Multi-Path Communication for Cloud Security

Diogo Miguel Marcos Raposo
diogo.m.m.raposo@tecnico.ulisboa.pt

Instituto Superior Técnico

Advisors: Professor Miguel Nuno Dias Alves Pupo Correia
Professor Miguel Filipe Leitão Pardal

Abstract. This project aims at providing complete solutions to address private communication from client to cloud, i.e., to transfer data over channels without exposing its content. The idea is that cloud infrastructures, despite all the advantages, raise privacy and data integrity issues that are not satisfactorily addressed by existing technologies. Revelations made by former NSA contractor Edward Snowden show that Internet traffic is not routed through the physically most direct path, but through the path considered the "cheapest", suggesting that this may be used to intercept on traffic. To address these issues, this project aims to provide a mechanism to improve confidentiality in communications by splitting packet flows (each flow) in different physical paths. This makes snooping attacks less effective. The document will consider the provisioning of multi-path TCP through multi-path routes. In practice, multi-path communication requires making IP packets follow different routes, which is not possible at network level using the standard Internet routing. Mechanisms will be provided to select paths based on *diversity*, measured in amount of physical paths used, and *trust*, measured in geographic location and amount of packets dropped.

Keywords:
Multi-path Routing, Cloud Security, Communication Confidentiality, Eavesdropping

Table of Contents

1 Introduction 2
1.1 Overview 2
1.2 Objectives 3
2 Related Work 4
2.1 Multi-path Communication for Path Diversity 4
2.2 Achieving Path Diversity 5
2.2.1 Multi-homing 5
2.2.2 Overlay Routing 7
2.2.3 Multi-path TCP 10
2.3 Multi-path Routing for Security 12
2.3.1 Wireless Mesh Networks 13
2.3.2 Internet 16
2.3.3 Network Coding 17
2.3.4 Secret Sharing 19
3 Architecture 20
3.1 Component Structure 20
3.2 Behavioural Structure 21
4 Evaluation 24
5 Schedule for Future Work 25
6 Conclusions 25
References 26

1 Introduction

1.1 Overview

Cloud systems are increasingly used for data sharing and storing between devices with connection to the Internet, whether it is for personal or business use. They offer a reliable data storage alternative with the advantages of removing the concern of hardware maintenance and treatment and having the information accessible in any device that is connected to the Internet.

However, placing information on the Internet, whether it is in the cloud or in any other type of web applications, has the disadvantage of making it vulnerable to eavesdropping by unauthorized third parties. In the case of entities where confidentiality is critical (government, military, medical, etc.) this problem is decisive.

In general, implementing cryptographic systems provides the security needed to keep information confidential, but it is possible that certain devices are unable to implement such systems. This inability is a problem that is already addressed in wireless mesh networks, since this type of networks is built upon devices that are not capable of implementing any kind of cryptographic systems [40]. Recent facts also show that is possible to break these systems in certain conditions. As an example, Adrian *et al.* [1] investigated a new flaw in the Diffie-Hellman key exchange that manages to downgrade a TLS connection for a specified 512-bit group. The authors also explain how a nation-state has the computational power to attack 1024-bit groups, which would allow decryption of many channels that implement this method.

Another problem faced by implementing *multi-path routing* is the possibility of having a conditioned number of channels where to distribute the information due to path failures. For example, for a network composed by ten possible paths from a certain source to destination, only two paths are available to send the data. These path failures can be caused by different reasons, but, in this case, it is important to study the denial of service or selective forward as the main malicious cause for these failures.

Summing up, two problems are addressed. The first is to achieve confidentiality assuming no cryptographic systems are used by splitting the information among channels. The second is to defend against the attacks that aim to force unwanted paths selection.

This research will, therefore, focus on providing an additional layer of security by distributing the communication through several channels. In order to do so there are *four mechanisms* that should be implemented in the system. The communication starts with a *path discovery*, where network nodes (routers or other devices with routing capabilities) are discovered, i.e., the source acquires information on which nodes it can send the information through. This discovery will be based on an upper and lower bound of geographic location. Using a topology aware and trust based decision algorithm, several nodes from different Internet Service Providers - This approach is called *1.Multi-homing* - are chosen, according to their location and trust level. The trust level is calculated according

to the amount of packets dropped and response time to the source in the discovery phase. This selection and the number of nodes used will be based in the amount of physical paths available and the amount of information to be sent. Each node will create a single-hop overlay link to the destination, generating an *2.Overlay Network*. Posteriorly the packets are split among the different overlay links that were previously created and sent to the destination using Multi-path TCP - *3.MPTCP*. This protocol is an extension of the generic TCP implementation that has the ability to distribute and send data between different interfaces or IP-addresses. Therefore, it will be used to distribute the data among all links in the overlay network. In the case when not enough channels are acquired to provide eavesdropping resilience, *4.Network Coding* is applied to the packets in order to provide more resistance to these attacks.

After describing the goals in Section 1.2, this report is organized as follows. Section 2 presents an overview of solutions related to this work and methodologies that help achieving these goals. Section 3 presents the proposal for the architecture of the system. Section 4 presents the methodology to evaluate the proposed architecture. Section 5 presents the schedule for future work and Section 6 presents the conclusions of this research.

1.2 Objectives

This research aims to address the problem of providing a new level of security to communications, assuming cryptographic mechanisms are not enough to defend against eavesdropping attacks. The goal is to provide a system where all data is transferred among several paths over the Internet without requiring the need for changes in its structure. The requirements of this system are:

- The path selection should be the most physically disjoint possible;
- The path selection should be randomized according to lower and upper bounds of geographic location;
- The framework should resort to network coding when there are not enough paths to use;
- The security of communication should always overcome the performance, however the minimum performance must be the acceptable.

This approach will use the MPTCP protocol in order to distribute the packets among channels, therefore it is assumed there will be no problems with packet bursts and reordering in case of unordered delivery. It also assumes that the network nodes chosen in the path selection are able to redirect the packets to the destination. The result will be the implementation and testing of a functional framework that follows the above mentioned requirements and should provide the expected level of security.

2 Related Work

When talking about path diversity, the goal that the majority of the studies tend to achieve is resilience and performance, either on the Internet itself or on Wireless Sensor Networks (WSN). However, it is important to notice that multi-path routing brings advantages in the security of communications. Some studies have been made in the sense of implementing diversity of channels with emphasis on security. This section addresses all these approaches that aim to achieve resilience, performance and security when using several channels to communicate. Section 2.1 defines what is path diversity and how it can be measured. Section 2.2 presents different methods to achieve path diversity: multi-homing, overlay routing and Multi-path TCP. Section 2.3 provides a better insight of studies which aimed to use multi-path routing for security and states the new challenges that this method brings.

2.1 Multi-path Communication for Path Diversity

Multi-path communication is the ability to divide a certain amount of data to be sent over a network and split it across two or more channels, spreading on the source and sinking on the destination. The most generic motivations to study it are the performance and robustness benefits. Selecting multiple paths for reaching one destination and using an intelligent balancing algorithm is how a connection can survive individual path failures and maximize the use of bandwidth in each channel [43]. For example, VoIP can use channels that minimize delay and file sharing applications can use channels with high throughput. However, achieving a wide network multi-path routing faces two main challenges: the data storage and computational overhead increment and the difficulty of controlling routing in private networks which this network can greatly depend on [7].

Path diversity is the quality that defines and measures the use of multiple routes to reach a destination, whether they are physical or virtual. When there is path diversity, instead of using a single path from a source to a destination, it is possible to split the traffic among several paths and nodes before reaching the destination. In other words, it consists on applying multi-path routing to communications.

It is important to measure path diversity according to the amount of physical paths' used, i.e., the level of *physical layer disjoint topology* [43]. For example, if only a single physical path is used, but it contains two or more virtual paths, in the case where the physical path is being eavesdropped, no security is added to the communication. In general, maximizing the number of physical paths according to the amount of virtual paths used will lead to a more efficient and secure communication. In fact, in a good scenario, dividing the traffic over two or three extra physical paths is enough for reacting to path failures, balancing the communication load and improving security.

