# COOPERATIVE NETWORK ARCHITECTURE FOR NEXT GENERATION TECHNOLOGIES

A Dissertation
Presented to
The Academic Faculty

By

Razvan Cristian Voicu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Engineering
Department of Electrical and Computer Engineering

Georgia Institute of Technology

May 2022

# COOPERATIVE NETWORK ARCHITECTURE FOR NEXT GENERATION TECHNOLOGIES

Thesis committee:

Dr. John A Copeland
Electrical and Computer Engineering
*Georgia Institute of Technology*

Dr. Yusun Chang
Electrical Engineering
*Kennesaw State University*

Dr. Raheem Beyah
College of Engineering
*Georgia Institute of Technology*

Dr. Henry L. Owen
Electrical and Computer Engineering
*Georgia Institute of Technology*

Dr. Douglas M Blough
Electrical and Computer Engineering
*Georgia Institute of Technology*

Dr. Muhammad Hassan Tanveer
Robotics and Mechatronics Engineering
*Kennesaw State University*

Date approved: January 13, 2022

**Progress** is accomplished not by the wise who learn to give up but by the fools who forget to stop trying.

*Old Proverb*

To great mentors (Dr. Yusun Chang, Dr. Raheem Beyah, Dr. John Copeland)

To my parents (Lucian Voicu, Maria Roxana Voicu)

To my brother Lucian Bogdan Voicu

To Amanda Jean Fudge

& Everyone who have supported me

# ACKNOWLEDGMENTS

I want to extend my most heartfelt gratitude to Dr. Yusun Chang, Dr. John A Copeland, and Dr. Raheem Beyah for their mentorship, guidance, and continuous support. Without the exposure and ability to sit in during weekly discussions, I would never have considered graduate studies. The ability to receive advice and assistance at any time is indispensable and eternally appreciated.

The same support and belonging existed in the Communication Systems Center (CSC), Communications Assurance and Performance Group (CAP), and Vehicular Ad-Hoc Networking (VANET) groups at Georgia Tech. I am thankful to all of the colleagues in the CSC, CAP, and VANET groups for their professional advice and assistance.

Additionally, I am grateful for all of the faculty members at Georgia Tech during my course work. I am thankful to Dr. Daniella Staiculescu for always providing administrative support throughout my graduate career. I could not navigate through all of the requirements without Dr. Staiculescus's aid.

Finally, I would like to thank Amanda Jean Fudge and my family, Lucian Voicu, Maria Roxana Voicu, Lucian Bogdan Voicu, for always supporting me during hardships and good times.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

## LIST OF ACRONYMS

**3gpp** 3rd Generation Partnership Project

**6LowPAN** IPv6 Low-Power Wireless Personal Area Network

**ACK** Acknowledgement

**AI** Artificial Intelligence

**AMF** Access and Mobility Management Function

**AMM** Access and Mobility Management

**ANs** Access Networks

**API** Application Programming Interface

**ARP** Address Resolution Protocol

**ATSSS** Access Traffic Steering, Switching and Splitting

**BSMs** Basic Safety Messages

**C2C** Car to Car

**CCM** Client Connection Manager

**CDNs** Content Delivery Networks

**CNN** Convolutional Neural Network

**CoopNet** Cooperative Networking Architecture

**CPRI** Common Public Radio Interface

**CRAN** Centralized/Cloud Radio Access Network

**D2D** Device to Device

**DC** Data Collection

**DHCP** Dynamic Host Control Protocol

**DL** Deep Learning

**DNSSEC** Domain Name System Security Extensions

**DSRC** Dedicated Short Range Communication

**eNodeB** Evolved Node B

**ETSI** European Telecommunications Standards Institute

**FFNN** Feed Forward Neural Network

**FHS** Frequency Hop Selection

**IaaS** Infrastructure as a Service

**ICMP** Internet Control Message Protocol

**IETF** Internet Engineering Task Force

**IoT** Internet of Things

**IP** Internet Protocol

**IPSec** Internet Protocol Security

**LAN** Local Area Network

**LEO** Low Earth Orbit

**LoRA** Long Range

**LTE** Long Term Evolution

**M2M** Machine to Machine

**MAC** Media Access Control

**MAMS** Multi-Access Management Services

**MANET** Mobile Ad-Hoc Networking

**MEC** Multi-Access Edge Computing

**MIMO** Multiple-Input Multiple-Output

**ML** Machine Learning

**MPLS** Multiprotocol Label Switching

**MPQUIC** Multipath QUIC

**MPTCP** Multipath Transmission Control Protocol

**MRANs** Multipath Radio Access Networks

**N3IWF** Non-3GPP Interworking Function

**NCM** Network Connection Manager

**NDP** Neighbor Discovery Protocol

**NFV** Network Function Virtualization

**NN** Neural Network (NN)

**NR** Next Radio

**NS3** Network Simulator V3

**NSAs** NetShare Advertisements

**NSCs** NetShare Correction

**NSRs** NetShare Requests

**NSSs** NetShare Solicitations

**NSVs** NetShare Validations

**PAN** Personal Area Networking

**PC** Policy Control

**PDU** Packet Data Unit

**PON** Passive Optical Networks

**QOE** Quality of Experience

**RAN** Radio Access Network

**RAT** Radio Access Technology

**RED** Random Early Detection

**ReLu** Rectified Linear Unit

**RF** Radio Frequency

**RFC** Release for Comments

**RNIS** Radio Network Information Service

**RNN** Recurrent Neural Network

**RSUs** Road Side Units

**RTT** Round Trip Time

**SCTP** Stream Control Transmission Protocol

**SD-WAN** Software-Defined Wide Area Networking

**SDNs** Software-Defined Networks

**SM** Session Management

**SNIR** Signal to Interference and Noise Ratio

**SNR** Signal-to-Noise Ratio

**SoC** Systems-on-Chip

**TCP** Transmission Control Protocol

**TDM** Time Division Multiplexing

**TLS** Transport Layer Security

**UAVNs** Unmanned Aerial Vehicle Networks

**UAVs** Unmanned Aircraft Vehicles

**UDP** User Datagram Protocol

**UE** User Equipment

**URLL** Ultra-Reliable Low Latency

**VANETs** Vehicular Ad-Hoc Networks

**WBA** Wireless Broadband Alliance

**WDM** Wavelength Division Multiplexing

# SUMMARY

The internet is evolving, with the number of devices connected rapidly increasing. The 5G/6G wireless implementation promises several Gbps plus massive connectivity with ultra-reliable low latency capabilities. However, 5G and 6G provide solutions by integrating the cellular core. Satellite internet aims to provide deep area coverage through a mesh topology of Low Earth Orbit (LEO) stationary satellites.

Due to ephemeral communication links, satellite internet routing and forwarding need to be decentralized, agnostic and offer the lowest latency possible. Like decentralized and centralized synergy, similar necessities arise in differing environments, including Vehicular Ad-Hoc Networks, Internet of Things communication, Infrastructure-less networks, and more. The analogous communication systems denote a need for restructuring the traditional network communication architecture and frameworks.

This research helps develop the structure and proof of concept implementation of a horizontal communication architecture, termed Cooperative Networking Architecture (CoopNet). CoopNet introduces a programmable platform for developing innovative mechanisms to improve and provide parallel multipath communication. CoopNet depends on node cooperation allowing end nodes and systems to become part of the network infrastructure and offer networking functions through a Network Function Virtualization (NFV) approach. CoopNet with NFV allows regular nodes to assist in routing, forwarding, security, and other functionality using physical and virtual links.

CoopNet proposes to enable advanced network functionality to all nodes, including edge nodes, thereby becoming part of the network development and enrichment. CoopNet allow nodes to craft packets in proprietary formats, such as cellular communication, even without a proprietary Radio Access Technology (RAT) or interface. Hence, hosts can use intermediary nodes as Radio Access Network (RAN) translation points from one radio type to another. RAN translation yields a new infrastructure as a service platform.

# CHAPTER 1

# INTRODUCTION AND BACKGROUND

There is a need to develop novel architectures for future generation networks to meet integration, expansion, and usability demands. This thesis presents CoopNet, a proposed network architecture to create and maintain multipath routes through a cooperative node approach. Collaborative nodes assist in the network development and provide network features, such as forwarding messages using idle or underutilized channels.

Historically, all improvements in telecommunication have been a stimulus for technological advancement. The development of the internet, evolution in networking, and more recently, the Internet of Things (IoT), also known as everything connected [1], are significant advancements that push the research community in discovering methods of transferring data faster and more reliably.

The 5G cellular network [2] aims to meet the evolutionary demands, having higher speeds, better range, and a more significant number of devices connected. Vehicular Ad-Hoc Networks (VANETs), cellular Device to Device (D2D), and Unmanned Aerial Vehicle Networks (UAVNs) [3] have begun to use ad-hoc networks due to the increasingly dynamic topologies. Ad-hoc communication allows infrastructure-less networks to exist through the direct D2D communication [4]. Satellite internet focuses on an ad-hoc mesh hybrid that allows for links between satellites, while the ground station can communicate with a satellite directly.

Currently, communication development primarily branches in different directions with a limited focus on merging the various Radio Access Networks (RANs) and utilizing multipath communication. Undoubtedly targeted advancements are optimal in different environments. For example, flying ad-hoc networks, developed by Unmanned Aircraft Vehicles (UAVs), are ordinarily temporary networks used in military, commercial, and civil applications. Such ephemeral networks can be used in disaster recovery, search and rescue missions, covert operations, sensory data collection, and more. Vehicular Ad-Hoc Networks VANETs and cellular Car to Car (C2C) networking aid in communication between vehicles [5].

Commonly, UAVNs and VANETs function via broadcasting multi-hop messages [6]. With broadcasting-based communication, over usage of the channel leads to broadcast storms [7]. Broadcast storms cause network flooding leading to unusable channels. Whereas some approaches, including VANETs and UAVNs, are ideal for particular circumstances, such as ultra-low latency communication, the network type is not optimal for infotainment applications. A best-fit solution is to combine multipath multi-radio access network access. However, how would such an environment look? How would it work? One unique solution is CoopNet.

Other works in progress include frameworks and solutions by differing agencies and committees. The Internet Engineering Task Force (IETF) provides Release for Comments (RFC) 8743: Multi-Access Management Services (MAMS) completed in March 2020. 3rd Generation Partnership Project (3gpp) plans to release Access Traffic Steering, Switching and Splitting (ATSSS) with the 5G Next Radio (NR) core. Other forums, such as Wireless Broadband Alliance (WBA), European Telecommunications Standards Institute (ETSI) Multi-Access Edge Computing (MEC), aim to provide solutions in preparation for multipath multi-radio use [8] [9] [10].

The IETF RFC 8743, MAMS, explains dynamic programable selection and intelligent combination of Access Networks (ANs) [9]. MAMS defines the terminology and best practice recommendations, including maintaining backward compatibility and use of Multipath Transmission Control Protocol (MPTCP) and Stream Control Transmission Protocol (SCTP) [9].

MAMS introduces a Multi-Access (MX) Convergence Layer sitting below the network layer to maintain backward compatibility. MAMS proposes several new concepts in the latest March 2020 release, including Radio Network Information Service (RNIS). RNIS is an initial step of sharing control messages between the Network Connection Manager (NCM) and Client Connection Manager (CCM). Signaling messages include path quality information and network capabilities.

NCM defines packet distribution over available ANs managing the user-plane treatment, such as tunneling, encryption of the different flows. CCM handles exchanging signaling messages with NCM to configure the various paths from existing ANs on the node. MAMS recommends separating the control plane and the user plane. The control plane handles the negotiation of agnostic downlink and uplink networks, and the user plane manages the specific protocol requirements. RFC 8743 suggests a policy-based optimal path, service discovery, lossless path switching or handling, adaptive access with multipath support.

CoopNet differs from MAMS in the foundational structure. It does not rely on a convergence layer. CoopNet integrates at the application layer in traditional networks to offer backward compatibility. At the same time, CoopNet can operate as a stand-alone architecture in a custom network environment. CoopNet aims to persuade the development of a programable network API to control all network functions, including discovery.

Another popular solution supporting MAMS is 3gpp's ATSSS [11]. More featured than MPTCP, 3gpp's ATSSS offers coexisting 3gpp-cellular and non-cellular connections. ATSSS is a multiple-layer framework working between several 5G Core functions, including Access and Mobility Management (AMM), Session Management (SM), and Policy Control (PC) [12]. Undoubtedly, ATSSS will revolutionize communication, providing parallel transmission through multiple interfaces. However, ATSSS dictates all devices to use 3gpp's 5G Core. Moreover, ATSSS is still in the infancy stage of implementation.

Most multipath implementations push responsibilities to the end-host/User Equipment (UE). CoopNet goes a step further and provides an architecture that runs on all nodes, enabling network development and communication assistance through network function virtualization (NFV) using three cooperation classes [13]. The proposed CoopNet architecture consists of five modules, including Discover, Decision, Utilization, Data Collection, and Dynamic Adaptation depicted in Figure 1.2.

Module 1, Discovery, and module 2, Decision, are core components that handle the network awareness and data flow policies analogous to the steering, switching, and splitting in ATSSS. The benefits of CoopNet are multifold. It is implementable in existing networks and does not require a proprietary core framework, such as 3gpp's 5G Core. It is useable in different environments, and it can provide new Infrastructure as a Service (IaaS) through resource sharing.

The motivation is to support unique and novel communication requirements, including IoT deployment of low resource devices, Machine to Machine (M2M) big data, and delay intolerant communication for areas such as Vehicular Ad-Hoc Networks (VANETs). More vital requirements are cellular data offloading and on-demand support for the classically defined end-user access. The benefits of CoopNet extend to healthcare, military, commercial, and consumer applications. It is a stepping stone to develop self-organizing, self-managing, using parallel Multipath Radio Access Networks (MRANs) providing optimal spectrum efficiency [14].

## 1.1 Cooperative Network Architecture: CoopNet

Communication networks advance at an expedited rate. Newly deployed satellite mesh networks aim to provide internet access at promised rates of Gbps. 5G network research focuses on discovering ways of using network slicing [15], better usage of traditional systems, heterogeneous Radio Access Technologies (RATs) [16], higher frequency modulation with mmWave technologies [17].

In parallel, research is progressing mobile networking such as VANETs [6, 7, 18, 5], Mobile Ad-Hoc Networking (MANET) [19], UAVNs [3, 20], Long Range (LoRA) networking[21], Visual Light Networking [22], and much more. Devices' computational power improvement, reduced manufacturing costs, multi-radios on a chip (SoC) have made it easier to create communication networks where infrastructure is nonexistent or sparse. Since building infrastructure is a long and costly process, it is vital to use device-to-device communication.

Different multi-hop techniques aid D2D communication [6, 7]. Data sharing provides a new venue for network assistance, such as sharing sensory information to improve the routing capabilities [18, 21]. NextGen networks in [16, 23] aim to use Radio Access Networks in a heterogeneous network manner to handle the speed demands of 5G standardization using machine learning, Signal to Interference and Noise Ratio (SNIR), and probabilistic approaches to determine which network is best.

In most literature, machine and reinforcement learning seem to have promising results. However, while machine learning performs admirably in specific environments, machine learning can be more harmful than basic approaches with incorrect or little training data. Likewise, the dependency on using physical layer information alone is insufficient to keep up with rapid dynamics.

CoopNet is a proposed architecture that combines different protocols to discover and develop multiple links using nodes in the network route development. A centralized approach is limited in maintaining a stable network in a highly dynamic environment where several thousand devices are connected.

## 1.2 Architecture Background

CoopNet recommends implementation standards and protocols for enhancing communication, especially in multipath multi-radio infrastructure-less networks. A brief description of how communication occurs in traditional networks is necessary to depict the difference best. Traditionally, a communication device has a unique MAC address given by the manufacturer and generally will acquire an Internet Protocol (IPv4/IPv6) address for network layer addressing.

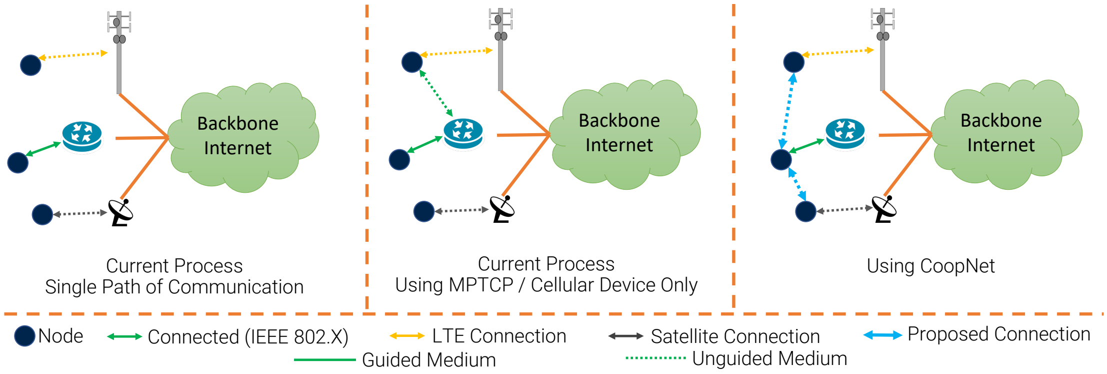


Figure 1.1: Current Internet Working — MPTCP — CoopNet Implementation

Using protocols such as Dynamic Host Control Protocol (DHCP), Neighbor Discovery Protocol (NDP), or Address Resolution Protocol (ARP), a host or access point can broadcast a message to get address information about where to send session establishment requests. Once a node collects the necessary information, communication can occur using specific transport protocols such as Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). A set of applications will compile the necessary messages and send them using a specific physical layer modulation.

Two nodes must establish a TCP connection using what is known as three-way handshaking. TCP connections provide some benefits such as congestion control, data reliability, and some network fairness. Upon establishing a TCP connection, nodes can transmit application layer messages. MPTCP is an enhancement to TCP that allows multiple TCP connections to join together. However, cellular systems use MPTCP as a handoff mechanism for cellular traffic offloading in actual implementation. Parallel use of numerous radios still requires further progress with different releases, including MAMS, ATSSS, and CoopNet.

Software-Defined Networks (SDNs) and Network Function Virtualization are two pioneering methodologies to improve communication. In traditional networks, routers represent the backbone of the internet. Routers are in charge of creating routing paths and storing forwarding tables. As with anything from a global perspective, a single router does not have all possible routes since the current global network in its entirety is massive and rapidly changing. SDNs provide a moderately centralized top-level approach handling route development and next-hop transmission using a match-plus action method.

SDNs permit generalized forwarding where packet transmission can use additional information beyond the destination address for determining the correct link to use. With SDNs accomplishing NFV is easier. For example, firewall implementation can be considered a policy in the forwarding definitions of an SDN controller. Moreover, SDNs assist in implementing MPTCP by providing a centralized decision of routes through data flows [24].

SDNs are ideal for many environments. However, for more dynamic networks such as UAVNs or VANETs, establishing a transport-level connection such as TCP and maintaining an Internet Protocol (IP) address assigned by a central base station is unrealistic due to the high mobility. Even in cases of handoff mitigation, the exchange between the different base stations, also known as Road Side Units (RSUs) for VANETs, would cause unnecessary flooding of messages. CoopNet allows granularity through programable control of all network formulation and functionality aspects.

Figure 1.1 depicts the current communication structure, with MPTCP, and through the use of CoopNet. Most internet traffic represents a setup shown in Figure 1.1A where communication uses a single path, and a single RAN for the entire process. Alternatively, much of the communication using both WiFi and cellular networks utilizes MPTCP for cellular offloading of traffic, as shown in Figure 1.1B, alleviating the cellular network congestion and reserving resources for other users.

Alternatively, communication depicted in Figure 1.1C does not yet exist except in specialized testbeds, even though the capabilities are there. CoopNet allows using multiple radios in parallel by creating links and developing new frameworks for data access. Nodes (hosts) create traditional and virtual communication links developing new routing paths cooperatively.

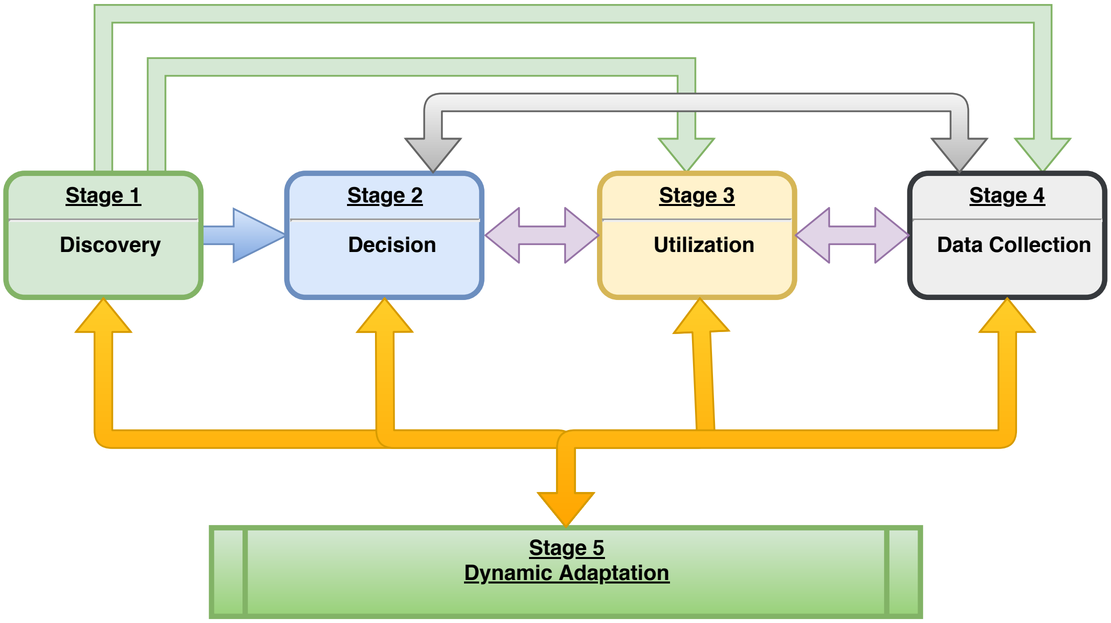


Figure 1.2: Architecture Flow Chart

In CoopNet, intermediary nodes can remain transparent and forward packets without modification or alter the messages by inserting their address for return messages providing the possible added benefit of a new level of security. With nodes having multiple interfaces available, parallel communication can use the idle network radios. Thus, inactive nodes can act as relays and play an essential role in cooperative network development.

## 1.3 Architecture Design

The architecture, by design, is cyclic with continuous adjustment providing a constant update of available paths and their states. Figure 1.2 presents the generalized architecture flow chart with five modules containing the following components: A) Discovery, B) Decision, C) Utilization, D) Data Collection, and E) Dynamic Adaptation.

Modules are scalable, allowing additional frameworks, module deactivation, or custom modules implementations depending on the environment and requirements. CoopNet uses

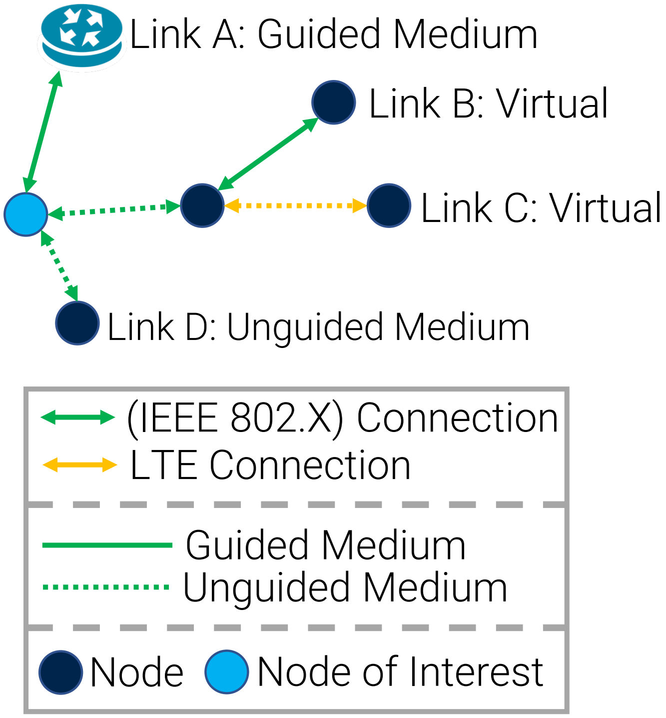


Figure 1.3: Link Definition

links for communication to receive or transmit data. For best understanding, the thesis uses the following definition for links.

**Links** are permanent, semi-permanent, or temporary active connections enabling communication, including physical and virtual connections. A connection can be a guided media (wire) connection between two interfaces, unguided media such as a wireless radio connection, or a virtual connection over two hops. Figure 1.3 depicts four different links, one guided (wired) link, one unguided (wireless), and two virtual links through a wireless interface. The virtual links can use the same wireless interface or different interfaces. Virtual links allow intermediary nodes to act as RAN translation from one access network type to another, enhancing the reach and capabilities.

**Paths** indicate the direction traffic transmission or reception takes over specific links.

Discovery handles the network awareness functionality using different protocols and frameworks with unique and independent mechanisms. The Decision module manages the path selection characteristics and use. Developing novel path selection and usage techniques allows communication to use multipath in parallel. Utilization maintains performance metrics, collects link utilization data, and aids the path decisions or other modules, including Dynamic Adaptation.

Moreover, the utilization module handles the network sharing frameworks, including NetShare, discussed in [25]. NetShare, developed as part of this thesis, is a network awareness-sharing mechanism for providing neighboring nodes with network dynamics information. Link utilization is independent of path decisions, and its importance is to ensure correct usages of links since a single interface may create multiple virtual links. Data collection collects, preprocesses, and formats data for the other modules to use.

Dynamic Adaptation implements heuristic, algorithmic, stochastic, or machine learning mechanisms to improve path selection and use decisions. Communication networks are environment-dependent, rarely static, and rarely achieve a steady-state situation except for sensor networks where the data collected and transmitted follow a predictable implementation. Hence, more dynamic networks such as UAVNs, VANETs, Machine to Machine (M2M) communication, and traditional networks require more frequent use of the Dynamic Adaptation mechanisms since the topology, link availability, and requirements are rapidly changing.

Soon, cities will deploy innovative communication-capable devices. Infrastructure will exist for a more comprehensive array of access networks (ANs), including newly developed wireless communication ANs. Wireless communication is more likely to advance through different ANs, such as satellite, next-generation cellular networks, microwave, visible light, and even unlicensed band communications.

Currently, each ANs uses proprietary protocols. Hence, future devices, even low-cost sensor devices, will contain multi-radio capabilities built into a single chip. CoopNet aims to use the multi-radio technologies in parallel through a programable system pushing for generalized universal link access instead of a subscription service approach.

In preparation, CoopNet extends the communication capabilities by enabling all node cooperation in network development and function implementation. Node cooperation enables a novel approach to RAN translation and novel link creations. Node cooperation help advance the following items:

- **Newly Developed Paths**: Non-network specific devices can implement network functionality, including routing, forwarding, and management through network function virtualization.
- **Enchanced Cybersecurity**: Intermediary nodes can act as middleware boxes integrating different types of security measures, including firewall capabilities, intrusion detection and prevention capabilities, and privacy enhancements.
- **Radio Access Network Translation**: Intermediary nodes can translate packets from one proprietary access network to another or forward messages transparently. Two forms of RAN translation include modifying every message to support the new communication standards or directly forwarding the message if the original sender preassembles the packet to match the translated access network.
- **Increase Performance**: Parallel communication through node cooperation increases the maximum bandwidth to aggregate bandwidth and can drastically improve throughput through the usage of two radios simultaneously.

### 1.3.1 Node Cooperation

An essential concept for CoopNet is the necessity of node cooperation in developing a network link. Node cooperation provides nodes at the edge an opportunity to aid in packet

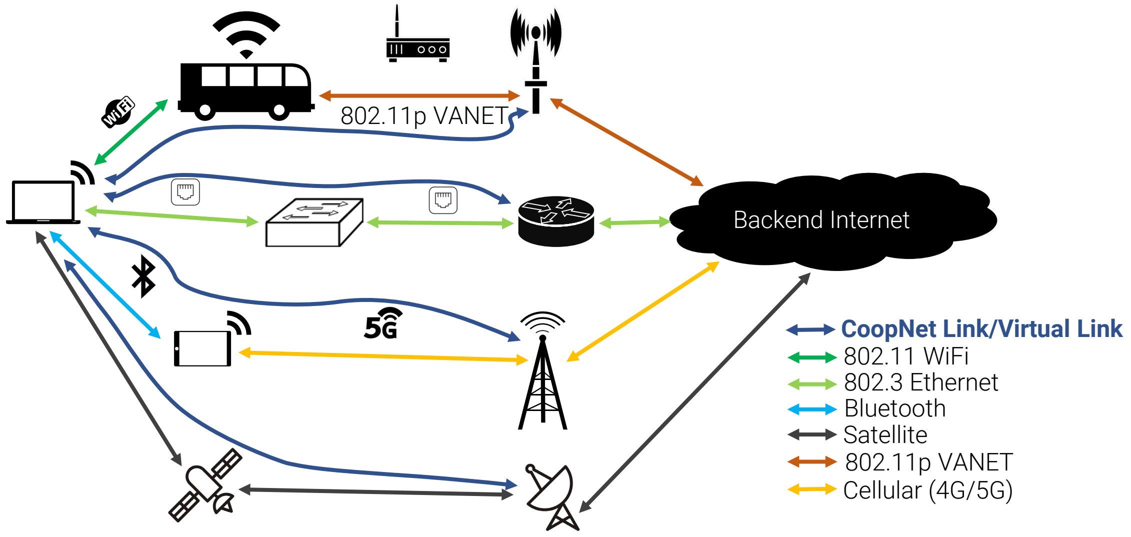


Figure 1.4: CoopNet: Cooperative Nodes Create Virtual Links

forwarding, switching, replaying, routing, and communication functions. Figure 1.4 depicts a use case of how two idle nodes, an intelligent vehicle node and a cellular device, assist in the communication initiated by the client.

Simultaneous parallel dissemination of messages can exist using multiple paths developed through the commonly available node capabilities. Chapter 3 covers several methods for parallel communication integrated into CoopNet. To create a scenario like Figure 1.4, node cooperation is vital. Hence, CoopNet introduces and enables three cooperation classes: public, private, and no-cooperation.

**Public** cooperating nodes may assist in all networking functions, including RAN translation, forwarding, routing, and different applications such as network management, firewall, load balancing limited only by the available hardware and resources. Public cooperation indicates that a node is accessible and willing to participate in the communication. The sender, receiver, or both can identify publically cooperating nodes and request assistance. Like privately cooperating nodes, publically cooperating nodes can transmit data transparently and translate messages from one protocol to another.

The initiating node performing all packet formattings is ideal for best implementation, increasing performance by alleviating packet modification at the intermediary node. The cooperating node can use its address for message transmission for privacy protection. The originating node may append its address in an encrypted message envelope.

Afterward, the receiving node can transmit a reply to either the originating node or the intermediary node using dual encryption. The intermediary node can decrypt the first envelope and forward the message to the intended recipient upon receiving the reply. Moreover, if a sender request security features, such as intrusion or deep packet inspection, the original sender can omit the insertion of an address. Instead, the intermediary node can handle providing the message exchange.

Lastly, with the rapid IoT deployments offering extended coverage, a public node's responsibility can extend to provide adaptive network functionality based on needs. Public nodes can implement existing works utilizing cluster head selection for delivering routing functionality and scheduling by implementing the mechanism into CoopNet.

Similarly, static systems can implement software-defined network capabilities, traditional routing algorithms, or policy-based decisions using different CoopNet modes. Hence, public cooperation provides several benefits, including requesting the parameters for communication and which features to implement for better overall control.

**Private** cooperating nodes assist transparently, including direct packet forwarding, RAN translation, and more. A privately cooperating node can update header information in more limited situations but omit any self-identifiable information. Analogous to self-learning switches, private cooperating nodes can gather information about the sender and the receiver.

The intermediary node uses that data to determine the best action based on implementation covered in chapter 4. Certain functionalities do not exist with private cooperation, such as routing functions. However, most consumer-grade cooperation presumably provides private collaboration instead of public cooperation due to limited resources.

**No-Cooperation** indicate nodes that do not wish to participate in networking functions for differing reasons, including limited resources, privacy, or policy.

**Radio Access Network (RAN) translation** is possible with publicly and privately cooperating nodes and is ideal for improving communication by extending the capabilities to include neighboring node capabilities. RAN translation is similar to packet inspection, where the intermediary node can change the communication protocol and header information to allow communication to take place using a different access network.

For example, suppose a node has two radios available, an IEEE 802.11 WiFi interface and an IEEE 802.3 interface. A neighboring node has access to both WiFi and cellular communication interfaces. Using CoopNet, a sender can create a virtual cellular link using the neighbor's cellular access network. Data transmission between the sender and its neighbor uses WiFi as the communication medium.

In public cooperation, if the sender has information about the access network type two hops away and can generate the proprietary access network packets, the sender will create packets in the specific format required by the cellular access network. In private cooperation, or if the sender cannot generate the proprietary message structure, the intermediary node will update each message from the sender and any replies. Thus, creating a translation feature in networking.

There is a higher performance gain if the original sender can generate the packets correctly and create a virtual link two hops away. Otherwise, the intermediary nodes incur computational overhead from packet modification. However, using an idle interface to create a link for parallel communication provides a meaningful boost even with a less than optimal implementation.

RAN translation can enhance the discovery and communication access at the edge and core internet infrastructures. Moreover, node cooperation can aid in developing agnostic communication for use in the future and implement the five different modules of CoopNet.

### 1.3.2 Discovery

Discovery manages the path, link, interface, and node detection. The value of a programable approach is to permit an open-sourced style addition and removal of different mechanisms with ease. Node discoveries can include active techniques, such as hello packets, blind transmission, heuristic, algorithmic, and machine learning.

Similarly, passive approaches may consist of passive listening, historical data preservation, probabilistic computations, and more. Self-information such as functional interfaces and communication capabilities are necessary to determine link and path.

CoopNet offers a platform and path agnostic form of communication. CoopNet does not dictate the use of any specific protocol for discovery. Instead, nodes can use any enabled

**Algorithm 1** Discovery: The Initial Start

```
Input: Historical Data, Feedback, NetShare Data, Self-Capabilites
Output: Available Links
Discover and Determine Usable Links
if ($Historical\ Data = AVAILABLE$) then
    if ($Aging\ Time\ Not\ Expired\ AND\ Relevant$) then
        Validate Link
    end if
end if
While loop for adding and updating links
while TRUE do
    for $i = 1$ to $Number\ of\ Discovery\ Algorithms$ do
        if ($Discovery\ Algorithm(i)] = Enabled$) then
            Push(New Link) or Update(Existing Link)
        else
            Update/Add From Feedback Received
        end if
    end for
    if ($New\ Neighbor\ Data\ Exists(NetShare)$) then
        Push(New Link) or Update(Existing Link)
    end if
    if ($Self\ Capabilities\ Changed$) then
        Push(New Link), Update(Existing Link) or Remove(Existing Link)
    end if
end while
return $Available\ Links$
```

frameworks for performing the discovery. As Algorithm 1 depicts a skeleton structure of the programable discovery module. Historical data and available built-in interfaces provide the best source of possible link information upon initialization. Historical data needs validation since most links will no longer exist in highly dynamic topologies.

