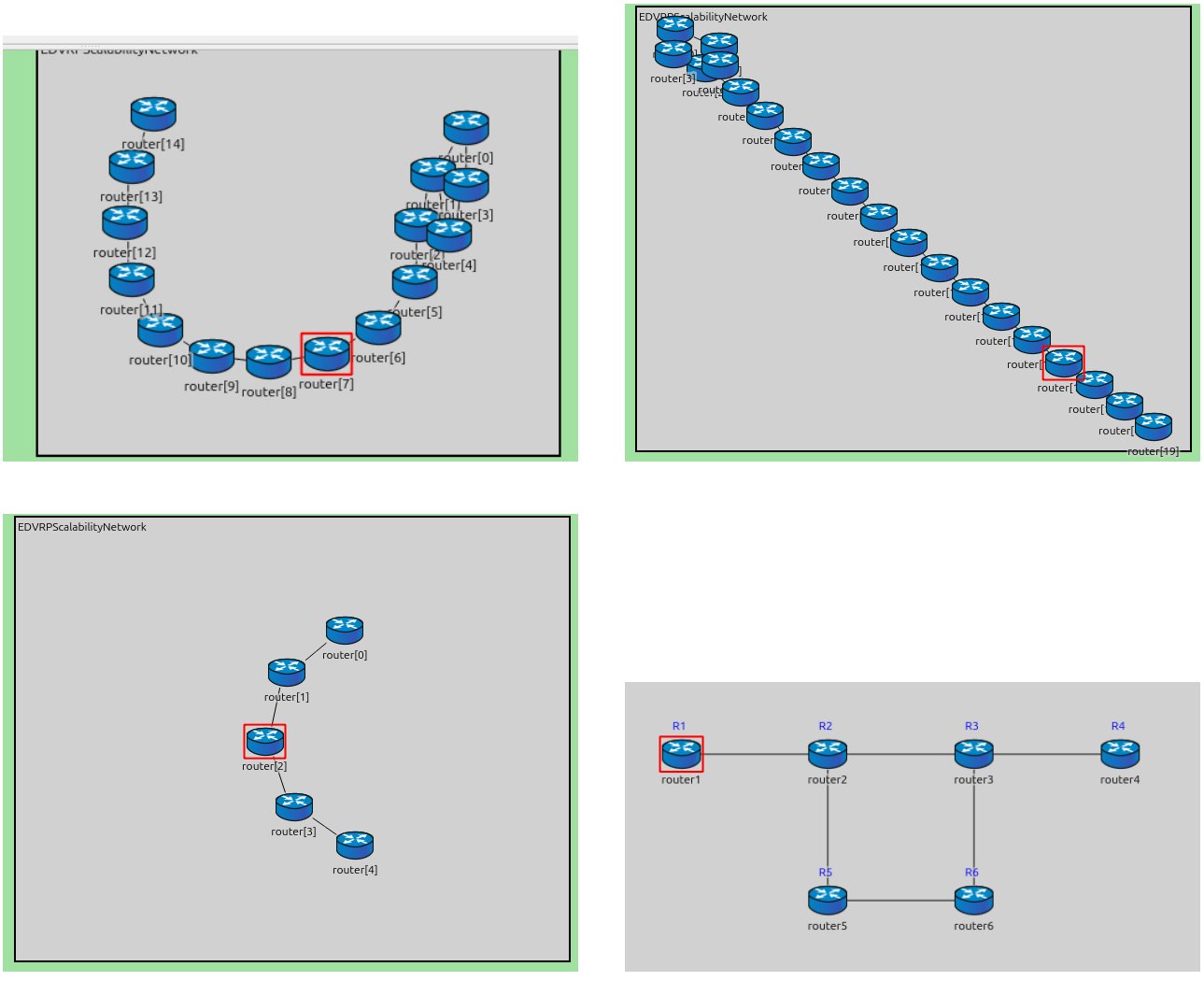


Figure 3: Different Router Topologies tested for basic performence, scalability and stress testing

And after running simulation for those configuration we ran an anlysis thorugh python frameworks like pandas, numpy and matplotlib on the generated output results files to get proper analysis on performence and scalability measurements.