To avoid congested paths, router failures and adversary packet dropping attacks, it is necessary to have a system that can quickly switch paths and distribute information. Several techniques are presented [22]:

- *Round-robin*: Switches the traffic according to the granularity of files. It can be very accurate to distribute small files. However if the paths used have different delays between the source and destination, some TCP packets that belong to the same flow will arrive out of order. TCP protocol will, therefore, consider that the network is congested, adding extra overhead to the communication.
- *Hashing*: Starts by dividing the amount of information in partitions. Each partition has a size directly related to each path's throughput and the packets are distributed corresponding paths. It is good for packet arrival ordering, however, it fails on correctly splitting the packets according to channels throughput.
- *Flow cache*: A flow is defined by having a source address and port, destination address and port and transport protocol in the header. It manages the transfer of different sizes of data. This technique keeps track of which path each flow is sent on making it a good choice for packet ordering. It provides a system to dynamically allocate a new flow to each path leading to a better control on the path's congestion balance.
- *Flowlet cache*: Similar system to Flow cache, this technique also provides a mechanism that divides the traffic according to packet-burst granularity instead of single packets, meaning less memory is used for computations in comparison to Flow cache.

Path failure correlations can also cause many paths to fail, emphasizing the need to have intelligent path selection (focusing on choosing disjoint paths). In the case of not choosing completely physically disjoint paths, congestion or failure of a physical path or router will highly affect the availability and performance of connections [18].

Several approaches have been studied in order to introduce diversity in communications. Even though it is highly believed that most approaches, such as overlay routing, multi-homing and multi-path TCP, provide significant benefits, their effectiveness depends on the level of path diversity between two end-points at the physical layer. Han *et al.* [18] present a study proving that in many cases implementing either multi-homing or overlay networks can have a bad outcome in communications. Due to the lack of physically disjoint paths in some cases, it is possible that implementing path diversity might bring performance inefficiency.

It is very important for this project to measure path diversity in the communication. It will not only avoid several resilience problems, but also provide a more versatile path selection mechanism.

These approaches and others are explained in the next section, addressing the issues that arise when trying to achieve adequate levels of path diversity.

2.2 Achieving Path Diversity

2.2.1 Multi-homing

A method of achieving path diversity is to use *multi-homing*. This concept is simply defined as having a customer network linked to two or more Internet

Service Providers (ISP) as shown in Fig. 1. Resilience and performance are the main advantages of this method. Different providers offer different performance levels to different parts of the network, so choosing the "right" provider will result in a performance increase by itself [2, 3]. Akella *et al.* [3] evaluate the use of multi-homing, using HRTT (Hand-shake Round Trip Time) as a measurement unit for data centers and enterprises. These studies [2] conclude that by simply using two providers the performance is increased by at least 25% and that the improvements are very small beyond four providers. The same authors in 2008 [2] did a similar but deeper k-homing study, measuring the performance based on the RTT and throughput of small (10 KB) and medium sized (1 MB) file transfers. In general, the same conclusion is reached in all multi-homing studies: the use of two or three providers increases the performance in 15-25%.

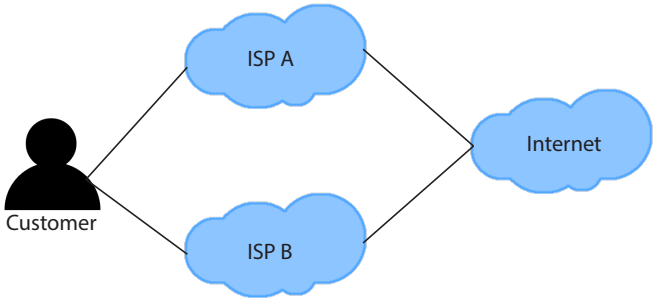

Fig. 1. Example of a multi-homing connection.

The Stream Control Transmission Protocol (SCTP) is an example of a transport protocol that natively supports multi-homing. This protocol with the addition of the Concurrent Multipath Transfer (CMT) [32] extension allows not only to connect to different destination addresses, but also through and to different interfaces. Another option to implement multi-homing is the MONET web proxy system (Multi-homed Overlay Network) [5] that aims to improve the client access availability specifically on websites. In order to do so, each address bound by the DNS name in a replicated site is related to a different server machine or Internet path. According to the evaluation made on [5] this approach has proven to be very good to mask and recover from path-failures comparing to the traditional DNS queries with low time to live (TTL) that force the client to refresh the mapping.

In theory, this methodology is full of advantages and presents a very good choice to achieve path diversity. However, multi-homing might not always provide the desired level of diversity [19, 21]. In fact, based on the test made in [19], 80% of 80,000 connections to the same destination using different providers

overlap. This happens due to the lack of control from each hop to the next one and the architecture of the Internet's core, to which no solution has been found yet, unless there is massive Internet restructuring.

2.2.2 Overlay Routing

Overlay routing is a mechanism that allows the creation of a virtual network on top of an already existing network infrastructure without modifying it. An overlay link is created between two nodes making them virtually connected, independently of the physical paths that it is crossing in the underlying network. For example, a peer-to-peer network is an overlay network, since each peer connects to another peer using the Internet. A level of abstraction is added to the underlying physical topology. At the physical level the packets travel in the actual physical structures that contain this virtual network. An indirect virtual path is composed by several physical devices, however the abstraction created by the overlay network allows it to be treated as an end-to-end link.

Resilience is the major advantage brought by this ability of sending packets through indirect virtual paths. It allows a fast path failure recovery and congestion control [4]. An example is shown below in Fig. 2.

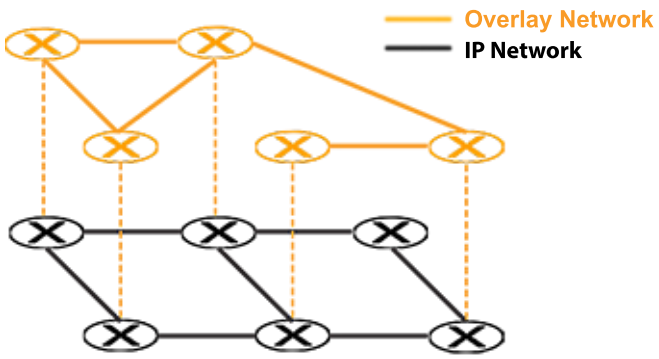

Fig. 2. Example of an overlay network.

Several proposals exist for overlay routing mechanisms that aim to maximize the resilience of communications. Andersen *et al.* present RON (Resilient Overlay Network) [6], an application layer overlay network on top of the Internet. It has many nodes deployed over the Internet that form an application-layer overlay network. Each node periodically probes its links to other nodes so that it can quickly detect path failures and monitor the quality of the connection. The main advantage of this architecture is the ability to recover from path failures within seconds, which is a major benefit comparing to current wide area routing

protocols like Border Gateway Protocol. The fact that RON is small sized compared to the Internet (a net constituted by 50 or 60 nodes is usually appropriate) allows the protocol to keep information of latency, packet loss and throughput of a link (this might be a disadvantage when trying to scale to a large network, since there is a lot of information to be kept). The architecture proved to be able to recover from more than 60% of the outages in an average time of 18 seconds.

Amir and Danilov [4] propose a reliable hop-by-hop overlay that recovers from losses only on the overlay hop that caused the outage, enabling a fast recovery. In order to do so each link must have its own set of acknowledgements of packets so that it is possible to know exactly where the packet failed to send. Moreover, in order to increase the congestion window and deal with packet bursts, the nodes also implement buffers. Processing in intermediate nodes is required to deploy this protocol. In the authors experimental evaluation this solution presented jitter and delay times of about 3 or 4 times smaller than the end-to-end TCP connection for the same scenario.