After initialization, CoopNet provides the platform for checking which discovery implementations exist and are enabled. For example, node beaconing is the most popular form of discovery, where nodes will send periodic session establishment messages. External inputs for possible link information include NetShare [25], supplementary data provided with the received packets, or network sensing. NetShare is a framework developed as part of this thesis allowing nodes to gather data an additional hop away.

For implementation, separate threads for each different interface type allow simultaneous discoveries to function. Chapter 3 provides several forms of discovery implementations, including methods of detecting nodes that provide private cooperation through a blind transmission and a timeout period.

### 1.3.3 Decision

The Decision module provides the application developers with maximum flexibility to use multiple paths in parallel when needing such granularity. Simultaneously, CoopNet provides several self-selecting utilization modes for different traffic types and scenarios for parallel multipath communication discussed in chapter 3.

The Decision stage provides default options for the path selection. In addition to default options, CoopNet allows using other mechanisms from the literature on single path selection. For example, path selection is possible through packet separation into classes [26], game theory [27], network and user characteristics such as Signal-to-Noise Ratio (SNR) or Quality of Experience (QOE) [28], machine learning, and more.

In CoopNet, when the application requests a specific path selection, Decision uses the requested parameters, even if not optimal. Some scenarios where requested path selection can be helpful include wireless sensor networks, [1], or delay-intolerant environments requiring a channel to stay clear for emergency message propagation [5]. Algorithm 2 depicts the path selection and arrangement overview of the Decision module.

Without specified path selection, CoopNet attempts to provide the optimal preference

**Algorithm 2** Decision: Network Usage

```
Input: Available Links, Feedback (From Other Stages), Messages For Transmission
Output: Link Usage Decision
Determining Best Link Usages
if ($!Available\ Link$) then
    Notify Message Generating Entity
    Call Discovery
else
    if ($Manual\ Path\ Selection$) then
        Transmit Messages Via Specified Path
    else
        for $i = 1$ to $Number\ of\ Available\ Links$ do
            Use Optimal Path Selection Frameworks
            if ($Machine\ Learning\ Best$) then
                Use Accurate Weighted Link Separation
            else
                if ($Traffic\ Class\ Required$) then
                    Separate or Request Traffic Based on Traffic Type
                else if ($Static\ Data$) then
                    Use Differential Data Transmission/Reception
                else if ($Multimedia\ Or\ Similar$) then
                    if ($Streaming$) then
                        Use Event Triggered Multipath Selection
                    else
                        Use Adaptive Window
                    end if
                else if ($Real\ Time\ Traffic$) then
                    Use Real Time Adaptive Selection
                end if
            end if
        end for
    end if
end if
Return Feedback
Schedule Periodic Metric Sharing Messages On All Links
return $Decision$
```

based on different categories. For example, a machine learning model can provide more accurate link selection than heuristic or event-triggered selections. However, machine learning is more useful in static environments since dynamic environments can have significant parameter variations leading to improper use.

Moreover, not all nodes have the resources or capabilities to implement a machine learning model. Hence, several different possibilities exist in chapter 3, including an event-triggered mode, differential data transmission, adaptive sliding window, or real-time selection.

Since the concept of parallel communication is still in the infancy stage, significant potential exists in the Decision module for network traffic engineers to develop frameworks for multipath communication. For example, traffic engineers can develop multipoint-to-point communication for Content Delivery Networks (CDNs). Different CDNs can provide content simultaneously to the various existing links or interfaces at the receiver end.

### 1.3.4 Utilization

The utilization module collects and uses metrics, including congestion, throughput, goodput, latency, hop count, power, and more. The utilization module has several primary functions. First, it periodically records the self-information, including current battery levels, interface usage, average use, and more, and shares the information with the other modules. Self-information is vital since a node must be aware of its capabilities, such as energy consumption, open connections, and data rates for correct link usage and cooperation.

The second function of the utilization stage is sharing metrics with other nodes. For intelligent networks, data sharing is vital in developing a fair network by informing all nodes of existing conditions. Hence, as part of this research and CoopNet, this thesis introduces NetShare, a framework that defines several message types to share metrics, request information, and create virtual links two hops away covered in [25].

Without a data-sharing platform, there is no way to know if communication capabilities

**Algorithm 3** Utilization

**Input:** Self-Information, Metric Data, Feedback, External Shared Data
**Output:** Utilization Measurements, NetShare Messages, Data Stream

```
Stage 1: Self Measurement
while ($Enabled$) do
    if ($Measured\ Metrics\ Functions\ Enabled$) then
        Measure, Compute, & Record
        Instantenous & Average Throughput, Latency, Rount Trip Time ...
    end if
    if ($Metrics\ Functions\ Enabled$) then
        Record Metrics
        Congestion, Packet Info (Count, Size, Type) ...
    end if
    if ($Network\ Metrics\ Functions\ Enabled$) then
        Record Network Metrics
        SNR, Tx/Rx Power, Utility, Availability, Passive Listening Captures ...
    end if
    For Advanced Metric Computations, Processing, Imputation, Reduction Pass to Data Collection
    if ($Received\ Data\ From\ Data\ Collection$) then
        Update Stored Metrics
        Transmit Dedicated NetShare Advertisement If Altering Neighbor Information
    end if
    return $Metrics$
end while
Stage 2: Sharing Framework
while ($New\ Data\ Received$) do
    Update Network Awareness
    if ($Link\ Establishment\ Possible$) then
        Send NetShare Request For Additional Information
        Send NetShare Solicitation To Establish Link
    end if
    if ($Neighbot\ Information\ Differs$) then
        Update Link Paramters
    end if
end while
while ($NetShare\ Enabled$) do
    Assemble NetShare Advertisments
end while
if ($Information\ Request\ Received$) then
    Send Response With Requested Information
end if
return $NetShareMessages$
```

or range are expandable using intermediary nodes for RAN translation or message relaying, leading to a loss of enhancement. Moreover, data sharing allows nodes to collect enhanced network awareness when they come online.

Since the Utilization module is increasingly computationally intensive and can have a significant overhead, by default, CoopNet disables most of the functionality to record information and compute different metrics. However, neighboring nodes and other modules can request a node to collect metrics for sharing and implementing specific mechanisms. Algorithm 3 depicts the functionality of the Utilization module.

The second responsibility of the Utilization module is to ensure transmitting periodic NetShare Advertisements with a minimum set of data to encapsulate a breadth of data while keeping the overhead to a minimum. The node can respond with a dedicated NetShare Response message when a neighbor requires supplementary information. A node can collect additional data more frequently when extra resources are available, increasing awareness accuracy and validity.

### 1.3.5 Data Collection

The data collection module is an essential part of the architecture. It is crucial to transmit, receive, buffer, queue, encapsulate, segment, and assemble data. Application developers have the flexibility to craft packets in any form. However, it is inefficient and challenging for the application layer to create packets in a wide range of formats. CoopNet defines two methods of accepting data from the application implementation, a direct and a preparation method.

**Direct** bidirectional communication allows an application to send and receive real-time data chunks or queue entire datasets without any preparation. Data collection polls the available links and, using the link parameters, will segment and assemble the data chunks to match the link limitations and requirements. Chapter 4 covers the link selection based on link ranking and machine learning.

**Algorithm 4** Data Collection

```
Input: Data Input, Resource Information, Available Links, Network Data, and Metrics
Output: Data Output and Processed and Formatted Metrics
    Data Received or Queued at Upper or Lower Layer
while $Data\ Available$ do
    if ($Data\ From\ Upper\ Layer$) then
        if ($Transmission\ Intent$) then
            Poll Available Link From Utilization
            if ($Available\ Links$) then
                Prepare Communication Parameters & Buffers
                Pass To Decision Module (Data, Traffic Type, Priority, Requirements...)
                Notify Upper Layer Of Communication Configurations
            else
                Notify Upper Layer & Call Discovery
            end if
        else
            Poll Available Link From Utilization
            if ($Available\ Links$) then
                Pass Data To Decision & Any Attained Knowledge of Traffic
            else
                Notify Upper Layer & Call Discovery
            end if
        end if
    else if ($Data\ From\ Lower\ Layer$) then
        if ($Data\ Assembly\ Required$) then
            Buffer Data, Assemble & Complete Missing Chunks
            Perform Data Reliability If Required
            Pass To Upper Layer Compiled Sets
        else
            Pass To Upper Layer
        end if
    end if
    Pass To Utilization Feedback For Metric Updates
end while
while ($Network\ Data\ Metrics || Internal\ Metrics$) do
    Check Request (Data Preprocessing, Normalization, Reduction, Imputation, and Formatting)
    Call Individual Implementation
    if ($Return\ Requested$) then
        return Processed Data
    else
        Pass to Dynamic Adaptation
    end if
end while
```

**Preparation** involves the signaling of data transmission intent to help deduce specification, type, class, priority, or other information. With preparation, buffer allocation and link reservation offer additional capabilities of transmitting data in parallel. Parallel communication requires rethinking data access and segmentation for transmission.

Currently, the notion of a sliding window with acknowledgment is ideal for sequential data transmission. However, with parallel communication, the sliding window maintained by individual links will contain gaps resolved by other connections.

Similarly, with the differential data mechanism, where data transmission exists in both directions from the beginning and end, the sliding window will have to increment normally and decrement towards the front. For this reason, data manipulation in CoopNet allows for multiple simultaneous implementations depending on the specific selection.

The second essential responsibility of the Data Collection module is to enable a programable platform for data preprocessing, normalization, reduction, imputation, and formatting. The data preparation platform offers two stages of programable control. Traffic engineers can modularly develop and code for the individual Data Collection stages and develop models combining the different pieces into a generalized structure. Chapter 4 covers the implementation of data preparation designed for machine learning and heuristic implementations.

For example, different data inputs can use exponential moving average normalization. One traffic engineer can develop a specific exponential weight average implementation that other traffic engineers can use. While the aggregated combination of the normalization, imputation and formatting for use in the Dynamic Adaptation can be a separate model that may provide usefulness to different machine learning models.

Similarly, data normalization can provide direct feedback for other modules, such as Utilization. Link utility is a normalization feature where normalized link utility among all links offers valuable information for NetShare and the Decision module.

The Data Collection module ties the application and message-generating entities with

CoopNet. CoopNet may function independently without following the traditional network TCP/IP Five-Layer Model. However, CoopNet inserts the Data Collection module between the application and transport layers to ensure backward compatibility.

Current communication standards operate by having the application layer dictate which transport layer to use and handle the opening of connections. However, in an agnostic network environment, the ideal situation is to provide a separate management entity to receive application layer messages and handle all transport layer functions since the different links will offer various properties and requirements.

In some cases, performance improvements exist if not all parallel links utilize the same transport layer. For example, a more stable connection can use features similar to TCP data reliability, but a less stable link can transmit direct messages utilizing an approach like UDP. Indeed, such a change will require rethinking data reliability implementations since currently, TCP uses a byte sequence and acknowledgment field for reliability. However, a selective repeat implementation can provide reliability while different links focus on performance.

Algorithm 4 depicts an overview of the Data Collection module. The upper layer can transmit data to Data Collection with intent or immediately. Data Collection polls for available links and segments data based on the individual link parameters and requirements when sending data directly.

However, initializing communication with intent provides several benefits. Communication initialization allows for proper buffer configuration and link reservation. With non-real-time data, data transmission can occur in parallel with different links transmitting data chunks from various locations in the object.

Upon receiving messages from the lower layers, instead of passing the data through without inspection, it is possible to perform all necessary and required functions, including data chunk assembly, reliability, security checks, and more. CoopNet can set up all essential buffers and parameters for the more advanced features such as object reassembly with connection preparation.

Moreover, Data Collection integrates with NetShare for sharing network information between the available nodes. The internal and network metrics collected from NetShare must be preprocessed and formatted correctly to improve network performance and provide a more accurate link ranking.

### 1.3.6 Dynamic Adaptation

The Dynamic Adaptation module offers different implementations for network improvement, including machine learning approaches, heuristic, algorithmic, or stochastic. For a programmable network, a modular implementation enables the research community, network engineers, and hobbyists to rapidly develop and test new network improvement solutions.

CoopNet does not limit the Dynamic Adaptation to path-selection decisions. Instead, Dynamic Adaptation solutions can aid any of the functions provided by the different CoopNet modules. A few examples of Dynamic Adaptation solutions include the following:

- **Network Awareness**: Passive discovery allows for channel sensing by listening to channel use and noise and collecting low-level signal data. The implementation of the Data Collection module can preprocess, digitize, normalize, and format the recorded information. A machine learning model can enhance network awareness through pattern recognition and profile fingerprint.

- **Path Selection**: Different implementations can offer more optimized path selections. A heuristic approach can select an optimal path based on available bandwidth, throughputs, utility, and more. Similarly, as shown in chapter 4, machine learning can provide load balancing weights to use links.

- **Best Neighbor**: Dynamic Adaptation can use data received from the Utilization module of neighboring nodes and available resources to determine the best node for implementing different network functions, including cluster-head selection, routing, and security. Similarly, Dynamic Adaptation plays an active role in sharing information and caching.

- **Adaptive Buffering and Parameter Setting**: Using the information output from Data Collection can allow the development of a model to more accurately provide dynamic buffer allocations and parameter setting in the link reservation and use. For example, it can help determine which link to use for reliability and which one to use for performance.

Algorithm 5 shows the overview functions for the Dynamic Adaption. Dynamic Adaptation provides several internal and external benefits. For example, the Discovery module implements different protocols and frameworks for network awareness, while Dynamic Adaptation provides a few features applicable to Discovery, including correct network fingerprinting. Moreover, link information extraction such as link quality, aging, routing, and more benefit from autonomous adjustments.

Much of the benefit of Dynamic Adaptation is recognizable in combination with the Decision module, where adaptive bit rate, data rate, baud rate, and optimal link selection improve overall network performance. Moreover, Dynamic Adaptation assists NetShare through optimal neighbor information sharing. A single NetShare Advertisement cannot contain all current information in many cases. For this reason, Dynamic Adaptation provides a mechanism for selecting which data to share and how.

## 1.4 Simulation Implementation

It is challenging to produce a large testbed for testing improvements in communication in research. Hence, often, performing simulations are a better solution. A popular active tool in academia is Network Simulator V3 (NS3). Initial simulations in NS3 of the modules with an equal split help check the validity and possibility of implementing a parallel form

**Algorithm 5** Dynamic Adaptation

```
Input: Responses From All Phases, Feedback Control
Output: Individual Feedback To All Phases, Historical Data
while $Discovery$ do
    if ($Link\ Discovered$) then
        Extract Link Aging Time, Routing Capabilities, Link Quality ...
    end if
    Aid In Next-Hop Selection, Parallel Routes, Link Discovery
    return $Discovery\ Feedback$
end while
while $Decision$ do
    if ($Non - Optimal\ Link$) then
        Determine Optimal Link, Send New Link Selection
    else
        Adjust Parameters For Better Performance
    end if
    if ($External\ Network\ Condition\ Poort$) then
        Adjust Link Selection To Eliviate Network Ailements (Congestion, Noisy Channel Reduction, ...)
    end if
    return $Decision\ Feedbak$
end while
while $Utilization$ do
    if ($New\ Metrics$) then
        Call Data Collection Data Preprocessing & Update Dynamic Adapation Model
        if ($Affecting\ Neighbor$) then
            Transmit Dedicated NetShare Advertisement Message
        end if
    end if
    return $Feedback$
end while
while $Data$ do
    if ($Data\ Queued$) then
        Determine Best Buffer Sizing & Overflow Techniques
        Notify Network
    end if
    if ($Data\ Incoming$) then
        Implement Functions (Security, Periodic Packet Information Extraction)
    end if
    if ($Data\ Transmission$) then
        Track Connections Parameters For Adjustments
    end if
    return $Data - FeedBack$
end while
```

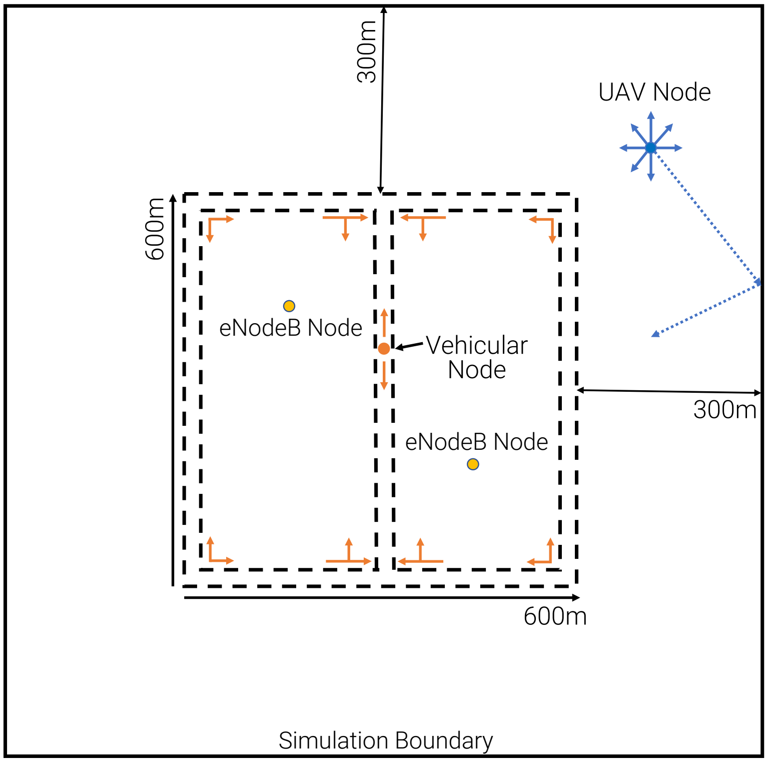


Figure 1.5: CoopNet Mobile Simulation Environment

of communication.

NS3 is a discrete-event simulator with numerous models and open-source code, including cutting-edge implementations for 5G wireless communication, device-to-device (D2D) communication, vehicular ad-hoc networking, cybersecurity, and more.

CoopNet simulations in NS3.26 helped demonstrate parallel communication's utility and overall improvement in a highly dynamic environment. The simulation environment consists of two hundred fifty nodes simulating vehicular nodes traveling in a predefined pattern shown in Figure 1.5. The vehicular nodes travel within the dotted lines and randomly enter and exit the central road.

The simulation environment uses two nodes operating as Long Term Evolution (LTE) Evolved Node B (eNodeB) depicted in Figure 1.5. Additional nodes are simulating UAVs traveling randomly within the simulation boundary at different elevations. When a UAV

Table 1.1: Simulation Parameters NS3 Simulation (CoopNet)

| **Attribute** | **Value** |
|---|---|
| | **VANET/UAVN** |
| Data Rate (VANET/UAVN) | (6Mbps / 9Mbps) OFDM |
| Tx Range (VANET/UAVN) | (300 / 600) meters |
| Speed (VANET/UAVN) | (120 /200) kph |
| Fading Channel | Rayleigh Fading Channel |
| | **LTE** |
| Fading Channel | Friis Fading Channel |
| Number of eNb | 10 |
| Mobility model | Constant Velocity Mobility |
| Road dimensions | 600m X 600m |
| Node density | 50-250 nodes |
| Packet Sizes | 400/700/1000/1300 bytes |
| Simulation Time (per run) | 15 seconds |

Node hits the perimeter, it travels back in a randomly selected direction. All nodes except the eNodeB static nodes have access to three access network types, including VANETs at 5.9GHz, UAVNs with WiFi D2D at 2.4GHz, and LTE.

Table 1.1 described the simulation parameters. The VANETs channel uses the implementation developed in ”Fast and Reliable Broadcasting in VANETs using SNR with ACK Decoupling” [6] to transmit the message to the intended recipient.

VANETs and UAVNs implement the Rayleigh fading channel model. The UAVNs channel uses a similar backbone implementation to VANETs using multi-hop communication, with an increased coverage distance of approximately 600 meters. The VANETs transmission range covers close to 300 meters.

Nodes using the VANETs access network communicate through a multi-hop approach, where intermediary nodes handle the packet retransmission in case of lost packets. UAVs use the 802.11n WiFi configuration implementing a device-to-device communication format with the UDP transport layer.

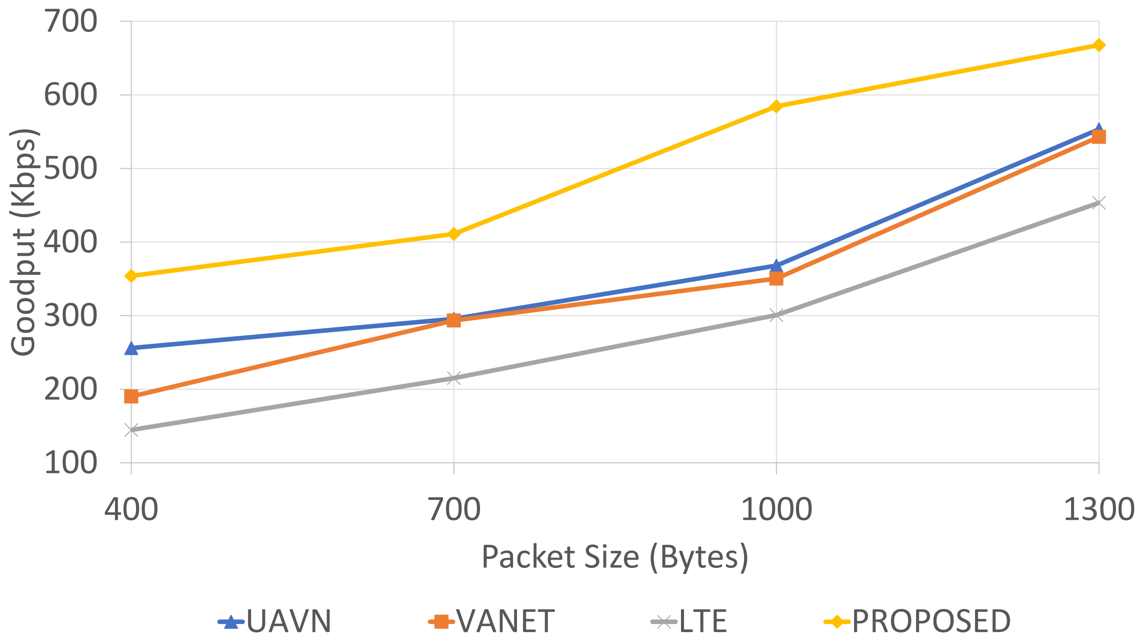


Figure 1.6: Goodput(Kbps) vs Packet Sizes(Bytes)

The intended sender and receiver using LTE communicate directly through the eNodeB. The simulation compares a single access network, where nodes transmit messages containing a random number of packets sequentially to parallel multipath transmission. With CoopNet implementation, nodes send messages in parallel simultaneously. The end nodes have access to all interfaces to access the different ANs.

Discovery uses the LTE UE establishment procedure, WiFi Ad-Hoc connection setup, and blind transmission for VANETs. The simulation implements a decision policy of equal split packets among the various ANs.

The Utilization module captures several influential network and communication metrics, including goodput, latency, and access network utility. Data Collection handles the buffering and reassembly of packets into messages.

Figure 1.6 provides the goodput of LTE, VANETs, UAVNs access networks, and the combined goodput for the proposed architecture, CoopNet. As expected in a controlled environment, the overall goodput shows significant improvement as the total available bandwidth across the different ANs provides an aggregated total larger than a single AN.

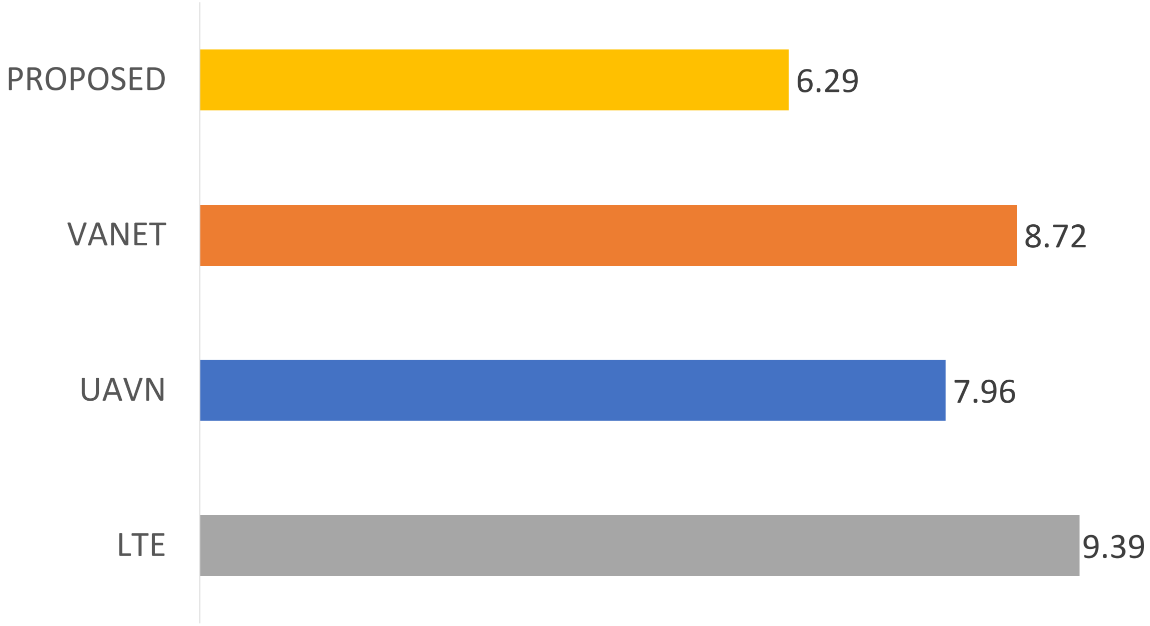


Figure 1.7: Average Message Latency (mS)

Smaller-sized packets show a more significant improvement of more than 38.3% in goodput. Larger packets show a gain of over 20.7%. Furthermore, the simulation provides a few additional vital insights. Splitting network traffic reduces congestion with fewer packets dropped due to possible collisions during transmission. Individual access network background traffic shows a reduction since fewer retransmissions exist.

CoopNet reduces the latency throughout all transmissions, except when small messages transmit through a single path. Parallel communication channels can disseminate packets simultaneously without interruption. The slowest link affects a message latency the most when using parallel communication since a receiver must receive all packets for a complete message. CoopNet implementation leads to a reduction in latency of more than 20.9% through the different ANs. Figure 1.7 depicts the average latency with messages ranging from one to ten packets in size.

The equal split Decision implementation limits the improvement in the simulation. A different implementation using a weighted link selection can provide better overall results. LTE has a larger latency in the simulation due to the built-in Acknowledgement (ACK).

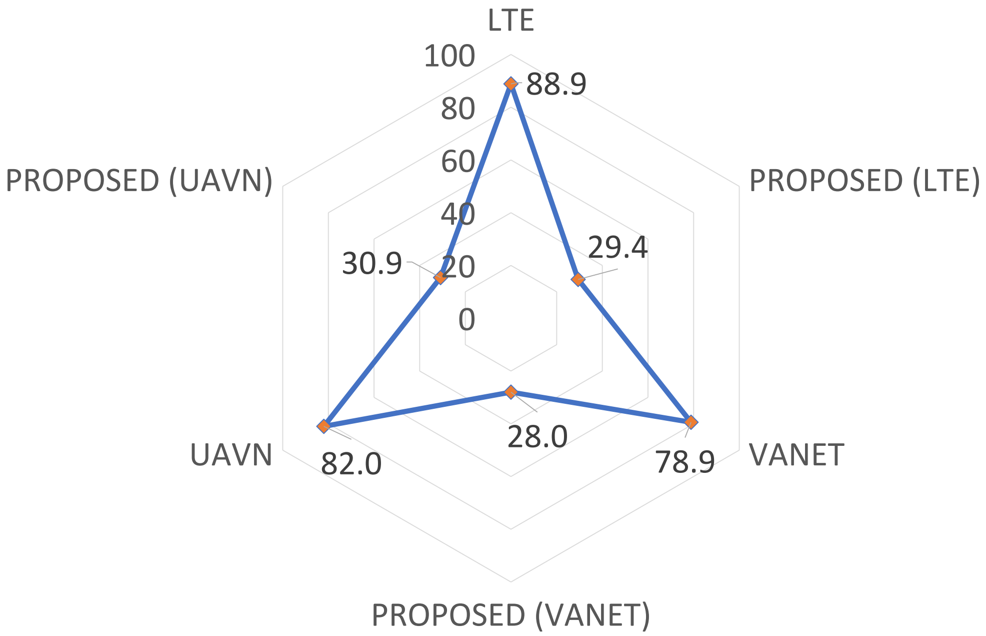


Figure 1.8: NS3.26 Simulation Results of Individual Channels Vs Proposed Architecture

VANETs and UAVNs do not use the same ACK feature. Instead, an implicit ACK of the overheard retransmitted message acts as an ACK to the original sender[6].

The most valuable benefit of CoopNet is the reduction in individual access network load. Cellular ANs strive for WiFi offloading to enhance overall cellular network performance and enable more nodes to exist within a specific area. CoopNet's use of idle ANs can provide drastic communication advancements to use otherwise wasted resources.

Figure 1.8 presents the channel utilization. Without CoopNet, the minimum access network utility exists in VANETs with 78.9% continuous usage, with 88.9% LTE utility. As the traffic intensity of an access network increases, the congestion and backend network queuing also increase, causing an exponential increase in delay.

It is difficult to recover from a congested scenario without incurring significant packet loss. CoopNet can assist in two ways. First, CoopNet can use idle ANs to transmit data. Second, CoopNet enables parallel communication for transmitting and receiving messages simultaneously. Figure 1.8 shows a drastic access network utility reduction from 88.9% to 29.4% for LTE, 82% to 30.9% for UAVNs, and 78.9% to 28% for VANETs.

## 1.5 Thesis Outline

Chapter 1 provides the introduction and motivation for a programable and horizontal architecture termed CoopNet. CoopNet provides a framework for developing NextGen networks, emphasizing parallel multipath communication.

The architecture can operate without modifications to existing hardware, focusing on ease of implementation and usage. Parallel multipath communication provides higher goodput, lower individual channel utilization, and reduces the delay in simulation.

Chapter 2 introduces address agnostic packet headers to improve CoopNet. Parallel, multipath communication exists in an environment where different interfaces containing different addresses must work together to transmit single messages of multiple packets. For this reason, universal headers provide a benefit to ensure the correct reception of all packets. Separately, universal packets can improve the notion of Multiprotocol Label Switching (MPLS), where path labels aid routing instead of generalized or address-based routing.

Chapter 3 defines the first few modules of CoopNet, including Discovery, four different Decision models, and NetShare. NetShare is a network-sharing framework part of Utilization that allows information exchange between neighboring nodes. Additionally, NetShare assists greatly with RAN Translation and enables the creation of virtual links two hops away.

Chapter 4 provides the data preprocessing framework for Data Collection and a machine learning implementation for more optimized link weight abstraction. Chapter 5 provides the ending discussion, future works, and conclusion in chapter 6.

# CHAPTER 2

# AGNOSTIC NETWORKING: UNIVERSAL PACKET HEADER

## 2.1 Introduction

Traditionally, communication procedures implement a point-to-point connection format where one node either transmits a message addressed directly to another node or first establishes a connection. Behind the scenes, a socket is created that establishes two-way communication between nodes. Socket programming uses an identifier, such as an Internet Protocol v4 (IPv4) or v6 (IPv6) address and a port number to identify which application handles the communication.

There are two types of sockets, a Transmission Control Protocol (TCP) and a User Datagram Protocol (UDP) socket. Both UDP and TCP sockets use the address to identify the node. However, TCP uses a four-tuple socket, where different combinations of the source address and port number and destination address and port number generate a unique socket. A single application can utilize multiple distinct sockets.

Socket programming is both advantageous and disadvantageous. Sockets provide an easy-to-use Application Programming Interface (API) where one or both nodes can open a socket and listen for data reception on a specific port number or transmit data using an API method for sending. However, standard socket communication does not provide a native implementation for parallel communication using different radio access networks (RANs). To use multiple Radio Access Networks (MRANs) simultaneously, implementation of Multipath Transmission Control Protocol (MPTCP) or Multipath QUIC (MPQUIC) is necessary [29].

MPTCP and MPQUIC offer a joining connection feature. Unfortunately, creating an application to use multiple ubiquitous paths implementing different access networks or protocols is currently not feasible due to the lack of systems implementing standards and mechanisms for parallel communication. Additionally, the dependency on the traditional network infrastructure forces any applicate solutions to circumvent the existing frameworks.

TCP connections use specific buffers and counters, such as a continuous sequence and acknowledgment counter based on received bytes. Adopting parallel communication with different frameworks requires the application developer to create a unique networking paradigm or the research community to build a new architecture, such as CoopNet. The design needs to consider a unique coupling between the transport and application layers to overcome the limitations, such as the sequence and acknowledgment buffers.

As depicted in shown in Figure 2.1, standard packet headers exist with every packet, including message headers, transport layer segment headers, network datagram headers, and link-layer frame headers, including a possible footer. Security features such as encryption, Internet Protocol Security (IPSec), Domain Name System Security Extensions (DNSSEC), or Transport Layer Security (TLS) appends additional header overhead on the encapsulated envelope. Sometimes several layers of redundant headers exist when tunneling information of different protocols, thus leading to inefficient network use.

The lack of performance and inefficiency drives the need for continuous advancements in communication to meet the demand for connectivity, security, and unique environments created by technological advances. The advent of everything connected through the Internet of Things (IoT), increased machine-to-machine communication, and big data have made the traditional networking framework obsolete.

Different communication advancements, including increasing the frequency spectrum [30], deploying analog fiber to the antenna [31], deploying lower earth orbit (LEO) satellite internet, and advancing the cellular communication core frameworks to support non-cellular links are in the initial phase of release. Each new communication improvement provides both advantages and drawbacks.

Higher Radio Frequency (RF), specifically the millimeter wave spectrum in 5G Next

Radio, is undoubtedly the next step for wireless communication improvements providing data rates in the Gbps. Millimeter-Wave shows much promise, but not without challenges. Millimeter-wave suffers from a necessary line of sight, reduced transmission range, and a lack of penetration. Moreover, to reuse millimeter-wave RF, smaller cells such as femto, nano, pico-cells to a no-cell implementation requires consideration for future infrastructure development [30].

Due to the drawbacks of higher RF, 3gpp plans on using dual network accessibility. ATSSS, part of the 5G proprietary core, enhances communication through non-cellular connections. Simultaneously, satellite communication with low earth orbit satellites is becoming a good communication platform to consider. Satellite mesh networks offer extended internet access but lack the necessary data rate capabilities and suffer from increased propagation delays.

Alternatively, CoopNet extends multiple access networks without a proprietary core and can utilize different access networks in an address-agnostic manner through programable networks. Enabling programable network features can produce novel implementations rapidly and efficiently. Open source networks provide several benefits, including shifting the focus towards a ubiquitous packet transmission employing a universal header.

CoopNet provides two communication handling techniques. First, it can handle socket generation, link selection, application message segmentation, and message assembly using traditional communication protocols, such as TCP or UDP. Concurrently, it can use custom transport layer protocols for transmitting data with universal headers.

Universal headers provide several benefits in a dynamic network environment dealing with a mixture of address styles and differing protocols, including reducing total overhead created by standard protocol headers and an address-independent form of transmission. In a programmable approach, universal headers allow picking and choosing only the options pertinent to the specific demand.