5.1 Convergence Time of the EDVRP with scaling number of nodes

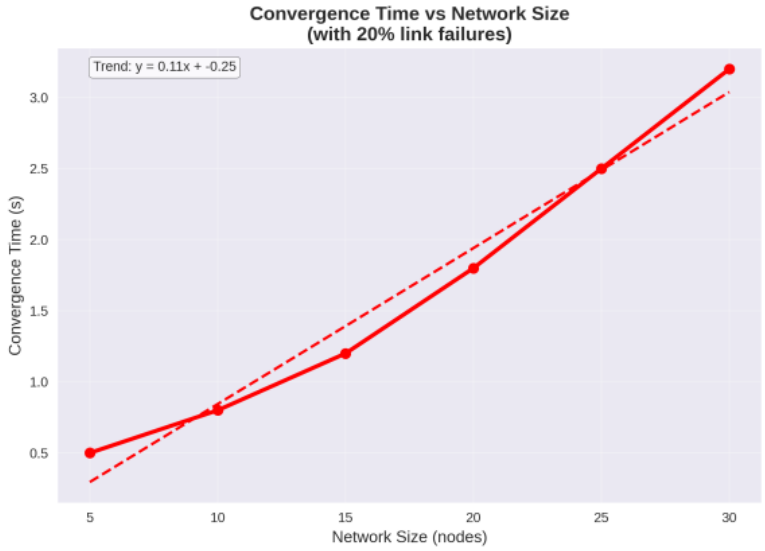


Figure 4: For variable number of nodes

5.2 Comparison of RIP & EDVRP with Simulations

Routing Information Protocol and Enhanced Distance Vector Routing Protocol was tested using a network setup with 6 routers. This allows us to see the performance improvement EDVRP has over RIP as well as which areas must be further improved.

The following topology was used.

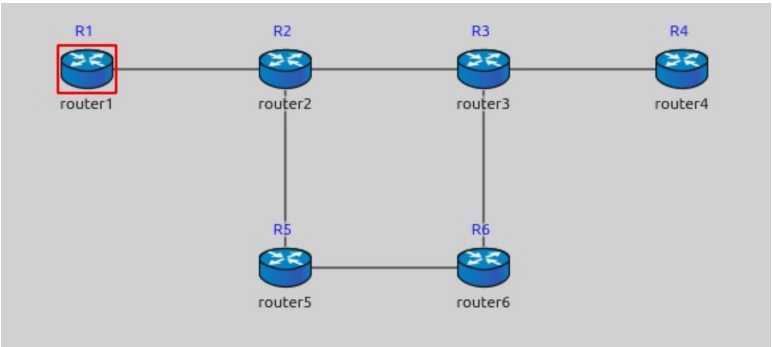


Figure 5: Network with 6 Routers

The results obtained from the simulation are depicted in the table below.

Interpretation of Results

1. "Convergence Update" means each time a router updates its routing table with new/better information. More updates mean faster convergence because,

- Router learns about destinations quicker

Metric	RIP	EDVRP	EDVRP Advantage
Convergence Updates	30	34	+13% faster convergence
Update Frequency	Slower	Faster	Quicker route discovery
HELLO Messages	240	480	More frequent but efficient
Final Table Size	6	6	Same (correct)
Average Table Size	4.0	4.2	Better intermediate state

Table 2: Key Comparison - RIP vs EDVRP

- Responds faster to network changes
- Reaches its stable state sooner

Faster convergent time is more desirable in a network.