Another proposal is the one presented by Snoeren *et al.* [39] which is a mesh-based content routing using XML. Its architecture is based on an overlay network that transports XML streams. Each client or device that joins the network sends a query informing which types of packets they are available to send and the overlay network re-arranges itself to deliver the desired packets. The advantages of this method are:

- It allows the network to correctly interpret client data;
- The protocol allows the calculation of which packets are going to be processed together, aiding the network scheduling;
- XML is a easy language to parse and many tools exist for it, making it easy to receive and send queries among clients or devices.

It should also be noticed that adding extra XML information requires more information to be sent. Other than using XML in router's core, this architecture also implements a diversity control protocol (DCP). In this protocol, the diversity is achieved by allowing the receiver to reassemble packet streams from several senders. The same stream of packets is sent from different sources with a certain level of redundancy, however, instead of re-transmitting all the received packets it will only send each packet of that stream once, the first to arrive with no errors. It provides a resilient mesh communication that allows a receiver to reassemble a packet stream coming from different sources. Again the evaluation made presents a throughput increase of about 5 times comparing to an end-to-end TCP. No fault recovery evaluation was performed.

Overlay routing is one of the main approaches studied to increase path diversity in multi-path routing, however, later studies prove that many path outages are unavoidable when using the protocols shown before and others that do not select overlay nodes considering the underlying topology [19–21]. Therefore it is necessary to study how is it possible to maximize the path diversity when using overlay routing.

2.2.2.1 Topology-Aware Overlay Routing

Topology-Aware Overlay Routing is an approach that aims to select the overlay links and nodes based on their topological disjointness in order to maximize the path diversity. Han *et al.* [20] provide a topology-aware framework for overlay networks that aims to maximize physical independence without demoting performance. The key aspect is to select a subset of routers and ISPs that present a good level of topological diversity and performance. In order to so, it is necessary to measure the routers regarding two metrics from a set of sources to destinations:

- *Router path diversity*: measured based on the amount of shared routers between overlay paths and a physical direct paths;
- *Router latency*: defined by the difference of RTT using the overlay layer and using the physical most direct path.

The nodes between a source and destination node can be set either statically or dynamically in order to get an extensive evaluation. Choosing routers dynamically (i.e., selecting the optimal overlay node in terms of diversity and latency according to the pair of source and destination being tested) has shown to provide high level of diversity and provides better latency than direct physical paths.

Once all routers are rated by its neighbours it is necessary to know which ISPs contain routers that are worth using, but the comparison is not straightforward. From each ISP, the k most diverse routers are chosen and each ISP is evaluated for k nodes (where k varies from one to ten), according to the evaluation methodology stated above. In general, studying each ISP for each possibility of k provided the same results for the different ISPs. After the evaluation for a single ISP is made, there is a need to study the path diversity gain from increasing the number of ISPs. For example, for two ISPs, all the possible combinations of a pair of ISPs should be studied. Based on the authors experiments, the most significant gain in what matters to physical diversity is changing from one to two ISPs. On the other hand, regarding latency, using only one ISP proved to perform as well as deploying the network in all ten ISPs used in the experiment.

The authors final evaluation shows that it is possible to recover up to 87% of path outages with the correct node selection which is a better value comparing to the ones obtained before.

2.2.2.2 Single-hop vs. Multi-hop Overlay Routing

Another decision to have into account is the amount of hops used in each path, i.e., single-hop vs. multi-hop paths. RON is an example of a protocol that uses a complex multi-hop system, yet the experimental studies made in [6] showed that most outages were avoided by using a single node. For the topological aware framework, the authors also studied the performance achieved on a direct physical path, a single-hop path and a multi-hop path [20]. The metric used for comparison is the RTT, which in a direct physical path is easy to acquire. For a single-hop it needed to add the RTT from the source to the chosen hop and that same hop to the destination. In the last case, where multiple hops

are used, the best scenario is assumed when: the source chooses the best entry node with least overlapping routers; the best "exit" node to destination is also always selected; the path between the entry and exit nodes do not present any overlapping. According to the results presented, single-hop overlay paths suffice in what matters to path diversity and latency [20].

Gummadi *et al.* [17] also present a single-hop source routing (SOSR) that aims to recover form path failures using a single hop from the source. The hops are chosen based on a random-4 policy, which chooses four random hops next to the source and only then redirects the packets through an indirect path. In order to arrive to the number 4 for the policy, the authors made a random-k study and the results revealed that this is the best value regarding over-head and amount of path failures recovery. This protocol is able to recover from 56% of network failures while adding low overhead and providing good scalability.

Table 1 shows an analysis of overlay routing protocols in terms of physical topology and fault-recovery.

Overlay Routing Protocols	Topologically disjoint	Fault-recovery
Mesh-based XML [39]	✗	-
RON [6]	✗	60%
SOSR [17]	✗	56%
Topological Aware Framework [20]	✓	87%

Table 1. Comparison of overlay routing protocols.

2.2.3 Multi-path TCP

A third approach to achieve path redundancy and diversity is the use of *Multi-path TCP* (MPTCP) which is an extension of the TCP protocol that enables the use of simultaneous IP addresses and interfaces when communicating between end points [41, 42]. The protocol discovers which paths are available to use, establishes a connection and splits the traffic among them. It presents the same programming interface as TCP, however the data is spread across several subflows. The option field in the regular TCP protocol is filled with MPTCP data structures in order to keep track of all packets status (sent/failed) through the different subflows. Since this protocol is implemented over multi-homing or overlay networks, the main advantage is, as mentioned before, the resilience on communication.

Path Discovery: In order to implement MPTCP, it is important to maximize the throughput to the sum of all available channels. Therefore, the path discovery algorithm must not only choose the paths with most efficiency and lower load, but also maximize the number of paths used, i.e, not as many as possible, but

the number of paths that achieve the highest performance of the data transfer. Several algorithms were implemented [30] that have shown to achieve the same amount of distinct routes for each one:

– *Flood Method*: Since routers can only have knowledge about their direct neighbours, the method floods the neighbourhood information from the emission node to the destination node with the aid of Discovery Packets. These packets contain the Flood Time To Live (so that the propagation stops in case it can not reach the destination), a list of addresses already visited (so that it does not repeat and a weight of the route). Based on that weight and number of hops visited, the routes are chosen on the destination node.

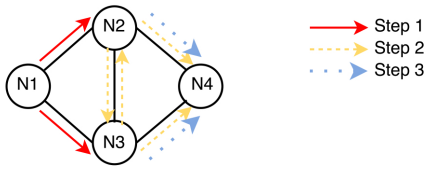

Fig. 3. Example of path discovery from N1 to N4 using the flood method.

The example in Fig. 3 shows how the paths are discovered, from N1 to N4 using flood method:

1. N1 floods his neighbors and generates paths {N1, N2} and {N1, N3};
2. From {N1, N2}, N2 floods his neighborhood and generates {N1, N2, N3} and {N1, N2, N4}. From {N1, N3}, N3 floods his neighborhood and generates {N1, N3, N4} and {N1, N3, N2}. This step already provides two possible paths from N1 to N4;
3. From {N1, N2, N3}, N3 floods his neighborhood and generates {N1, N2, N3, N4}, since N2 is already in the list, the packet reaches N2 but it is dropped. The same happens in the last incomplete path generating {N1, N3, N2, N4}. The method is complete and four possible paths were generated.

– *Modified version of k-Dijkstra's Algorithm*: It is based on the original shortest path Dijkstra's algorithm [12], but the initial weights are the real topology. Other than changing the links weight as the original algorithm does, a constant that represents the maximum weight between two nodes is added to the links. With this feature, the probability of having a link appearing twice in the calculation of new shortest paths is low.

– *Modified version of Bellman-Ford's Algorithm*: The Bellman-Ford's algorithm [9] is applied to all direct neighbours of the emission node where the destination is the destination node. The algorithm is also applied to all

direct neighbours of the destination node where the destination is the emission node. Posteriorly, all routes are merged generating new ones and the shortest path is calculated from those.

