المدرسة الوطنية العليا في الأمن السيبراني
NATIONAL SCHOOL OF CYBERSECURITY

# END SEMESTER PROJECT (NETWORK FOUNDATION )

2025

Presented for :

**MOUZAI MUSTAPHA**
**Mohamed Amine Riahla**

Presented by :

**Hassani Fateh**
**Kassoul Mohammed Ali**
**Hernane Sami Youcef**
**Gheribi Abdenour**

# Introduction

The rapid evolution of wireless communication has brought about a growing need for efficient and reliable routing protocols to manage data transfer in dynamic and decentralized environments . One such protocol is the **Dynamic Source Routing (DSR) protocol**, which is specifically designed for use in Ad Hoc Networks networks that are characterized by their lack of fixed infrastructure and dynamic topology.

This report aims to explore the fundamental principles, operational mechanisms, and practical applications of the DSR protocol. The protocol's unique approach to route discovery and maintenance makes it a pivotal component in enabling seamless communication within wireless networks, particularly in scenarios where network nodes are highly mobile and infrastructure is absent.

To provide a comprehensive understanding of DSR, the report also delves into the broader context of Ad Hoc Networks and wireless communication protocols, highlighting the technical challenges and innovations that define this field.

Through this study, we aim to not only analyze the theoretical aspects of the protocol but also discuss its **real-world implications** and performance considerations.

This report serves as a record of our collective effort and a resource for anyone seeking to understand or work with the DSR protocol in the future.

By studying DSR and its associated technologies, we aim to contribute to the growing body of knowledge in the field of wireless communication and inspire further research and innovation .

# 1. Introduction to Ad Hoc Networks

## Definition :

- Mobile Ad Hoc Networks (MANETs) are decentralized **wireless** networks formed **dynamically** without the need for any fixed infrastructure or centralized administration.
- Devices, or nodes, in a MANET **can move freely** and organize themselves into the network.

## Examples :

- **Disaster Recovery**: In natural disasters like earthquakes or floods, MANETs enable communication among rescue teams where traditional infrastructure is unavailable.
- **Military Operations**: Soldiers and vehicles form tactical networks in remote areas for secure communication.
- **IoT Communication:** Smart devices communicate directly in applications such as smart homes or industrial automation.

## Characteristics :

1. **Infrastructure-less**
   - No reliance on base stations or centralized control points.
   - Nodes themselves handle routing and data transmission.
2. **Dynamic Topology**
   - Nodes can join, leave, or move around freely, leading to frequent topology changes.
3. **Decentralized Operations**
   - Each node operates independently, and decisions are distributed across the network.

## Challenges :

1. **Frequent Topology Changes**
   - The constant movement of nodes causes routes to break, requiring frequent updates.
2. **Limited Resources**
   - Nodes typically have constraints like limited battery power, memory, and processing capacity.
3. **Energy Constraints**
   - Efficient routing is critical as nodes often run on limited energy sources.

# 2.DSR

# DEFINITION :

**Dynamic Source Routing (DSR)** is an **on-demand source routing** protocol designed specifically for MANETs.

# KEY FEATURES OF DSR :

## On-Demand Routing

- What It Means:
  - Routes are established **only** when there is data to send.
  - Eliminates the need for periodic updates, unlike proactive protocols.
- Why It's Useful:
  - Saves **bandwidth** and **energy** by avoiding unnecessary communication.
  - Ideal for dynamic networks where nodes frequently join, leave, or move.

## Source Routing

- **What It Means:**
  - The source node determines **the full path** to the destination and includes this path in the header of each packet.
  - Intermediate nodes forward the packet based on the included path, without needing to maintain their own routing tables.
- **Advantages:**
  - Simplifies intermediate node operations (no routing tables).
  - Makes route tracing and debugging easier.
- **Limitations:**
  - Adds overhead to packets in large networks due to long headers.

## Self-Configuring

- **What It Means:**
  - DSR **adapts dynamically** to changes in the network topology.
  - Nodes **automatically adjust** to route breaks and establish new paths without manual intervention.
- **Why It's Crucial:**
  - **Ensures reliability** in networks where node mobility leads to frequent changes in connectivity.

# 3. HOW DSR WORKS ?

# ROUTE DISCOVERY :

## Objective

- **Finding a Route to the Destination:**
  - The main goal of the route discovery process in DSR is to **identify a valid path** from the source node to the destination node.
  - This process is initiated only when the source node needs to send data and does not already have a route to the destination in its route cache.