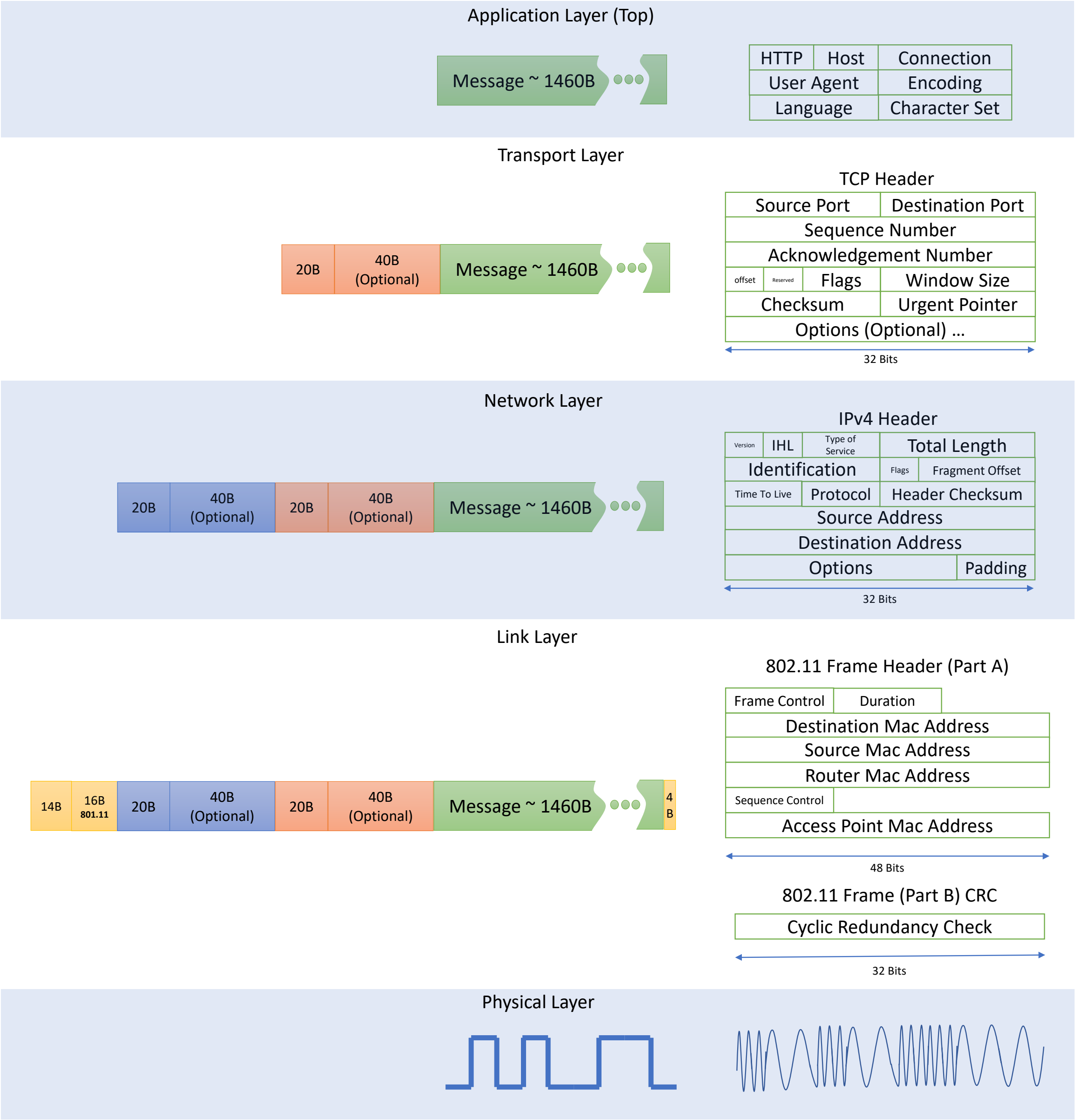


Figure 2.1: TCP 5 Layer Header Demonstration

## 2.2 Related Works

The advancement in network communication in the late 1980s and early 1990s paved the way for Web 1.0 to interpret static content that computer screens could display. Web 2.0 became popular with an ever-increasing improvement in technology and limited mobile communication for more dynamic two-way content delivery using Representational State

Transfer (REST), Simple Object Access Protocol (SOAP), and similar frameworks.

More recently, an increase in image and video data demand due to social media led companies such as Google, Amazon, Netflix, and Microsoft to deploy Content Delivery Networks (CDNs) for more optimal delivery of popular content to consumers. Due to the nature of CDNs, it is possible to leverage parallel communication using a multi-source to single-destination approach with minimal changes to the current network infrastructure.

In addition to CDNs, a new computing methodology, cloud computing, has become a popular concept. Cloud allows for synchronization, anytime access to data from any device, and extensive computational resources. Cloud computing and hosting empower new generation applications to exist, including applications to facilitate the use of the Internet of Things (IoT) [1]. IoT devices can sense, actuate, communicate, and upload information to a central location.

Traditionally, users generated the most significant communication demand. With IoT, machine learning, and Artificial Intelligence (AI), users alone do not motivate the requirements of next-generation communication such as 5G. Machine demands have become a significant factor in considering next-gen communication standards and protocols[32].

5G/6G releases aim to provide internet speeds in the range of Gbps, massive connectivity, ultra-reliable low latency communication, low energy consumption, deep awareness, and enhanced security [33][2]. Researchers present many approaches to enhance 5G. Some of the more popular methods are digital fiber to the last hop [34] and direct analog fiber to the antenna[30].

Passive Optical Networks (PON) with Wavelength Division Multiplexing (WDM) and Time Division Multiplexing (TDM) are a promising approach for the future to replace all electrical communication lines [31]. However, such an endeavor has an enormous cost prolonging massive deployment.

Heterogeneous networking [16] [23] is popular to assist in aggregating data into one transmission line via a medium like fiber[35]. However, regardless of the advances in

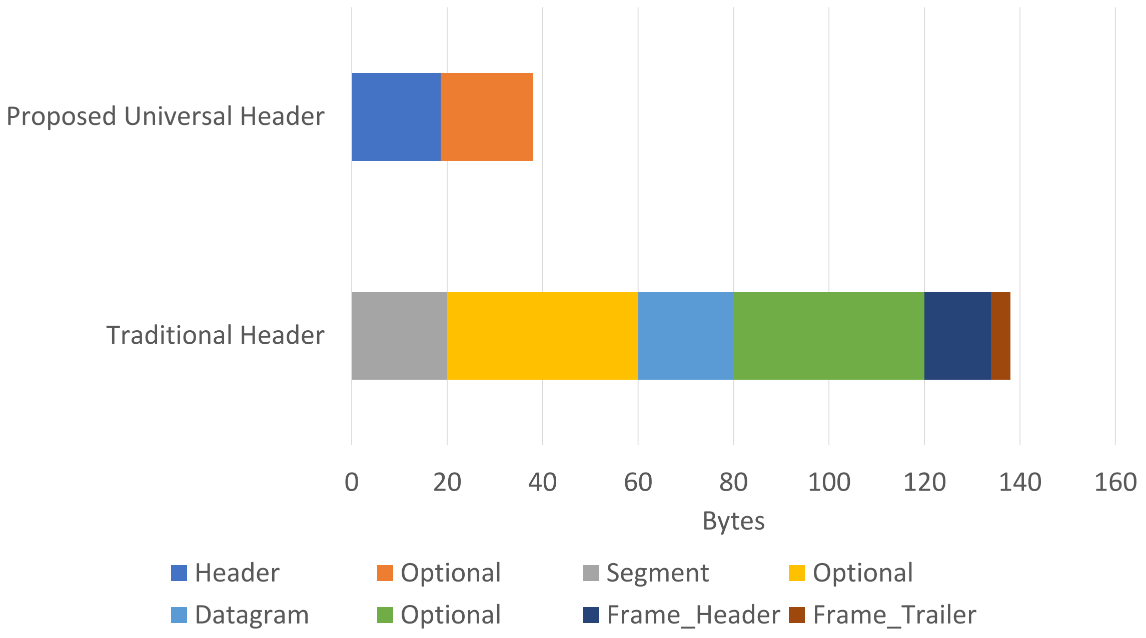


Figure 2.2: Header Size Proposed Vs. Traditional

communication, the standards and protocols for communication use the same TCP/IP five-layer model.

Some advanced communication developments, including Vehicular Ad-Hoc Networks (VANETs), take the initiative and implement non-traditional communication since the TCP/IP five-layer approach is unsuitable for meeting latency and reliability needs. Similarly, more environment-specific systems consider new strategies using cross-layer techniques and ad-hoc devices to device communication based on broadcasting [6].

## 2.3 Contribution & Motivation

In addition to cross-layering, the newly developed mechanisms for communication implement unique identifiers instead of using traditional internet Procol v4/v6 addresses, motivating a universal packet format to enhance address-agnostic communication. A universal packet format can improve performance by reducing overhead and simplifying the network dynamics.

Multiprotocol Label Switching (MPLS) is the closest implementation of an address-agnostic routing and forwarding method. MPLS does not dictate upper-layer implementation or specific use. Instead, MPLS encapsulates a frame and forwards the information based on a small label, reducing the lookup complexity while increasing performance.

With MPLS, developing virtual end-to-end circuits and backup paths without recomputing the path algorithms is possible. MPLS exists between the network and link layers, enhancing and adjusting the routing capabilities. While MPLS offers a good user experience and bandwidth allocation, it is a costly solution, not suitable for utilizing MRANs.

A rapid replacement for MPLS is Software-Defined Wide Area Networking (SD-WAN) [36]. A programmable approach like SD-WAN is expendable and allows custom implementations for networking. Moreover, SD-WAN is designed with cloud connectivity in mind, offering security, performance, and accessibility features.

Nevertheless, with SD-WAN, the traditional frameworks persist. Similarly, to use SD-WAN, the devices must be Software-Defined capable. Instead, an environment with a wide variety of clouds, numerous applications, several differing protocols, numerous nodes with multiple RANs capabilities can significantly benefit from a universal format rather than switching between countless proprietary protocols.

### 2.3.1 Motivation

Figure 2.1 provides a brief demonstration of the current traditional TCP five-layer network architecture and example headers associated with each layer. Application layer headers are usage and developer-dependent necessary in any architecture. Hence, the focus of chapter 2 is to provide reasoning on implementing a universal header for programmable networking to remove the overhead created by the traditional three layers, transport, network, and link.

Figure 2.1 shows the standard sizes for the different layered protocol headers using TCP for the transport layer and IPv4 for the network layer. Figure 2.2 displays a quick comparison of the universal header size and the standard headers.

Equation 2.1 depicts the packet size of the standard headers, and Equation 2.2 depicts the packet size of the proposed universal header. L is the message length in bits. $\alpha$ is the constant determining the TCP segment header (S) size where S is 20 Bytes, and $\alpha$ is between 1 and 3. The value of $\alpha$ is in increments of .2 since the TCP header must end at 4-byte increments above the required 20 bytes. For UDP headers, (S) is 8 bytes. $\beta$ is the constant determining the size of the network datagram header (D) where D is 20 Bytes for IPv4, and $\beta$ is between 1 and 3. Since IPv4 can grow between 20 bytes to a max of 60 bytes depending on the flags and optional settings. For IPv6, D is 40 Bytes, and $\beta$ is 1.

Similarly, $\gamma$ provides the size of link-layer frame (F) headers, and $\Omega$ depicts the link-layer frame trailer (T). By comparison, Equation 2.2 shows the universal proposed header size summation with L being the message size in bits and $\delta$ is the weight constant providing the size of the universal header ($U_h$) including optional fields. Since most communication traverses over numerous hops from the source to the destination, the communication overhead amplifies over a more extended range.

$$Traditional_{Packet_{size}} = L + \alpha S + \beta D + \gamma F + \Omega T \tag{2.1}$$

$$Universal_{Packet_{size}} = L + \delta U_h \tag{2.2}$$

Utilizing a universal packet does not ensure the same level of features as using the standard header. By default, the universal headers do not offer any of the TCP guarantees, such as data reliability, flow control, network adaptation, or sequencing. Implementing the required services is up to the application layer or communication management framework. The overhead may grow beyond the universal header size depending on the necessary services. The universal header aims to provide a minimum set of information essential for communication. The remaining features can be enabled as needed in a programmable communication approach.

Therefore, the universal packet format considers security, privacy, latency, throughput, and alleviating unnecessary congestion caused by traditional packet overhead. For security, the universal packet standard supports an anonymous communication platform by removing the readable source address requirement and making it optional. However, for cybersecurity attribution, the source address is necessary. Alternatively, populating the universal header with a direct link source address can benefit data reliability on a link-to-link basis.

## 2.4 Header Design

CoopNet can function as a supportive architecture between different layers providing benefits for parallel multipath communication or can operate independently. In the standalone implementation of CoopNet, universal headers offer several benefits, including reduced overhead, address agnostic transmission for MRAN use, and protocol-independent use.

In supportive use, CoopNet can use the universal headers accepting data from the applications. However, all packets require the standard headers to exist for communication. Suppose CoopNet uses universal packet headers and communication requires backward compatibility. In that case, the application layer message adds unnecessary overhead to standard headers for the added benefit of using multiple RAN simultaneously.

CoopNet supports encrypting data before passing the message for transmission. The Data Collection module offers the ability to develop and use a link-to-link or end-to-end encryption algorithm for added security. Setting encryption preferences is possible both from the application layer request and during link generation.

However, some environments require cleartext packets. For example, in Vehicular Ad-Hoc Networks, emergency messages and periodic Basic Safety Messages (BSMs) need to be in cleartext for all receiving nodes to use the information promptly [37]. Therefore, a packet header format needs to account for transmitting both encrypted messages and cleartext messages.

Figure 2.3 presents the proposed universal packet used in CoopNet. From the left, the

Figure 2.3: Universal Header

”Type” two-bit indicates if the message is encrypted or cleartext, allowing to speed up processing time by omitting the parsing of a message content if encrypted. If a message is not encrypted, a recipient may check the data for personal use.

The following 4 bits identify the address type. Currently, 15 address types suffice to include the primary address formats, including IPv4, IPv6, Media Access Control (MAC) address, content-centric address, IoT identifiers, MPLS labels, and custom addresses. A custom address allows private networks to implement independent identification instead of commonly known addresses.

The address type option considers next-generation content-centric networks and current network use of IPv4/6 addressing. If a specific address type becomes a universal standard, the address option can be removed and further reduce the header size. Since future network developments focus on a programmable approach, updating network equipment to use an updated packet header that includes additional bits for address types is more manageable. Like software-defined networking equipment, programmable equipment allows in-place updates to change its operational mode and changes to the core functionality. It is not feasible to change the universal header format in older equipment that does not implement a programmable environment.

The message size has 16 bits allocated. This message size field is necessary for determining if a packet is complete allowing the application layer to handle message corruption. The destination address takes up 128bits. The size of the destination address can encompass IPv6 addresses and naming conventions for content-centric communication.

The following 128 bits are optional for the source address. For privacy, the source address is not necessary. The source address can be omitted and left up to the application

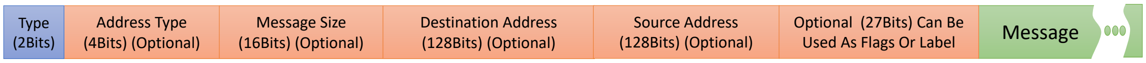


Figure 2.4: Universal Clear Text Header

layer to include the address as part of the encrypted message. However, for cybersecurity attribution, the source may remain a required option. The following 27 bits include optional flags and are large enough to fit a cyclic redundancy check for link-to-link retransmission without waiting for a sender's time out.

The optional field is for multi-use. It can encompass an additional identifier for content distribution centers allowing content reception from multiple sources simultaneously. One bit in the optional field provides forward congestion control alerting. Moreover, the optional field can assist with RAN translation by addressing two hops away from the sender. Finally, the message is of variable size of fewer than 65 Kbytes. Not allocating more bits to the message is recommended since performance increases with multiple smaller packet transmission than one large packet over numerous paths or many hops.

Assuming that most packets will be encrypted, more responsibilities can be delegated to the application layer or the network architecture to handle acknowledgments, congestion control, packet sequencing, or even establish symmetric encryption channels for the remainder of the communication between two users.

Allowing CoopNet to handle the link creation and connection establishments can provide added benefits. CoopNet can utilize parallel multiple-path communication with different modulation, technologies, and protocols through individual service selection. A universal packet header aids in using ubiquitous paths implementing various Radio Access Networks without forcing the use of a particular protocol.

### 2.4.1 Clear Text Format

The same universal packet standard exists for cleartext private messages with all fields as optional or reserved. In cases such as emergency message dissemination, a specific user does not always exist. The destination address of the packet standard is optional in cleartext since packet dissemination may be to all recipients. If no addresses are assigned, it is feasible to remove the address type, further reducing the message overhead. The message portion is left for the application to handle its formatting and presentation.

For example, critical safety messages disseminated by vehicles, BSMs, have a specific format. With BSMs, there is no need for any addressing since the application includes Vehicle_ID as part of the message, nor is the message size important. Therefore, the universal header adds only a two-bit overhead. Message size is essential in some cleartext cases, such as sensor data sharing. Hence, sensor messages add 18bits(bits) of overhead. Attaching the message size to the type implies the lack of address. Figure 2.4 shows the proposed cleartext universal standard.

In practice, the type can indicate the utilized fields. However, the type must grow from two to five bits to include various options. For example, the first bit indicates is a packet is cleartext or encrypted. The following four bits indicate the specific field added to the header. The following four bits with encrypted packets indicate using a source address and optional field and denote if the option field is a checksum or other selections. With cleartext, the four bits combination expresses the inclusion of specific fields. Finally, the options field needs to increase depending on the number of possibilities to distinguish the option and data associated.

## 2.5 Analysis & Evaluation

$$Universal_{Msg_{Latency}} = \sum_{i=1}^{N_p} \frac{L + \delta U_h}{R_{avg_i}} + T_{NTD_i} \tag{2.3}$$

$$R_{avg_i} = \frac{R_{link_{hop(1)}} + R_{link_{hop(2)}} + ... + R_{link_{hop(n)}}}{n} \tag{2.4}$$

$$T_{NTD_i} = T_{prop} + T_{proc} + T_{queue} + T_{ILR} \tag{2.5}$$

Equation 2.1 provided in the motivation section describes the traditional message latency based on bits divided by bit rate. Equation 2.3 provides the calculated latency using the universal header, where $\delta$ is the constant dictating the size of a single ubiquitous packet header ($U_h$). $N_p$ is the number of packets disseminated per message, L is the size of the message in bits, $R_{avg}$ is the individual link rate average given by the rate of individual link hops $R_{link_{hop(n)}}$ divided by the number of hops the packet traverses.

The first component of the summing function is equal to the transmission delays. The total communication delay given by transmission delay, propagation delays shown as $T_{prop}$, processing delays shown as $T_{proc}$, queuing delays shown as $T_{queue}$, as well as inner link retransmission delays shown as $T_{ILR}$.

$$Pkt_{Throughput} = \frac{1}{\dfrac{1}{R_{avg_i}} + \dfrac{T_{NTD_i}}{L + \delta U_h}} \tag{2.6}$$

$$Avg_{Throughput} = \frac{\sum\limits_{i=1}^{N_m} \dfrac{\sum\limits_{i=1}^{N_p} Pkt_{Throughput}}{Universal_{Msg_{Latency}}}}{N_m} \tag{2.7}$$

Retransmission generally occurs with a lack of acknowledgment with a timeout period or a negative acknowledgment. However, to eliminate overall network traffic, inner link retransmission of a specific frame exists when a packet is corrupt, adding additional delay noted as $T_{ILR}$. Since queuing delay calculations are different based on the different queuing approaches applied, such as Last In First Out, or random based such as Random Early

Detection (RED) queue[38], the queuing delay analysis differs.

Overall throughput uses the message latency and packet throughput provided in Equation 2.7, where Equation 2.6 provides the individual packet throughput. Intuitively, latency and throughput improvements exist due to smaller packet sizes. Even with CoopNet adding additional information for sequencing and guarantees, such as reliable data transmission, the size increase with the universal headers is smaller than traditional header overhead due to the redundant data, such as checksums, and duplication of useable information, including MAC address and IP addresses as different identifiers.

## 2.6 Simulation

Network Simulation V3 simulations depict the latency, throughput, and packet sizes between universal and standard headers. Since the conventional five-layer does not support parallel multipath communication natively, the simulation environments use single path implementations to compare a universal header fairly. Figure 2.5 provides the simulation environment topology for comparing standard packet headers and universal headers.

The simulation environment uses six nodes with three different radio access technolo-

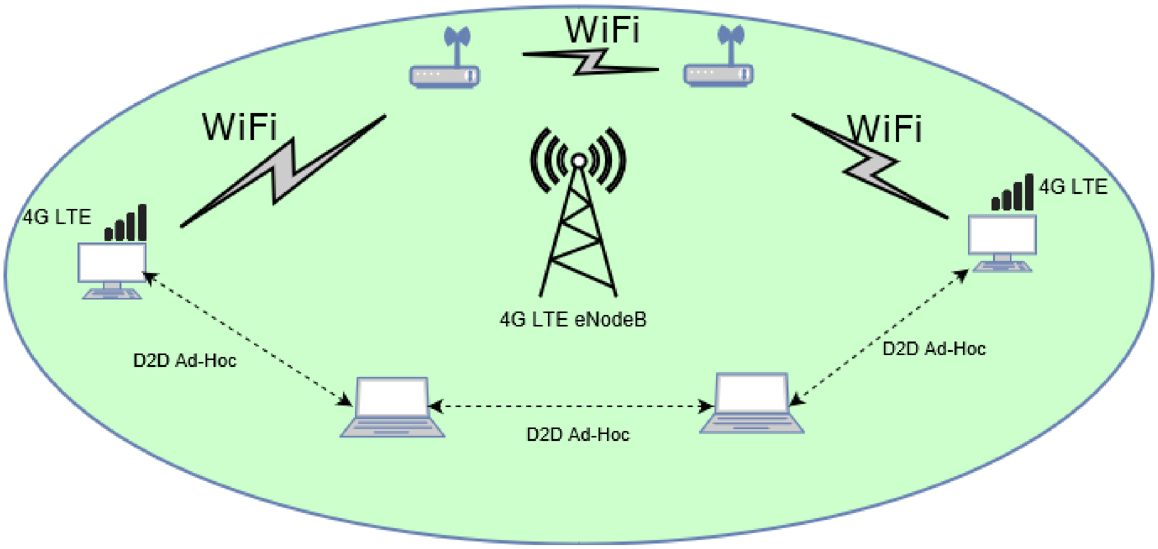


Figure 2.5: Simulated Topology

gies and one node acting as an enhanced node base station (eNodeB) for Long Term Evolution (LTE) communication. The three access networks employed are WiFi, LTE, and multihop vehicular ad-hoc communication. Two intermediary nodes have access to WiFi 802.11n.

Two additional intermediary nodes access direct ad-hoc device-to-device broadcasting capabilities using 802.11p Dedicated Short Range Communication (DSRC) standards from Vehicular Ad-Hoc Networks (VANETs). The two end nodes have access to all technologies.

Table 2.1 provides the NS3 parameters for the simulation, including the data rate, packet

Table 2.1: Simulation Parameters

| **Attribute** | **Value** |
|---|---|
| | **VANET** |
| Data Rate | 6Mbps OFDM |
| Tx Range | 300 meters |
| Fading Channel | Rayleigh Fading Channel |
| | **WiFi** |
| Data Rate | 54Mbps OFDM |
| Tx Range | 200 meters |
| Mode | Adhoc WiFi |
| Fading Channel | Friis Fading Channel |
| | **LTE** |
| Fading Channel | Friis Fading Channel |
| Path Loss | Const Propagation Loss Model |
| Number of eNb | 1 |
| Mobility model | Constant Position Mobility |
| Dimensions | 800m X 800m |
| Node density | 7 nodes (1eNb Node) |
| Packet Sizes | 100-1500 bytes |
| Simulation Time (per run) | 10 seconds |

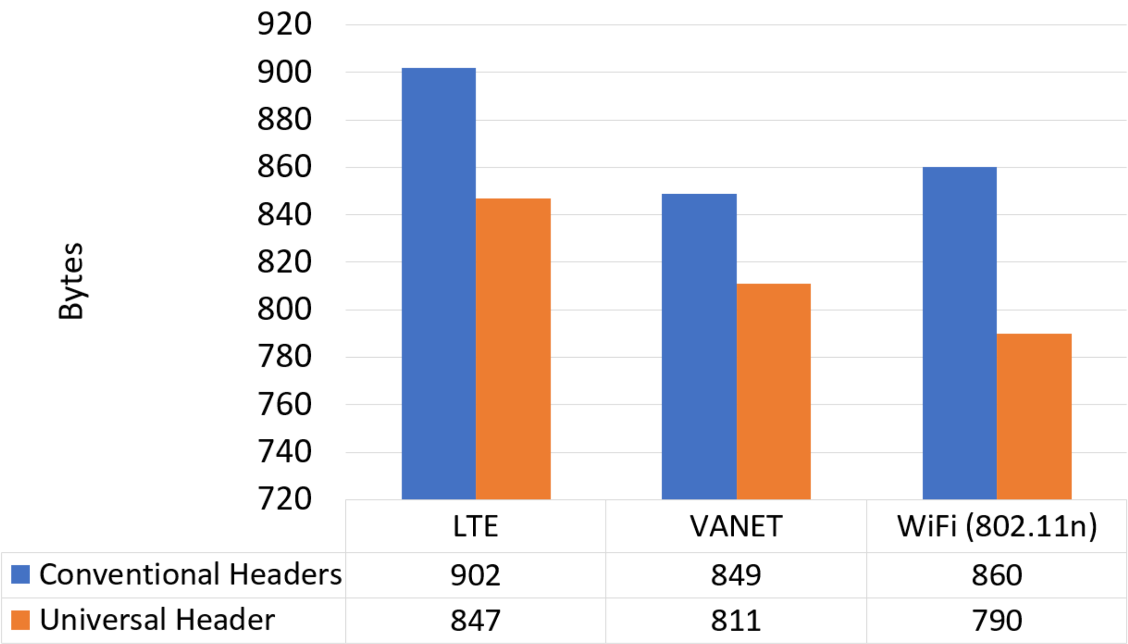


Figure 2.6: Average Packet Size

size, fading channel model, and approximate transmission ranges. The simulation uses standard and universal headers to compare the packet reception, latency, and throughput in three communication links.

### 2.6.1 Packet Sizes

The simulation environment randomly selects packet sizes between 100 bytes to 1500 bytes to simulate different traffic types, including multimedia, IoT, and data transmission. Figure 2.6 shows the average packet size in all three communication channels, as well as its counterpart using only the universal header. A reduction of 6.1% in LTE, 8.2% in WiFi, and 4.5% in VANETs exists using the universal packet header.

For the LTE implementation, specific headers information is necessary to employ the LTE modules of NS3, leading to a lower packet reduction in LTE than in WiFi. VANET packet size is smaller than LTE and WiFi since VANET employs an already reduced packet size as part of the protocol. Basic safety messages (BSMs) in VANETs contain specific data about a vehicle, and service packets utilize a cross-layer approach for forwarding. A

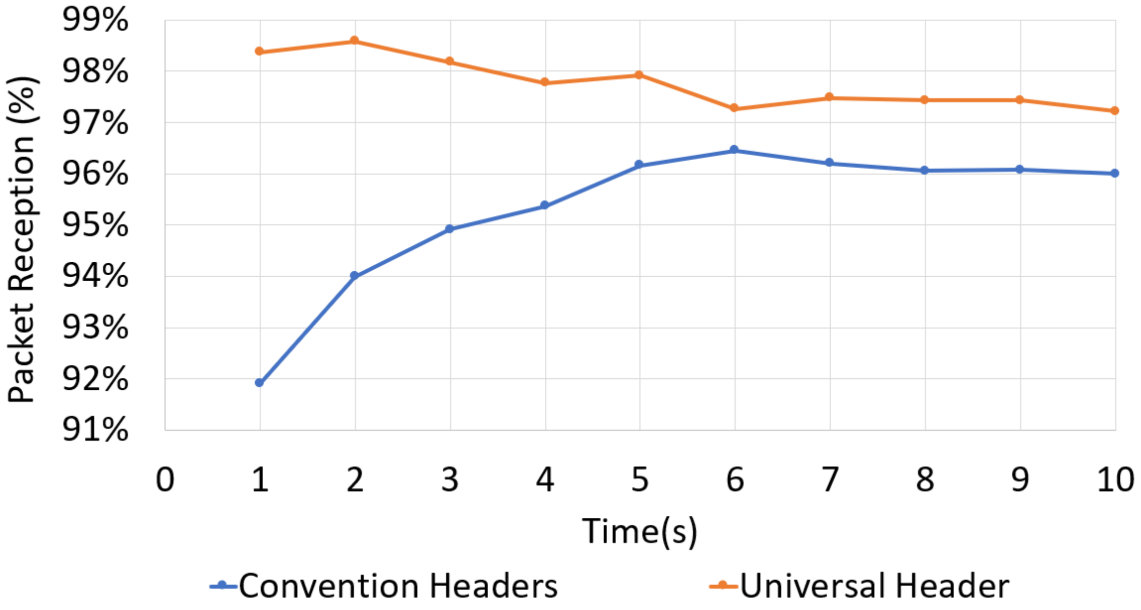


Figure 2.7: LTE Packet Reception

similar approach for infotainment and data packets uses the same mechanism.

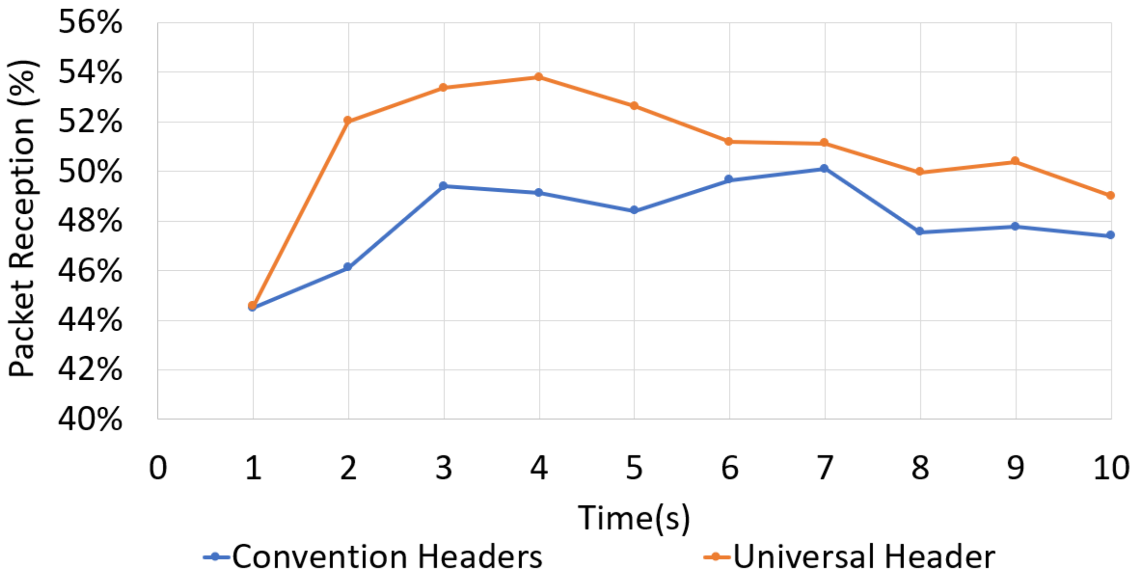


Figure 2.8: VANET Packet Reception

### 2.6.2 Packet Reception

Packet reception is vital, especially in harsh conditions such as an overly congested channel. With high congestion, smaller packets have a higher probability of transmission due to queue alleviation and overall congestion reduction under the same number of packet transmissions. The simulation depicts the packet reception in Figure 2.7, Figure 2.8, and Figure 2.9 over a simulation runtime of ten seconds.

The packet reception is low for VANETs and WiFi due to the heavy background traffic and continuous bidirectional transmissions. The average improvement in packet reception using VANETs is around 2.8%, while the packet reception improvement in WiFi is approximately 4%. Moreover, the simulation presents an LTE packet reception improvement of around 2.5%. WiFi does not provide any link recovery methods for ad-hoc communication, while VANETs uses a contention window approach for retransmissions.

The packet reception for VANET depicts an initial increase due to packet retransmission acting as ACKs for the previous transmission. If a forwarder does not overhear the retransmitted packet, it will retransmit upon timeout. More commonly, if a node in the vicinity did not overhear the transmission, the node forwards the message when its forwarding count down occurs. The packet reception slows to level out after a short period.

WiFi depicts an increase in packet reception, possibly due to the use of smaller-sized packets. The total congestion for WiFi is smaller due to a lack of retransmission. As the simulation depicts, a more significant packet reception improvement exists while using communication methods without built-in mechanisms for recovery, either link-level or end-to-end.

LTE packets reception with conventional headers increases since the queue begins to be processed. Similarly, since the LTE universal headers do not use a queuing mechanism, the reception drops as the simulation progresses but is initially higher. LTE maintains its recovery mechanisms even with the use of the universal header with the removal of unnecessary fields to reduce the size of the packet header instead of replacing all headers

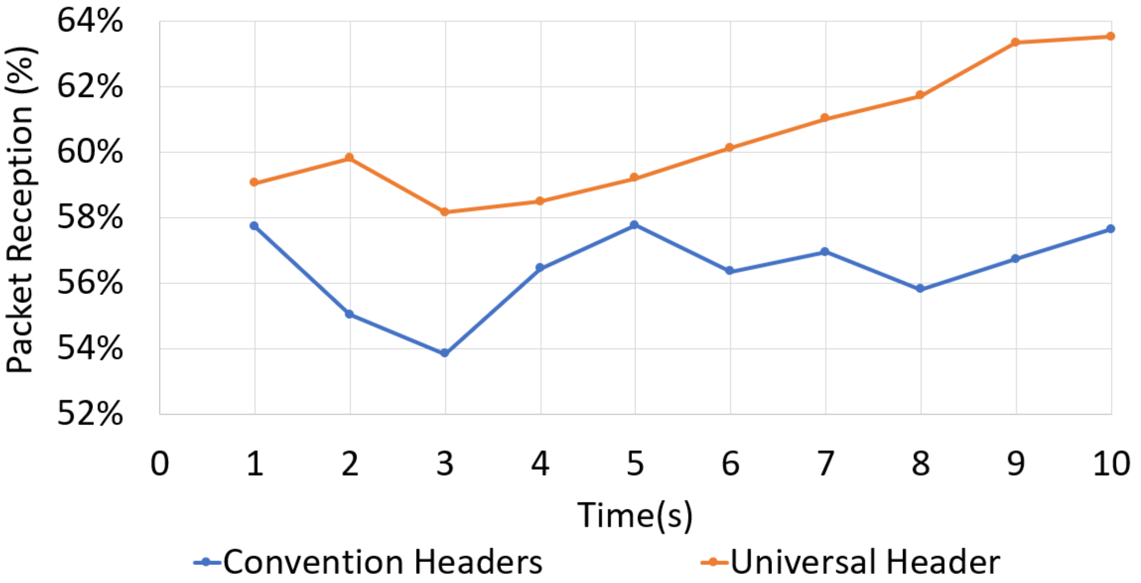


Figure 2.9: WiFi Packet Reception

in the LTE implementation. Otherwise, implementing the existing LTE NS3 module is not possible.