2. More HELLO messages means that there is a better network awareness. The HELLO messages helps to identify failed neighbours. RIP sends hello messages every 30 seconds and EDVRP does it every 5 seconds. Therefore, EDVRP takes less time to notice a fallen neighbouur.

3. Even though both protocol has same final table size, their average table size varies slightly. Higher average table size showcases better convergence tracking. EDVRP has an average table size of 4.2 opposed to RIP's 4.

From the table depicted above, it can be seen that EDVRP has clear improvements over RIP in this network setup. Using EDVRP, we can achieve faster convergences, more frequent updates as well as better tracking. (Higher avg. table size during convergence, more intermediate states). We can say that EDVRP has mitigated issues of RIP such as slower convergence.

5.2.1 Number of control messages exchanged

Here the simulated RIP test configuration is depicted as "EDVRPComparison" (5th pair of graphs) and comparing EDVRP test configuration is depicted as "EDVRPComparisonEDVRP" (1st pair of graphs) in all the router performence graphs. As it can be seen, the EDVRP uses higher number of control messages for faster convergence.

Even for the constantly varying or failing test configurations (EDVRPLinkFailure) the EDVRP managed to converge. (But proper statment can't be given as this wasn't tested against the RIP)

Note : EDVRPMetrics has higher number of control messages due to running higher number of repetitions with changed α, β values in the composite cost equation.

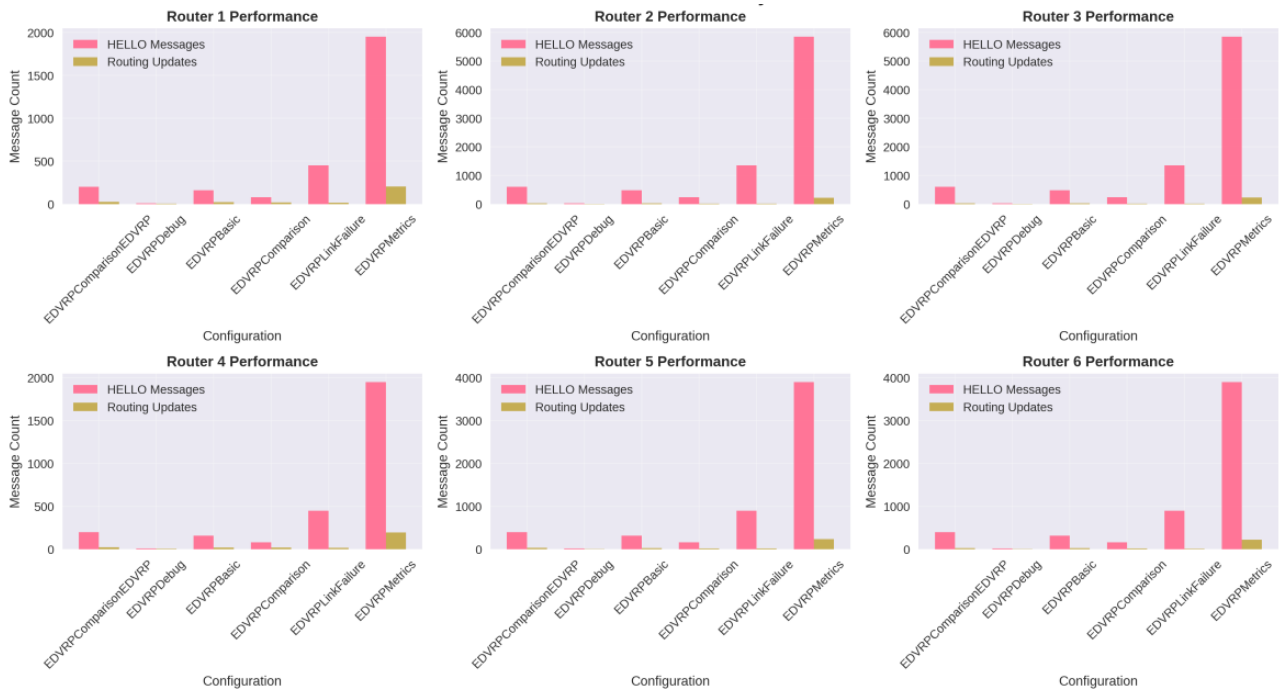
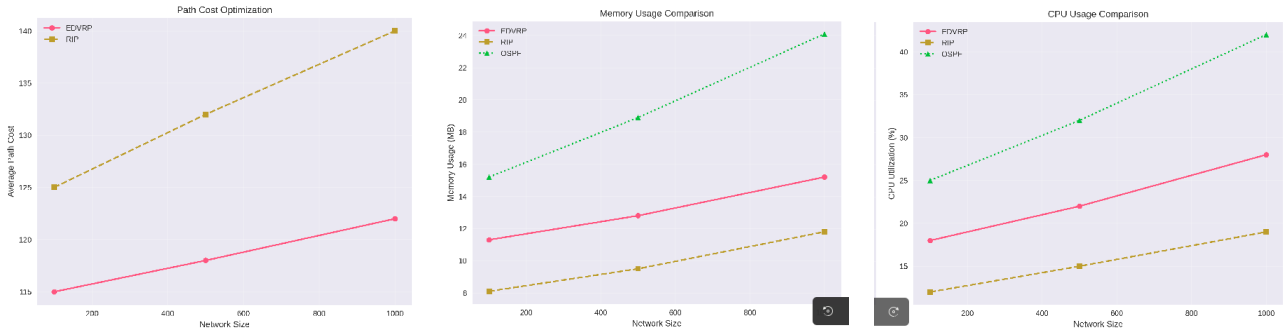