## Process

### 1. Route Request (RREQ)

- **Sent by the Source Node:**
  - The source node broadcasts a Route Request (RREQ) packet to all its neighbors to discover a route to the destination.
  - The RREQ contains:
    - Source node address.
    - Destination node address.
    - A unique identifier (to avoid processing duplicate RREQs).
    - A list of addresses representing the route accumulated so far.
- **Broadcast Nature with Route Accumulation:**
  - As the RREQ propagates through the network:
    - Each intermediate node appends its own address to the route field in the RREQ.
    - The packet is forwarded further until it either reaches the destination or an intermediate node with a cached route to the destination.
  - To avoid redundancy, nodes drop duplicate RREQs received earlier.

### 2. Route Reply (RREP)

- **Sent by the Destination (or Intermediate Node with Cached Route):**
  - When the RREQ reaches the destination, the destination node sends a Route Reply (RREP) back to the source using the accumulated route in the RREQ.
  - If an intermediate node has a cached route to the destination, it can respond with an RREP instead of forwarding the RREQ further.
- **Unicast Nature:**
  - The RREP is sent **unicast (directly)** along the reverse of the route accumulated in the RREQ.
  - Intermediate nodes don't need to modify the RREP, as the route **is already known.**

# ROUTE MAINTENANCE

## Objective

- **Ensuring Valid Routes:**
  - The goal of route maintenance in DSR is to ensure that routes remain valid and usable as nodes move or links break.
  - This involves detecting and addressing broken links quickly to maintain seamless communication.

## Process

### Monitoring Link Status

- **How Nodes Monitor Links:**
  - Nodes continuously monitor the status of their links with neighbors.
  - This monitoring can be achieved through:
    - Acknowledgements: A node expects an acknowledgment from the next hop after sending a packet.
    - Periodic HELLO Messages: Some nodes periodically send small messages to confirm link connectivity.
- **Detecting Failures:**
  - If a node does not receive an acknowledgment or detect a HELLO message, it concludes that the link to its neighbor has failed.

### Route Error (RERR)

- **When RERR is Generated:**
  - If a node detects a link failure, it creates a Route Error (RERR) message to notify the source node about the broken route.
- **How RERR Works:**
  - The RERR is sent back along the reverse path to the source node.
  - Each intermediate node receiving the RERR removes the invalid route from its route cache.
- **Triggering New Route Discovery:**
  - When the source node receives the RERR, it either:
    - Uses an alternative cached route to the destination.
    - Initiates a new route discovery process if no alternative route is available.