Reducing the protocol header overhead from existing packet transmissions is beneficial by allocating space for additional application data. In low-resource IoT environments, reducing header overhead can enhance long-term usage by reducing power consumption.

### 2.6.3 Throughput

Reducing the header overhead improves the throughput for several reasons, including lowering collisions, congestion, and packet loss. The processing delay is negligible in comparison to other delays incurred during communication. Reduction in the transmission and propagation delays exists due to reducing the number of bits over the bandwidth, or additional application layer data can occupy the saved overhead space.

The simulation depicts a throughput improvement of 3.5% in LTE, 1.8% in VANET, and 5.6% in WiFi. The VANET improvement of 1.8% shown in Figure 2.11 is a smaller increase due to employing a cross-layer approach that provides an already reduced set of

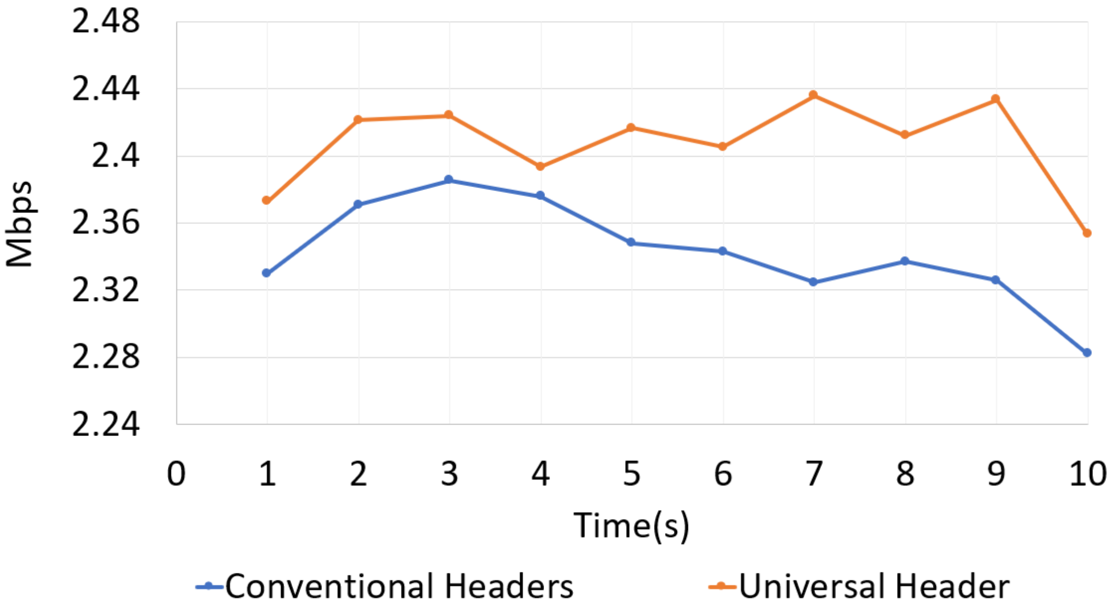


Figure 2.10: LTE Average Throughput

headers and VANETs incur higher packet drops. VANETs incur higher packet drops due to random channel switching between control and service channels every 50ms. The results are shown in Figure 2.10, and Figure 2.12. Similarly, the LTE throughput improvement in Figure 2.10 is smaller than WiFi in Figure 2.12 since header reduction in LTE is limited for proper implementation.

The optimal packet size for ad-hoc DSRC is under 100 bytes. Smaller packet sizes increase the probability of uncorrupted packet reception and reduce the overall congestion [39]. Thus a smaller packet utilizing ad-hoc WiFi would lead to a similar outcome. Since the simulation uses single path communication to compare universal headers, combining the different RANs would allow transmitting data in parallel, reducing the overall individual link utility and congestion and leading to a better result.

### *Latency Comparisons*

Figure 2.13 provides the average end-to-end packet delay. In all three occurrences, LTE, WiFi, and VANET, the overall delay improvement is more than 2.67% in simulation. The

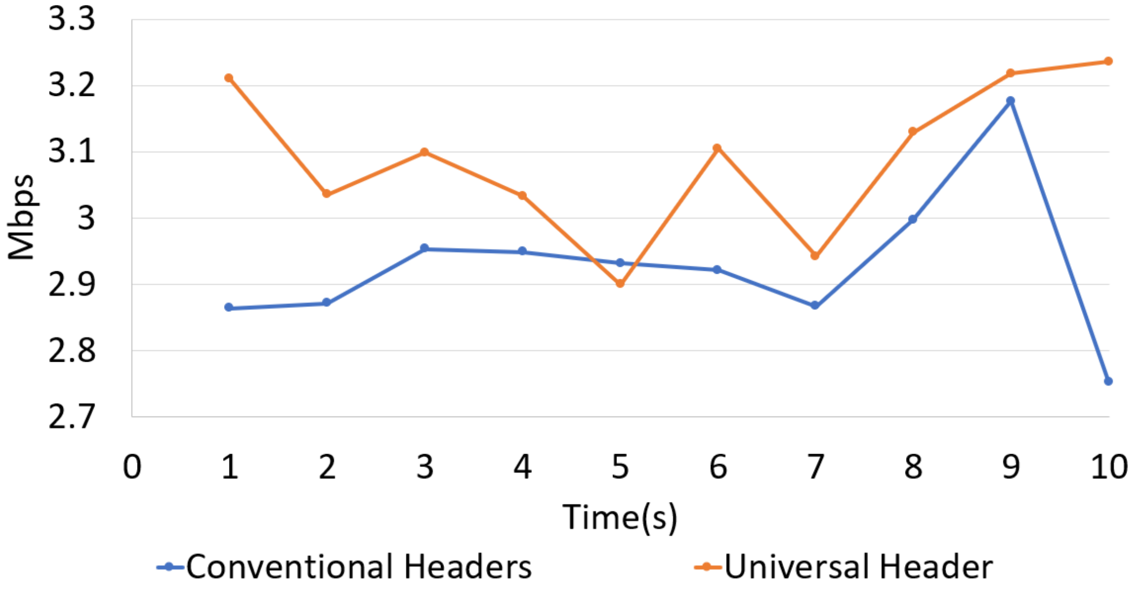


Figure 2.11: VANET Average Throughput

VANET channel depicts a lower overall delay while at the same time a lower latency improvement.

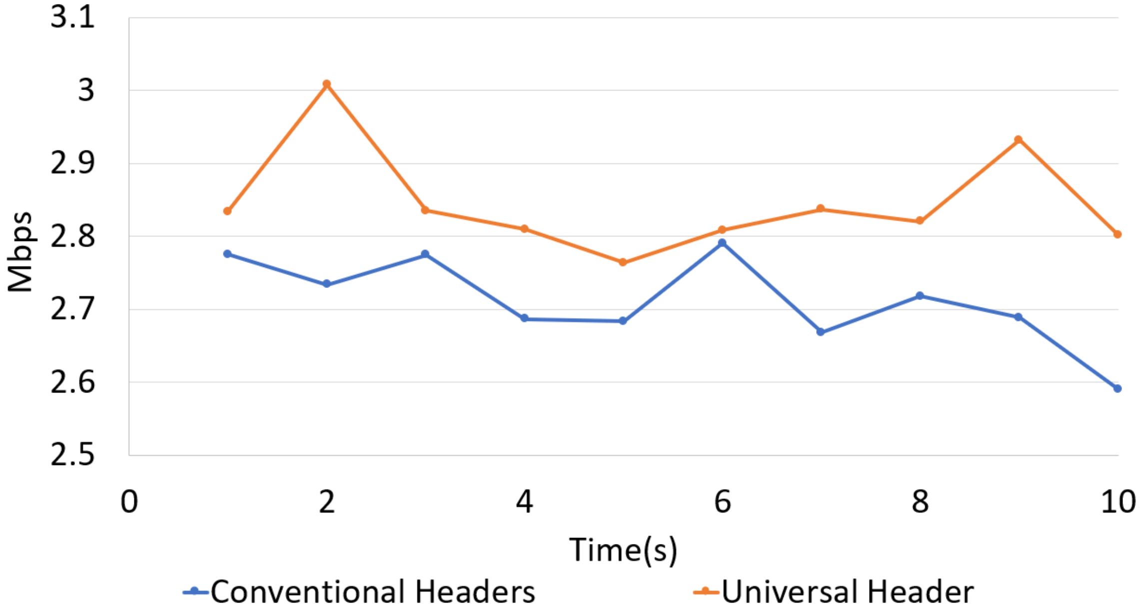


Figure 2.12: WiFi Average Throughput

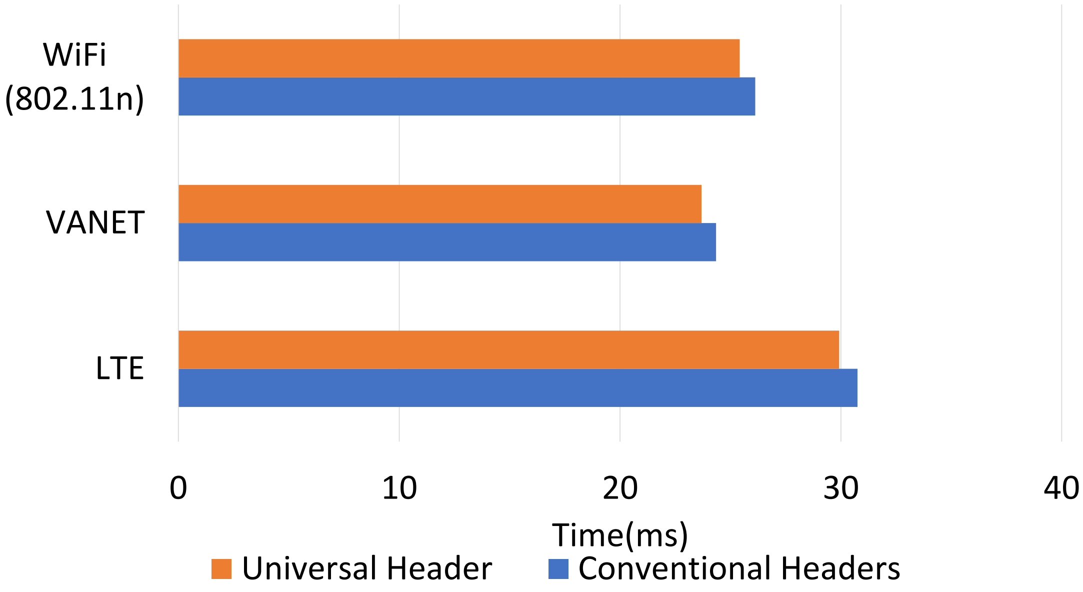


Figure 2.13: Average Packet End To End Delay

By design, the broadcasting scheme intends to disseminate messages as quickly as possible due to a highly dynamic topology subject to rapid changes. However, collisions and packet loss are acceptable in VANET. VANET uses small packet sizes with a reduced set of headers to reduce the possibility of collision. Hence, the total end-to-end delay improvement is more negligible in VANET than WiFi and LTE when using a universal header, minimizing the overhead.

## 2.7 Summary

Instead of implementing a protocol-agnostic approach between the link and network layers like Multiprotocol Label Switching (MPLS), it is helpful to implement a ubiquitous mechanism at the top layer between the application and transport layer for parallel multipath communication. Separating the protocol dependency at the top level allows each path to use independent implementations for services such as data reliability, bandwidth guarantees, security, adaptive rates, link selection, and connection setups.

Due to different link dynamics and capabilities, link use may differ significantly, even more so with new developments, including IoT Ad-Hoc Communication, Satelite Communication, and short-lived local communication. Defining a universal packet standard that can function across different Access Networks can reduce unnecessary overhead while offering easy integration to the various parallel link conditions. The proposed packet format assists next-generation parallel multipath heterogeneous networks, cooperative multi-point network architectures, and CoopNet.

For programmable networks, instead of depending on the packet headers for delivering particular services, such as guaranteed data delivery, throughput or bandwidth limitation, and more, it is feasible to allow the managing entity to provide the services using alternative measures. The managing architecture can implement similar features such as TCP to provide inorder byte correction, buffering, and reliability or employ TCP as the transport layer. If using TCP sockets can be managed by the architecture. Otherwise, a translation between the packet destination to a particular socket identifying a process needs to exist for backward compatibility.

The tradeoff of using a universal header is better local communication performance, a higher level of control for the application layer or managing architecture, and the removal of unnecessary archaic fields from the inception of the internet model at the cost of losing access to the traditional services provided by the conventional models. Losing access to traditional mechanisms implies that implementing a new packet format is not backward compatible, requiring an overhaul to the internet infrastructure for operability.

In practice, beyond specialized networks, such as sensor networks or VANETs a universal packet format is not feasible. Since existing hardware depends on specific layer formatting to accept or drop packets as part of the networking functionality, a universal header must continue using the standard headers, which negatively affects the overhead. Hence, implementing a universal format in practice is only possible when the supporting systems allow for complete programable control over the networking functionality.

# CHAPTER 3

# DISCOVERY, DECISION, AND UTILIZATION

## 3.1 Introduction

It is possible to classify current communication trends into two different approaches, vertical and horizontal. Both have certain advantages and disadvantages. Vertical communication heavily relies on existing standards, processes, and proprietary implementation—alternatively, horizontal approaches aim to develop a dynamic communication platform that allows for more advanced network engineering through a programmable API methodology. Figure 3.1 depicts the differences between vertical communication with and without cross layering, software-defined networking, and a horizontal approach.

Predominantly, the current implementation uses vertical communication in the backend

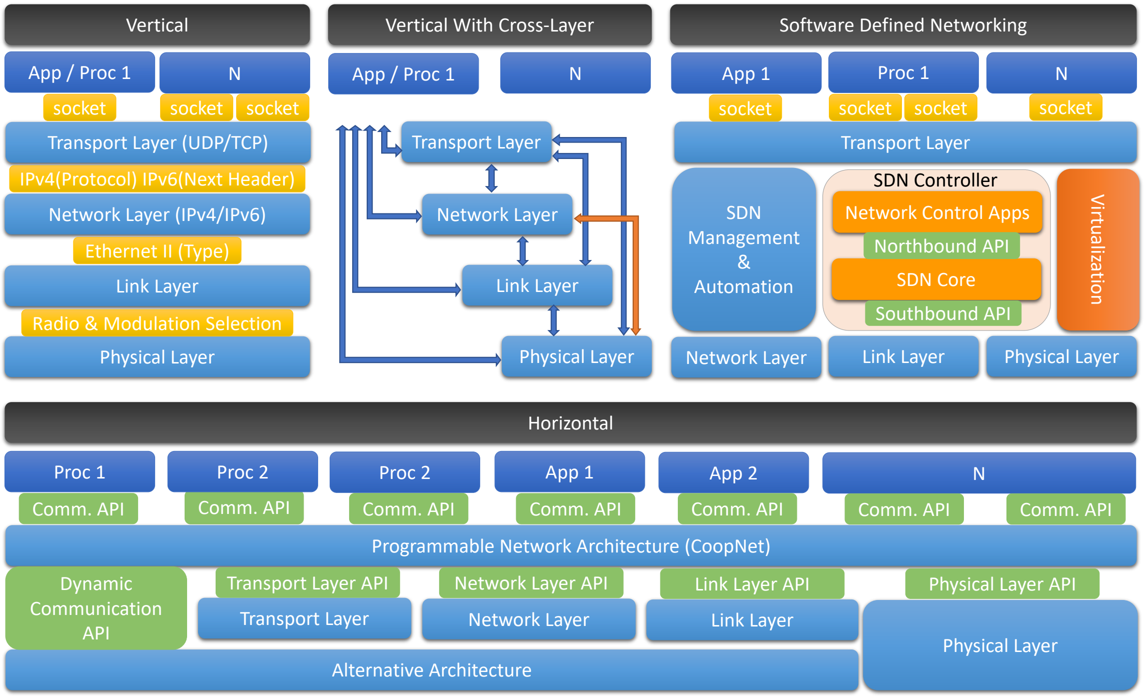


Figure 3.1: Vertical vs. Horizontal Communication Approaches

and limited software-defined networking (SDN), a programable approach at the edge. The vertical method requires using predefined protocol standards at each of the levels. The application layer message can use any application layer protocol, including a custom message structure.

However, the transport layer requires selecting either the Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). The network layer uses Internet Protocol version 4 or 6 for packets to be routable on the backend, not including the messaging protocols for network feedback, such as Internet Control Message Protocol (ICMP). Finally, the link layer is more diverse, with most traffic using the Ethernet II protocol.

The cross-layer approach allows omitting specific layers and using layers out of conventional order. In select environments, implementation of all the layers leads to performance degradation and inhibits intended operation. Using all layers is not optimal with highly dynamic environments, such as transportation. In a high mobility environment, the network dynamics change rapidly, and maintaining active transport layer connections for continuous transmission is impractical. Obtaining a TCP connection leads to unnecessary latency degradation for Ultra-Reliable Low Latency (URLL) applications, such as emergency message dissemination.

With limited resource IoT, using all layers affects power consumption. Moreover, due to significant header overhead with protocols, such as IPv6, several IoT environments implement a modified network layer protocol, IPv6 Low-Power Wireless Personal Area Network (6LowPAN), to reduce the address size for local sensor communication. With a modified header, the network equipment must be capable of using the address changes in 6LowPAN. Otherwise, packets are not routable, causing a backend dilemma.

Software-Defined Networking (SDN) is the most popular solution for generalized forwarding and routing. It allows match-plus-action implementations in a more horizontal approach to communication using SDN-capable devices. Instead of strictly forcing decisions based on layer-dependent header information, SDN devices can simultaneously use data from the headers of any layer.

With SDNs, it is possible to develop new network engineering applications through extensive matching capabilities, including security and reliability functions. Additionally, it is simpler to implement cross-layer techniques while using SDN as any fields from the transport to the link layer are accessible for decision making. Nevertheless, SDN requires using the existing protocols, and multipath communication focuses on using a single end node interface.

Strict standards are ideal for several reasons, including inter-manufacturer support, maintainability, and universal operability. Simultaneously, a rigid stack inhibits performance and communication in differing environments. A horizontal communication approach, as depicted in Figure 3.1, enhances communication by providing advanced communication interfacing while adding self-management capabilities for the communication parameters in different environments and network dynamics.

A programmable approach for communication can provide additional features in addition to traditional socket communication. Instead of opening a socket or multiple sockets with an intended destination, an application can request a connection establishment request and allow an architecture like CoopNet to set up the necessary connections for utilizing multiple parallel interfaces.

Afterward, the application transfers the message content to CoopNet and allows CoopNet to send the data using any possible means. In turn, CoopNet informs the application of successful transmission, passes the acknowledgments if necessary, requests additional data, and provides bidirectional connection parameters updates.

The reason why traditional approaches are limiting is due to the intended use during the initial development. Since the protocol stack development and standards consider only single point-to-point communication, implementing multipath transmissions requires non-optimal solutions, such as Multipath Transmission Control Protocol (MPTCP). MPTCP allows multipath communication through a join function beneficial in data centers and with cellular offloading.

However, MPTCP does not utilize parallel communication optimally since it uses the same initial TCP connections. Using the same socket requires the packets to be sequential for the application layer. After the transport layer, between different sub-flows, the sequence and acknowledgment numbers can differ in MPTCP but continue to depend on the traditional core TCP implementation. When devices have access to multiple interfaces and networks, utilizing a sequential form of communication leads to inefficient use of the available spectrum.

CoopNet can use sockets, UDP, TCP, and MPTCP for data transmission in a parallel multipath approach. Simultaneously, MPTCP permits using custom transport layer implementations practical with next-generation IoT and Machine to Machine demands communication. Through a programable API approach for communication, applications can expand their capabilities using advanced connection management.

Applications can request a connection, and the framework can handle the parallel multipath selection or single path selection if attainable. Network engineers and open-source hobbyists can develop novel frameworks to improve spectrum use through an API approach, removing vertical stack limitations.

Idle and dynamic spectrum access are popular topics in academia and industry. Heterogenous networking is more popular than ever, with mobile offloading being the biggest motivator. Current communication aims to implement several new protocols, including 802.11u WiFi Certified Passpoint and 5G Core Access Traffic Steering, Switching, and Splitting (ATSSS). CoopNet provides an alternative approach for enabling parallel multipath communication for upcoming communication devices.

For implementation, different agencies and committees provide specific frameworks. The Internet Engineering Task Force (IETF) provides Multi-Access Management Services (MAMS) in Release For Comments (RFC 8743). 3gpp's offers Access Traffic Steering, Switching, and Splitting (ATSSS) as part of the 5G core. Several other agencies, such as

Wireless Broadband Alliance (WBA) and European Telecommunications Standards Institute (ETSI) Multi-Access Edge Computing (MEC), aim to use many features similar to MAMS for advanced communication [8][9][10][13].

## 3.2 Related Works

In addition to low-cost devices, communication benefits healthcare, military, commercial, and consumer applications. Sustaining a massive amount of data exchange and seamless connections with minimal to no user input are two of the core challenges [40]. A current focus is to develop self-organizing, self-managing, auto-connecting, and parallel MRANs to provide optimal spectrum efficiency [14].

The IETF Release For Comments (RFC) 8743, MAMS, discusses the dynamic selection and combination of Access Networks (ANs) [9]. MAMS aims to establish an agnostic user plane where the usable interfaces can implement any transport layer protocols separating the control plane from the user plane. The control plane handles the negotiation of agnostic downlink and uplink networks, and the user plane manages the specific protocol requirements. Therefore, MAMS recommends independent uplink and downlink configurations.

Like CoopNet, MAMS considers utilizing path measurements and analytics for dynamic adaptation of path utilization and uses JSON as a delivery format over web sockets for the control plane. Moreover, MAMS suggests a policy-based optimal path, service discovery, lossless path switching or handling, adaptive access with multipath support. However, IETF does not currently provide an implementation supporting the specified features.

In addition to CoopNet [13], 3gpp's ATSSS [11] uses features similar to MAMS. Whereas WiFi Alliance's Certified Passport implements seamless connections for WiFi Hotspots [41]. The primary focus of parallel MRAN implementation is cellular plus WiFi synergy and offloading [42]. Manufacturers, such as Apple, employ MPTCP for homologous interface use. In addition to MPTCP, Multipath-QUIC provides applications more control over traffic policies and reliability [29].

3gpp's ATSSS offers coexisting 3gpp-cellular and non-cellular connections. However, even though ATSSS will revolutionize communication, providing parallel transmission through multiple interfaces, ATSSS dictates all devices to use the proprietary 3gpp's 5G Core.

Newly developed mechanisms, like ATSSS and MAMS, focus on the end-node/user equipment (UE). Similarly, CoopNet emphasizes the end nodes but may expand to intermediary and backend systems. CoopNet provides mechanisms that run on all hosts enabling node assistance in network development and communication through network function virtualization (NFV) using three cooperation classes [13].

As chapter 1 depicts, the CoopNet architecture consists of five modules. The Discovery and Decision stages are core components that handle the network awareness and data flow policies analogous to the steering, switching, and splitting in ATSSS.

The benefits of CoopNet are multifold. It is implementable in existing network testbeds and does not require a proprietary core framework, such as 3gpp's 5G Core. It is useable in different environments, and it can provide new infrastructure as a service (IaaS) through resource sharing.

## 3.3 Discovery

There are two types of link establishments, manual and automatic. Manual connections require a user or entity to connect using a set of credentials or procedures. Autonomous connections require no user input and are the main form of cellular systems. Underneath, nodes use a set of messages to establish sessions and handoffs in the case of mobility.

A node must gather network awareness before the message exchange for session establishments. In wired environments, both active and passive discoveries are conventional. In wireless settings, most detection is proactive using request and response protocols. For example, WiFi may use the Network Discovery Protocol (NDP) for network exploration.

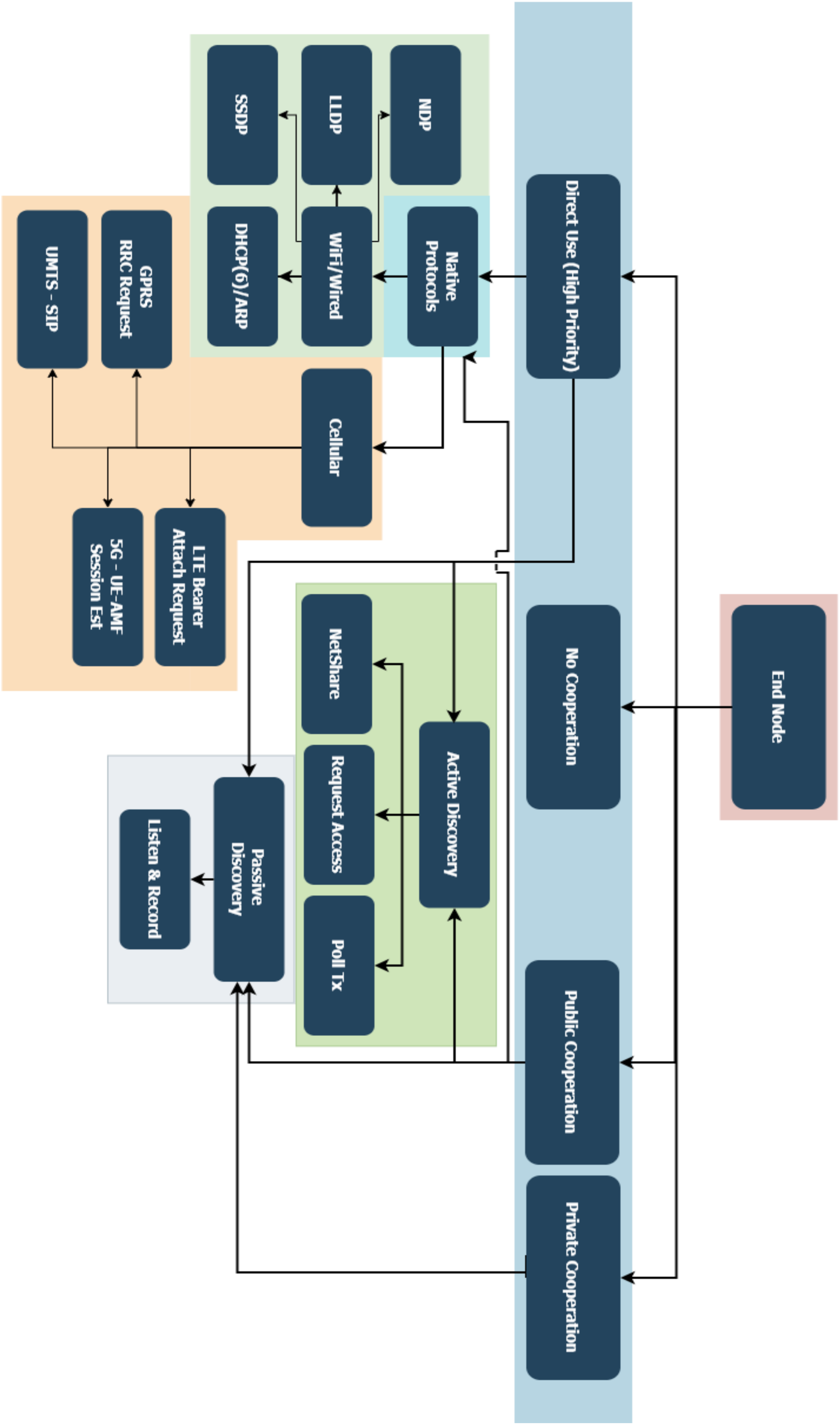


Figure 3.2: Discovery And Link Establishment Protocols

CoopNet does not inhibit existing discovery implementations. Instead, it offers additional discovery support based on existing interfaces and available radios. Native devices utilize discovery protocols shown in Figure 3.2 which periodically reach out for network awareness. Figure 3.2 depicts many of the traditional network discovery mechanisms and the additional mechanisms enabled by CoopNet. The same discovery protocols can be called directly from CoopNet. Likewise, CoopNet periodically checks for active interface connections when connecting manually.

As shown in Figure 3.2, CoopNet offers several discovery mechanisms for public and private cooperation, including blind transmission, transmission polling, a network awareness sharing framework named NetShare, and passive discovery based on listening and recording. subsection 3.3.2 provides the definition and implementation for NetShare.

CoopNet uses NetShare for public and private cooperation. Transmission polling is helpful for public collaboration, and blind transmission aids with discovering a privately cooperating node. CoopNet actively collects link information, such as keep-alive time and information relating to the link stability, availability from its hardware, and maintains neighbor awareness information for developing a network environment where node collaboration assists with networking functions.

A node with public cooperation indicates its availability to assist in network functions, including forwarding, routing, security, and more. A public node's responsibility is to be a translation relay, access point, or transparent relay for forwarding. In routing, the node's responsibility is to establish routing tables, address identification, and assist network management. Moreover, a public node can aid in security through a trusted network approach and security feature offerings.

Inherently, parallel communication improves cybersecurity with the node in the middle attacks since an attacker must capture traffic from multiple interfaces using different paths and the order of the packets if using encryption. CoopNet further improves security by using privately cooperating nodes leading to a more complex condition in determining path usage. Chapter 5 covers security advantages in more detail and possible future implementation.

### 3.3.1 Active Discovery

Most networking awareness employs an active approach that uses received information with and without an initial request. Request-based discoveries begin with a session establishment, attach request, neighbor advertisement, or broadcast message. The initial message informs the network of a nodes' presence to establish a connection.

One benefit of active approaches is implementing security, validation, and link parameter exchange. CoopNet uses several network awareness mechanisms and introduces a sharing framework, NetShare, for advance awareness and link establishment.

### 3.3.2 NetShare

NetShare extends existing approaches such as channel sensing, access point broadcasting, and session requests to improve network awareness by sharing self and neighbors' data. Figure 3.3 presents the flow diagram for the NetShare Framework. NetShare introduces several message types, including NetShare Advertisements (NSAs), NetShare Requests (NSRs), NetShare Correction (NSCs), NetShare Validations (NSVs), and NetShare Solicitations (NSSs).

NetShare transmits periodic transmission of NSAs containing accessible interface information shown in Figure 3.4. The type defines the different NetShare messages. There are eight bits reserved for specific options, including network cooperation, high or low priority, preference for encryption, verification, and authentication assistance. The interface count provides the number of interfaces belonging to the node sending the advertisement.

The interface options offer information, including wired or wireless, interface availability, next-hop exists, cooperation status, congestion, interface reserved, and two additional flags reserved. The reserved bits depict the validated and authenticated state when sharing

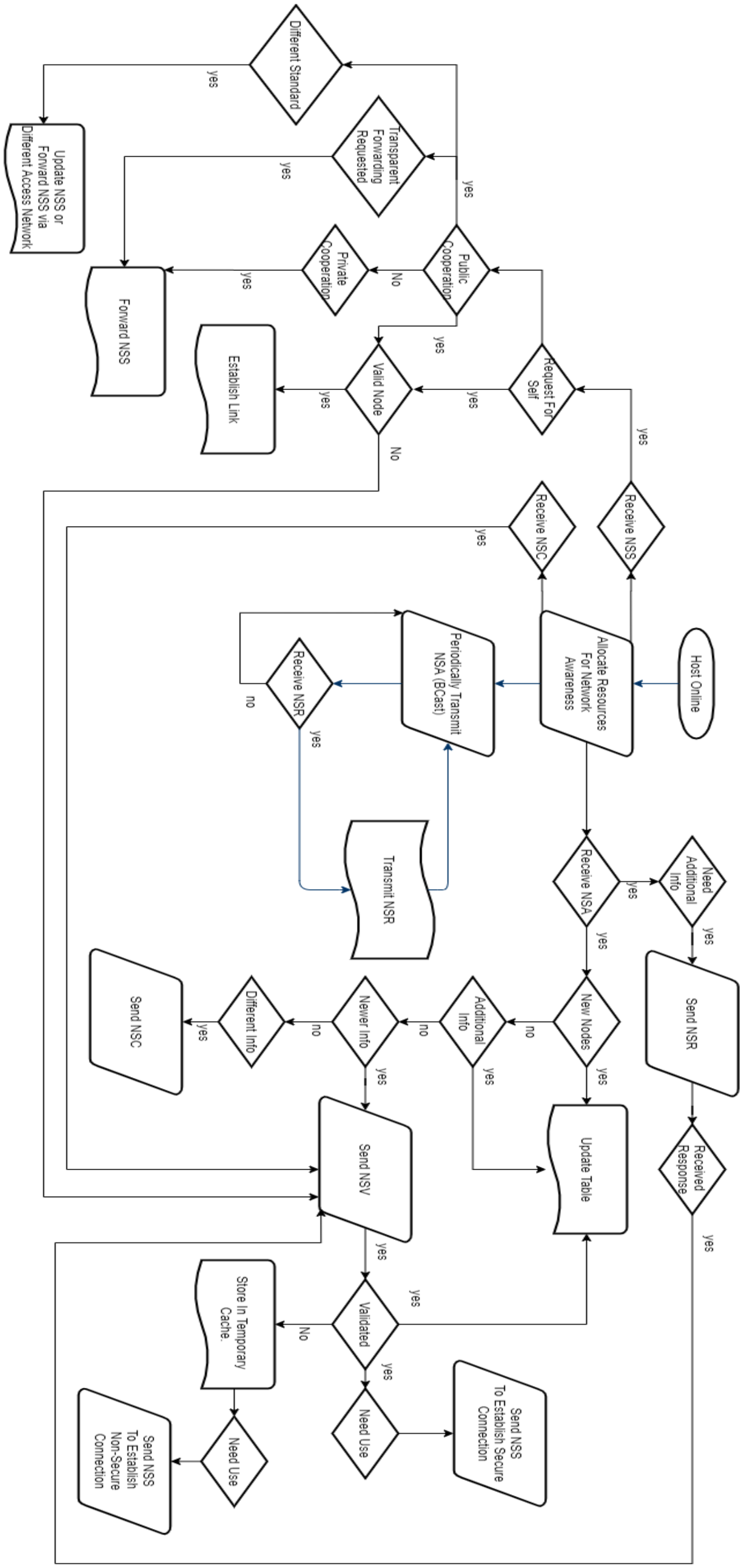


Figure 3.3: NetShare Framework Flow Diagram

neighbors' information. The identification type defines the format to use.

For example, some interfaces use Internet Protocol Version 6 (IPv6), which dictates a 16-byte address. The NetShare Advertisement's remaining space contains a semi-random set of neighbor interface information with the most recently verified and active links given a higher priority. NSAs are limited to sharing only first-hop neighbors.

Upon reception of the advertisement, a specific node can send a NetShare Request for additional information, a NetShare Solicitation for creating a link, a NetShare Validation, shown in Figure 3.7, to validate or authenticate an existing interface, and a NetShare Correction to update the advertised information.

NSRs, shown in Figure 3.5, utilize the id type and id to identify the interface of interest. Unlike the NSAs, the request for information in NSRs uses JSON Format. JSON allows for flexibility depending on the environment and the required information. NSCs and NSVs follow a similar format as NSRs.