Congestion Control: The implementation of this protocol also comes with the need of having a good congestion control algorithm. In fact, it is only worth using MPTCP if a good congestion control system is used [42]. In general, the key to implement a good congestion control scheme is to calculate a congestion control window. For example, one solution is to have a connection with several sub-flows where each sub-flow has an algorithm that calculates the congestion window based on the round-trip time (RTT) of that same sub-flow. Other deeper mathematical solutions have also been studied for the congestion window calculation, such as the one presented in [18] that studies the stability of the congestion control loop based on the Nyquist Condition.

The mechanisms mentioned above and the algorithms used allow the users to use the best paths available and are optimal in static settings. Though they might have some problems, such as the failure of detecting free capacity due to not probing sufficiently paths with high loss probability, randomly flipping the packets through channels if there are multiple good channels, users with MPTCP will reduce the throughput of others with regular TCP, etc. These cases have already been studied and Khalili *et al.* [24] presents an algorithm that solves them, by correctly dividing the throughput between clients with MPTCP and TCP and making the RTT a characteristic with more importance on the channel decision.

All three mechanisms presented in this section are related. Multi-homing provides a greater amount of nodes to select when creating an overlay network, therefore allowing to have a greater level of path diversity. Once the overlay network is created, MPTCP is applied between the end points to distribute the data between all the overlay links. In the architecture proposed in this document all three mechanisms are used. It is also important to notice that the overlay network will be topologically aware and will also consist on several single overlay hop links, i.e., when creating an overlay link it is only necessary that the communication goes through one overlay node (independently of other underlying physical nodes).

2.3 Multi-path Routing for Security

So far the implementation of path diversity has been studied with the objective of increasing the availability, resilience and reliability of networks. However, the aim of our research is the use of several paths in order to increase the security of connections. Assuming the attacker manages to spy on a single end-to-end channel, the communication is totally vulnerable to eavesdropping among many other types of attack. Using multiple channels jeopardizes only part of the information. Studying multi-path routing protocols with emphasis on security might be the key to achieve higher levels of security in the Internet [26]. For critical

infrastructures (military, health care, government, etc.) a trade off between performance and security must be taken into account where the second might need to be privileged [40].

Several protocols presented below apply multi-path routing with a focus on security. Most of them accomplish the objective based on node selection according to their rate on a certain characteristic. In general, each node has a different trust level and decides which neighbour is the next hop based on their trust level. This depends on different characteristics, such as geographic location, packets forwarded, response time, among others. There are other characteristics that might condition the next-hop decision, however, trust is the most used conditioned so far. It is important to notice that most of the studies were made for wireless networks that can not handle other security measures. The capability of having, for example, a node with a geographic locator device is possible in wireless sensor nodes but can be difficult when using wired devices. These situations have not yet been given enough attention in wired networks due to the existence of cryptography mechanisms in most WAN devices.

Since wireless sensors are unable to implement these mechanisms, it is necessary to find different methods to obtain security, namely multi-path communication.

Security is now the goal of implementing multi-path routing and trust is the metric used in nodes selection. As mentioned above, it is important to measure trust. Different authors present several options to measure it for each node of the network.

The multi-path routing protocols can be applied in different networks. The sections below present different protocols for Wireless Mesh Networks or for the Internet, Section 2.3.1 and Section 2.3.2 respectively. In addition, Section 2.3.3 and Section 2.3.4 present Network Coding and Secret Sharing respectively, providing an insight of how they can help achieving eavesdropping protection and other security measures.

2.3.1 Wireless Mesh Networks

Wireless Mesh Networks is the most studied type of networks when implementing multi-path routing, since the wireless device susually lack the capability of implementing cryptographic mechanisms [40]. The generic methodology to achieve multi-path communication in this type of networks is to assign a certain characteristic to the nodes and classify them accordingly. Jiang *et al.* [23], for example, classifies them according to their capability. This characteristic rates the probability of a node to create a multi-hop communication that avoids any unnecessary delay.

Since the goal is security and not performance, trust is a more adequate characteristic. This way, several protocols have been studied to rate the trust of each node, such as the ones mentioned above. Besides these, there are several others such as a trust model where the trust is given by the neighbour nodes or other embracing entities.

An option presented by Bao *et al.* [8] is a hierarchical solution for wireless sensor networks, composed by two levels of trust storage, a sensor node level and a cluster head level. Each sensor evaluates all his neighbours and each cluster head evaluates all sensors in its cluster based in four components: intimacy, honesty, energy and selfishness. Intimacy is calculated based on interaction history; honesty refers to the belief of a node in another; energy indicates the percentage of the node's remaining power; selfishness is acquired by overhearing and snooping neighbour nodes. Although the authors provide a quite complete solution in terms of security, if a node gets compromised, its snooping capabilities on other nodes might create a chain of compromised nodes.

Another option is using geographic routing (GR) [27] in wireless sensor networks with a trust-based decision algorithm. Each node contains a mechanism that indicates its geographic location and the decision for the next-hop is based on both the neighbours proximity to the destination and the amount of packets successfully forwarded. Nodes that drop more packets will be punished and if in any case a node lies about its location it will be immediately deleted from the forwarding set. The punishment calculation system must be correctly implemented in order to achieve a high delivery rate. Otherwise unintentional packet drops might be punished leading to the same delivery rate as if no trust mechanism was used. The trust-based mechanism allows the communication to defend itself from "black hole"[1] and "selective forward"[2] attacks, which only increases the integrity and availability. Calculating the trust based on the chance of having an eavesdropping attack is another possible approach to achieve the highest level of security possible with an acceptable performance. Although a great part of the protocol is based on a physical mechanism, the logical connection of nodes is still arranged in the data link layer. This protocol is the most noteworthy, because it is the one that is most topologically aware.

Other methodologies are possible that do not involve the node's trust level. An example is hybrid routing where proactive routing mechanisms are used in some areas and reactive routing is used in the rest of the network. In proactive routing, routes are computed automatically and independent of packet arrivals. In reactive routing, the routes are discovered on demand when packets arrive to a node. The performance given by this hybrid method allows the use of lightweight cryptography mechanisms for a security compensation [38]. Lightweight cryptography is a method that trades security for computational power, i.e., it aims to reach high levels of security using small computational power.

Since security of communication is the aim of the project, studying the benefits of implementing multi-path routing with a focus on security is not enough. The implementation of the protocols mentioned before brings new problems that should be addressed in order to keep communications secure. Stavrou *et al.* have

[1] A black hole attack is a type of denial-of-service attack that forces the router to discard all received packets.

[2] A selective forward attack is an attack that compromises a node, making it behave like a normal node, however it selects the packets that are forwarded.

given a better insight of these issues, specifically for wireless sensor networks (WSN) which are defined in three main sectors [40]:

- *Redundant routing*: with redundant messages circulating over the network, more opportunities are given for an attacker to intercept them.
- *Routing attacks*: several attacks can be executed to condition the discovery of paths, such as a "denial of service" or "selective forwarding" attack.
- *Survivability*: extending the lifetime of a wireless sensor in terms of energy consumption is something to worry about.

This provides a good idea of which problems come up with multi-path routing and how they affect the main security requirements (confidentiality, integrity, reliability). However this might not be the case in wider networks. The lifetime of a sensor has also revealed to be a smaller problem since the authors studied it. New technologies have appeared to increase the battery of a sensor allowing them to have more processing power, although not enough to provide adequate cryptography systems. Lightweight systems might still be implemented as mentioned in previously in this section.

Depending on which protocol is used to provide multi-path routing, these issues become more or less relevant. For example, the hierarchical trust-based routing protocol with geographic routing proposed by Bao *et al.* [8] protects from the compromised and selfish nodes that generate the routing attacks. The next hop is decided based on the node's trust level and proximity to the destination, instead of regular geographic routing that uses only the second option.