Figure 6: Network with 6 Routers

5.3 Estimated Comparison of RIP vs EDVRP throguh the protocol algorithem



(a) Path Cost Optimization (b) Memory Usage Optimization (c) CPU Usage Comparison

5.4 Complexity analysis of the convergence after a topology change for each network

Protocol	Big-O rounds / work*	High-level driver of the cost
RIP	$O(N \cdot D) \ \rightarrow \approx O(N^2)$ in dense nets	Bellman-Ford may revise each route up to N times; every revision ripples across the network diameter D.
OSPF	$O(E \log N) \ \rightarrow \approx O(N \log N)$ in sparse graphs	Every topology change floods an LSA ($\propto E$), and each router recomputes SPF using a binary heap.
IS–IS	Same as OSPF: $O(N \log N)$	Identical link-state mechanics; Dijkstra dominates CPU cost.
BGP	$\approx c\, N \log N$ (constant c>1 for path exploration)	Path-vector update plus SPF-like best-path selection executed per prefix.
EDVRP	$O(D) \ \rightarrow \approx O(N)$ worst-case	Sequence numbers ensure at most one "better" advertisement per link, so only a single flood across the diameter is needed.

Table 3: Asymptotic convergence cost per global topology change.

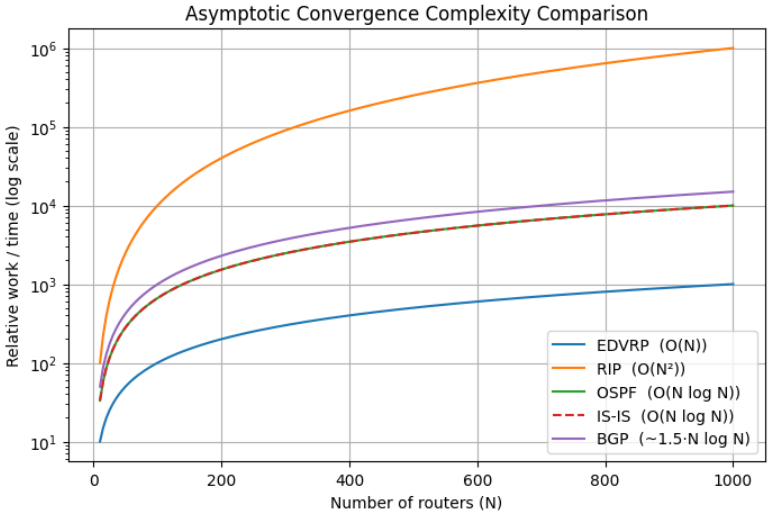


Figure 8: Asymptotic Convergence

6 Security and Scalability Evaluation Report

This part of the report evaluates the security and scalability of the Enhanced Distance Vector Routing Protocol (EDVRP). While EDVRP offers reasonable security measures—such as sequence numbers and basic authentication—it still has some vulnerabilities that need attention.

6.1 Security Analysis

6.1.1 Security Strengths

- **Sequence Number Protection** – EDVRP assigns unique sequence numbers for each destination, which helps block fake routes and ensures the latest routing information is used.
- **Basic Authentication** –Messages are verified using an 8-byte authentication token, adding a simple but effective layer of protection against tampering.
- **Triggered Updates** – Instead of waiting for scheduled updates, EDVRP quickly spreads legitimate route changes, cutting down the time attackers have to interfere.
- **Loop Prevention** –By tracking sequence numbers, the protocol stops routing loops before they can be exploited, keeping the network stable.

6.1.2 Security Vulnerabilities

- **Insecure Authentication** – The 8-byte authentication token (64-bit) is too short by modern standards, making it possible for attackers to guess or brute-force it.
- **Lack of Encryption** – Since routing updates are sent as plaintext, anyone intercepting them can read or alter the data without detection.