# 4. DSR ALGORITHM

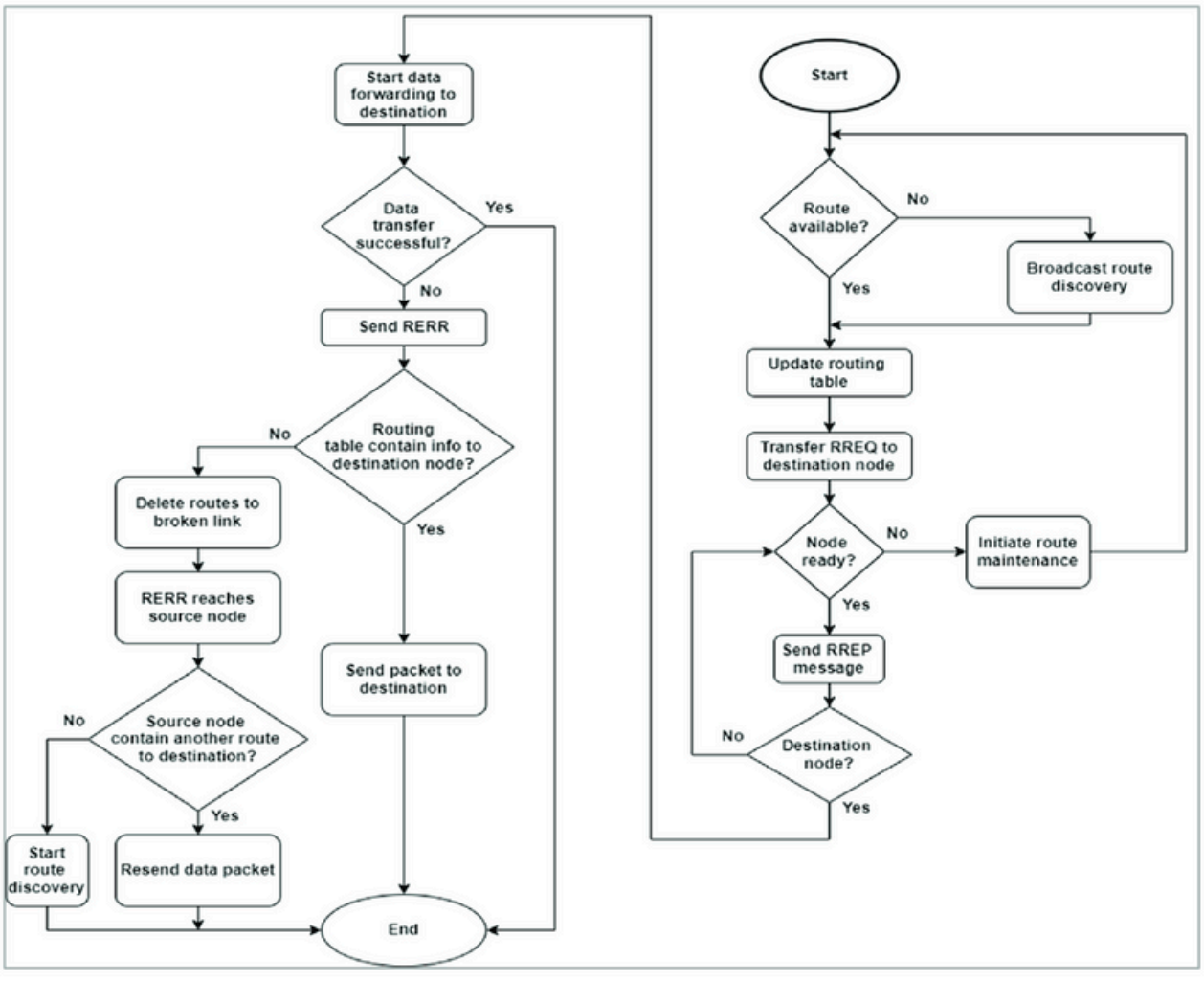
# CODE IN C :

```c
#include <stdio.h>
#include <string.h>

#define MAX_NODES 5
#define INFINITY 999

typedef struct {
    char name[10];
    char route_cache[MAX_NODES][10];
    int num_hops;
} Node;

typedef struct {
    Node *source;
    Node *destination;
} RREQ;
typedef struct {
    Node *source;
    Node *originator;
    Node *destination;
    char route[MAX_NODES][10];
    int num_hops;
} RREP;

typedef struct {
    Node *source;
    char broken_link[10];
} RERR;

Node nodes[MAX_NODES];
```

```c
void initialize_nodes() {
    strcpy(nodes[0].name, "Node0");
    strcpy(nodes[1].name, "Node1");
    strcpy(nodes[2].name, "Node2");
    strcpy(nodes[3].name, "Node3");
    strcpy(nodes[4].name, "Node4");
}

void send_rreq(Node *source, Node *destination) {
    printf("%s is sending RREQ to find route to %s.\n", source->name, destination->name);
    RREQ rreq = {source, destination};
    forward_rreq(&rreq, source);
}

void forward_rreq(RREQ *rreq, Node *sender) {
    printf("%s forwarding RREQ to neighbors.\n", sender->name);
    for (int i = 0; i < MAX_NODES; i++) {
        if (strcmp(sender->name, nodes[i].name) != 0) {
            receive_rreq(&rreq, &nodes[i]);
        }
    }
void receive_rreq(RREQ *rreq, Node *receiver) {
    printf("%s received RREQ from %s.\n", receiver->name, rreq->source->name);
    if (strcmp(rreq->destination->name, receiver->name) == 0) {
        printf("%s has route to %s. Sending RREP.\n", receiver->name, rreq->destination->n
        RREP rrep = {receiver, rreq->source, rreq->destination, {0}, 0};
        strcpy(rrep.route[rrep.num_hops], receiver->name);
        rrep.num_hops += 1;
        send_rrep(&rrep);
    } else {
        printf("%s doesn't have route to %s. Forwarding RREQ.\n", receiver->name, rreq->de
        forward_rreq(rreq, receiver);
    }
}
```

```c
void send_rrep(RREP *rrep) {
    printf("RREP sent by %s to %s.\n", rrep->source->name, rrep->originator->name);
    receive_rrep(rrep->originator, rrep);
}

void receive_rrep(Node *receiver, RREP *rrep) {
    printf("%s received RREP from %s.\n", receiver->name, rrep->source->name);
    strcpy(receiver->route_cache[receiver->num_hops], rrep->route[rrep->num_hops - 1]);
    receiver->num_hops = rrep->num_hops;
}

void send_rerr(Node *source, char *broken_link) {
    printf("%s detects broken link %s. Sending RERR.\n", source->name, broken_link);
    RERR rerr = {source, broken_link};
    for (int i = 0; i < MAX_NODES; i++) {
        if (strcmp(source->name, nodes[i].name) != 0) {
            receive_rerr(&nodes[i], &rerr);
        }
    }
}
void receive_rerr(Node *receiver, RERR *rerr) {
    printf("%s received RERR. Updating route cache.\n", receiver->name);
    for (int i = 0; i < receiver->num_hops; i++) {
        if (strcmp(receiver->route_cache[i], rerr->broken_link) == 0) {
            for (int j = i; j < receiver->num_hops - 1; j++) {
                strcpy(receiver->route_cache[j], receiver->route_cache[j + 1]);
            }
            receiver->num_hops -= 1;
        }
    )
}

int main() {
    initialize_nodes();

    send_rreq(&nodes[0], &nodes[3]);  // Node0 wants to send data to Node3

    send_rerr(&nodes[2], "Node2 <-> Node3");  // Simulate broken link

    return 0;
}
```