A good option to fill all the security requirements is SEIF protocol [33], which prevents attacks that affect confidentiality and integrity. This protocol uses a sub-branching system for path discovery that allows the authentication of sub-branches and a freshness system in order to avoid replay attacks on messages. It also provides a good level of reliability due to lightweight computations. H-SPREAD is another solution presented by Lou *et al.* [28], that calculates the probability of a message being compromised based on the paths that were compromised. A path is considered compromised if any node in the path is considered compromised. A computationally more heavy protocol that still provides good security but lacks on reliability is INSENS [13]. It is based on three main principles: exploit redundancy to tolerate intrusions (without detecting compromised nodes), perform heavy computations at base stations and limit the scope of damage by limit flooding.

Table 2 shows these protocol evaluations made by Stavrou and Pitsillides [40].

It is important to note that all these studies consider wireless sensor networks, since these devices have low encryption capability. This might not be the case in networks using more capable devices, however, it is still important to study these protocols, due to the fact that they provide the needed security when lacking strong cryptographic mechanisms.

Protocols/ Security Requirements	Authentication	Integrity	Confidentiality	Reliability
SEIF[33]	✓	✓	✓	Good
H-SPREAD[28]	✗	✓	✓	Good
INSENS[13]	✓	✓	✓	Limited

Table 2. Analysis of WSN multi-path protocols based on their security requirements, adapted from [40].

2.3.2 Internet

When trying to implement multi-path routing protocols on the Internet it is important to notice that the protocols should adapt to the structure of the Internet itself other than creating massive changes in it.

Lee *et al.* [26] provide two distributed algorithms (in the sense that the routing decisions are not centralized) that aim to defend the communications against intruders from jeopardizing single-path communications. The Bound-Control algorithm aims to defend against attacks on a single link and the Lex-Control algorithm aims to defend from multi-link attacks. In order to do so, both protocols apply the minimax algorithm - an algorithm used to minimize the possible loss for a worst case scenario - to achieve the optimal value for an attack damage variable that is defined by the attack cost divided by a security constant. The second algorithm applies the first one iteratively, to identify several critical links.

Narula *et al.* [31] also present a specific protocol for mobile ad-hoc networks that uses a trust-based routing mechanism together with an encryption system where the files are used as keys. This protocol aims to address the issues of access control, confidentiality and integrity by using encryption systems and by calculating the trust level of each node. For example on their computation power (a more powerful machine is more capable of cracking the encryption of a file, since it has higher processing power to execute brute force or dictionary attacks). This method adds a good level of security to the communication, but it brings new issues. The key exchange system (centralized system) used in it is not the most appropriate, the keys might not reach every node, or the handshake might be captured by an untrusted third party. This protocol also allows to measure the trust of all nodes based on direct interaction and recommendations of its neighbours. If an acknowledge of a packet delivery is promptly received from a neighbour then it is assumed the node is not involved in a brute force or a denial of service attack. In general, ad-hoc networks are connected to the Internet, therefore it is important to study their implementation.

Other than studying examples of protocols that can be applied on the Internet, there are other methodologies that are worth noticing. Onion Routing is one of those methodologies that aims to provide security when communicating over insecure networks.

Onion Routing is an overlay network technique that offers resistance to eavesdropping and traffic analysis by providing anonymous communication. In other

words, it allows to privately communicate in a public network, since the sender remains anonymous. It is implemented in overlay networks since the technique starts by identifying a series of routing nodes which will add the layers of encryption to the message, creating the onion as shown in Fig. 4.

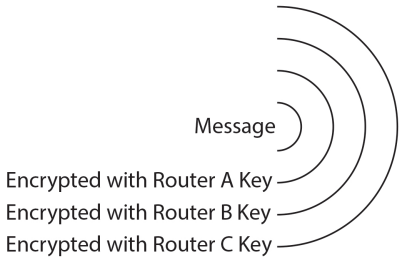

Fig. 4. Example of an onion in onion routing. The message is encrypted using the several onion router keys.

After the data is encrypted according to each nodes' key, it will be decrypted by every node until it reaches the destination in clear. This way, each node only knows the previous and following node, hiding the source of the packet. This overlay network is only implemented in one bi-directional virtual circuit with only two entities, an initiator and a responder [16,36]. This does not provide path diversity or multiple-path routing, although it provides confidentiality which is one of the project's goals. It is currently used by TOR [15], which is a service that works on the Internet and adds some characteristics, such as perfect forward secrecy, congestion control, integrity checking, among others. To do so, each onion router maintains a TLS [14] connection to every other onion routing and each user runs a service that fetches directories, establishes circuits and handles all connections ran by the user. Using Diffie-Hellman, the keys are exchanged with all onion routers that compose the circuit.

2.3.3 Network Coding

Usually, a router only has the options of forwarding the messages. *Network coding* consists in allowing a node to perform some computation and send a different "output". It normally consists on mixing or encoding messages/packets between nodes, for example:

- A node receives two packets, p1 and p2 and sends a packet that results from the addition of p1 to p2.
- Another node that receives p1+p2 and p1 can retrieve p1 and p2 by itself.

The advantages are generally performance related in terms of communication [11]. Maximizing throughput, minimizing delay and improving scalability are some of the advantages, however it is required that the nodes are capable of

such computation power to make network coding worth using. Fig. 5 shows the butterfly network which is a scheme that is usually used to represent linear network coding. This is a type of network coding where the set of nodes used between the source and the destination multiplies a combination of packets received by a certain set of coefficients. These coefficients may work as a secret on the communication, since they add entropy to the output. When the nodes start sinking to the destination, they start recovering the original messages by performing Gaussian elimination. This way, it is possible for the destinations to deduce both packets without receiving them uncoded.

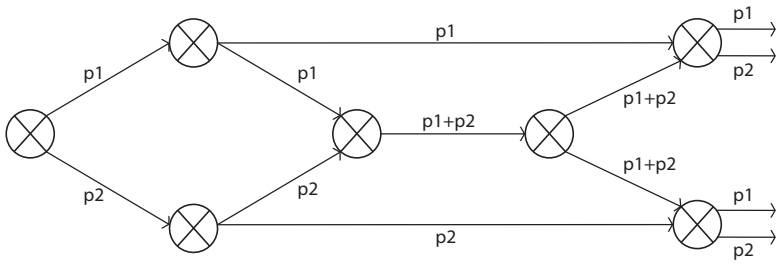

Fig. 5. Considering a link can only send one packet at the time, linear coding allows both destination nodes to receive both p1 and p2. If not network coding was used, the middle link would only send either p1 or p2. This way one of the destination nodes would only receive one of the packets twice.

Matsuda *et al.* [29] provide an insight of the possible network coding applications. Security is among them and it is divided in the protection against two types of attack, "eavesdropping" and "pollution" attacks. The second ones do not affect confidentiality in any matter, therefore, in the scope of this research, applying network coding is only considered to fight eavesdropping. The authors define a network coding scheme as information theoretically secure against eavesdropping if the attack cannot obtain any information from the coded packets. In this sense, Cai and Yeung [10] present a model called wiretap network where it is possible to include symmetric cryptography with network coding that proves to be information theoretically secure. The key to achieve this security is to have only one dealer with full access that shares the secrets with every participant so that only it is able to completely decode the packets. The presented result was completely theoretical.

This methodology is also indirectly protecting multi-path routing from eavesdropping when it is directly affected by routing attacks. A selective forward attack has the objective of forcing the multi-path routing protocol to choose paths

desired by the attacker. However, if the messages sent require network coding, resistance of eavesdropping is increased even if not many paths are used. In the case where an attacker can spy on most of the information, he would have also to break the entropy of the coded packets to get the original packets.

2.3.4 Secret Sharing

In this project, network coding will be used to protect the communication when the path selection is conditioned to a low number of paths. But, it comes to the problem that the coefficients used in it are transmitted in clear. For that reason, secret sharing is studied in this document. Using a function which requires all shares to discover the secret will allow to securely transfer these coefficients that can be considered a secret, though more computation is required.