Finally, a NetShare Solicitation, shown in Figure 3.6, initiates the establishment of a link. The NSSs packet structure maintains the standard fields plus a link count, an extended link option, source and destination identifier of the first hop, an optional second

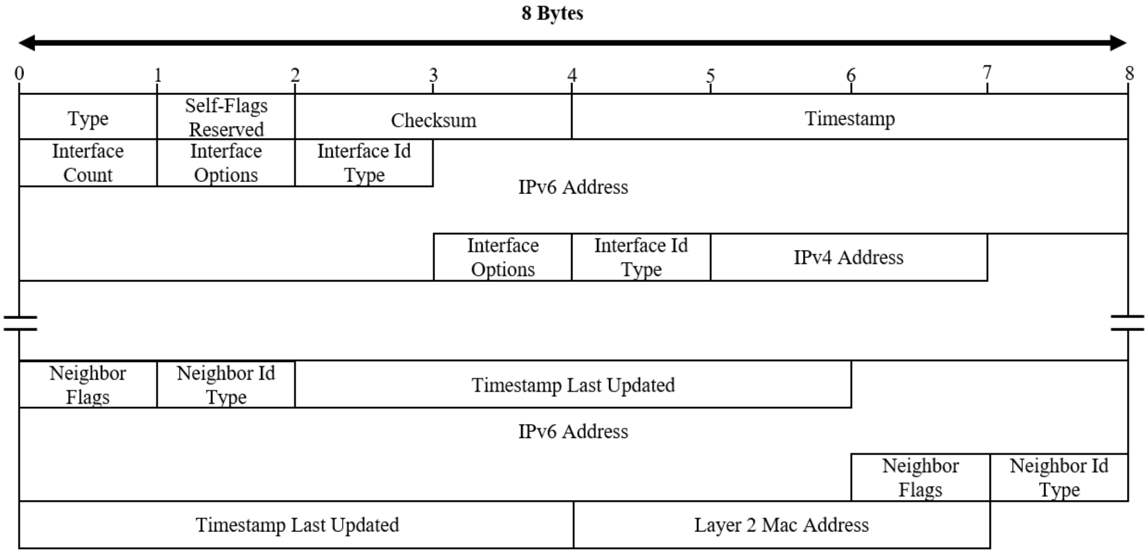


Figure 3.4: NetShare Advertisement

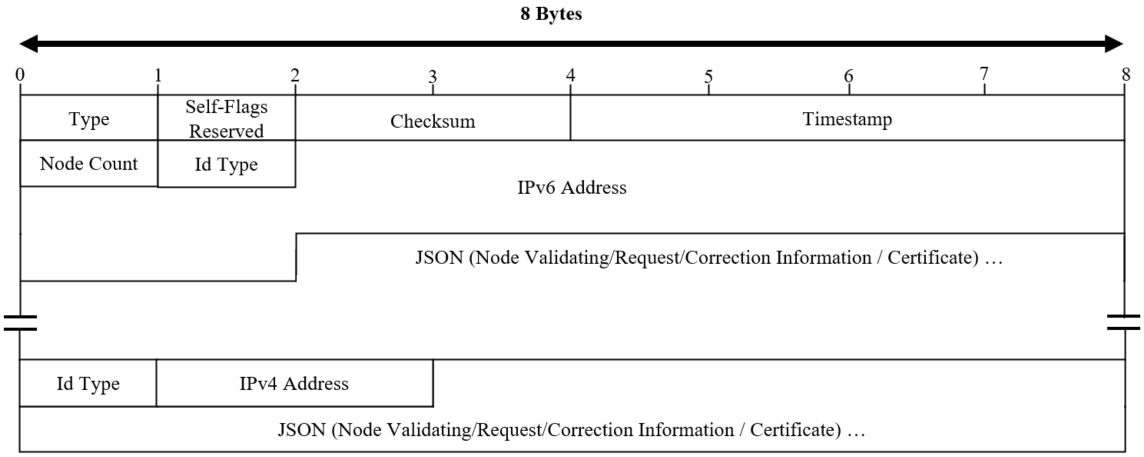


Figure 3.5: NetShare Correction/Response

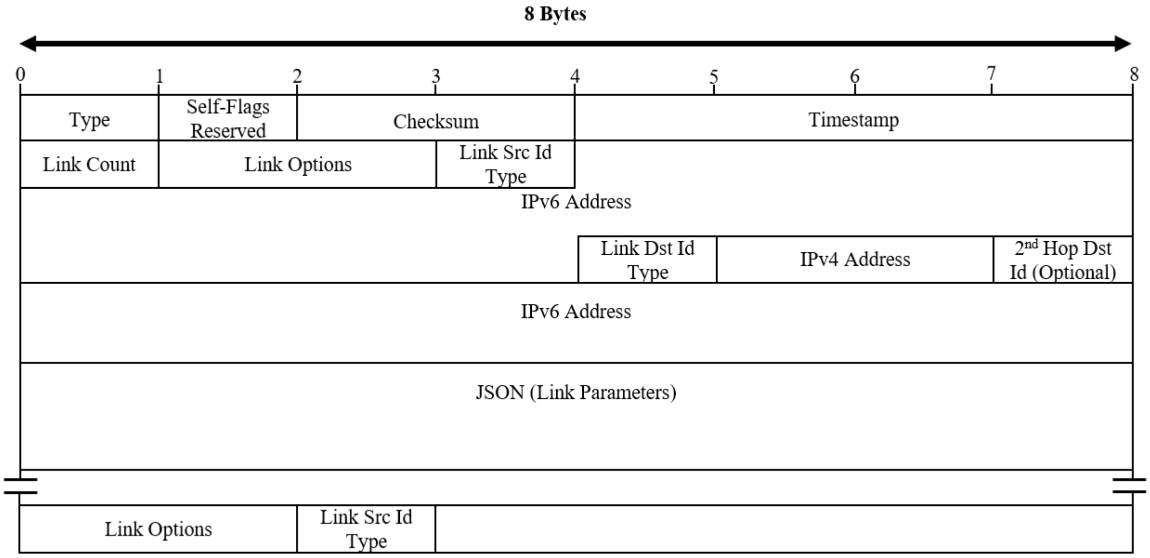


Figure 3.6: NetShare Solicitation

hop identifier, and the link establishment parameters. The extended link options allow for encryption preference, priority, transparent forwarding, routing, private, public, and the remaining reserved. The JSON parameters provide preferred link conditions, such as data rate, keep-alive period, timeout information, and specific security parameters.

NetShare empowers the possibility of having private, transparent forwarding, with links developed over two-hops, in addition to more traditional single-hop connections. Hence, a node can create a link outside of its communication range and access different RANs where

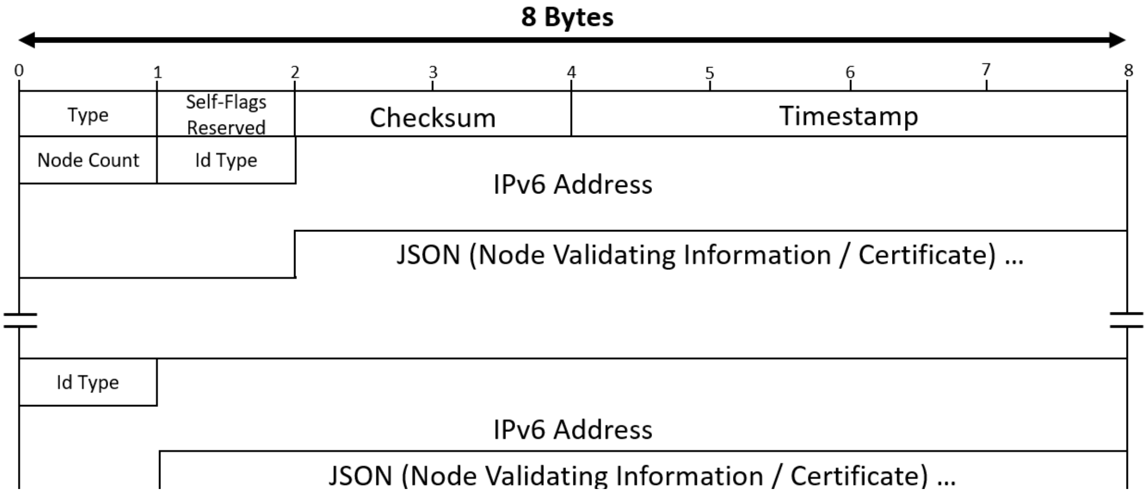


Figure 3.7: NetShare Validation

an intermediary node provides RAN translation shown in Figure 3.8.

The intermediate node alters the headers for outgoing and incoming packets with active collaboration. Whereas, if the end node can generate the correct Packet Data Unit (PDU) to match the second hop RAN, the intermediate node forwards the messages unaltered.

Transparent transmission of packets through an intermediate node leads to several challenges that require mentioning, including the dependency on a middle node for cooperation. If the central node disables cooperation or moves out of range, the end node must recover by establishing a new link.

If no other node exists, communication will shift to another active interface. Furthermore, translation adds considerable overhead since the middle node, in many cases, is not a dedicated networking device lacking a switching fabric. One solution is to ensure the ability to generate packets for different RANs in all end nodes.

### 3.3.3 Access Request

In addition to the NetShare framework, CoopNet uses traditional network discoveries and transmission polling. CoopNet implements a request and response method for a host to automatically or manually establish an ad-hoc connection with an idle interface. However, CoopNet defaults to automatic connection if it can validate a node's certificate using an arbitrary certificate authority or crowd validation.

A manual connection can provide an added security level if validating a node is unattainable. Nevertheless, CoopNet does allow connecting to non-secure nodes automatically forcing. CoopNet defaults to automatic encryption and sends messages through alternating links when connecting to an unvalidated link.

For establishing an ad-hoc connection, the process follows the following steps, similar to Figure 3.8. A node initiates either a unicast or broadcast message utilizing available information. The message will contain the preferred and non-preferred identifier types and information about its capabilities.

Upon receiving the payload, a neighbor will broadcast a message back, accept an identifier type, suggest a new address, keep the indicated initial address, or decline. If approved, the message reply will contain a suggested address for itself. The initiating host will transmit a packet acknowledging the provider's information or offer final changes.

### 3.3.4 Transmission Polling

CoopNet permits the use of intermediary nodes acting transparently as direct forwarding units. Transparent forwarding provides additional assistance when a node, such as a router, is outside the communication range or when a second hop node uses a different RAN. Transmission polling helps discover privately cooperating nodes. CoopNet introduces two transmission poll types, a loopback shown in Figure 3.9 and a blind transmission procedure.

In loopback's first polling procedure, a host periodically transmits a particular packet with no source identifier. The structure is similar to NSR packets utilizing a reserved type field. Upon receiving the message, a privately cooperating may retransmit the packet (unaltered). In wireless environments, the loopback retransmission requires a contention window time to reduce the possibility of collision if more than one privately cooperating node exists.

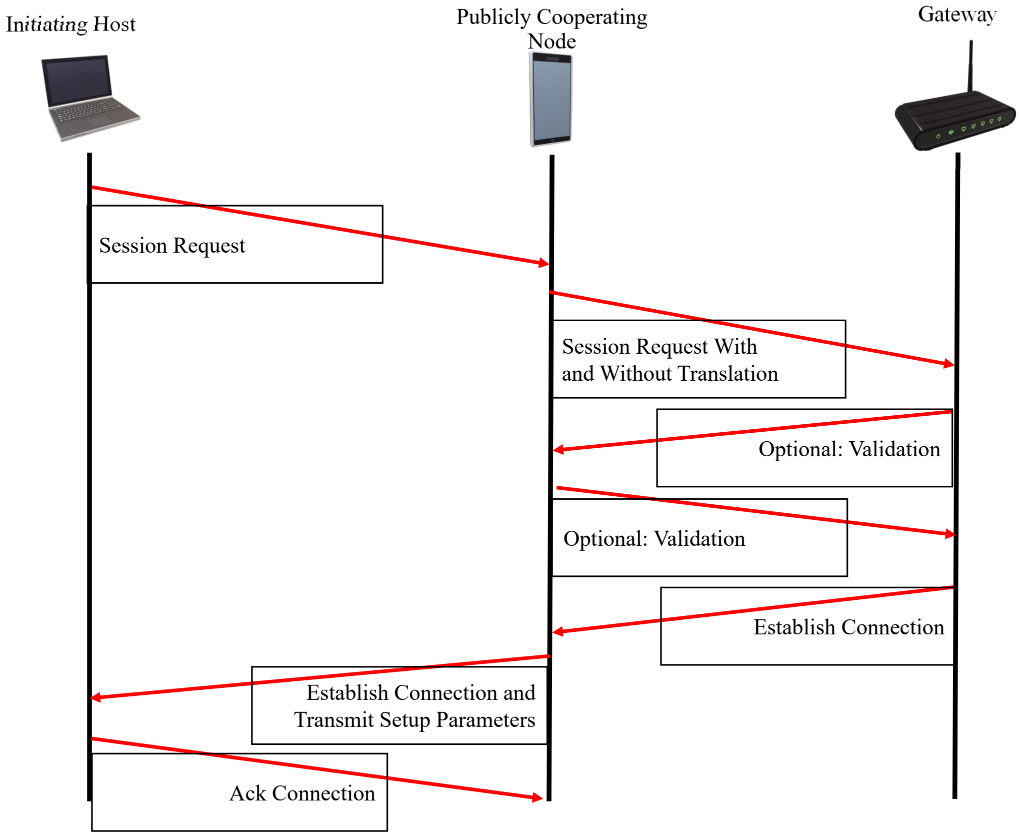


Figure 3.8: Public Node Link Establishment Over Two Hops

Equation 3.1 utilizes the SNIR to compute the maximum contention windows slots. Factor U defines the utility of a node's capability to cooperate. For example, if an intermediary node is not utilizing the interface for any other purposes, factor U will be closer to 1.

Factor U will decline if the receiving interface is the same as the forwarding interface or is currently in use. Finally, factor U will approach zero if no forwarding is possible due to a lack of connectivity. $SNIR_{thresh}$ defines the necessary threshold limitation.

The threshold selection differs based on the implementation and uses additional information inside the NSR to configure the threshold correctly. A node can select a random retransmit slot from the maximum contention window to inform the original host that forwarding is available based on the self calculation. $Base_C$ is the adjustable growth rate of

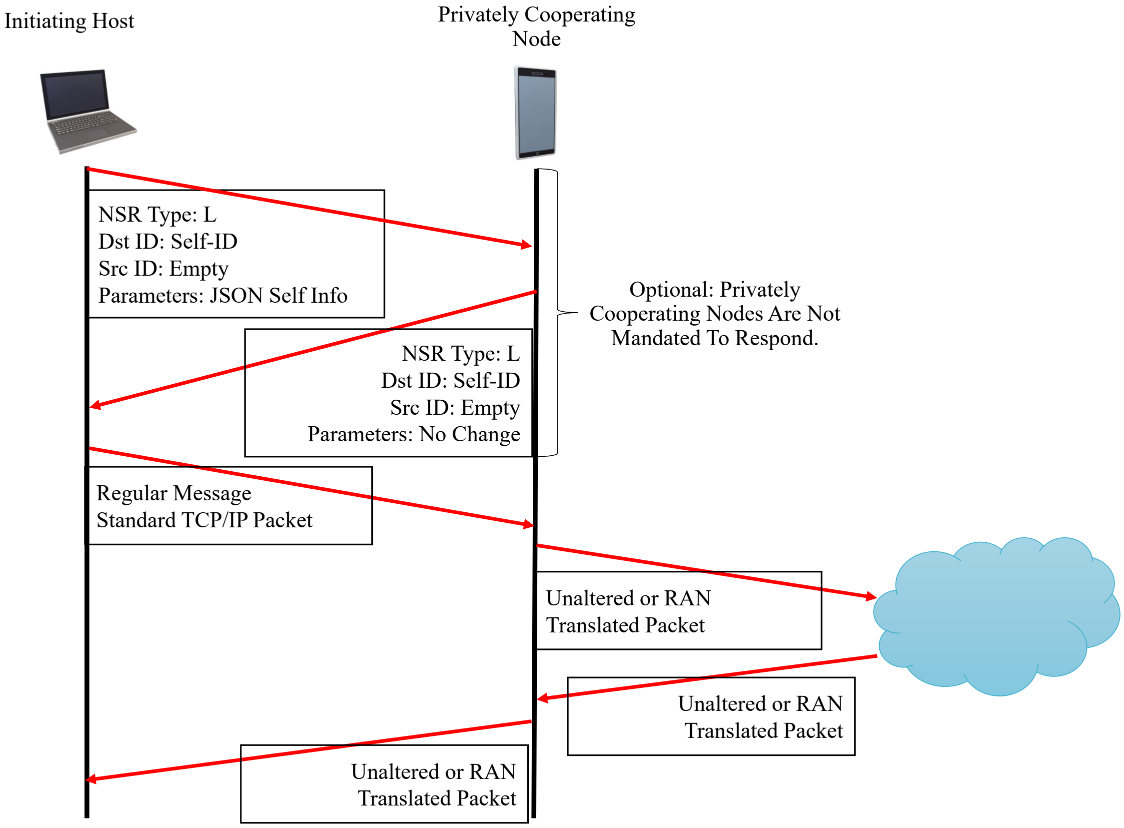


Figure 3.9: Private Node Cooperation & Loopback

the contention windows, and $\gamma$ is the factor adjusting the quality of SNIR. A similar proven contention window approach exists in [37].

$$Max_{Cwnd} = UBase_C^{\frac{SNIR - SNIR_{thresh}}{\gamma(dbm)}} \tag{3.1}$$

In addition to the loopback procedure, the CoopNet allows for direct message forwarding. For example, selecting a forwarding node using handshaking methods is inefficient in vehicular emergency message dissemination. Instead, the emergency message itself is transmitted, utilizing the overheard retransmission as an implicit ack [37]. In CoopNet, with every successful ack, the amount of data sent increases.

### 3.3.5 Passive Discovery

Active discovery methods are still preferential for creating connections and establishing links offering direct message exchange for correctly allocating resources. However, passive discovery plays a vital role in nodes wishing to assist in communication privately and transparently.

Passive discovery aids in gathering network awareness when otherwise not possible. While an intermediary node cooperates privately, the original sender or second hop forwarder is unaware of the intermediary node. Hence, CoopNet dictates a host listen to the existing communication environment.

CoopNet implements a weighted random round-robin cycling through different channels. Upon channel selection, the loopback NSR process will actively seek cooperation. If no loopback message is received, the host will transmit a message and wait for a timeout period for an acknowledgment.

While waiting, the host will monitor the channel to assess the network dynamics and perform possible fingerprinting. Later stages of CoopNet handle the more advanced fingerprinting. After discovering and establishing links, using the links for parallel communication is essential.

Algorithm 6 runs on all available interfaces. When a node is online, it checks any previously collected data about the network environment and activity levels. The Utilization module of CoopNet handles the collection and sharing of network metrics to use in the private detection algorithm. Suppose a particular interface is available for use and is currently not transmitting any messages. In that case, the interface begins listening to ongoing traffic to determine the network activity levels on a particular channel.

CoopNet maintains a centralized neighboring node database with data, including packet type counters, average SNIR, and packet generation frequency. It runs through a fingerprinting process to identify which node emits traffic when receiving information.

The built-in fingerprint process uses the signal to interference and noise ratio, a direct

**Algorithm 6** Passive Discovery: Initialization

```
Input: Interface Access, Communication Traffic (Noise)
Output: Enhanced Awareness, Link Establishment & Use The following algorithm is run
    through all avaialbe interfaces and through each existing channel.
if ($Interface\ != AVAILABLE || Interface.Use == MAX\_USE\_ALL\_CHANNELS$) then
    Return
end if
if ($Interface == IDLE$) then
    Listening.Enable
else
    Wait For Ongoing Communication
    Listening.Enable = TRUE
end if
While loop for passive listening of Network Traffic
while Listening.Enable do
    if ($Interface.Traffic\_Detected$) then
        - Fingerprint Using Network & Packet Parameters
            (SNIR, Packet Length, Packet Types,...)
        - Set Link Establishment Flags (Active & Blind TX)
        - Record Channel Characteristics & Activity
        - Listening.Enable = FALSE
    else
        Random Channel Selection (Previous Data Weights)
    end if
end while
while Listening.Disable do
    if ($Interface.Public\_Cooperation$) then
        Transmit Active Discovery, NetShare & Blind TX Messages
    else if ($Interface.Private\_Cooperation$) then
        if ($Interface.Activity\_Allowed$) then
            Generate Traffic
            if ($Link.Establishment.Blind\_TX$) then
                Transmit Message & Set TIMEOUT For Reply
            end if
        else
            Transmit Messages
        end if
    end if
end while
return $Available\ Nodes$
```

indicator to SNR quickly extracted from individual packets, and a heuristic approach to provide a probability indicator of the traffic generated by a new node. Further improvement is possible by applying a machine learning fingerprinting model for determining traffic type based on captured packet information [43] or deep packet inspection.

$$Ch_i = \frac{\gamma}{\sum_{n=1}^{N_{ch}} \frac{\gamma_n}{\gamma_{max}}} \tag{3.2}$$

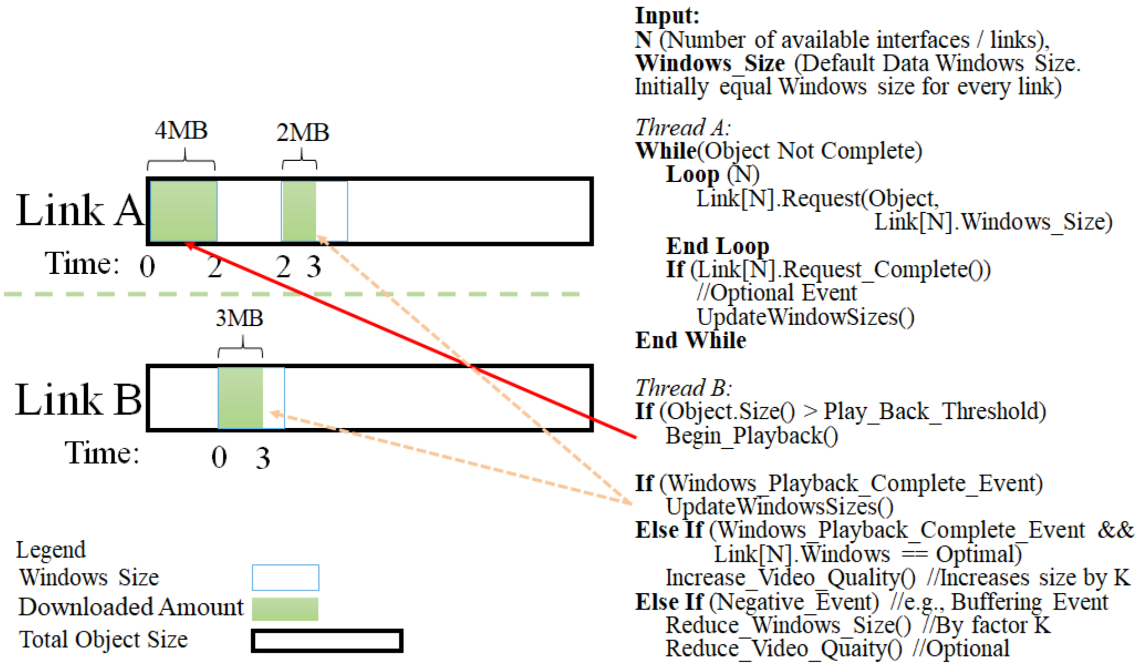
$$Ch_p = (\alpha)Ch_{i-1} + (1-\alpha)Ch_i \tag{3.3}$$

Equation 3.2 and Equation 3.3 assist the random selection of the channel, taking into account channel activity detection and use. A weighted sum assists in maintaining a more constant channel selection probability and removes the effect of single bursts in channel use. $\gamma$ depicts the current activity and use measurement, $Ch_i$ and $Ch_p$ represent the currently selected channel and channel probability weight, $n$ is the set of possible channels.

Often the summation of activity will result in a value less than one since a user cannot use multiple channels concurrently. However, since $\gamma$ considers channel traffic generated by neighboring nodes, the summation can be larger than one. In which case, the result will be a proportion of the total activity over all channels. Therefore, reducing the probability of including overly congested channels.

## 3.4 Decision

Once the discovery and link establishment procedure are complete, CoopNet uses the Decision module for utilizing the available links. CoopNet natively implements four parallel multipath communication frameworks for using the available links, including event triggered, differential access, application or traffic specific, and adaptive sliding windows.



```
Input:
N (Number of available interfaces / links),
Windows_Size (Default Data Windows Size.
Initially equal Windows size for every link)

Thread A:
While(Object Not Complete)
    Loop (N)
        Link[N].Request(Object,
                        Link[N].Windows_Size)
    End Loop
    If (Link[N].Request_Complete())
        //Optional Event
        UpdateWindowSizes()
End While

Thread B:
If (Object.Size() > Play_Back_Threshold)
    Begin_Playback()

If (Windows_Playback_Complete_Event)
    UpdateWindowSizes()
Else If (Windows_Playback_Complete_Event &&
        Link[N].Windows == Optimal)
    Increase_Video_Quality() //Increases size by K
Else If (Negative_Event) //e.g., Buffering Event
    Reduce_Windows_Size() //By factor K
    Reduce_Video_Quaity() //Optional
```

Figure 3.10: Adaptive Sliding Windows

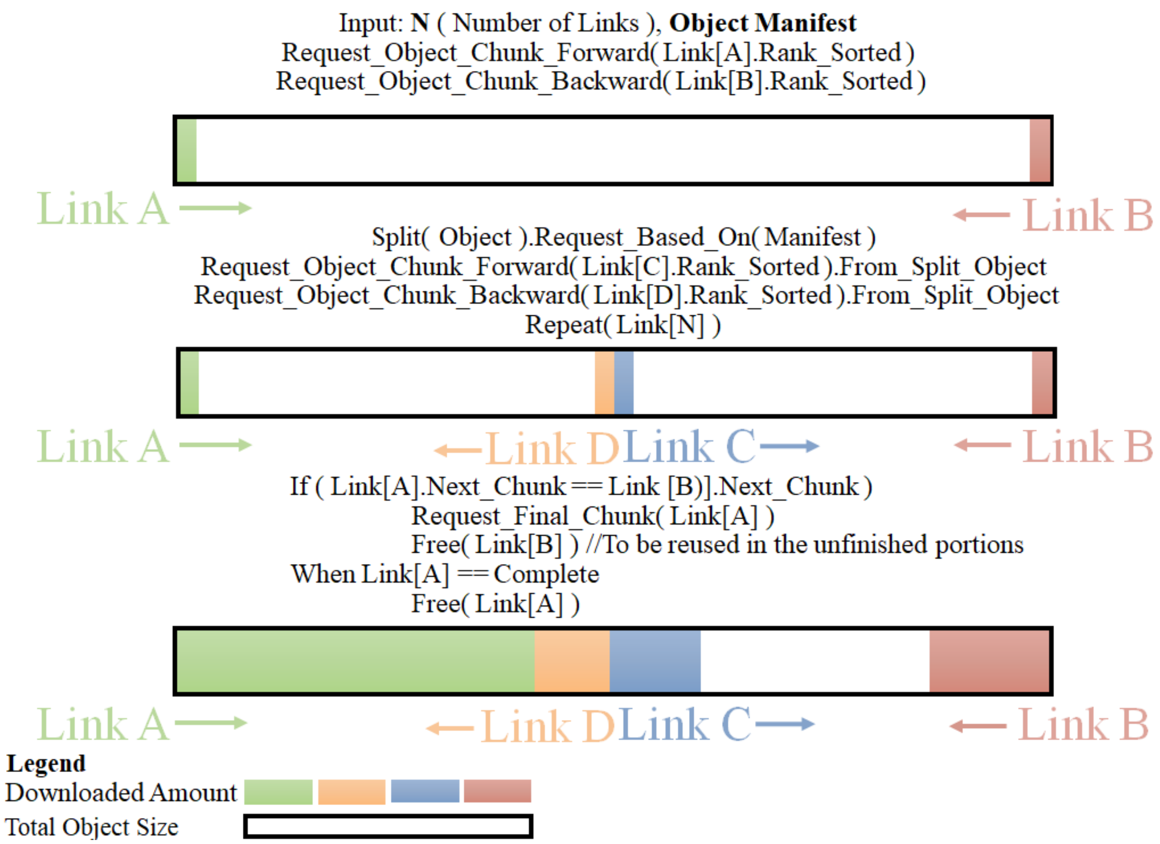
### 3.4.1 Event Triggered Adaptive Link Use

Social media and on-demand multimedia streaming are some of the most popular services offered on the internet. CoopNet introduces a novel approach to enhance service quality using multiple parallel links while still employing the existing protocols.

Since the interface and link properties differ, CoopNet provides an adaptive sliding window. For representation purposes only, the example depicts two parallel links for downloading a multimedia video file shown in Figure 3.10.

In parallel, both links will begin downloading an equal windows size, e.g., two seconds of video content. Link A downloads the first chunk, and Link B downloads the second equal-sized chunk. The video will begin playing the file as soon as an amount of data, such as one-half of the first chunk, is available.

When a node finishes downloading the first chunk, it begins to download the next piece. When the playback reaches the end of a downloaded section or buffers due to a lack of data, both links will adjust their download window size based on the amount downloaded thus



```
Input: N ( Number of Links ), Object Manifest
Request_Object_Chunk_Forward( Link[A].Rank_Sorted )
Request_Object_Chunk_Backward( Link[B].Rank_Sorted )
```

```
Split( Object ).Request_Based_On( Manifest )
Request_Object_Chunk_Forward( Link[C].Rank_Sorted ).From_Split_Object
Request_Object_Chunk_Backward( Link[D].Rank_Sorted ).From_Split_Object
Repeat( Link[N] )
```

```
If ( Link[A].Next_Chunk == Link [B)].Next_Chunk )
        Request_Final_Chunk( Link[A] )
        Free( Link[B] ) //To be reused in the unfinished portions
When Link[A] == Complete
        Free( Link[A] )
```

Figure 3.11: Differential Download

far.

If link A downloaded a total of six MB of viewable content before playback reaches the end of the first chunk, then the new window size will be six MB for link A and three MB for link B. Both links will finish downloading the window size before using the updated windows sizes.

A separate adverse event caused by buffering due to insufficient downloaded data will adjust the windows sizes to be proportional to the amount downloaded. If no events occur within a set number of periods, other actions are implemented, including increasing the video quality for a better overall experience or increasing the window size.

Adapting the link size is possible without direct feedback such as playback and buffering events. It can change its window size based on Equation 3.4. Equation 3.4 uses the balance between the alternative links and itself to get a value to increase its window size.

$CW$ represents the current contention window.

$\chi_{cw_{New_j}}$ is the new window size utilizing its instantaneous throughput times the fraction of the current link's window size and throughput. $\chi_i$ represents the amount downloaded on every other link in the same time block $t_i$. $\alpha$ offers manual adjustment based on policy.

$$T_{cw} = (1+\alpha)CW_{L_i} + (1-\alpha)\left(\frac{t_i L_{tput_i}}{CW_{L_i}}\left(\sum_{L(i)=1}^{n} \chi_i\right)\right) \tag{3.4}$$

$$\chi_{cw_{New_j}} = L_{throughput_j}\frac{CW_{L_i}}{L_{throughput_i}} \tag{3.5}$$

### 3.4.2 Differential Link Usage

Unlike streaming services, non-real-time downloading and uploading may benefit from a continual split feature. CoopNet introduces a mechanism for multiple links to continue downloading an object by separating them into chunks, as in Figure 3.11. First, two of the higher-quality links will begin downloading from the front and back.

If additional links are available, the proposed solution is to split the object in half. The fastest link available will download in the direction of the slowest link, and another link will download in the backward order towards the higher-quality link. The process continues until no more available links exist.

When a link downloads its designated portion, the same process repeats, splitting the remaining amount and utilizing the newly idle link. The proposed approach's complexity is in pre-allocating memory for the object's size, reassembling the downloaded chunks, and bidirectional communication to update the serving client on splitting the item and transmitting it.

### 3.4.3 Parallel Upstream and Downstream

CoopNet gives a solution for a third case utilizing application or traffic type for link selection. For example, with M2M or IoT sensor data that requires continuous data upload, a node may use the best-performing link for the data and an underperforming link for acknowledgments. Separating the upstream and downstream via a different link can ensure a constant and stable sensory data flow without congestion or pauses caused by bidirectional traffic. Nevertheless, it is possible to split traffic based on weighted factors from link properties and traffic class.

### 3.4.4 Additive Link Usage for Real-Time Data

Finally, the last test case is splitting real-time traffic. If data generation is higher than a single connection can handle, the data is divided based on instantaneous throughput and will utilize the maximum available resources. Transmitting data in parallel can increase the overall throughput, alleviate individual link congestion, and download the items with a lower total delay.

Alternatively, if the data generation is lower than an available link can handle, then the parallel transmission is weighted based on Equation 3.6. The $Optimal_{chunk-size}$ denotes the optimal buffered size of the real-time data before dispatch. $\nabla\tau$ represents the time block of the specific buffered chunk. The denominator is the summation of the total throughput between the different links simultaneously, $\nabla\tau$.

$$Link_{weight} = \frac{\dfrac{optimal_{chunk-size}}{\nabla\tau}}{\displaystyle\sum_{interface(i)=1}^{n} \frac{throughput_i}{\nabla\tau}} \tag{3.6}$$

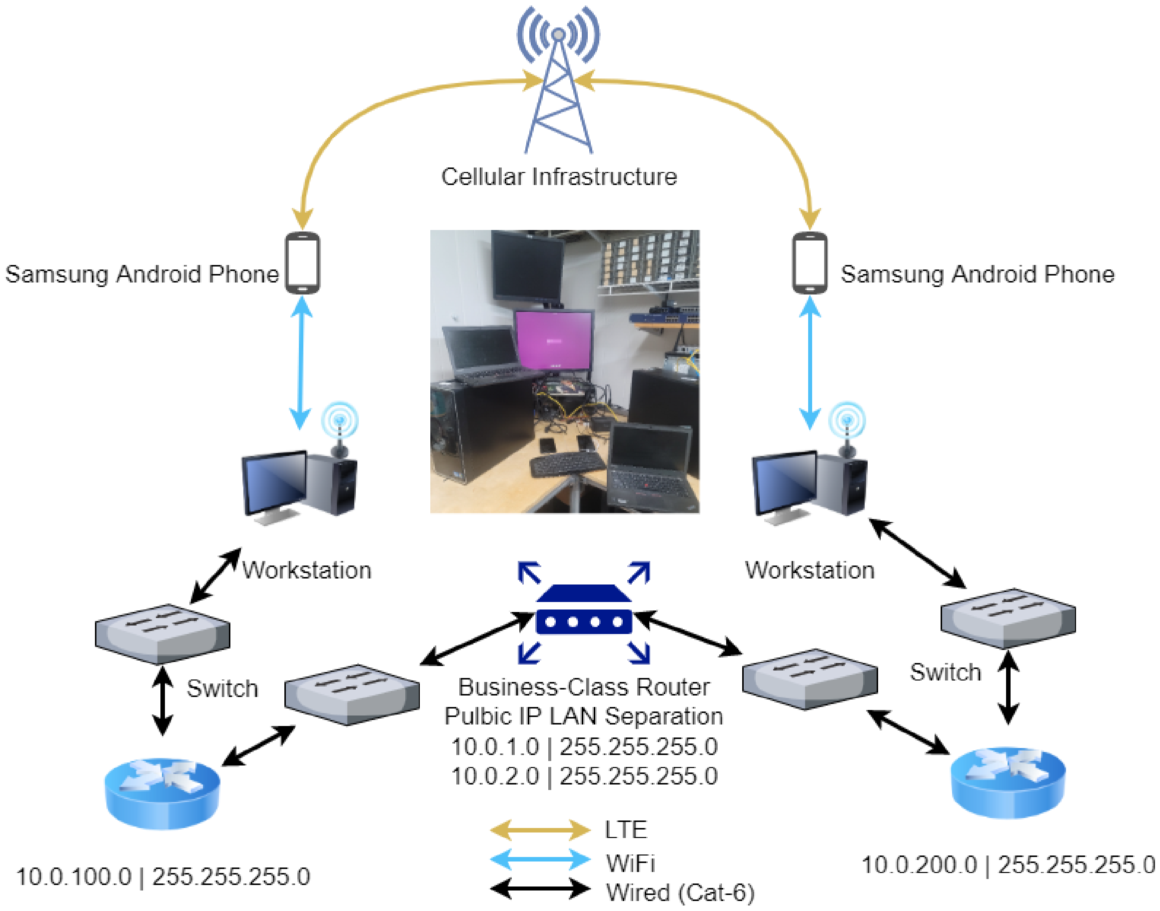


Figure 3.12: Testbed Topology Setup

## 3.5 Results

An experimental testbed, depicted in Figure 3.12, with two workstations and two cellular phones, develop two usable parallel links to illustrate the capabilities and proof of concept of CoopNet. The cellphones utilize LTE for communication using the cellular network and WiFi Direct to communicate with the workstations. The workstations use a wired connection through a local sub-network as a secondary link.