# SIMPLE ILLUSTRATION :



# 5. ADVANTAGES AND LIMITATIONS

## ADVANTAGES

**1 .Efficient in Dynamic Environments:**
DSR excels in Mobile Ad hoc Networks (MANETs) due to its ability to adapt to changes in topology quickly without relying on pre-established routing tables.

**2. Reduces Routing Overhead:**
Unlike proactive protocols, which maintain routing tables constantly, DSR only establishes routes when needed, leading to lower bandwidth usage and reduced routing overhead.

**3. Simplifies Protocol Design with Source Routing:**
By allowing the source node to determine the complete route to the destination, DSR eliminates the need for routing tables at intermediate nodes, simplifying protocol design and implementation.

## LIMITATIONS

**1. Header Overhead in Large Networks:**
Since DSR uses source routing, each packet carries its entire route in the header. This increases the size of packet headers, especially in large networks, leading to inefficient use of bandwidth.

**2. Potential for Route Cache Staleness:**
As DSR relies on caching routes at nodes, stale or outdated routes can lead to route errors, which can slow down communication in the network.

**3. Flooding During Route Discovery in Large Networks:**
In scenarios where route discovery needs to be initiated, DSR might experience excessive flooding, increasing the risk of collisions and delays, which is less efficient in larger networks.

## COMPARISON WITH OTHER PROTOCOLS

- **Source Routing vs. Hop-by-Hop Routing:**

DSR uses source routing, where the entire route to the destination is included in the packet header, whereas AODV uses hop-by-hop routing, where intermediate nodes rely on a route table to forward packets.

- **Differences in Routing Table Maintenance:**

DSR caches routes at intermediate nodes and the source, while AODV maintains routing tables by sending periodic control messages. AODV dynamically updates these tables using route request and reply mechanisms.

# Conclusion

**The Dynamic Source Routing (DSR) protocol** has proven to be an engaging and thought-provoking topic that warranted in-depth exploration. Through collective teamwork and productive collaboration among team members, we successfully covered numerous aspects of the protocol. To achieve the objectives of this project, we expanded our knowledge by delving into new concepts such as Ad Hoc Networks and wireless protocols , which greatly enriched our understanding and added substantial value to the project.

The work on this project commenced on **Monday, January 20, 2025**, with the initial stages involving defining objectives and gathering the necessary resources. Tasks were allocated efficiently to ensure optimal use of time and expertise while maintaining precision and quality throughout the process. The final preparation was completed, and the project was officially presented on **Wednesday, January 22, 2025.**

This project provided an invaluable opportunity to acquire new insights into networking and wireless protocols while also enhancing our research, analytical, and collaborative skills. It enabled us to confront technical and logistical challenges that contributed to the refinement of our practical abilities. We hope that this work serves as a foundation for future research in this vital area of technology for us .