Secret Sharing [25] is a methodology to distribute a secret among a group of entities, where each one receives a share of that secret. This secret can be reconstructed only when a sufficient number of shares of that secret are retrieved, meaning that individual shares are of no use on their own. Considering n as the total number of secret shares and m as the number of shares necessary to reconstruct the secret, a (n, m)-secret sharing scheme consists of two phases. The first is the distribution process where the secret is split into n shares and privately delivered to the participants. The second phase is the reconstruction process that, once the receiver acquires m shares, it is able to reconstruct that secret.

Shamir [37] presents a secret-sharing scheme that it takes k points to define a polynomial function of degree $k-1$, i.e., considering a function where the secret is divided in five parts, but only three are necessary to recover the secret, it is only necessary to create a parabolic function to recover the secret [37]. The secret will be split into five shares and the information that will be transmitted will be the number of that share and the result of applying that function.

$$f(x) = c_1 + c_2 x + c_3 x^2$$

On the previous function c_1 represents the secret itself and the constants c_2 and c_3 are random. Therefore, the information to be transmitted is $(x, f(x))$ where x starts in one or the secret would be clearly transmitted. For the reconstruction, the Lagrange basis polynomials are computed, obtaining $f(x)$ and, therefore, the secret c_1. This method is used in recent systems. Padilha and Pedone present Belisarius [34], a distributed byzantine fault-tolerant storage system with confidentiality that uses Shamir's secret sharing algorithm to protect the data among the several storages. This system is composed by clients and servers. In short, the secret for the client's data decryption is split among the servers and it has to request the share from the servers. Once it receives the information and the necessary number of shares, it is possible to decrypt the data.

3 Architecture

After the efforts to apply multi-path routing over the Internet mentioned in the previous section, it is possible to realize that none of them completely achieves all the goals of this project by itself. The next subsections presents the components and behaviour of the proposed architecture.

3.1 Component Structure

In order to provide the desired confidentiality, assuming there is no cryptography and the amount of physical paths may be conditioned, the following architecture is proposed at the application level. The components structured is as shown in Fig. 6.

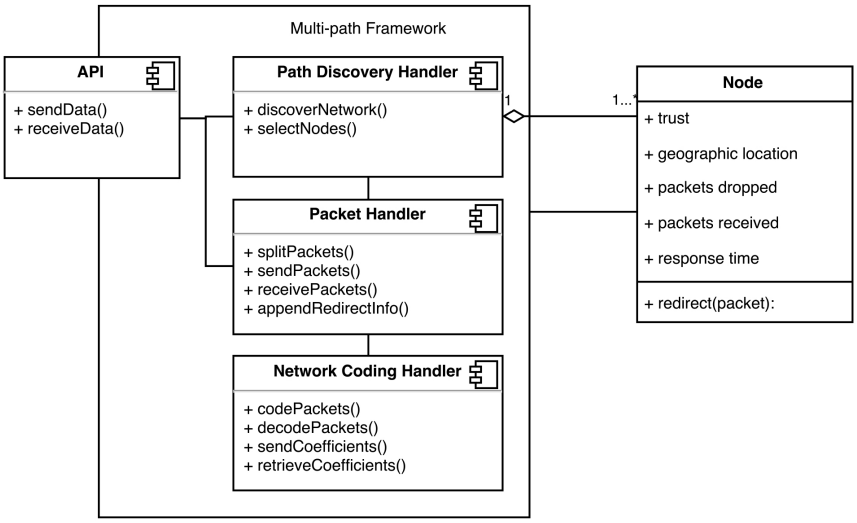

Fig. 6. Components structure of the framework.

- *API*: Connects the framework to the cloud application itself, working as a bridge of data that will trigger the discovery and selection of the overlay network on the part of the Path Discovery Handler;
- *Path Discovery Handler*: Once the API transmits a signal that is ready to send data, this component is in charge of discovering the list of network nodes upon which it is possible to build the overlay network on. Once the list is made it is necessary to rate the trust at each node. According to the previous rate, this component also decides which nodes will support the

overlay links in the network according to the level of path-diversity achieved by a certain selection. This decision is made with the algorithm described in Section 3.2;
- *Packet Handler*: This component is responsible for splitting the data given by the API among the links created by the Path Discovery Handler and, using the MPTCP protocol, to send the data to the nodes previously selected. It must also append the information of the destination so that the nodes know to where they should redirect the packets;
- *Network Coding Handler*: In the case the Path Discovery Handler does not provide enough links, the Packet Handler will trigger this component. Linear network coding is applied to the packets that will be later decoded by the destination. This component is also responsible for generating the packets that contain the coefficients for network coding after Shamir's secret sharing algorithm is applied;
- *Node*: Represents a virtual list of the network nodes information, containing the values for trust, geographic location, recorded packets dropped, recorded packets received and response time since the information was request by the Path Discovery Handler.

3.2 Behavioural Structure

The behaviour of the system consists mainly in two steps when transmitting information between two end-points: the path 'discovery and selection' phase and the 'transmission' phase.

On the first phase, the system will flood the nodes that can compose the network, i.e., the nodes that are able to process packets sent by the source and redirect them to the destination. This methodology is inspired on the flood method present in Section 2.2.3, when describing how MPTCP discovers paths. The packets are sent to neighbours of the source, and these neighbours will posteriorly send to their neighbours and so on, as shown on Fig. 7. The algorithm will only accept, however, the nodes that are over a constant lower bound and upper bound of depth. The lower bound is used to provide the physical disjointness, nodes further away from the source are more likely to be physical disjoint than ones that are closer. The upper bound allows the communication to be kept inside a desired range. For example, if a European company wants to keep its information inside Europe, this is possible due to this upper bound.

The trust-based topology-aware algorithm also has two phases of calculation. One that guarantees a node selection based on their trust and afterwards another that guarantees path-diversity. This algorithm is inspired in most of the protocols presented in Sections 2.3.1 and 2.3.2. On the first phase, each node's trust t is measured according to their geographic location, recorded packets dropped p_d, recorded packets received p_r and response time r_t. If no packets were recorded and hence dropped until the node's discovery, their values will be set to one so that the node is not punished during the selection. Once the information is normalized, it is finally possible to rate the trust. Since the response time can vary due to the distance to the source it will be set to either zero, if it overcomes a

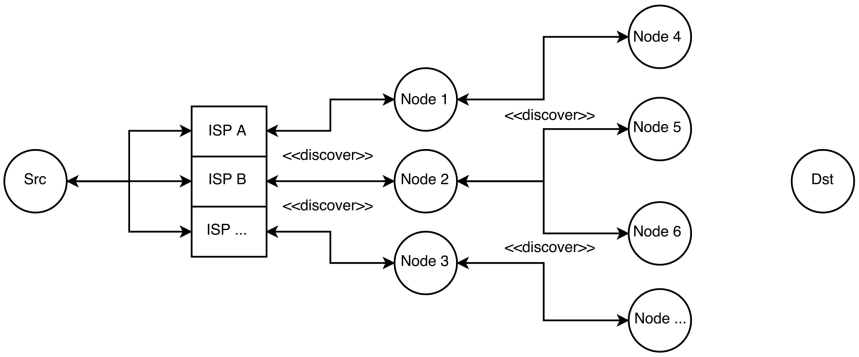

Fig. 7. Example of network flooding to retrieve nodes information.

constant c_t that defines the maximum acceptable RTT, or one, if it is acceptable, as shown in the function below.

$$r_t = \begin{cases} 1, & \text{if } r_t < c_t \\ 0, & \text{if } r_t \geq c_t \end{cases}$$

This constant c_t exists because the objective is not to achieve the highest performance, but to achieve an acceptable one while providing the level of security desired. Afterwards, the algorithm calculates the percentage of packets forwarded on each node so that it is possible to know which nodes should the system rely on. It is finally possible to retrieve the trust level by multiplying that percentage by the response time:

$$t = \frac{p_d}{p_r} * r_t$$

Once the packets are ordered by their trust level, the second phase starts. The nodes are selected having into account the path-diversity achieved. Before the selection starts, any geographic duplicate is deleted, prevailing the node with higher trust. When there are other restrictions, like the European company example, any node outside some preset boundaries is also removed. Only then, as expected, the node with higher trust level is selected (when there is more than one it is randomized) and posterior nodes are selected the same way until it reaches the limit set by a value that is defined according to the amount of data to be sent. In each selection the nodes must keep a distance from the nodes previously selected. If a node proves to be close to another then it is quite likely that a part of their path is physically the same, therefore, on each step of the node selection it is necessary to measure the path diversity d achieved. This value will contain the lower distance between two nodes in a finite set of nodes and each node can only be used if that value is over a certain constant.