The testbed separates the local network into subnetworks with a private IPv4 address range and netmask shown in Figure 3.12. The private subnetworks connect to a business class router providing Local Area Network (LAN) separation. The business class router uses Charter internet service with a block of static public IP addresses. The additional inline

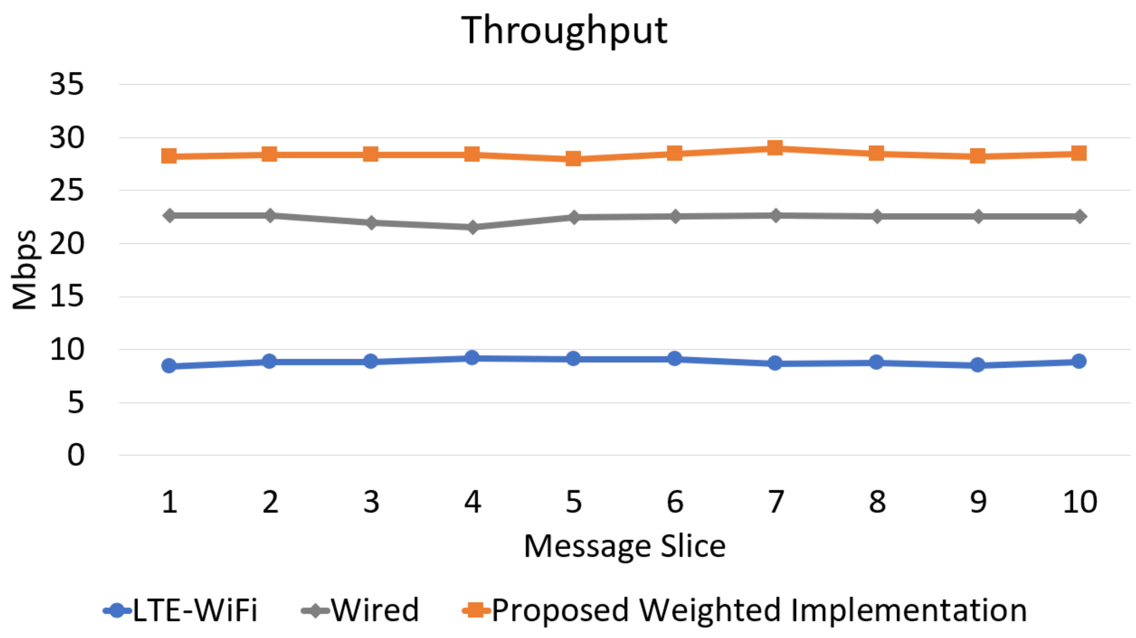


Figure 3.13: CoopNet Experimental Throughput

switches and standard routers induce background traffic through various connected devices. Throughput, latency, and link utility measurements help compare employing traditional communication to CoopNet.

Unlike a simulation environment, experimental implementations have several intentional and non-intentional limitations. Due to hardware and software restrictions, cellular devices utilize WiFi Direct for communicating with the workstations creating a peer-to-peer connection instead of using an Access Point (AP) intermediary approach.

Moreover, the current configuration does not support simultaneous WiFi Direct connections to multiple devices. Advanced implementations of CoopNet can provide modular end-node AP capabilities, numerous links using the same interface, and multi-protocol packet format assembly to reduce the need for RAN translation at the intermediary node.

The cellular devices act as parallel links and translational points for the experimental testbed, converting the communication modes. As mentioned in chapter 1, RAN translation allows for using Access Networks not native to a node. Packet by packet RAN translation is not optimal since it requires parsing and updating each packet. However, utilizing a

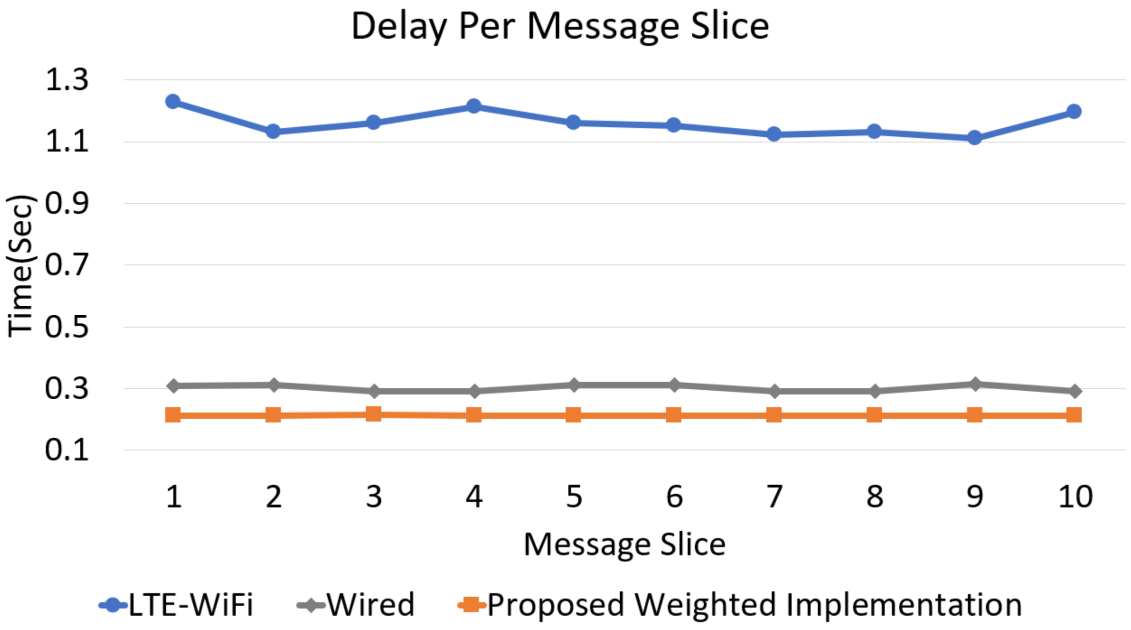


Figure 3.14: CoopNet Experimental Delay

parallel link for transmission, otherwise idle, provides significant improvement to standard implementations.

Figure 3.13 depicts the throughput comparison of using parallel communication with CoopNet and single path communication. Using CoopNet leads to an aggregated 26% improvement. Figure 3.13 provides the average throughput per message slide of 10Mb. It is possible to deduce the overall throughput per second from the delay and throughput plots where CoopNet leads to over 100Mbps.

The experimental testbed ensures that the cellular connection is idle and ready to accept traffic. Utilizing the cellular channel for other purposes leads to a restrained increase in throughput. However, developing a direct transmission module for CoopNet can lead to a more significant improvement in throughput.

As expected when using parallel communication, the average end-to-end latency is reduced by over 29%. Figure 3.14 provides the delay per message slice where each message slice is approximately 10Mb. The results include the translation time to cellular traffic. Larger message slices depict a multimedia scenario transmitting four-second video content

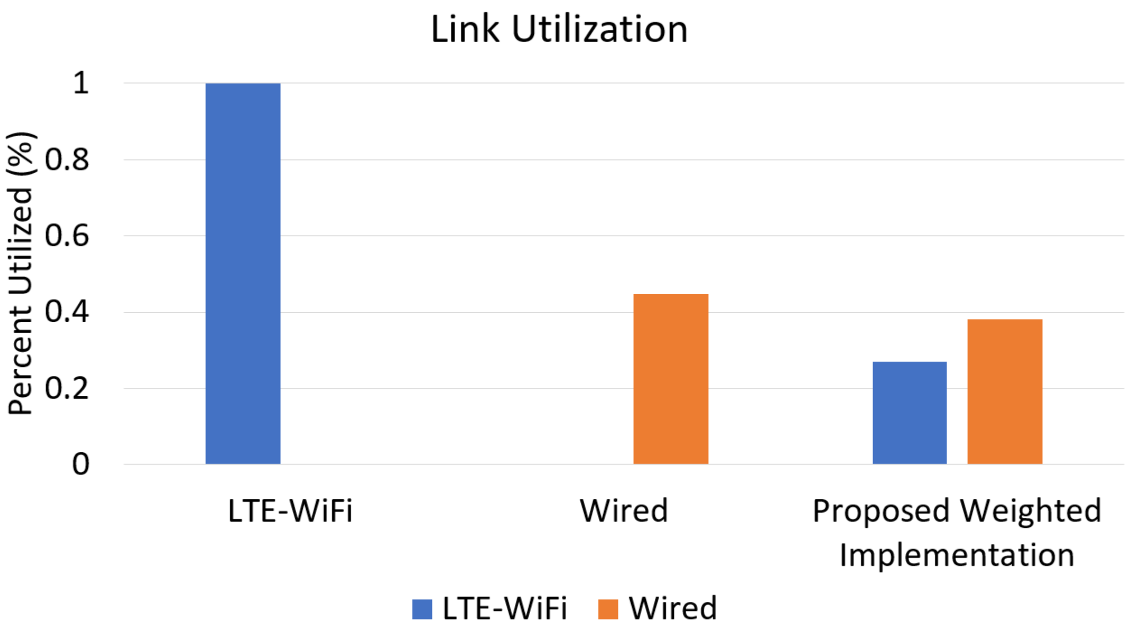


Figure 3.15: CoopNet Experimental Link Utility

at different qualities or transmitting raw sensor data for analysis. Moreover, more significant message sizes help demonstrate the benefit of parallel communication with a limited Access Network and a higher-performance Access Network.

When utilizing LTE alone, the overall link utility reaches close to maximum usage attempting to transmit packets back to back as shown in Figure 3.15. Wired link utility consumes close to 44% of the bandwidth. However, utilizing CoopNet, the link utility reduces by 14.91% and 73% using parallel connections allowing other applications to use the cellular network bandwidth. Even if the improvement is more negligible for the higher-performance link, it provides additional benefits by reducing the congestion and overutilization in lower-performance links with lateral communication.

## 3.6 Summary

CoopNet is different from cellular offloading. It aims to use all available Access Networks, not just WiFi and cellular, but also satellite, high mobility Ad-Hoc Access Networks, and any custom implementations in IoT or similar environment. Developing multipath parallel communication mechanisms makes it possible to reach an equilibrium between the different Access Networks usage.

However, a slow adjusting mechanism can lead to inefficiencies, and more dynamic adaptation mechanisms can assist in providing better overall network usage while ongoing traffic exists. Alternatively, more advanced dynamic adaptation solutions such as machine learning may be infeasible in specifically limited environments. Hence both advanced and basic adjustment mechanisms are necessary, providing an ideal reason for developing a programable network environment.

This chapter provides the implementation of NetShare for sharing network information and developing links up to two hops away and the discovery methods. Moreover, CoopNet delivers several mechanisms for parallel communication, including event triggered, additive link usage, and differential implementation. Event-triggered assist with applications providing node feedback, such as buffering conditions. Additive link use offers a solution for real-time traffic, and differential transmission helps improve static content transmission using simultaneous parallel paths.

Creating virtual links two hops away has several implications for utilizing underutilized or idle devices, extending the communication range, providing translational service between different access networks, and integrating networking functionality into all nodes. The performance improvement is lower than the direct communication use of the intermediary node. Depending on the type of intermediary node, the translational service adds a notable delay in processing the packet before forwarding.

Likewise, since multiple idle nodes are not continuously idle, the link stability and availability are lower than more permanent access network devices deployed. Moreover, the use of idle resources requires updating the network platform to support CoopNet and must be able to switch between different interfaces rapidly. Unfortunately, since most consumer-grade devices implement a single active and permanent connection, the devices must create an ad-hoc connection with the idle interface, which incurs higher inefficiency than existing approaches.

However, using idle interfaces for enhancing communication provides significant benefits since it extends the capability of a node and, in some cases, can use the slower connection for alternative application traffic. Moreover, through the continuous development and possible implementation of architecture like CoopNet, the network dynamics may change to allow multiple permanent connections and the integration of supportive communication mechanisms with the deployment of IoT devices. Lastly, parallel communication will enrich a node's communication ability by utilizing multiple services through different service providers and can provide a backup communication platform for always-on connectivity.

# CHAPTER 4

# DATA COLLECTION, PROCESSING, AND DYNAMIC ADAPTATION

## 4.1 Introduction

The advent of the Internet of Things (IoT) and machine-to-machine (M2M) communication provide a system for collecting and manipulating big data and a platform for sensing, actuating, and automating the environment. IoT, M2M communication, in addition to social networking and mass multimedia, places a severe strain on the communication infrastructure. Thus, the archaic communication frameworks require necessary improvements.

One such improvement is the simultaneous usage of parallel communication links of differing radio access networks. Chapter 4 presents a Machine Learning (ML) optimization for link selection and use in CoopNet, a horizontal programmable communication architecture. Programmable networking paves the way for advancing communication to improve performance, reliability, security, and policy-based applications, including network decoupling.

CoopNet strives to persuade shifting from a traditional networking structure to a programable and horizontal approach to provide developers with additional capabilities through a modular communication platform. The motivation is to create a parallel communication environment using all available Radio Access Technologies (RATs) and resources.

More and more manufacturers develop devices with multi-mode single-chip Systems-on-Chip (SoC) for communication. A single chip can embed several unique radio access technologies and frequency ranges for the physical layer access, such as Bluetooth, WiFi, Personal Area Networking (PAN), cellular, dedicated short-range communication (DSRC), and more.

Regardless of the accelerated hardware development, the existing standards limit the usage of the different access networks due to an archaic single point-to-point model. Cellular communication plans to improve on two fronts with 5G Next-Radio (NR) and 6G releases to overcome the traditional model limitations.

5G NR will utilize a higher frequency spectrum with more advanced modulation and cooperation between the base stations while simultaneously using cellular and non-cellular links through the Access Traffic Steering, Switching & Splitting (ATSSS) framework. 3gpp's ATSSS is a closed-source solution. With ATSSS, user Equipment (UE) can establish cellular and non-cellular sessions through the Access and Mobility Management Function (AMF).

AMF uses the Non-3GPP Interworking Function (N3IWF) for establishing the non-cellular connections. However, devices without cellular connectivity and the 5G core cannot use ATSSS to enable MRANs with multipath communication. Other solutions such as MPTCP provide some improvement with limited usage since most traffic in MPTCP is not parallel, rather opportunistic. MPTCP supports handovers more readily than establishing multiple paths with new TCP connections.

Alternatively, software-defined networking (SDN) is a game-changer facilitating MPTCP and implementing a semi-horizontal approach. SDN implementations typically occur after the end nodes pick a particular transport layer protocol to use. Afterward, SDN-capable devices have access to a wide array of header fields. Due to the lack of SDN integration into the end systems, the end node capabilities remain underutilized.

CoopNet considers the end nodes to maximize the performance. Moreover, CoopNet natively implements features including Radio Access Network (RAN) translation, a network sharing framework, all-node as infrastructure capabilities, node cooperation, different multipath parallel access, and new application-layer features. The three classes of collaboration include the following.

**Public** cooperating nodes may assist in all networking functions, including RAN translation, forwarding, routing, and different applications such as network management, cybersecurity, load balancing, limited only by the available hardware and resources.

**Private** cooperating nodes assist transparently, including direct packet forwarding, firewall functions, RAN translation, and more. It is best to leave the packet formatting to the originating host for the best implementation. However, a privately cooperating node can update header information as needed in more limited situations.

**No-Cooperation** indicate nodes that do not wish to participate in networking functions for differing reasons, including policy, limited resources, privacy, and regardless of the rationale.

CoopNet emphasizes links when establishing connections, including virtual links developed across two hops. Links across two hops improve reachability and capabilities by allowing Radio Access Network (RAN) translation. For developing virtual links, discovery and network awareness information sharing are vital, demonstrating a need for CoopNet's NetShare to share network metrics and establish communication.

CoopNet is modular to allow network engineers to develop new decision models. One benefit of a modular programable approach is the ability to implement different decision mechanisms. For example, CoopNet natively implements four decision modes, event-triggered, adaptive sliding windows, differential, and traffic specific. Moreover, instead of forcing the applications to create connections for bidirectional traffic, CoopNet handles the connection establishment and traffic transmission decisions through the Decision module.

**Event Triggered** indicates the use of events, such as video buffering or channel condition changes, for updating utilization. For example, with video content and buffering, a node may download different chunks using different links and buffering the downloaded content. CoopNet provides adaptive alteration of the chunk size and link use during specific events such as reaching the end of a downloaded chunk during playback or adverse events such as buffering.

**Adaptive Sliding Windows** allows a node to adaptively select and modify link use without event triggering by utilizing information such as round trip time (RTT), bandwidth, and throughput calculations for real-time continuous data transmission.

**Differential** access provide a unique mechanism for downloading static content via parallel paths. It allows downloading from opposing ends and splitting the content to match the number of available parallel links. The higher ranked links begin from the front and the slower link from the rear.

**Data & Application Specific** allows utilizing parallel paths based on traffic types or application requirements. For example, a node may use one link for continuous sensory data reception while sending the acknowledgment via a separate link in parallel.

CoopNet uses Data Collection to handle data modification, separation, assembly, and acquisition, including the packet header replacement and packet modification for cooperating nodes. Moreover, the Data Collection module prepares the utilization metrics and formats the collected information for proper use in the Dynamic Adaptation stage.

Standard mechanisms for communication are necessary for proper operation but not optimal. However, a single optimization technique for all network topologies and communication environments is not feasible due to dynamic network characteristics. Different optimization techniques, including machine learning, require distinct inputs and sets of data. Not all network optimization techniques apply depending on the existing network dynamics and limitations. Nor can all nodes gather all necessary data using the different optimization techniques.

CoopNet offers an open data access platform for obtaining network dynamics and metrics. Due to many factors, CoopNet uses two modules for advanced optimization techniques such as machine learning. One advantage of separating the components for advanced optimization is that it allows different network engineers to work on various segments separately.

This chapter presents a feed-forward neural network combined with a recurrent neural network for link optimization. The Data Collection module handles the data preparation, including normalization, reduction, imputation, and formatting for both neural network stages. The Dynamic Adaptation implements the ensemble machine learning model.

### 4.2 Literature Reivew

Several publications have proposed machine learning (ML) for network optimization, including ML for optimal modulation use of the physical layer [44], adaptive bitrate selection for multimedia applications [45] [46], network traffic control [47], network security [48], Radio Access Technology (RAT) scheduling [49], and for IoT [50]. However, limited literature exists implementing machine learning for parallel multipath communication. In part, the reason is due to the limited implementation and use of parallel multipath connections.

In [44] the authors demonstrate adversarial networks with multi-transmitter/receiver competing for capacity as Neural Networks. Moreover, the authors illustrate the use of Deep Learning (DL) for classifying modulation I(In-phase) Q(Quadrature) signals correctly.

The authors in [49] introduce a building block model for cross-system machine learning using both cellular and WiFi Radio Access Technologies (RATs). The publication depicts promising improvement in edge user equipment (UE) throughput and faster convergence than Reinforcement Learning. CoopNet provides a different solution by presenting a model for optimal link use instead of varying RAT.

Since CoopNet permits nodes to become part of the network infrastructure, it is possible to apply several existing machine learning models for physical layer modulation, traffic control, security, and more. Dynamic Adaptation can use a pre-trained model or provide in-place training and testing. Node with a higher computational set of resources can generate more reactive models when applied in different environments.

Moreover, CoopNet introduces the notion of RAN translation and links over two hops. A node may use the same RAT with differing links in some cases. For example, a node may use WiFi to send part of the traffic via the attached access point, and at the same time, the node may use WiFi Direct to send packets to a separate link created through a cellular node.

Using virtual links introduces a wide range of networking opportunities and challenges. Currently, network management processes allow for a single active connection per interface. However, multiple virtual links developed over a single interface can provide multipath access and connections through Frequency Hop Selection (FHS) or a time division approach. For optimal integration, network management software requires core changes to **enable multiple active and semi-permanent connections through a single interface**.

## 4.3 Data Collection - Capture, Processing & Formatting

As necessary as any optimization machine learning model, the preprocessing and data preparation is vital and, in many cases, more influential. Incorrect data segmentation, normalization, and formating can easily lead to improper fit, sparsity, variance, and unnecessary dimensionality.

CoopNet separates the data preparation module, Data Collection, and the optimization module, Dynamic Adaptation, to allow for a programmable network environment that supports continuous improvement and rapid adaptation of new protocols. The separation offers granular control for developing accurate models. For example, when using parallel links with multiple RANs, a normalization function requires special consideration in the data preparation.

In the case of throughput, wired connections tend to have higher limits for bandwidth and throughput. A simple ratio of throughput/bandwidth can indicate the current link activity but may offer the neural network misleading information if not paired correctly with additional data or normalized. Therefore a more advanced data preprocessing model can account for the ambiguity.

Data Collection (DC) uses all collected information, internal and external data, and for-

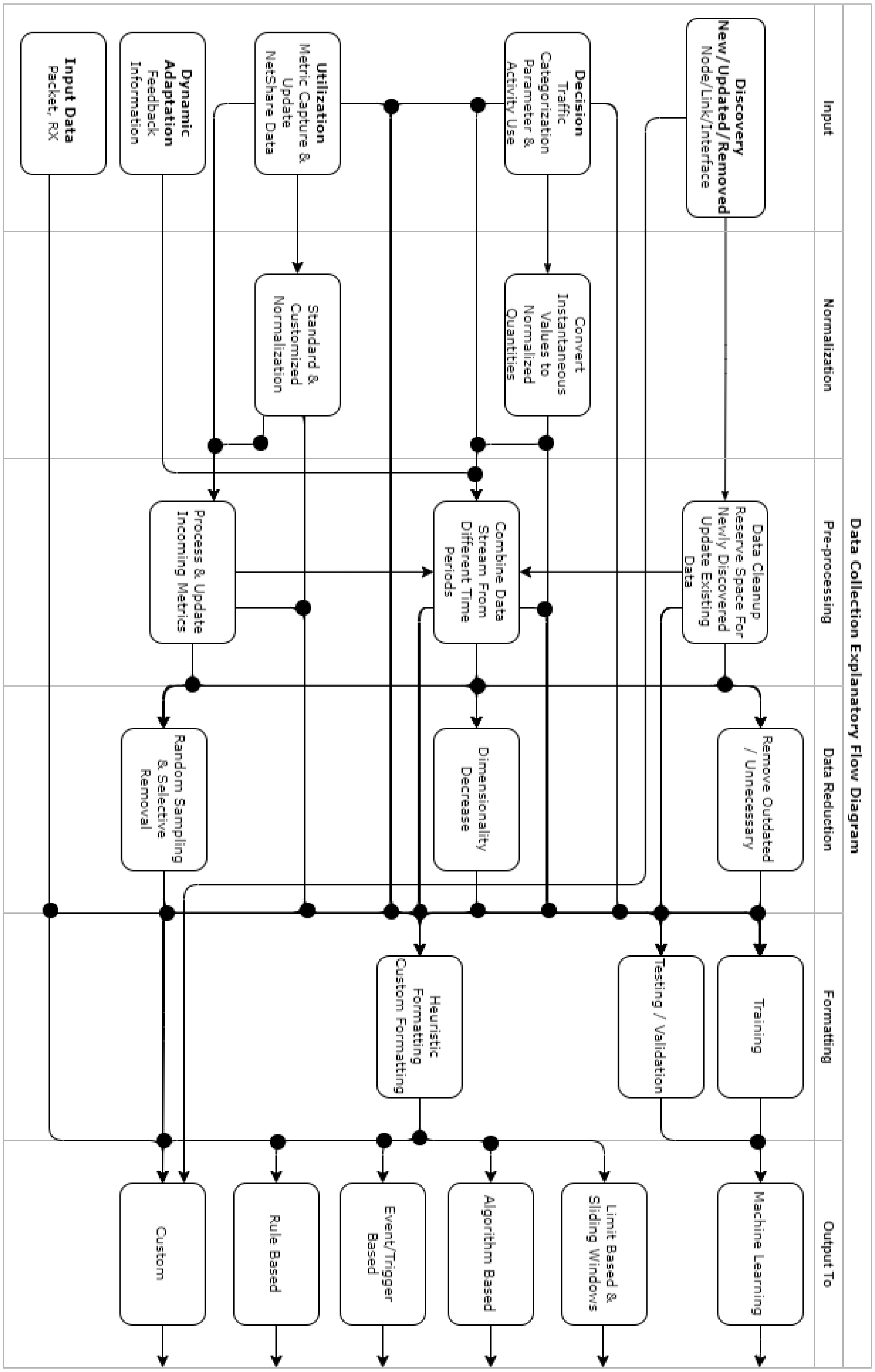


Figure 4.1: Data Collection Flow Diagram

mats it based on the requirement demand from Dynamic Adaptation for machine learning models or other optimization mechanisms. Newly discovered links, interfaces, and the loss of such links are vital to determine what resources are available.

As presented in Figure 4.1, a flow diagram of the Data Collection module separated by the different compartments, many of the inputs undergo a normalization. Equation 4.1 - Equation 4.3 depict three normalization functions in the DC modules used in the neural network relating to throughput. Other normalization functions follow a similar approach.

In Equation 4.1, the total interface (i) throughput sums all link ($\ell$) throughputs. The normalized link throughput is ($t_\ell$) over the interface rate (R). The feed-forward neural network limits the rate by ($\alpha$). Similarly, in Equation 4.3 the normalized interface throughput is the interface throughput divided by the sum of all available rates. Normalization employing all available rates highlights the proportionality.

$$t_i = \sum_{\ell=1}^{n_\ell} t_{i_\ell} \tag{4.1}$$

$$Norm_{t_l} = \frac{t_\ell}{\alpha R_i} \tag{4.2}$$

$$Norm_{t_i} = \frac{t_i}{\sum\limits_{i=1}^{n_i} R_i} \tag{4.3}$$

After normalization, a preprocessing compartment is necessary to modify various data sets, adjust expired data due to link termination or loss, and assemble the data received from NetShare or any raw metrics. The preprocessing can take in both normalized and raw data depending on the implementation requirements.

The next compartment handles the data reduction. Random sampling and data segmentation for training, testing, and validation are necessary for specific machine learning implementation. Similarly, it is acceptable to separate samples triggering an event for a heuristic implementation of event-triggered decisions.

Finally, the formatting ties the Data Collection module and Dynamic Adaptation mechanisms through a programable API approach. Formatting rules and regulations ensure the data provided will be in the exact composition for direct use. The different compartments allow for a middlebox approach that inputs and provides respective outputs through a programmable system.

## 4.4 Machine Learning Adaptation

Since link establishment can exist over two hops, many link parameters and information sets are not directly accessible. Hence, NetShare, introduced in [25], dictates two message types for sharing interface information, including NetShare Advertisements and NetShare Requests. The information received from NetShare supplements the existing capabilities and internal limitations. Thus, there is a multi-dimensional set of conditions. A programmable system is the best option given the number of exclusive parameters.

The Data Collection module compiles the information, including the hyperparameters, and feeds it into the Dynamic Adaption module. The data preparation component is vital to prepare the data appropriately, where an ensemble machine learning approach can provide more helpful link selection. For example, a classification will not help proportionally select a specific link in many scenarios. However, classification can help in other techniques, such as RAN ranking. Afterward, the updated information set may help determine the optimal link selection and scheduling.

A single communication solution is not feasible given many parameters and conditions, including differing environmental elements and resource demands. One way to progress with communication advancement is to consider programable networking from the physical layer up.

Mixed modulation techniques may prove to be more efficient and valuable at the physical layer by implementing machine learning models [44]. Similarly, generalized forwarding is becoming more prevalent, with cross-layering becoming a norm.

From a security and performance perspective developing adaptive networks is beneficial. Using parallel communication reduces the man-in-the-middle attack drastically since it would require an attacker to capture traffic via multiple links and in parallel. Moreover, new techniques for securing communication are possible by implementing security features at the RAN translation nodes.

Due to the complex nature of optimal link selection, the machine learning model can be considered semi-supervised regression or, more realistically, unsupervised. A dense Neural Network (NN) (NN) approach, such as Recurrent Neural Network (RNN), is helpful in optimal link selection since covariance between the different features may exist but can lead to improper link separation due to the constraints. For example, there is a direct correlation between delay and throughput. However, a low end-to-end delay alone does not signify a high throughput in parallel communication using different RANs. Nevertheless, a Convolutional Neural Network (CNN) may be helpful to consider in future works.

Consider a set of links $L_i = \{1, ..., L_i - 1, L_i\}$ that a node has access to, expanding over two hops. Let $P_i = \{1, ..., P_i - 1, P_i\}$ be a set of parameters, $X_i = \{1, ..., X_i - 1, X_i\}$ be a vector of link features, and $H_i = \{1, ..., H_i - 1, H_i\}$ be a set of hyperparameters. Hyperparameters can be direct user input, such as limitations, preferences, or boundaries.

Hyperparameters also represent derived information, such as node resources, link use requirements based on traffic type, or current use. $\omega_\iota$ defines the set of weight vectors attached to neural network layers, $\beta_\iota$ defines the bias vector, $\iota$ defines the set of layers, and $\sigma$ defines the output function. Equation 4.4 provides the multilayer perceptron of a fully connected NN. $\gamma$ defines the output vector, and $\ell$ represents links.

$$\begin{aligned} \gamma(X_\iota; (P_\iota + H_\iota)) &= \sigma_0(\omega_\iota X_\iota + \beta_\iota) \\ \gamma_\ell(X_\iota; (P_\iota + H_\iota); \gamma) &= \sigma_1(\gamma; \omega_\iota X_\iota) + \beta_\iota) \\ \iota &= 1 ... \iota - 1, \iota \end{aligned} \tag{4.4}$$

Several activation functions depicted in Table 4.1 exist to obtain different results, including some of the more popular softmax, binary, Rectified Linear Unit (ReLu), sigmoid, and Gaussian activation functions. Y is the output function fed into the activation functions, with (i) representing the Feed Forward Neural Network (FFNN) and (j) RNN.

$\psi$ in the sigmoid and softmax parameter limits the output. $\eta$ is an optimization function for determining the sigmoid response. $\kappa$, $\mu$, $\delta$ in the binary and Gaussian activation function are arbitrary values left for optimization in future works implementing different machine learning algorithms.

A sigmoid activation function provides the required output for individual interface utility for link optimization. However, since two separate links may exist on a single RAN interface, a two-mode ensemble NN helps determine the proportionality of the link selection. Figure 4.2 shows the RNN with a FFNN depiction.