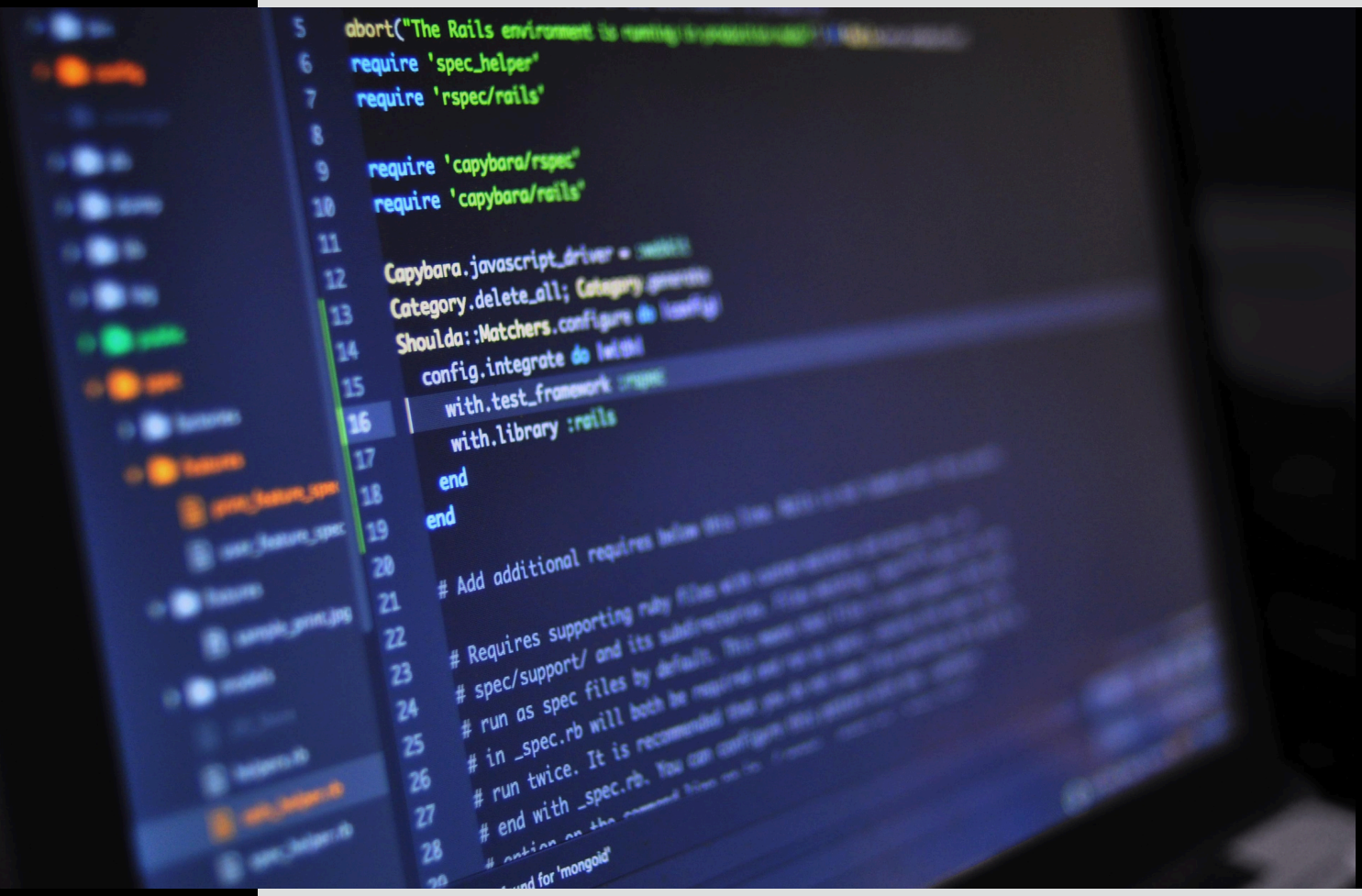
We recommend further study of other wireless protocols to analyze their performance in diverse environments and compare them with the DSR protocol.

Additionally, we implemented this protocol in practical simulations using programming tools to evaluate its performance comprehensively.

Our sincere gratitude goes to everyone who contributed and supported us in completing this project successfully.

END SEMESTER PROJECT

NSCS
المدرسة الوطنية العليا في الأمن السيبراني
NATIONAL SCHOOL OF CYBERSECURITY




# Thank you !

**For further information regarding this project, please contact me :**

# My Contact


f.hassani@enscs.edu.dz

s.harnane@enscs.edu.dz

m.kassoul@enscs.edu.dz

a.gheribi@enscs.edu.dz