Once the overlay links are set and the route established it is time to decide if the packets are going to be coded or not. In any case, having two channels is the minimum required to have a multi-path routing protocol and enough to apply network coding. However it is only possible to have a more accurate value after testing. When choosing this value it is important to notice the performance and security of applying network coding to a higher number of channels. Security should always prevail while there is an acceptable level of performance. There is also the problem of not knowing which type of network coding to use in the source and destination. The proposal is to use linear network coding with a shared secret set of coefficients. Since only the end-point nodes should have the power to completely encode and decode the packets, it is important to keep this set secret. In order to do so, Shamir's secret sharing algorithm is used. In this case, a polynomial function will be created with a number of shares necessary to retrieve the secret set equal to the number of total shares. For example, if only three channels are available the function should be parabolic so that all shares are required when decrypting.

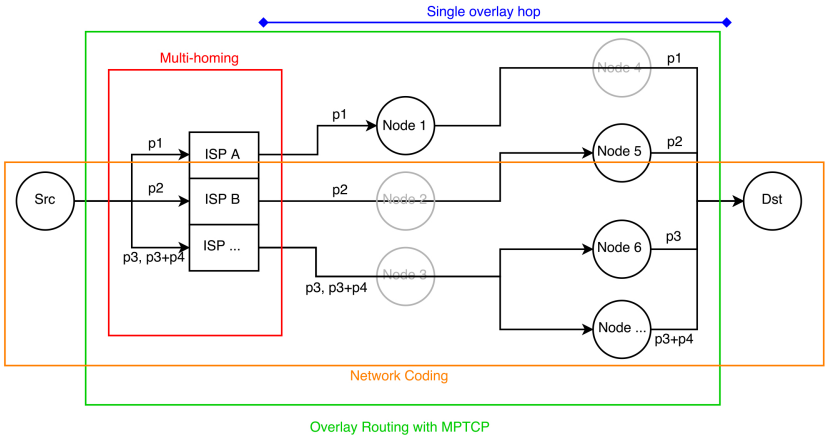

Fig. 8. Example of sending four packets using different channels, ISPs and applying network coding. The multi-homing section is only used to represent that different nodes might belong to different ISPs, they do not work as a hop. The water marked nodes are still part of the communication, however the system only forces the message to be sent over the non water marked nodes. The destination retrieves packet p4 by applying a decoding function on packets p3 and p3+p4.

Finally, all the packets are sent on the network. It should be considered that a path can fail on the meantime. Therefore the Path Discovery Component should provide $k+r$ paths where k represents the expected k links for a normal function of the communication and r represents the links that the Packet Handler will use to recover from failures on the previous ones. If it is not possible to provide

this extra value, then the rest of the data should be split among the available channels. As shown in Fig. 8, a single overlay hop is used which means that the application forces the packets to pass through a specific node. To reach it is necessary to go through other nodes.

4 Evaluation

The first step to evaluate this proposal is to select a platform composed by enough nodes so that the tests can be executed. The most reliable option for this is to use PlanetLab [35]: which is a global research network composed by over a thousand nodes that supports the development of network services.

The evaluation procedure will focus on two aspects: security and performance. Security is the main goal of this research and, therefore, will be the primary focus. Although, it is important to test the performance of the system so that it is possible to understand if it is deployable in real world applications.

The security tests will mainly consist on evaluating the ability of keeping confidentiality on data transmitted in clear and evaluating the ability to circumvent the lack of communication channels.

Regarding confidentiality, the appropriate methodology to test is to do a k-testing where k represents the amount of channels in a total t that are spied by an untrusted third party, as shown below.

$$k = \{1, ..., t-1\}$$

The maximum value of k is $t-1$,. If an attacker can spy on all channels, then it is assumed that all information is intercepted and, therefore, in that scenario this system adds no security.

On each iteration of the test, the metrics that will grade the effectiveness of maintaining confidentiality are the percentage of information jeopardized and the percentage that can be regenerated into non corrupt and reliable data.

On the performance aspect, it is important to test the optimized value of total channels used for a specified amount of data. Therefore, it is appropriate to apply another iterative test where the complete time of data transfer is evaluated for a small, medium and large size files. For example, when transferring a small 10kB file, two to five channels are tested. To measure it, the total of adding time of path selection and data transfer is the appropriate definition for the metric.

The previous tests are made without applying network coding, however it is also necessary to test the protocol when this mechanism is used. Regarding the confidentiality of the communication the same methodology as before will be applied to lower values of total channels used in the communication now applying network coding. This test will have another component when trying to regenerate the intercepted data, since it is also necessary to test if an attacker is able to retrieve the original packets by decoding the coded packets. Regarding the performance, the same methodology will be again applied to the same values, but when applying network coding. Therefore, the metric used will consist on

the total time of selecting paths, encoding packets, sending data and decoding packets.

Once both sections of testing are done - performance and security - it is necessary to combine them, so that it is possible to know what are the most adequate value that will maximize the confidentiality of the communications and provide a decent performance of communication.

These tests will prove if it is indeed possible to create a protocol that manages to defend itself from "selective forward" and "denial of service", which are indirectly correlated to the confidentiality of communications, and provide another layer of security against eavesdropping attacks that directly fight the confidentiality.

5 Schedule for Future Work

The work schedule is the following:

- January, 9th - March, 15th: Designing and implementing the proposed architecture. Including testing without network coding on PlanetLab.
- March, 16th - May, 1st: Experimental evaluation of the complete solution and analysis of the results on PlanetLab. Write the corresponding section of the dissertation;
- May, 2nd - May, 30th: Write a paper describing the project;
- May, 31st - June, 29th: Write the remaining of the dissertation document;
- June, 30th - Deliver the MSc dissertation.

6 Conclusions

This research analysed several methodologies that aim to distribute the communications through several channels, whether it is applied in wireless mesh networks or at the Internet itself. Multi-homing and overlay routing are currently the most used approaches to reach this goal, however, they are mainly focused on performance and failure recovery. Therefore, a different approach was also studied: the ability to implement multi-path routing with focus on security, most specifically in confidentiality. It is certain that there are many types of attacks that influence the confidentiality of communications directly or indirectly, but, in the scope of this research, it was considered that the attacks that mostly affect the confidentiality are "eavesdropping" or "denial of service" related. Breaking the cryptography of a channel and spying on the information that circulates on it is the attack that most directly compromises the confidentiality of the communication. In general, the idea to fight this is to distribute the communication among several channels so that only part of the information is compromised. However, this methodology is susceptible to a new types of attacks. "Selective forwarding" or "black-hole" are two types of attacks that aim to disable certain network nodes, forcing the communication to go through the nodes desired by the attacker. In this sense, the goal of this research is to provide and evaluate a

mechanism, implemented at the application level, that aims at mitigating both of these threats. By distributing the communication among several channels, using the methodologies previously stated, and applying network coding when the available channels are not enough, it is possible to improve the resistance to threats that aim to intercept communications.