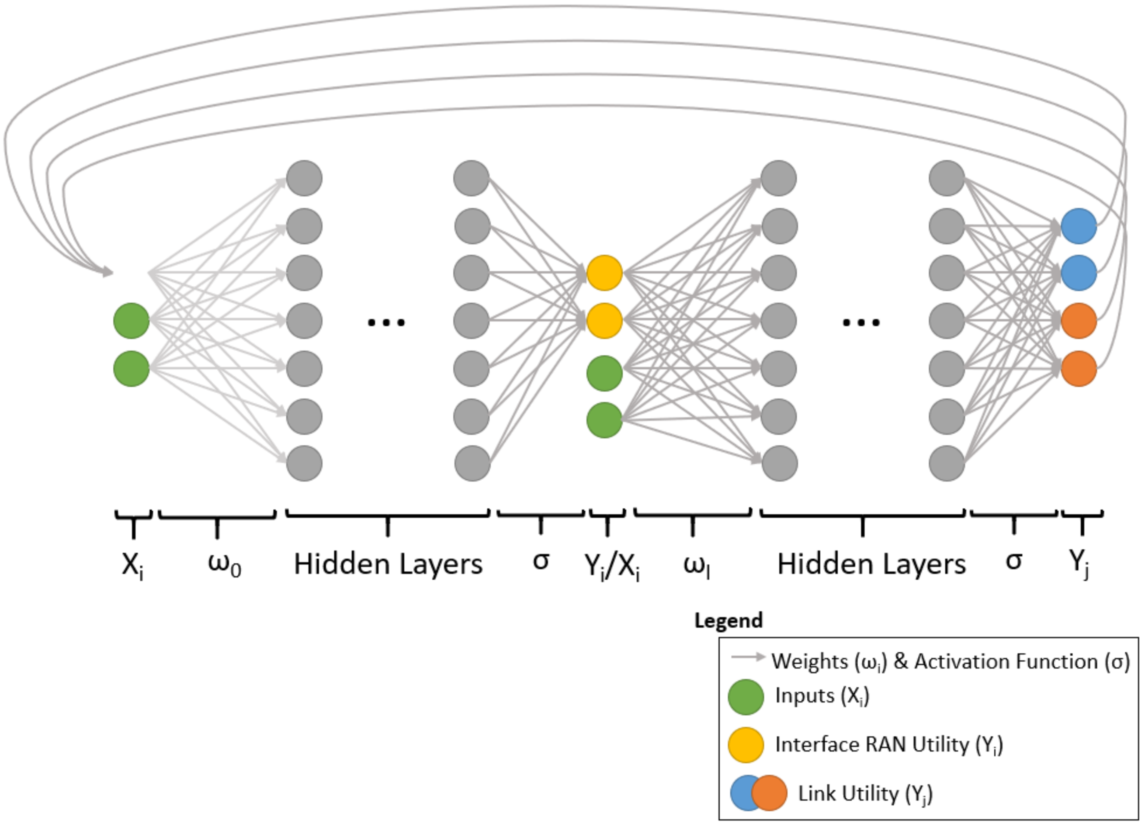
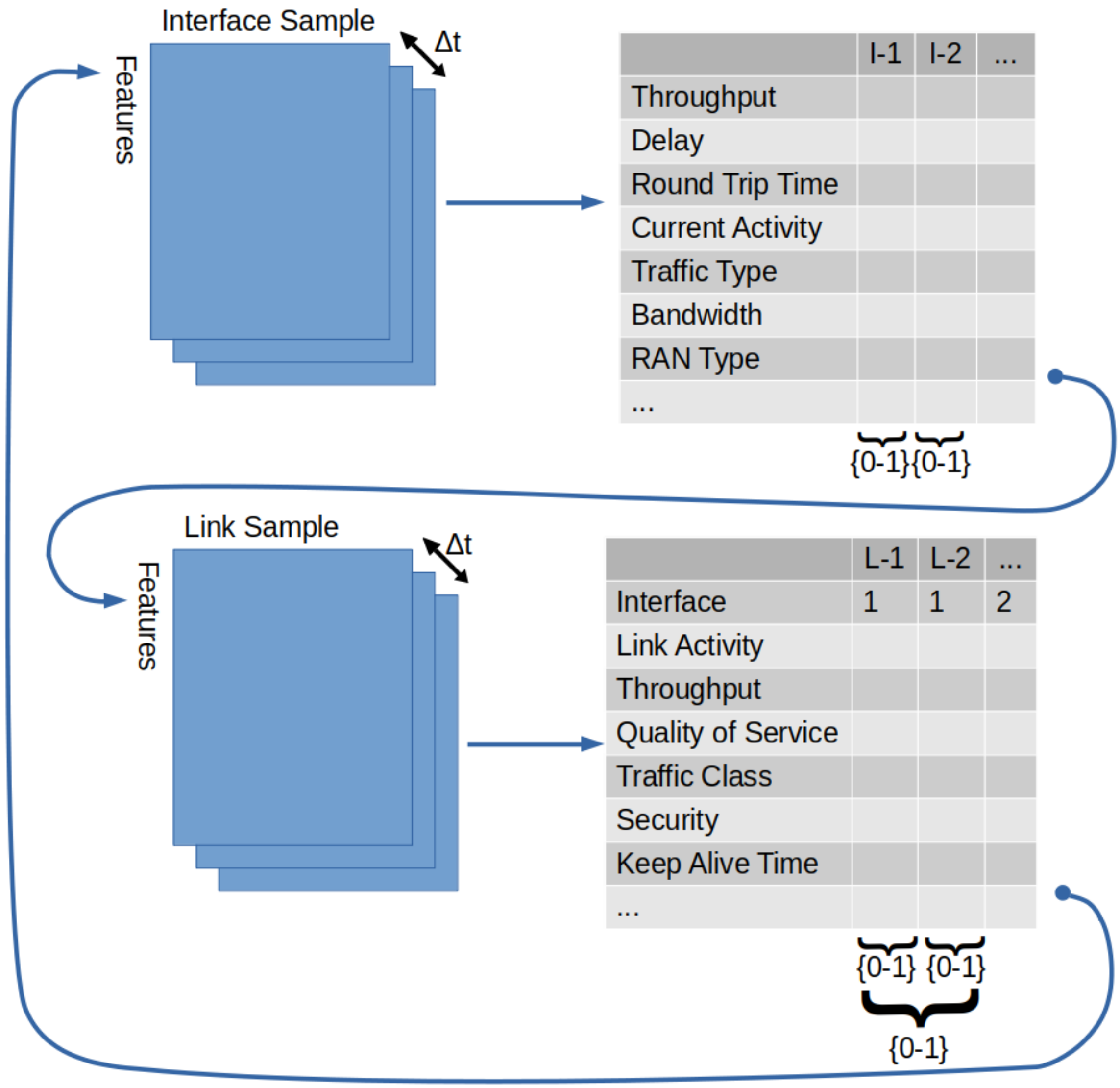


Figure 4.2: Feed-forward with Recurrent Neural Network



| | I-1 | I-2 | ... |
|---|---|---|---|
| Throughput | | | |
| Delay | | | |
| Round Trip Time | | | |
| Current Activity | | | |
| Traffic Type | | | |
| Bandwidth | | | |
| RAN Type | | | |
| ... | | | |

| | L-1 | L-2 | ... |
|---|---|---|---|
| Interface | 1 | 1 | 2 |
| Link Activity | | | |
| Throughput | | | |
| Quality of Service | | | |
| Traffic Class | | | |
| Security | | | |
| Keep Alive Time | | | |
| ... | | | |

Figure 4.3: Feature Sampling in FFNN

The initial forward neural network considers information relating to interface parameters and hyperparameters to provide a sigmoid response where the two interfaces' utility may range from 0 to 1. The output forward feeds into an RNN considering the established links with a reaction resulting in a summation between 0 to 1 per interface. An altered softmax activation functions best suits the link response limited by the available interfaces.

The training and testing depend on several features, with numerous features ideal for both ensemble stages and some specific to one. Figure 4.3 depicts an explanatory feature selection where the FFNN feeds the interface activity and output into the RNN branch.

Table 4.1: Activation Function

| **Name** | **Format** |
|---|---|
| Sigmoid ($\sigma_{FNN}$) | $\dfrac{Y^{\eta}}{\dfrac{\left(\dfrac{Y}{\psi}\right)^{\eta} + \left(1 + \left(\dfrac{Y}{\psi}\right)^{\eta}\right)}{\psi^{\eta}}}$ |
| Softmax ($\sigma_{RNN}$) | $\psi\left(\dfrac{e^{Y_j}}{\sum_{\ell(j)=1}^{\ell_J} e^{Y_j}}\right)$ |
| Smooth ReLU | $ln(1 + Ye^{Y})$ |
| Linear | $\psi Y$ |
| Binary ($\sigma_{RNN}$) | $\dfrac{1}{1 + e^{-2\kappa Y}}$ |
| Gaussian | $\delta e^{-\dfrac{(Y-\kappa)^2}{2\mu^2}}$ |

Figure 4.3 illustrates an example where RAN type is more beneficial in the first section, and the security selection is better suited for the links.

Some features may be unique or the same, depending on the condition. In the case of throughput, an interface's throughput would be the summation of the link throughputs. If a single link exists on the interface, the two throughputs can be the same. Alternatively, the link throughputs would be smaller than the interface throughput with every additional link using the same interface. Multiple virtual links can utilize a single interface.

The two-stage ensemble neural network accomplishes several goals, including limiting interface use, optimizing link selection, and controlling data streams. The goal of the first neural network branch is as follows:

- **Minimize** the interface usage to allow more time for the reception and new links while maximizing the total throughput between all interfaces.
- **Minimize** the end-to-end latency by maximizing the use of parallel link transmission.

There is a temporal limitation to achieving a lower end-to-end delay dependent on the end-to-end propagation delay. It is easier to discern propagation delay near the end nodes based on rate calculations. However, as the packets travel over longer distances and at different destinations using the same established link, backend internet-altering paths can significantly affect the total propagation delay.

Hence, the Round Trip Time (RTT) for the different interfaces become increasingly important to consider. For this reason, the feature concerning RTT differs from the standard TCP calculation based on Equation 4.5. Here, RTT discerns the average of the propagation delay rather than a time for retransmission.

The equation is a modified exponential moving average with a $\lambda$ factor shifting curve above or below. $RTT_\ell$ is the instantaneous throughput per each link summed. The summation of the $RTT_\ell$ over previous values aids by providing a smoothing factor and lagging response. Summing $RTT_i$ reduces outliner effects, and k is an exponential factor between 0 to 1.

$$RTT_i = \frac{\sum\limits_{i=i-m}^{i-1} \sum\limits_{n=0}^{n=\ell} RTT_\ell}{m} k^\lambda + \frac{\sum\limits_{i=i-n}^{i-1} RTT_i}{n} (1-k) \tag{4.5}$$

Equation 4.6 is the optimization function which the first neural network aims to achieve. The features include instantaneous bandwidth use, throughput, activity level, and information about the maximum expected bandwidth usage ($E(Max_{BW})$. The optimal solution is a set of weights indicating the minimum combination through all interfaces.

$$E(Max_{BW}) = argmin\left(\sum_{i=1}^{i_{max}} \omega_i Interface_i\right) \tag{4.6}$$
$$0 <= \omega <= 1$$

$$E(interface_{use}) = argmax\left(\sum_{n=1}^{n=\ell_{interface}} \omega_\ell \ell_n\right) \tag{4.7}$$
$$\sum_{\ell=1}^{\ell_{interface}} \omega_\ell <= 1$$

Alternatively, the second neural network aims to maximize the link selection to ensure maximum usage of the available resources. Equation 4.7 depicts the optimization function for the second neural network (RNN). $\ell_{interface}$ represents the maximum number of links per interface. $\omega_\ell$ depicts the weights for the links. $\omega_\ell$ sum to less than or equal to one for all links of a single interface. The goal of the recurrent neural network is as follows:

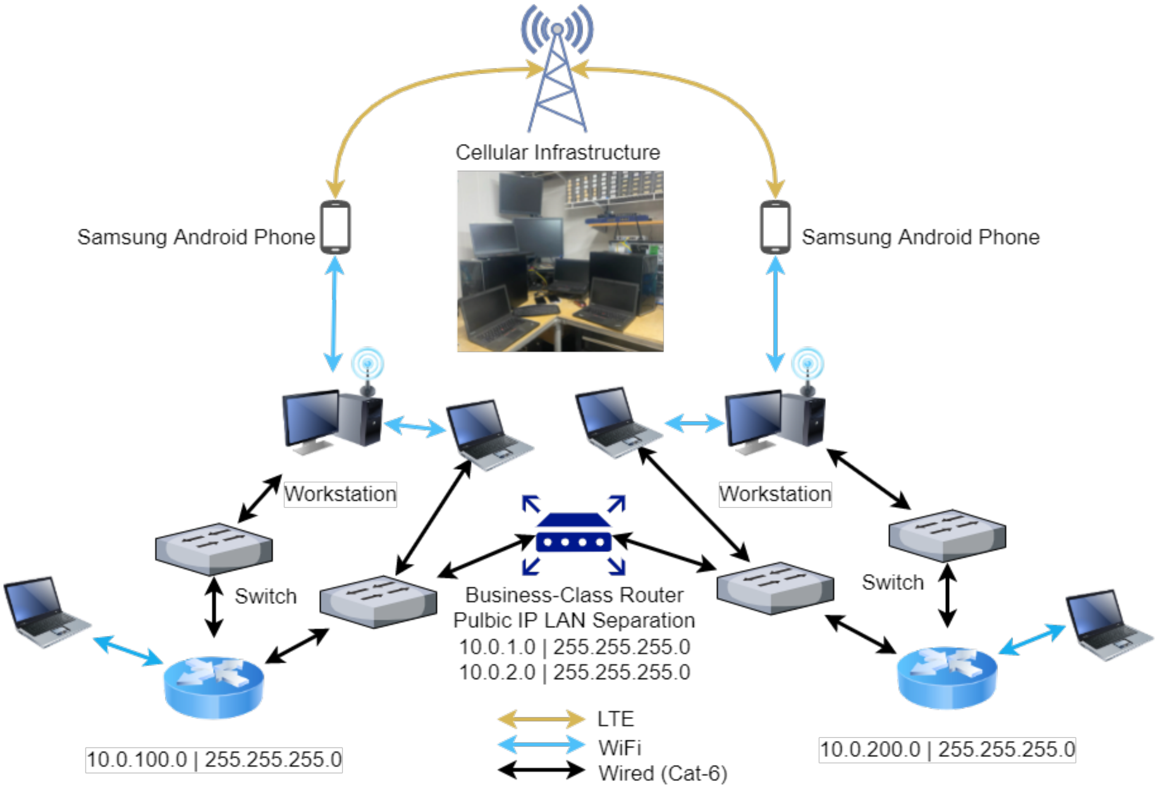


Figure 4.4: Testbed Topology Setup

- **Maximize** the link use providing feedback to increase or decrease the weights for the interface.
- **Classify** the reliability of each link by feeding the information to a classification model.

In addition to Machine Learning, other optimization mechanisms are possible, including heuristical, algorithmic, or stochastic. The same normalization functions, data formatting, and outputs can assist other techniques or be useful in different ML models as features for different traffic types, priorities, and necessities.

## 4.5 Experimental Analysis

The machine learning implementation and data preparation use data captured from a controlled testbed, portrayed in Figure 4.4. The testbed uses four laptop nodes, two cellular user equipment, two desktop nodes, and several networking devices to establish discrete

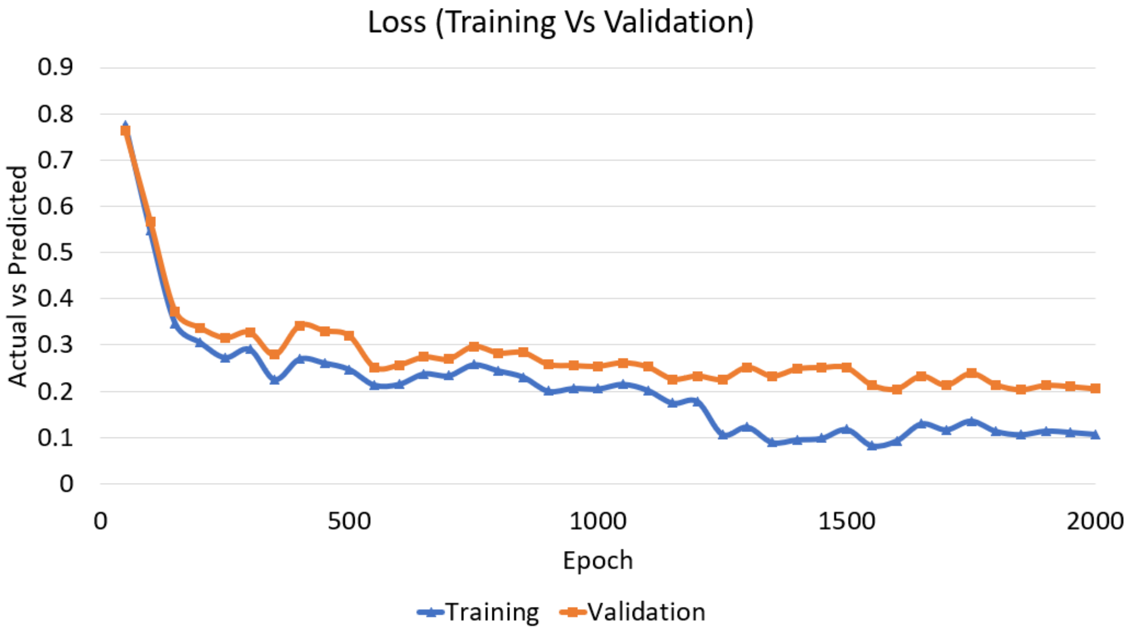


Figure 4.5: Training and Validation Loss

networks. The workstations have two WiFi interface cards connecting to the cellular phone and the laptop WiFi interface and a wired interface connected to the developed subnetwork.

The laptop and desktop nodes have dual-band wireless 802.11ac and gigabit Ethernet radio interfaces for use in the experiment. Two additional laptop nodes create traffic for data generation. The cellular nodes have WiFi interfaces and a 4G LTE connection through Straight Talk, a prepaid service.

Individually each node is limited to the maximum bandwidth available per interface. Through the implementation of CoopNet, parallel communication using all interfaces leads to a reduction in individual interface utility plus a higher aggregated throughput due to the combined bandwidth [25].

The data preparation uses the normalized Round Trip Time (RTT), instantaneous measurement of throughput, end to end delay measured from the packet header timestamp and received time, current interface activity, packet type (data, ACK, encrypted), traffic type, and average packet sizes as features and Radio Access Network type, and virtual link identifier as a label for the ML model.

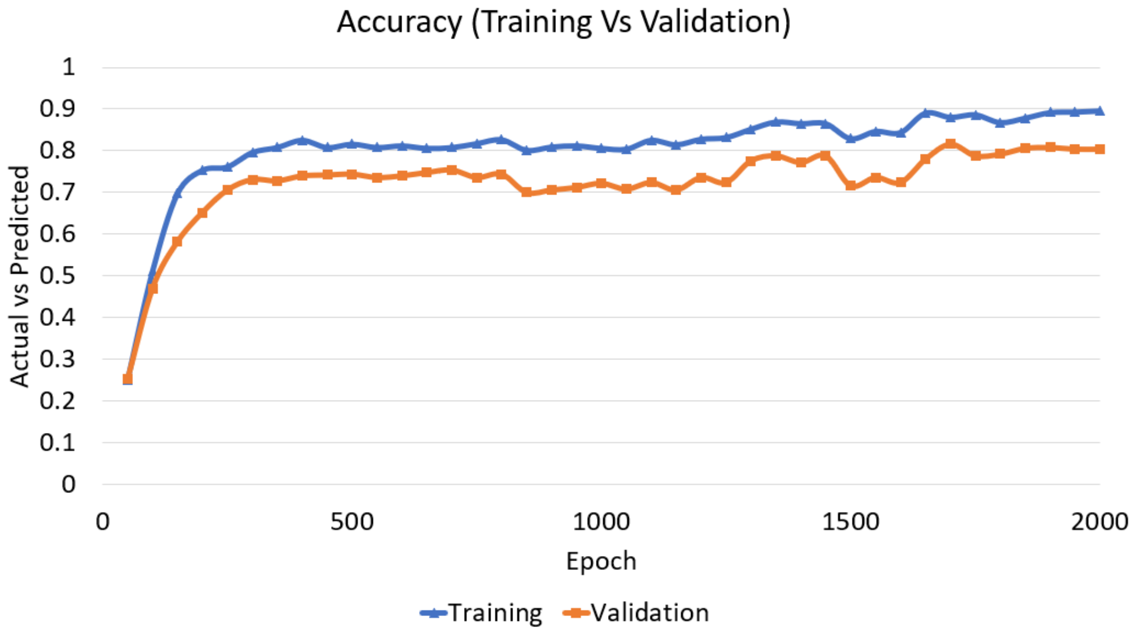


Figure 4.6: Training and Testing Accuracy

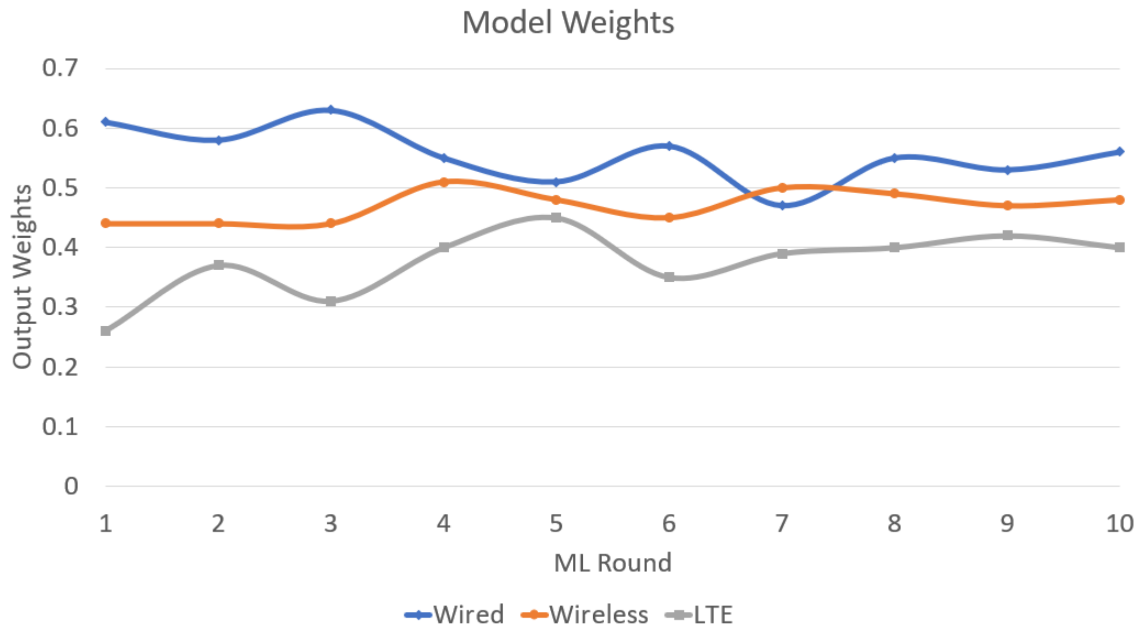


Figure 4.7: Two Stage Ensemble Machine Learning Output

It is essential to inform that the end-to-end delay inherits a clock skew between the devices, in some cases, leading to a negative value. Therefore the end-to-end delay normalization accounts for the time synchronization.

Figure 4.5 provides the ML model results showing the validation loss and accuracy over the number of epochs run. The hyperparameters and feature selection assist in accurately identifying the RAN type for the ground truth labeling. However, the more important set of information from the ML model is the predicted weights after training as provided in Figure 4.7. Setting the current link utilization to the predicted values improves communication performance compared to a heuristic selection.

Interestingly, the ML model reduces the utilization of the wired connection, which provides the highest set of resources by 9.18% in favor of using the LTE connection at the cost of increasing the LTE link use by 58.33% and WiFi link use by 11.9%. Figure 4.10 provides the individual link utility.

Comparing CoopNet to independent RANs using a heuristic implementation exists in [25]. The network performance compares the heuristic implementation of CoopNet with

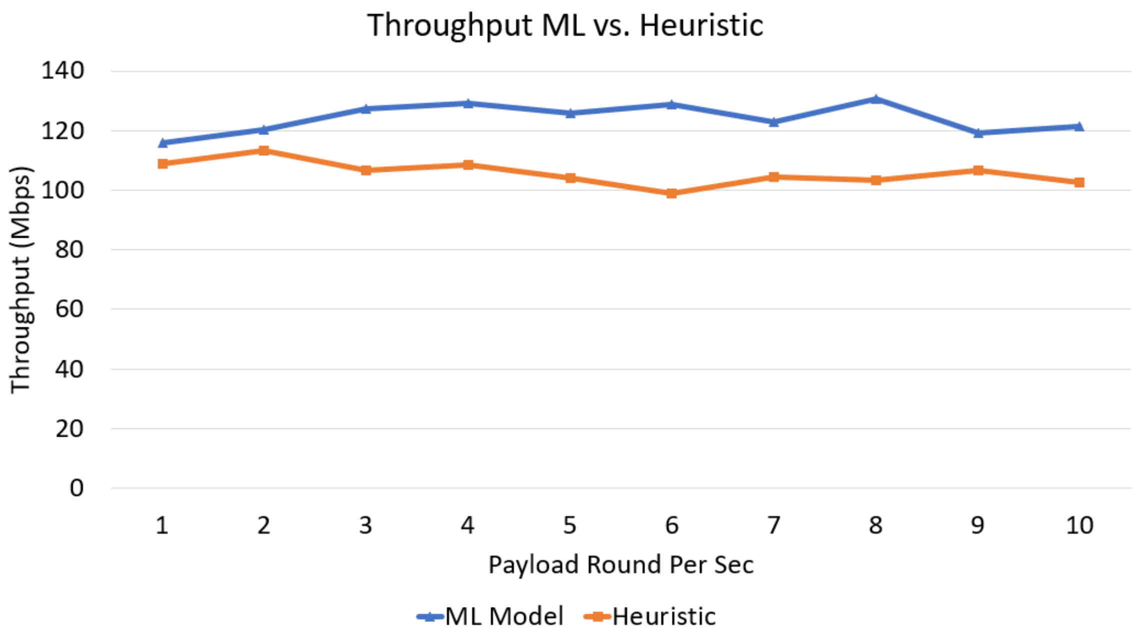


Figure 4.8: Throughput Machine Learning vs Heuristic

the ML model since the heuristic implementation is higher than traditional RAN usage.

Adjusting the traffic rate via the model output leads to an increase in both throughput and delay depicted in Figure 4.9, Figure 4.8. Both the delay and throughput are measured and averaged over data chunks of 10Mb. As Figure 4.9, Figure 4.8 portrays, there is a 17.39% throughput increase over the parallel links and an overall latency enhancement of 10.94%.

It is essential to note that the machine learning model is environment-specific and more likely to improve performance in a more static environment than a dynamic environment. The slow progression to a steady-state indicates that the model is non-linear and dependent on the current network dynamics.

Since the traffic condition changes based on ongoing communication, the model provides a different set of weights during each use. Hence, the following optimization step should include a time delta function limiting the link adjustment to eliminate the increased variability in traffic selection.

Moreover, developing an enhanced feedback system combined with an ensemble model

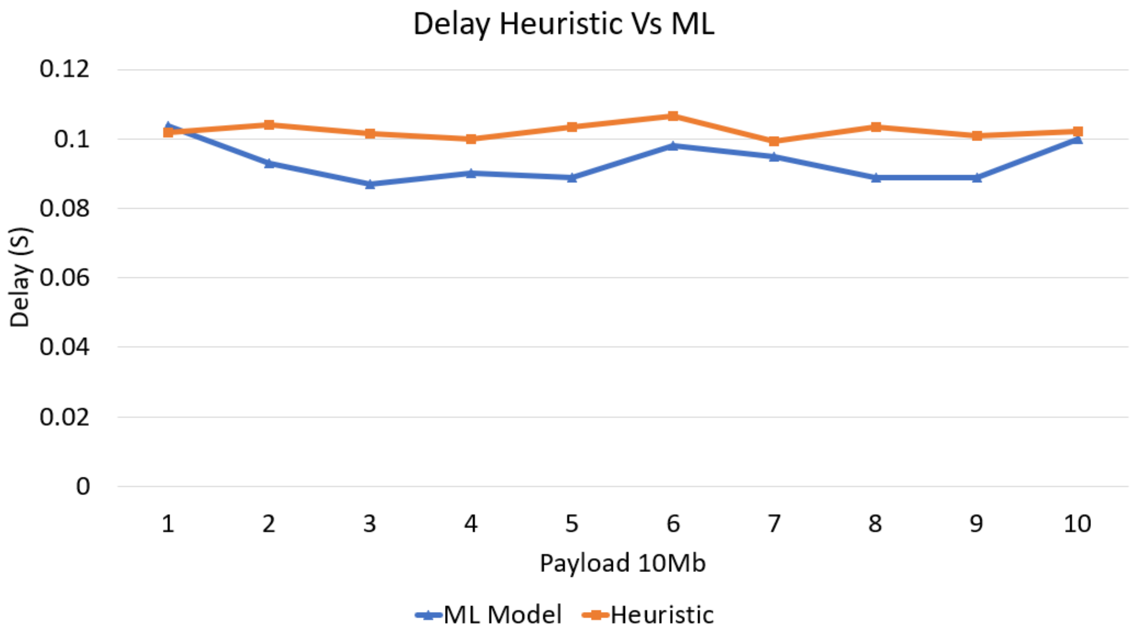


Figure 4.9: Delay Machine Learning vs Heuristic

that continuously updates may provide a real-time model for more diverse environments allowing it to auto-adjust. However, the training overhead may make such an approach infeasible with most edge devices.

## 4.6 Summary

The contributions of the chapter are twofold. The first main contribution introduces the final two stages of CoopNet, separating the responsibilities of data preprocessing, normalization, and formatting from the network dynamic adaptations. The separation allows for a programmable methodology where the data preparation implementations are achievable through a middlebox concept taking in inputs and providing outputs.

The middlebox's interworking can be anything from the default normalization function, direct passthrough, to customized machine learning approaches, including dimensionality decrease and encodings. This separation will allow network engineers to focus on network improvement development without affecting the existing functionality.

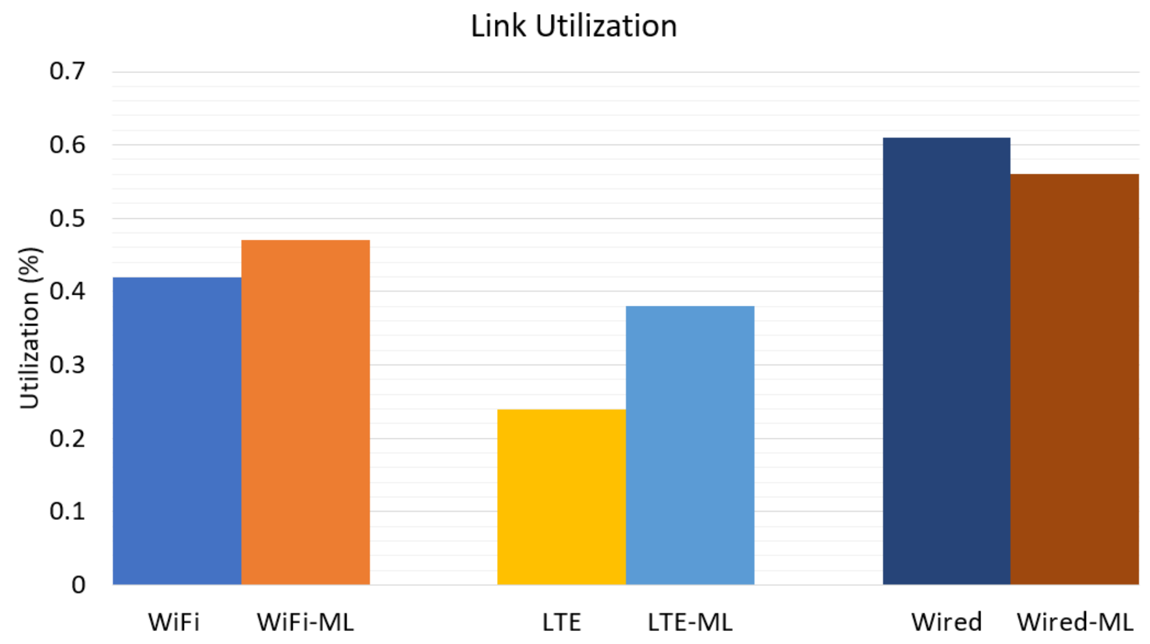


Figure 4.10: Link Utilization Comparison

Similarly, network engineers can develop novel Dynamic Adaptation heuristic, algorithmic, or machine learning implementations more readily by having access to an expected set of metrics, parameters, and options to use. The same programable concept follows all CoopNet modules to implement new node discovery, link establishment, security, or other features, including defining new metric and sharing frameworks.

The second contribution is the machine learning optimization for link utility. CoopNet defines links as any possible physical link, virtual link, communication channels, tunnels, connections up to two hops away. CoopNet allows RAN translation by using a middle node to translate from one interface to another privately and transparently or publically by actively modifying the messages.

Thus, multiple virtual links can exist on a single communication interface. For example, a node can connect with an IoT access point and simultaneously with a Cellular device using a single WiFi radio interface. Since RAN translation with virtual links is a newer concept, the publication contribution provides a machine learning optimization that can adaptively improve the network dynamics through periodic training [51].

# CHAPTER 5

# DISCUSSION

The internet architecture must evolve to meet the next-generational demands of cloud/fog computing, the Internet of Things (IoT), artificial intelligence, machine learning, and users. The traditional communication stack represents a point-to-point connection. However, big data, machines, sensors, and social content have made the conventional communication platform unfit to sustain the current needs.

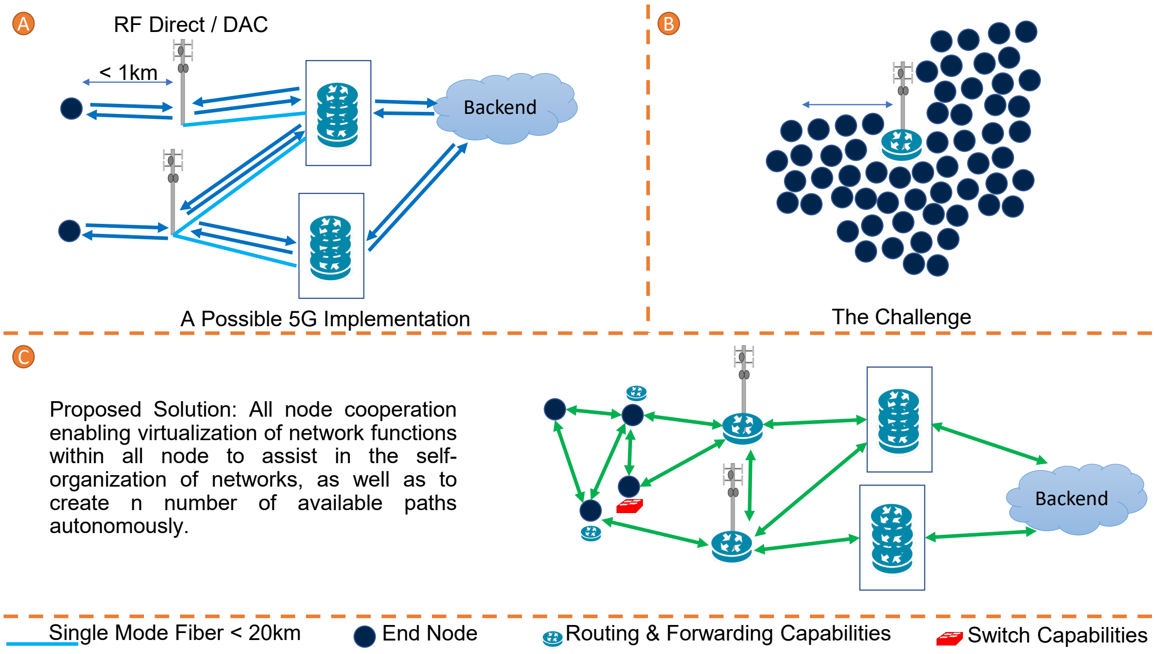
In large, the Internet backbone and protocols have remained unchanged. CoopNet introduces, enhances, and implements a semi-autonomous mutative parallel multipath architecture to meet future demands in data access, latency requirements, connectivity, and awareness. Furthermore, CoopNet improves spectrum efficiency by using idle resources through a cooperative approach.

Figure 5.1 is a visual representation of a current setup in communication (a), an existing challenge (b), and the possible outcome through the use of CoopNet (c). As portrayed in Figure 5.1a, next-generation 5G cellular communication requires installing base stations closer due to higher frequency use, less than one kilometer apart, to the end nodes and requires all end nodes to use the 5G core.

However, as shown in Figure 5.1b, deep network awareness and dealing with a massive increase in the number of devices requiring connectivity is still an ongoing challenge. Resolving the network awareness is increasingly challenging without providing active network metric sharing and more advanced discovery mechanisms, especially when nodes implement a variety of Radio Access Technologies (RATs).

Hence, CoopNet implements a programable platform to include different mechanisms of discovery for improving awareness. Figure 5.1c shows a possible outcome of CoopNet. Additionally, the node cooperation in Figure 5.1c allows all nodes to assist in communication adding virtual network functions, including routing, forwarding, and enhanced security.

Figure 5.1: Proposed Visual Representation



Proposed Solution: All node cooperation enabling virtualization of network functions within all node to assist in the self-organization of networks, as well as to create n number of available paths autonomously.

Current advances in communication in the works, including higher frequency spectrum, fiber to the antenna deployment using Common Public Radio Interface (CPRI) [34], direct analog radio frequency (RF) over fiber [31] [30], and Centralized/Cloud Radio Access Network (CRAN) [52]. CRANs store baseband units, ordinarily located next to the antenna, in a shared location.

Pooling baseband units together allow making use of resource sharing, network virtualization, and software-defined networking (SDN) approaches [53]. With a centralized radio access network, messages are disseminated cooperatively with more delicate control [54].

CRANs assist in making antenna deployment more simple and more economical and aid with features such as Multiple-Input Multiple-Output (MIMO), beamforming, and beam steering technologies that improve channel capacity while mitigating interference created at the edge nodes. Introducing higher radio frequencies lowers the transmission distance requiring smaller cells from nano cells to having a no-cell architecture [30].

However, CRANs are not ideal for every situation since localized communication must pass through the centralized station for packet forwarding leading to an increase in latency and possible bottleneck situations. A proper solution requires a mix of centralized and distributed approaches.

Permitting nodes to become a part of the network and adopting multiple parallel routing paths enhances both distributed and centralized approaches [13]. Centralized systems can accurately distribute content over different routes, alleviating the path-specific bandwidth, while distributed systems can implement more environment-specific communication.