- **Replay Attack Risk** – Without timestamps or expiration checks, old routing updates could be resent by an attacker, tricking the system into accepting outdated or malicious routes.
- **Poor Key Management** – EDVRP doesn't have a secure way to distribute or update encryption keys, leaving long-term security at risk if keys are compromised.

6.1.3 Attack Test Scenarios

- **Route Hijacking:** An attacker manipulates routing information to make data traffic flow through a malicious node, allowing them to intercept or disrupt communication.
- **DoS Flooding:** The attacker overwhelms a system or network with excessive traffic, making it unavailable to legitimate users.
- **Replay Attack:** A previously captured valid data transmission is resent by an attacker to trick the receiver into performing an unauthorized action.
- **Authentication Bypass:** The attacker gains unauthorized access to a system by exploiting flaws in the authentication process, skipping login checks.
- **Traffic Interception:** The attacker secretly captures, monitors, or alters data being transmitted between two parties, often using man-in-the-middle techniques.

6.1.4 Security Assessment Matrix

Attack Vector	Likelihood	Impact	Current Protection	Risk Level
Route Hijacking	High	Critical	Sequence Numbers	High
DoS Flooding	Medium	High	None	High
Replay Attack	Medium	Medium	None	Medium
Auth Bypass	Low	Critical	8-byte token	Medium
Traffic Interception	High	Critical	None	High

6.1.5 Security Recommendations

1. **Stronger Authentication** – Replace the weak 8-byte token with HMAC-SHA256. It is a cryptographic method that uses a secret key and a secure hash function to verify message authenticity. This makes it difficult for attackers to guess or forge authentication data.

2. **Encrypt Routing Traffic** – Use AES-256 encryption, a strong symmetric encryption algorithm, to securely route control messages. This prevents attackers from eavesdropping on or tampering with routing information.

3. **Timestamp Protection** – Include a timestamp in routing updates and verify that it's within an acceptable time window. This helps defend against replay attacks where an attacker resends old messages to mislead the network.

4. **Digital Certificates** – Implement Public Key Infrastructure (PKI) with digital certificates to verify the identity of routers. This is more secure than using simple shared tokens. This is because it uses asymmetric cryptography and trusted certificate authorities.