References

1. David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin Vandersloot, Eric Wustrow, and Santiago Zanella-béguelin Paul. Imperfect Forward Secrecy : How Diffie-Hellman Fails in Practice. *22nd ACM Conference on Computer and Communications Security (CCS '15)*, 2015.
2. Aditya Akella, B. Maggs, S. Seshan, and A. Shaikh. On the Performance Benefits of Multihoming Route Control. *IEEE/ACM Transactions on Networking*, 16(1):91–104, 2008.
3. Aditya Akella, B Maggs, S Seshan, A Shaikh, and R Sitaraman. A measurement-based analysis of multihoming. *Computer Communication Review*, 33:353–364, 2003.
4. Y. Amir and C. Danilov. Reliable communication in overlay networks. *2003 International Conference on Dependable Systems and Networks, 2003. Proceedings.*, pages 511–520, 2003.
5. David Andersen, H Balakrishnan, M F Kaashoek, and R N Rao. Improving web availability for clients with MONET. *NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, 2:115–128, 2005.
6. David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient overlay networks. *Symposium on Networked Systems Design & Implementation*, 32(1):66, 2001.
7. David Andersen, Alex C. Snoeren, and Hari Balakrishnan. Best-path vs. multi-path overlay routing. *Proceedings of the 2003 ACM SIGCOMM conference on Internet measurement - IMC '03*, page 91, 2003.
8. Fenye Bao, Ing-ray Chen, Moonjeong Chang, Virginia Tech, and Jin-hee Cho. Hierarchical Trust Management for Wireless Sensor Networks and Its Application to Trust-Based Routing. *Information Sciences*, pages 1732–1738, 2011.
9. Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16:87–90, 1958.
10. Ning Cai and Raymond W. Yeung. Secure Network Coding on a Wiretap Network. *IEEE Transactions on Information Theory*, 57(1):424–435, 2011.
11. Philip Chou and Yunnan Wu. Network Coding for Wireless Networks. *Signal Processing Magazine, IEEE*, 24(July):77–85, 2007.
12. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Dijkstra's Algorithm. In *Introduction to Algorithms*, chapter 24.3: Dijk, pages 595–601. MIT Press, McGraw-Hill, second edition, 2001.
13. Jing Deng, Richard Han, and Shivakant Mishra. INSENS: Intrusion-Tolerant Routing in Wireless Sensor Networks. *Tech. Report CU CS-939-02*, 2006.
14. T Dierks and E Rescorla. The Transport Layer Security (TLS) Protocol, Version 1.2, (RFC 5246), 2008.

15. Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. *SSYM'04 Proceedings of the 13th conference on USENIX Security Symposium*, 13:21, 2004.
16. David Goldschlag, Michael Reed, and Paul Syverson. Hiding Routing information. *Information Hiding*, 1174:137–150, 1996.
17. Krishna P. Gummadi, Harsha V. Madhyastha, Steven D. Gribble, Henry M. Levy, and David Wetherall. Improving the reliability of internet paths with one-hop source routing. *Operating Systems Design and Implementation*, page 13, 2004.
18. Huaizhong Han, Srinivas Shakkottai, C. V. Hollot, R. Srikant, and Don Towsley. Multi-path TCP: A joint congestion control and routing scheme to exploit path diversity in the internet. *IEEE/ACM Transactions on Networking*, 14(6):1260–1271, 2006.
19. Junghee Han and F. Jahanian. Impact of path diversity on multi-homed and overlay networks. *International Conference on Dependable Systems and Networks, 2004.*, (Dsn):29–38, 2004.
20. Junghee Han, David Watson, and Farnam Jahanian. Topology aware overlay networks. *Proceedings - IEEE INFOCOM*, 4:2554–2565, 2005.
21. Junghee Han, David Watson, and Farnam Jahanian. An experimental study of internet path diversity. *Dependable and Secure Computing, IEEE Transactions on*, 3(4):273–288, 2006.
22. Jiayue He and Jennifer Rexford. Toward Internet-Wide Multipath Routing. *IEEE Network Magazine Special Issue on Scalability*, (April):16–21, 2008.
23. Zhen Jiang, Zhigang Li, Nong Xiao, and Jie Wu. CR: Capability Information for Routing of Wireless Ad Hoc Networks in the Real Environment. *2010 IEEE Fifth International Conference on Networking, Architecture, and Storage*, pages 155–164, 2010.
24. Ramin Khalili, Nicolas Gast, Miroslav Popovic, and Jean Yves Le Boudec. MPTCP is not pareto-optimal: Performance issues and a possible solution. *IEEE/ACM Transactions on Networking*, 21(5):1651–1665, 2013.
25. Hugo Krawczyk. Secret sharing made short. *Advances in Cryptology—CRYPTO '93*, pages 136–143, 1993.
26. Ppc Lee, Vishal Misra, and Dan Rubenstein. Distributed algorithms for secure multipath routing in attack-resistant networks. *ACM Transactions on Networking*, 15(6):1490–1501, 2007.
27. Ke Liu, Nael Abu-Ghazaleh, and Kyoung-Don Kang. Location verification and trust management for resilient geographic routing. *Journal of Parallel and Distributed Computing*, 67(2):215–228, 2007.
28. Wenjing Lou and Younggoo Kwon. H-SPREAD: A hybrid multipath scheme for secure and reliable data collection in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 55(4):1320–1330, 2006.
29. Takahiro Matsuda, Taku Noguchi, and Tetsuya Takine. Survey of network coding and its applications. *IEICE Transactions on Communications*, E94-B(3):698–717, 2011.
30. Foued Melakessou, Ulrich Sorger, and Zdzislaw Suchanecki. MPTCP : Concept of a Flow Control Protocol Based on Multiple Paths for the Next Generation Internet. *International Symposium on Communications and Information Technologies (ISCIT)*, pages 568–573, 2007.
31. Prayag Narula, Sanjay Kumar Dhurandher, Sudip Misra, and Isaac Woungang. Security in mobile ad-hoc networks using soft encryption and trust-based multi-path routing. *Computer Communications*, 31(4):760–769, 2008.

32. Preethi Natarajan, Nasif Ekiz, Paul D. Amer, and Randall Stewart. Concurrent Multipath Transfer during path failure. *Computer Communications*, 32(15):1577–1587, 2009.
33. Abdelraouf Ouadjaout, Yacine Challal, Noureddine Lasla, and Miloud Bagaa. SEIF: Secure and Efficient Intrusion-Fault Tolerant Routing Protocol for Wireless Sensor Networks. *2008 Third International Conference on Availability, Reliability and Security*, pages 503–508, 2008.
34. Ricardo Padilha and Fernando Pedone. Belisarius : BFT storage with confidentiality. *10th IEEE International Symposium on Network Computing and Applications (NCA)*, pages 9–16, 2011.
35. PlanetLab. https://www.planet-lab.org/, 2007.
36. Michael Reed. Anonymous Connections and Onion Routing. *Selected Areas in Communications, IEEE Journal*, 1998.
37. Adi Shamir. How to Share a Secret. *Communications of the ACM*, 27:228–234, 1979.
38. Muhammad Shoaib Siddiqui, Syed Obaid Amin, Ho Kim Jin, and Seon Hong Choong. MHRP: A secure multi-path hybrid routing protocol for wireless mesh network. *Proceedings - IEEE Military Communications Conference MILCOM*, pages 1–7, 2007.
39. Alex C. Snoeren, Kenneth Conley, and David K. Gifford. Mesh-based content routing using XML. *ACM SIGOPS Operating Systems Review*, 35(5):160, 2001.
40. Eliana Stavrou and Andreas Pitsillides. A survey on secure multipath routing protocols in WSNs. *Computer Networks*, 54(13):2215–2238, 2010.
41. Damon Wischik. Sora Promises Lasting Impact: Technical Perspective. *Communications of the ACM*, 54(1):98, 2011.
42. Damon Wischik and Costin Raiciu. Design, implementation and evaluation of congestion control for multipath TCP. *Networked Systems Design and Implementation*, pages pp. 1260–1271, 2011.
43. Xin Zhang and Adrian Perrig. Correlation-resilient path selection in multi-path routing. *IEEE Globecom*, pages 1–6, 2010.