A programable communication platform allows for more dynamic communication and autonomy, facilitating both implementation and abilities. However, maintaining backward compatibility is challenging and requires careful consideration during deployment.

One solution for backward compatibility is implementing the communication improvement at the end systems allowing end nodes to select between the different modes, to use standard headers and protocols or updated techniques. For example, when developing new applications, the usage of CoopNet can provide additional features, such as opening connections, sending or receiving data, and more. At the same time, existing applications can continue to use traditional forms of communication or allow CoopNet to handle the usage of conventional networks.

Opening connections allows CoopNet to generate its flows using multiple interfaces for parallel communication. Another benefit of programable networking is implementing mechanisms for splitting and assembling flows and data optimally.

Alternatively, merging connection flows is more complex for existing applications, primarily when packets use encryption, which most internet traffic currently implements. Older generation applications typically share a symmetric encryption key for the particular connection and often require in-sequence packet reception. Stripping a message of the existing headers and transmitting the packet via multiple paths requires additional investigation and development.

Capturing outgoing packets and forwarding them unaltered via a different interface is feasible. However, the current networking interworking requires a node to receive feedback on a particular interface to determine round trip times and packet sequence. Hence, even unaltered packets stripped from a single interface require additional consideration if backward compatibility is vital.

Another possible consideration is to enable single interfaces to generate more than one active and permanent connection allowing a single interface to transmit to different networks while retrieving the acknowledgments on the same interface. Network management software and most operating systems do not support more than a single permanent and active connection.

Therefore, additional software development contribution beyond a forceful transmission through an interface is necessary. Such a contribution would be helpful for CoopNet and standard communication by allowing a single interface to communicate with different gateways simultaneously. CoopNet assembles packets not intended for the existing connection and pushes the message through the interface as a workaround. The forceful transmission is not ideal since TCP connections rapidly terminate while using UDP increases packet drops.

Even if no architecture that implements a programable platform exists, the ability for multiple active connections would advance communication capabilities. MPTCP may expand to utilize various links using the same interface. Thus, allowing application-specific link selection, traffic splitting, or any other novel implementation, such as synergic multiple link connections similar to MPTCP that can operate over parallel paths.

A primary concern during CoopNet development is enhanced network awareness to facilitate the integration of massive IoT deployments. CoopNet integrates both active and passive network awareness. Active discovery mechanisms include request and response methods, polling, and NetShare. Netshare is a network-sharing framework for enhancing network awareness between neighboring nodes as part of this research contribution [25].

To reduce the overhead of NetShare, CoopNet transmits minimal advertisements if a node can assist in communication and is actively using an interface below a set threshold of the available bandwidth. For inactive useable interfaces, NetShare sends additional advertisements and requests.

Active discoveries and data sharing are easier to accomplish in a programmable manner. Transmitting packets using a broadcast address can allow neighboring nodes to capture the traffic without dropping the packets. However, passive discovery implementations require significant contributions since most end nodes and systems do not monitor or collect channel noise and traffic. For proper execution, a node must be capable of switching the wireless device into promiscuous mode and enable wireless capturing using open-source libraries, such as the Aircrack-NG suite.

The wireless device and existing drivers must support the open-source library. However, most of the devices do not natively support implementing passive discovery. Even if the device supports passive discovery, the operating system must allow such an implementation through the hardware access processes, with windows providing minimal to no passive discovery capabilities. Utilizing a Linux environment with an open-source library and development offers limited proof of concept capabilities for research purposes.

Aircrack-NG suite provides an API for capturing wireless traffic and is ideal for wireless investigations and security. However, implementing communication features alongside the open-sources library is not as straightforward since the library did not intend to transmit and capture data simultaneously. Hence, the usage of passive discovery does not extend beyond a proof of concept research implementation restricted by the current networking limitations.

Figure 5.2 and Figure 5.3 provide preliminary results comparing passive discovery with blind transmissions or unicast transmissions after successful node discoveries using a channel sensing passive approach. The results present the capture accuracy of different packet

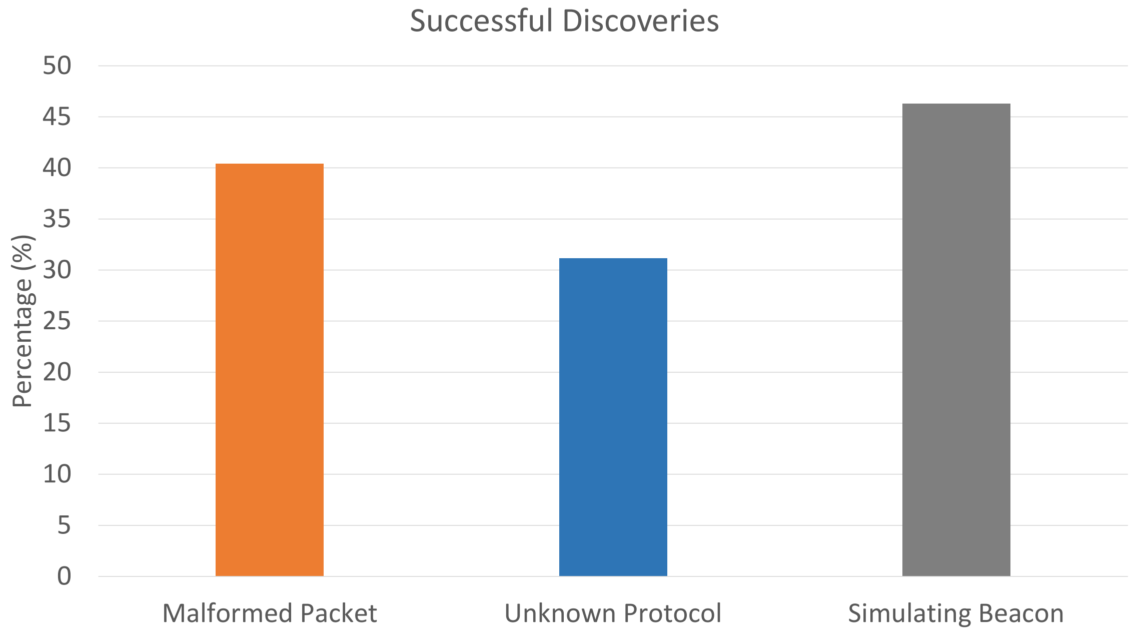


Figure 5.2: Correct Packet Reception

types by using the open-source library.

After an accurate capture, the rate of successful transmission using directed transmission is approximately 70%. However, using blind transmission with public node cooperation to forward the traffic is less than 10%.

Therefore, persuading the research community to support channel sensing and blind transmission integration and further developing communication capabilities of end nodes can drastically improve communication in environments such as IoT. Passive discoveries can play a vital role in an infrastructure-less scenario where nodes do not have a centralized communication platform for generating connections and transmitting data.

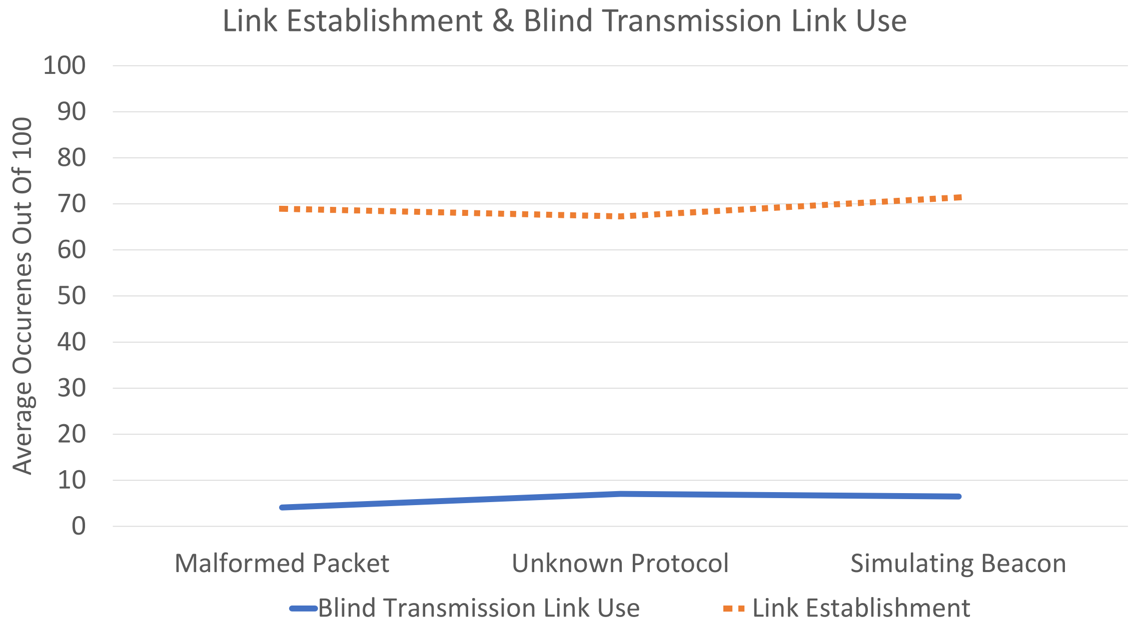


Figure 5.3: Successful Link Establishment

## 5.1 Infrastructure as a Service

An open, programmable platform for communication can extend into a new service-generating industry of always available pay-per-use communication. Currently, cellular service providers offer the only communication platform with large-scale connectivity coverage, especially with roaming.

Certain service providers, including cable internet providers, offer in-range connectivity. However, users must be subscribed to the provider's service and must connect using the provided credentials. More recently, low earth orbit (LEO) satellite internet enables accessing a satellite network for connectivity with a broader range of coverage.

All of the above examples require having a specific provider service. Thus, in most cases, the connectivity capabilities are limited to using a single interface and access network. Instead, an open communication platform assisted by an architecture such as CoopNet can provide a new Infrastructure as a Service (IaaS) system where nodes can use any available interfaces and share internet access with neighboring nodes.

The sharing can be complimentary or a paid service. Users may join a group that offers extended internet access for users, similar to peer-to-peer downloading. Alternatively, a new payment system can provide users with the ability to provide a payment method and pay the user sharing their services based on usage.

The benefit of an IaaS approach is using all available resources and interfaces simultaneously, enhancing communication performance. Moreover, users may access the internet upon need instead of requiring a subscription for internet connectivity. Lastly, the likelihood of finding users providing public cooperation and internet access increases if users receive compensation for the collaboration.

## 5.2 Security

Nodes providing networking functions can extend into a new array of cybersecurity protections. Intermediary nodes can implement deep packet inspection, firewall, or security applications providing an enhanced node mechanism for traffic monitoring and privacy.

Intermediary nodes can provide a certain level of anonymity by transmitting messages using their unique identifier instead of the generating node. An encrypted message block can contain the originating source's address to inform the receiver of the generating node.

Additionally, link-hopping and multiple parallel paths allow sending packets via differing routes, forcing an external attacker to know the link selection use. In some cases, finding the exact link selection may be statistically improbable for the attacker since different nodes can implement unique link selection methods.

## 5.3 Protocol Agnostic Communication

Finally, there is a vital need for considering protocol-agnostic communication and networking. With the advent of Satellite internet and custom protocol development for IoT, the conventional communication protocols no longer suffice.

The satellite mesh network implementations handle routing and forwarding in a satellite network. The ground station units ofter communicate directly with the satellite network, removing the need for a central office tieing all gateways and termination points together.

Similarly, for Vehicular Ad-Hoc Networks, IoT deployment, and Sensor Networks, adhoc device-to-device communication is becoming more popular. Often, each environment implements a unique set of protocols and frameworks.

A possible solution is to allow communication to use any available protocols or provide a protocol-less implementation. Such a system offers the end nodes advanced capabilities to generate communication messages in any available scheme and allows the neighboring node to forward the message correctly.

Finally, the necessity for automated connections is more prevalent than ever to reduce the need for manual user connectivity. Instead, neighboring nodes can assist in validation and node verification for autonomous connectivity. Concepts derived from existing platforms, such as wireless HotSpot 2.0 and cellular session establishments and handoffs, can guide the realization of an open communication paradigm [2][9][11][28][55].

# CHAPTER 6

# CONCLUSION

This research helps develop the structure and proof of concept implementation of a horizontal communication architecture, termed Cooperative Networking Architecture (CoopNet). CoopNet introduces a programmable platform for developing innovative mechanisms to improve and provide parallel multipath communication.

CoopNet depends on node cooperation allowing end nodes and systems to become part of the network infrastructure and offer networking functions through a Network Function Virtualization (NFV) approach. NFV allows regular nodes to assist in routing, forwarding, security, and other functionality. The three classes of cooperation in CoopNet are:

**Public** cooperating nodes assist in all networking functions, including RAN translation, forwarding, routing, and different applications such as network management, cybersecurity, load balancing, limited only by the available hardware and resources.

**Private** cooperating nodes assist transparently, including direct packet forwarding, limited security functions, RAN translation, and more. CoopNet recommends leaving all packet formatting to the originating host for best implementation. However, a privately cooperating node can update header information as needed in more limited situations.

**No-Cooperation** indicate nodes that do not wish to participate in networking functions for differing reasons, including policy, limited resources, privacy, and regardless of any rationale.

CoopNet provides five different modules, Discovery, Decision, Utilization, Data Collection, and Dynamic Adaptation. Discovery focuses on implementing network awareness through existing approaches and newly presented methods.

CoopNet implements both active and passive discovery mechanisms. Active discovery mechanisms include request and response, polling, and network awareness sharing. Request and response mechanisms are the most common by providing a session establishment procedure or request for connectivity.

Passive discoveries mechanisms include feedback loop, blind transmission, or channel sensing. A feedback loop includes transmitting a specially formatted packet and receiving the same message as a reply. Channel sensing provides passive discovery based on ongoing radio communication and channel noise fingerprinting. Channel sensing required a radio to switch between listening and transmitting.

Due to the operating system and hardware limitations, the channel sensing mechanism requires further development for a release-ready solution. Finally, CoopNet introduces a network awareness sharing framework termed NetShare for providing neighboring nodes with information about the network dynamics.

CoopNet provides several Decision mechanisms for parallel communication, including event-based, adaptive sliding windows, differential, application/traffic specific, and machine learning link selection and utilization.

**Event Triggered** indicates the use of events, such as video buffering or channel condition changes, for updating utilization. For example, with video content and buffering, a node may download different chunks using different links and buffering the downloaded content.

**Adaptive Sliding Windows** allows a node to adaptively select and modify link use without event triggering by utilizing information such as round trip time (RTT), bandwidth, and throughput calculations for real-time continuous data transmission.

**Differential** access provide a unique mechanism for downloading static content via parallel paths. It allows downloading from opposing ends and splitting the content to match the number of available parallel links. The higher ranked links begin from the front and the slower link from the rear.

**Data & Application Specific** allows utilizing parallel paths based on traffic types or application requirements. For example, a node may use one link for continuous sensory data reception while sending the acknowledgment via a separate link in parallel.

**Machine Learning** models allow proportionally selecting links for data communication based on several features, such as round trip time, instantaneous throughput, end to end delay, activity, link-state, and more.

Utilization handles both internal and external metric collection and the NetShare Framework. NetShare is a network-sharing framework enabling information exchange between neighboring nodes. NetShare assists with RAN Translation and reachability by providing the ability to create virtual links two hops away.

Radio Access Network (RAN) translation extends the capabilities of a node to include neighbor capabilities. Using CoopNet, a sender can create virtual links using the neighbor's access network. Data transmission between the sender and the intermediary node, the neighbor, uses the shared communication access network. Upon receiving packets to be forwarded, the intermediary node can forward a packet unaltered or update the protocol to support a different access network.

In CoopNet, physical and virtual links can be permanent, semi-permanent, or temporary enabling active connections for communication, including physical and virtual links. Virtual links allow nodes to create a relationship two hops away through intermediary nodes. Intermediary nodes can act as RAN translational nodes to convert packets from one access network type to another or forwarding nodes, enhancing the reach and capabilities.

The Data Collection module manages the segmentation, assembly, and modification of received and transmitted data. Moreover, the Data Collection module implements preprocessing, segmentation, assembly, formatting, and translation mechanism needed for optimization mechanisms or other modules.

Incorrect data segmentation, normalization, and formating can easily lead to improper fit, sparsity, variance, and unnecessary dimensionality. Similarly, improper assembly or translation leads to increased packet loss. For this reason, CoopNet separates Data Collection, the data preparation module, from other modules.

The Data Collection module is splint into independent compartments where data flows from the input to the output, progressing through several different mechanisms when enabled. Packets received in an intermediary node proceeds to formatting to check if the packer required header and protocol updating.

Afterward, the Data Collection module places the packet in the queue for transmission. Similarly, data received at the Data Collection relating to internal or external metrics flow through several normalizations, preprocessing, reduction, and formatting mechanisms for usage with Dynamic Adaptation.

Dynamic Adaptation offers the implementation procedures for transmitting data efficiently using parallel multipath links, including heuristic, event-triggered, or machine learning approaches. Often, a single machine learning implementation in networking is not sufficient due to the dynamic nature and extensive set of parameters.

Ensemble machine learning allows for staged processes implementing different machine learning algorithms while updating the features and inputs dynamically. A staged approach is essential for CoopNet due to virtual links through an intermediary node.

Not all advanced mechanisms are useable in all environments. Resource-limited environments such as the Internet of Things and wireless sensor networks cannot sustain complex designs. Unique conditions mandate using different network optimization and adaptation mechanisms, including possible hardcoded procedures or policy-based. Hence both advanced and basic models are necessary, providing an ideal reason for developing a programable network environment.

CoopNet implements an API design between the transport and application layers to support receiving data from specific processes for backward compatibility. In turn, CoopNet establishes the connections and maintains the flows via different paths, unless otherwise dictated from the application layer during the setup procedures.

Similarly, CoopNet operates as a separate architecture in a custom network environment and provides direct physical layer modulation calls. Acting as an independent architecture can be helpful in military applications where complete detachment is necessary for communication.

Ultimately, the goal is to achieve an address and protocol agnostic form of communication, allowing nodes to sustain communication in various scenarios without rigid frameworks. Traditional packet standards and overhead inhibit communication performance and advancements by requiring the complete implementation of a protocol sequence.

For example, instead of forcing the use of conventional packets headers, universal packet headers are beneficial by reducing the overhead and allowing the use of a range of identifiers instead of standard addresses, such as Internet Protocol v4/6 addresses. Ubiquitous packets can operate like Multiprotocol Label Switching (MPLS), where path labels aid routing instead of generalized or address-based routing. Moreover, a universal packet format can benefit specific environments that do not require the same header scheme as traditional standards.

Separating the protocol dependency at the top level allows each path to use independent implementations for services such as data reliability, bandwidth guarantees, security, adaptive rates, adaptive link selection, and connection setups. Due to different link dynamics and capabilities, implementations may differ significantly, especially with new advancements in IoT, Ad-Hoc, Satelite, and short-lived local communication. Hence, CoopNet uses an agnostic networking method to function across different access network types reducing unnecessary overhead and limitations.

CoopNet is unique from cellular offloading since it can use all available access networks, including satellite, high mobility Ad-Hoc access networks, and any custom implementations in IoT or similar environments. It is possible to use various access networks by allowing a management architecture to control the creation and usage of communication.

Lastly, a programmable platform provides tremendous advantages, including additional cybersecurity defenses, Infrastructure as a Service, efficient access network use, and a wide range of unique network developments, such as isolated and detached networks. CoopNet can be improved to integrate an autonomous connection platform using crowd validation and further implement self-healing networks with a considerable contribution.

# REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, Fourth Quarter 2015.

[2] F. Hu, B. Chen, and K. Zhu, “Full spectrum sharing in cognitive radio networks toward 5g: A survey,” *IEEE Access*, vol. 6, pp. 15 754–15 776, 2018.

[3] İ. Bekmezcia, O. K. Sahingoza, and Ş. Temel, “Flying ad-hoc networks (fanets): A survey,” in *Ad Hoc Networks*, vol. 11, 2013, pp. 1254–1270.

[4] A. Asadi, Q. Wang, and V. Mancuso, “A survey on device-to-device communication in cellular networks,” *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1801–1819, Fourth Quarter 2014.

[5] J. B. Kenney, “Dedicated short-range communications (dsrc) standards in the united states,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[6] R. C. Voicu, H. I. Abbasi, H. Fang, B. Kihei, J. A. Copeland, and Y. Chang, “Fast and reliable broadcasting in vanets using snr with ack decoupling,” in *2014 IEEE International Conference on Communications (ICC)*, IEEE, 2014, pp. 574–579.

[7] H. I. Abbasi, R. C. Voicu, J. A. Copeland, and Y. Chang, “Performance optimization of a contention based broadcasting algorithm in vanets,” in *2015 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2015, pp. 1–7.

[8] A. Network, “3rd generation partnership project; technical specification group services and system aspects; release 16 description; summary of rel-16 work items (release 16),” *3gpp Technical Report*, Sep. 2020.

[9] S. Kanugovi, F. Baboescu, J. Zhu, and J. Seo, “Multi-access management services (mams),” *IETF RFC 8743*, Mar. 2020.

[10] O. Elloumi and et al, “Harmonizing standards for edge computing - a synergized architecture leveraging etsi isg mec and 3gpp specifications,” *ETSI White Paper 36*, Jul. 2020.

[11] J. Ha and Y.-I. Choi, “Support of a multi-access session in 5g mobile network,” in *2019 25th Asia-Pacific Conference on Communications (APCC)*, IEEE, 2019, pp. 378–383.

[12] M. Condoluci, S. H. Johnson, V. Ayadurai, M. A. Lema, M. A. Cuevas, M. Dohler, and T. Mahmoodi, “Fixed-mobile convergence in the 5g era: From hybrid access to converged core,” *IEEE Network*, vol. 33, no. 2, pp. 138–145, 2019.

[13] R. C. Voicu, J. A. Copeland, and Y. Chang, “Multiple path infrastructure-less networks a cooperative approach,” in *2019 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, 2019, pp. 835–841.

[14] L. Doyle, J. Kibiłda, T. K. Forde, and L. DaSilva, “Spectrum without bounds, networks without borders,” *Proceedings of the IEEE*, vol. 102, no. 3, pp. 351–365, 2014.

[15] T. Yoo, “Network slicing architecture for 5g network,” in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2016, pp. 1010–1014.

[16] J. Li, X. Zhang, S. Wang, and W. Wang, “Context-aware multi-rat connection with bi-level decision in 5g heterogeneous networks,” in *2017 IEEE/CIC International Conference on Communications in China (ICCC)*, Oct. 2017, pp. 1–6.

[17] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, “End-to-end simulation of 5g mmwave networks,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[18] H. I. Abbasi, R. C. Voicu, J. A. Copeland, and Y. Chang, “Cooperative bsm architecture to improve transportation safety in vanets,” in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Jun. 2017, pp. 1016–1022.

[19] R. Mishra, P. Verma, and R. Kumar, “Gateway discovery in manet using machine learning and soft computing: A survey,” in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Mar. 2017, pp. 1–6.

[20] J. Li, Y. Zhou, and L. Lamont, “Communication architectures and protocols for networking unmanned aerial vehicles,” in *2013 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2013, pp. 1415–1420.

[21] H. C. Lee and K. H. Ke, “Monitoring of large-area iot sensors using a lora wireless mesh network system: Design and evaluation,” *IEEE Transactions on Instrumentation and Measurement*, pp. 1–11, 2018.

[22] A. M. Cailean, B. Cagneau, L. Chassagne, V. Popa, and M. Dimian, “A survey on the usage of dsrc and vlc in communication-based vehicle safety applications,” in

*2014 IEEE 21st Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Nov. 2014, pp. 69–74.

[23] J. S. Perez, S. K. Jayaweera, and S. Lane, "Machine learning aided cognitive rat selection for 5g heterogeneous networks," in *2017 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, Jun. 2017, pp. 1–5.

[24] K. D. Joshi and K. Kataoka, "Sfo: Subflow optimizer for mptcp in sdn," in *2016 26th International Telecommunication Networks and Applications Conference (ITNAC)*, Dec. 2016, pp. 173–178.

[25] R. C. Voicu and Y. Chang, "Stages of coopnet: A multipath parallel link architecture for next-gen networks," in *2021 International Wireless Communications and Mobile Computing (IWCMC)*, IEEE, 2021, pp. 592–597.

[26] R. I. Meneguette, E. R. M. Madeira, and L. F. Bittencourt, "Multi-network packet scheduling based on vehicular ad hoc network applications," in *2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualiztion management (svm)*, Oct. 2012, pp. 214–218.

[27] F. Bendaoud, M. Abdennebi, and F. Didi, "Network selection using game theory," in *2015 3rd International Conference on Control, Engineering Information Technology (CEIT)*, May 2015, pp. 1–6.

[28] S. Maaloul, M. Afif, and S. Tabbane, "Handover decision in heterogeneous networks," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, Mar. 2016, pp. 588–595.

[29] Q. De Coninck and O. Bonaventure, "Multipathtester: Comparing mptcp and mpquic in mobile environments," in *2019 Network Traffic Measurement and Analysis Conference (TMA)*, IEEE, 2019, pp. 221–226.

[30] G. Chang, M. Xu, and F. Lu, "Fiber-wireless convergence for next generation heterogeneous mobile data communications," in *2017 IEEE Photonics Society Summer Topical Meeting Series (SUM)*, Jul. 2017, pp. 117–118.

[31] G. Chang and L. Cheng, "Fiber-wireless integration for future mobile communications," in *2017 IEEE Radio and Wireless Symposium (RWS)*, Jan. 2017, pp. 16–18.

[32] M. Li, F. R. Yu, P. Si, E. Sun, Y. Zhang, and H. Yao, "Random access and virtual resource allocation in software-defined cellular networks with machine-to-machine communications," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6399–6414, Jul. 2017.

[33] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. D. Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, “5g: A tutorial overview of standards, trials, challenges, deployment, and practice,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201–1221, Jun. 2017.

[34] A. de la Oliva, J. A. Hernandez, D. Larrabeiti, and A. Azcorra, “An overview of the cpri specification and its application to c-ran-based lte scenarios,” *IEEE Communications Magazine*, vol. 54, no. 2, pp. 152–159, Feb. 2016.

[35] J. Zhang, M. Xu, J. Wang, F. Lu, L. Cheng, H. Cho, K. Ying, J. Yu, and G. Chang, “Full-duplex quasi-gapless carrier aggregation using fbmc in centralized radio-over-fiber heterogeneous networks,” *Journal of Lightwave Technology*, vol. 35, no. 4, pp. 989–996, Feb. 2017.

[36] O. Michel and E. Keller, “Sdn in wide-area networks: A survey,” in *2017 Fourth International Conference on Software Defined Systems (SDS)*, IEEE, 2017, pp. 37–42.

[37] H. I. Abbasi, R. C. Voicu, J. Copeland, and Y. Chang, “Towards fast and reliable multi-hop routing in vanets,” *IEEE Transactions on Mobile Computing*, 2019.

[38] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[39] H. I. Abbasi, R. C. Voicu, J. A. Copeland, and Y. Chang, “Cooperative bsm architecture to improve transportation safety in vanets,” in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, 2017, pp. 1016–1022.

[40] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, “A survey on 5g networks for the internet of things: Communication technologies and challenges,” *IEEE access*, vol. 6, pp. 3619–3647, 2017.

[41] W. Alliance, “Hotspot 2.0 specification package,” in *Wi-Fi Alliance Passpoint Release 3*, 2012.

[42] W. Xu, H. Zhou, Y. Bi, N. Cheng, X. Shen, L. Thanayankizil, and F. Bai, “Exploiting hotspot-2.0 for traffic offloading in mobile networks,” *IEEE Network*, vol. 32, no. 5, pp. 131–137, 2018.

[43] T. T. Nguyen and G. Armitage, “A survey of techniques for internet traffic classification using machine learning,” *IEEE communications surveys & tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[44] T. O'shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[45] B. Hayes, Y. Chang, and G. Riley, "Omnidirectional adaptive bitrate media delivery using mptcp/quic over an sdn architecture," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, IEEE, 2017, pp. 1–6.

[46] ——, "Adaptive bitrate video delivery using http/2 over mptcp architecture," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, 2017, pp. 68–73.

[47] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[48] K. A. Taher, B. M. Y. Jisan, and M. M. Rahman, "Network intrusion detection using supervised machine learning technique with feature selection," in *2019 International conference on robotics, electrical and signal processing techniques (ICREST)*, IEEE, 2019, pp. 643–646.

[49] M. Bennis, M. Simsek, A. Czylwik, W. Saad, S. Valentin, and M. Debbah, "When cellular meets wifi in wireless small cell networks," *IEEE communications magazine*, vol. 51, no. 6, pp. 44–50, 2013.

[50] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019.

[51] R. C. Voicu and Y. Chang, "Optimizing multipath parallel communication using a cooperative architecture, coopnet," in *2022 International Conference on Communications (ICC)*, IEEE, 2022 — Pending Review.

[52] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, "Heterogeneous cloud radio access networks: A new perspective for enhancing spectral and energy efficiencies," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 126–135, Dec. 2014.

[53] M. Peng, Y. Li, Z. Zhao, and C. Wang, "System architecture and key technologies for 5g heterogeneous cloud radio access networks," *IEEE Network*, vol. 29, no. 2, pp. 6–14, Mar. 2015.

[54] Z. Chen, J. Lee, T. Q. S. Quek, and M. Kountouris, "Cooperative caching and transmission design in cluster-centric small cell networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 3401–3415, May 2017.

[55] W. Xu, H. Zhou, Y. Bi, N. Cheng, X. Shen, L. Thanayankizil, and F. Bai, “Exploiting hotspot-2.0 for traffic offloading in mobile networks,” *IEEE Network*, vol. 32, no. 5, pp. 131–137, 2018.

# VITA

## Razvan Cristian Voicu
## Electrical and Computer Engineering — Georgia Institute of Technology

| EDUCATION | UNIVERSITY | TERM | GPA |
|---|---|---|---|
| Ph.D. ECE | Georgia Institute of Technology | Spr'22 | 4.00 |
| M.S. ECE | Georgia Institute of Technology | Sum'19 | 4.00 |
| B.S. EE | Kennesaw State University | Spr'13 | 4.00 |
| B.S. CET | Kennesaw State University | Spr'11 | 3.98 |

| ACCREDITATION | ENTITY | TERM | LICENSE |
|---|---|---|---|
| Fundamentals of Engineering | Georgia Secretary of State | Sum'11 | EIT024944 |

### EXPERIENCE

**Kennesaw State University** Limited Term Full Time Faculty January 2021-Present

- Responsible for delivering Computer Networking, Intermediate Programming for Mechatronics, and Mechatronics Fundamental courses.
- Deliver and maintain laboratories for numerous subjects, including Feedback and Controls, Senior Design, Intermediate Programming for Mechatronics, and Mechatronics Fundamental.
- Mentor internship students and assist as a faculty advisor for Senior Design project in Electrical Engineering.

**Kennesaw State University** Part Time Faculty August 2020-January 2021

- Responsible for delivering Computer Networking course, assist with student project, and cover networking topics from the application layer to the physical layer and associated protocols.
- Python programming introduction for Socket and network programming.

**Fortiphyd** Consultant January 2019-Present

- Software development design and implementation in Operational Technology (OT) / Cyber-Physical Security
  - C++ back-end development for network monitoring and deep packet inspection
  - Implement proprietary algorithms.
  - Python back-end development for implementing consumer ready application for OT.
  - Front end development and integration.

**Georgia Institute of Technology** Graduate Assistant January 2018-September 2019

- Software development for Industrial Network Cybersecurity.
    - Deep packet inspection from the link layer up to the industrial application layer.
    - Implemented backbone of security software.
    - Developed initial frontend dashboard.

**Georgia Tech & KSU** Research Associate May 2013-December 2016

- Led development of software and hardware implementation for GDOT Research Project No. 13-02 in Vehicular Ad-Hoc Networks (VANETS)
- Directly contributed to research findings and publications.
- Mentored new undergraduate members in developing skills necessary for research.
- Developed network algorithms, simulations, and applications for Dedicated Short Range Communication testbed experiments.

**World Technology, LLC** Partner December 2004-Present

- Built customer and vendor relationships increasing sales growth by over 40%
- Provide IT consulting to both domestic and international customers
- Manage company with 600K+ gross annual sales budgets for operation and maintenance expenditures
- Develop Standard Operating Procedures to ensure consistency of work output
- Design and develop rapid prototype devices for medical, home automation, IoT, and more

## SKILLS

- Network protocol implementation (hardware & simulation), Technical writing (Paper / Journal Publications), Software Development (C/C++, Python, PHP, CSS, HTML, SQL, InfluxDB, Latex, Java, Ladder Logic/STL (PLC)), CAD Design, Robot Operating System (ROS), Network Simulator v2/3 (NS2/NS3), Control Systems, Automation, Biomaterial Understanding, Biomedical Understanding – Tissue Engineering, Micro Electro Mechanical Systems (MEMS), System Design, Electronic Device Development (Filter, Signal Generators, Modulation)

## MENTORSHIP & LEADERSHIP

- Mentored and tutored undergraduate & graduate students in preparation for success during graduate studies. Provided employee, customer training seminars as well as conference paper presentations. Finance management (oversee sales, expenditures, and investments). Time management to ensure prompt project completions.

## PUBLICATIONS

- R. C. Voicu, H. I. Abbasi, H. Fang, B. Kihei, J. A. Copeland, and Y. Chang, “Fast and reliable broadcasting in vanets using snr with ack decoupling,” in 2014 IEEE International Conference on Communications (ICC). IEEE, 2014, p.574–579.

- H. I. Abbasi, R. C. Voicu, J. A. Copeland, and Y. Chang, “Performance optimization of a contention based broadcasting algorithm in vanets,” in 2015 IEEE Global Communications Conference (GLOBECOM). IEEE, 2015, pp. 1–7.
- H. I. Abbasi, R. C. Voicu, J. A. Copeland, and Y. Chang, “Cooperative bsm architecture to improve transportation safety in vanets,” International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE, 2017, p.1016–1022.
- R. C. Voicu, J. A. Copeland, and Y. Chang, “Multiple path infrastructure-less networks a cooperative approach,” in 2019 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2019, p.835–841.
- K. Li, R. C. Voicu, S. S. Kanhere, W. Ni, and E. Tovar, “Energy efficient legitimate wireless surveillance of uav communications,” IEEE Transactions on Vehicular Technology, vol. 68, no. 3, p. 2283–2293, 2019.
- R. C. Voicu and Y. Chang, “Network packetization in multi-path environments next-gen networks,” in 2019 15th International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE, June 2019
- H. I. Abbasi, R. C. Voicu, J. A. Copeland, and Y. Chang, “Towards fast and reliable multi-hop routing in vanets,” in IEEE Transactions on Mobile Computing, Pending Second Review Round, 2019
- R. C. Voicu, Y. Chang, “Stages of CoopNet: A Multipath Parallel Link Architecture For Next-Gen Networks,” In 2021 International Wireless Communications and Mobile Computing (IWCMC) (pp. 592-597). IEEE. (2021, June).
- A. Clark, D. Nguyen, L. Patel, R. C. Voicu, Y. Chang, C. Ham “Wheelchair Conversion & Automation,” In 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET) IEEE. 2021.
- R. C. Voicu, Y. Chang, “Optimizing Multipath Parallel Communication Through CoopNet,” in 2022 IEEE International Conference on Communications (ICC). IEEE (Pending Review)