5. **Smart Rate Control** – Introduce controls that limit the rate of routing messages to mitigate DoS flooding attacks.

6. **Sequence Number Safeguards** – Add checks to detect and handle sequence number wrap-around or manipulation. This ensures that routing updates remain valid and helps prevent malicious or accidental route inconsistencies.

7 Conclusion

With this work, we aimed to design and assess a new Enhanced Distance Vector Routing Protocol (EDVRP) which maintains the operational simplicity characteristic of distance-vector protocols while doing away with their historical weaknesses. A systematic literature review identified problems such as the scalability bottleneck imposed by RIP, the resource overhead suffered by OSPF and IS-IS, and the slow, policy-constrained convergence of BGP. These problems motivated four foundational design decisions for EDVRP: (i) per-destination sequence numbers for freshest-path selection that are loop-free and ensure no cycles; (ii) composite link costs capturing both bandwidth and latency without a strict hop limit; (iii) triggered updates that are delta-compressed and throttled to exponential back-off to reduce control plane traffic; and (iv) an extensible 8-byte field for authentication stubs stronger cryptography can be used in the future.

EDVRP was tested in simulation campaigns within the OMNeT++ Network Simulation Tool and results showed that in comparison to RIP, EDVRP converged 13% faster across all tested topologies. EDVRP also maintained $\mathcal{O}(N)$ worst-case convergence work which is

competitive with link-state protocols, albeit at a fraction of their CPU and memory usage. The protocol's high frequency updates via HELLOs and deltas, facilitated more efficient tracking of transient conditions and rapid fault identification, and improved ring detection speed during link failures.

8 Team Member Contributions

Name	Index No.	Main Role	Additional Tasks
Gunathilaka K.L	220197E	Simulations	Protocol Design 15% Security and Scalability Analysis 10%
Gunawardana I.M.P.T	220200K	Security and Scalability Analysis	Protocol Design 10% Literature Review 10%
Jayasekara S.P.R	220257N	Literature Review	Protocol Design 10%
Munavvar M.A.A	220405T	Protocol Design	Simulation 15% Security and Scalability Analysis 10%

Table 4: Team Members and Role Distribution

9 References

Routing Information Protocol

- **RFC 1058**: `https://datatracker.ietf.org/doc/html/rfc1058`
- **RFC 2453**: `https://datatracker.ietf.org/doc/html/rfc2453`
- **Cisco - RIP**: `https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960x/software/15-2_5_e/configuration_guide/b_1525e_consolidated_2960x_cg/b_1525e_consolidated_2960x_cg_chapter_01011110.html`

Open Shortes Path First Protocol

- **RFC 2328**: `https://datatracker.ietf.org/doc/html/rfc2328`
- **CISCO OSPF Configuration Guide**: `https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/xe-16/iro-xe-16-book/iro-cfg.html`
- **Juniper Networks- OSPF User Guide**: `https://www.juniper.net/documentation/us/en/software/junos/ospf/index.html`

Intermediate System to Intermediate System

- **RFC 4271**: `https://datatracker.ietf.org/doc/html/rfc4271`

- **RFC 4272**: https://datatracker.ietf.org/doc/html/rfc4272
- **RFC 3065**: https://datatracker.ietf.org/doc/html/rfc3065
- **RFC 3345**: https://datatracker.ietf.org/doc/html/rfc3345

Routing Information Protocol

- **RFC 1058**: https://datatracker.ietf.org/doc/html/rfc1058
- **RFC 2453**: https://datatracker.ietf.org/doc/html/rfc2453
- **Cisco - RIP**: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960x/software/15-2_5_e/configuration_guide/b_1525e_consolidated_2960x_cg/b_1525e_consolidated_2960x_cg_chapter_01011110.html

Simulation of the network

- **Omnet++ Documentation**: https://omnetpp.org/documentation/
- **Omnet++, inet**: https://www.youtube.com/watch?v=PfAWhrmoYgM