Faculdade de Engenharia da Universidade do Porto

Designing the Internet II

Software Systems Architecture

Team T24:

André Costa Lima up202008169@up.pt

Henrique Oliveira Silva up202007242@up.pt

Rui Filipe Cunha Pires up202008252@up.pt

Diogo Miguel Ferreira Costa up202007770@up.pt

João Pedro Matos Araújo up202007855@up.pt

Pedro Miguel Nunes up202004714@up.pt

Index

1. Introduction

In the ever-changing world of technology, the design and structure of the Internet reflect innovative thinking and careful architectural planning. The paper "Extracting The Essential Simplicity of the Internet" explores the fundamental principles and decisions that have shaped the Internet as we know it today. This essay aims to break down the main ideas highlighted in the paper, including the four-layer architecture, service models, and important mechanisms like routing, reliability, and resolution. We seek to provide a thorough analysis of how these elements have contributed to the Internet's success and compare these findings with our own design, in order to gain a deeper understanding of the Internet's complex yet straightforward framework.

2. Design Choices

The internet, as we know it, has been constructed through **various modular mechanisms and design choices**. Among these, the article emphasizes a service model, a four-layer architecture, and three critical mechanisms: routing, reliability, and resolution. However, two of the most significant design choices that have contributed to the internet's success are the Four Layer Architecture and Reliable Delivery.

2.1. Four Layer Architecture

The Internet's architecture is designed in a modular way, with each layer serving a specific purpose and interacting with adjacent layers to enable communication between devices on the network. This **modular approach** is a foundational principle in network design, providing several advantages such as scalability, simplicity, flexibility, and the ability to reuse components.

Modularity entails breaking down complex systems into smaller, more manageable modules or components, which can be designed, tested, and maintained independently. This approach promotes an efficient, adaptable, and maintainable system architecture. It allows for easier expansion and scaling, simpler understanding and maintenance, and greater adaptability to changes. Moreover, it enables the reuse of components, reducing the need for redundant development efforts.

Each layer in the Internet's architecture has its **own set of protocols**. These protocols are **encapsulated** as a payload by the layer below, which means that changes to one layer's protocols do not affect those of adjacent layers. This encapsulation ensures that each layer operates independently of the others, allowing for the evolution of protocols over time and across layers.

The architecture consists of four main layers:

1. **Physical Layer (L1)** : This layer is responsible for transmitting data bits through physical communication media, such as copper cables, fiber optics, or radio waves. It deals with digital-to-analog signal conversion, error correction, and other tasks related to the physical transmission of data.

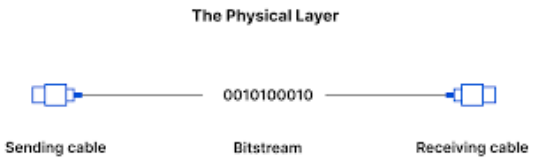

Fig. 1 - Physical information transmission channel.

2. **Network Layer (L2)** : The network layer is responsible for forming data packets from the bits received from the physical layer and adding packet headers that indicate the data's destination. It is also responsible for routing the packets to the correct destination within the local network. In non-broadcast networks like Ethernet, this layer is implemented by switches that forward packets to their destinations.

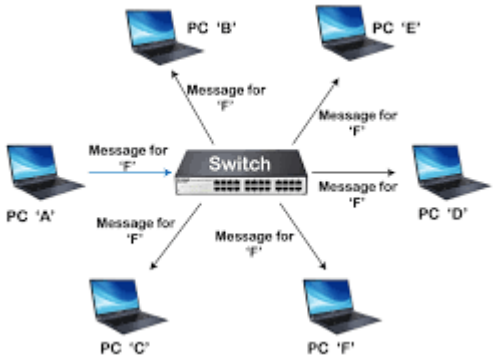

Fig. 2 - Communication of a switch with its local network hosts via mac addresses.

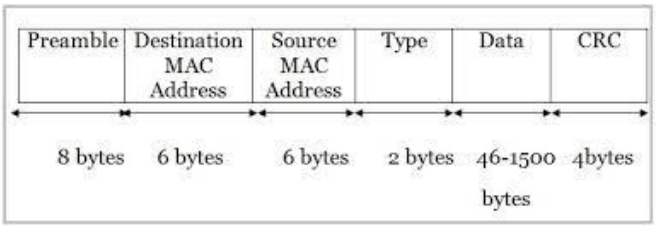

Fig. 3 - Ethernet frame format.

3. **Internetworking Layer (L3)** : This layer is responsible for routing packets between different networks. It uses the information contained in the packet headers to determine the most efficient path for the data to reach its destination. Routers are the devices that implement this layer, connecting two or more networks and forwarding packets between them.

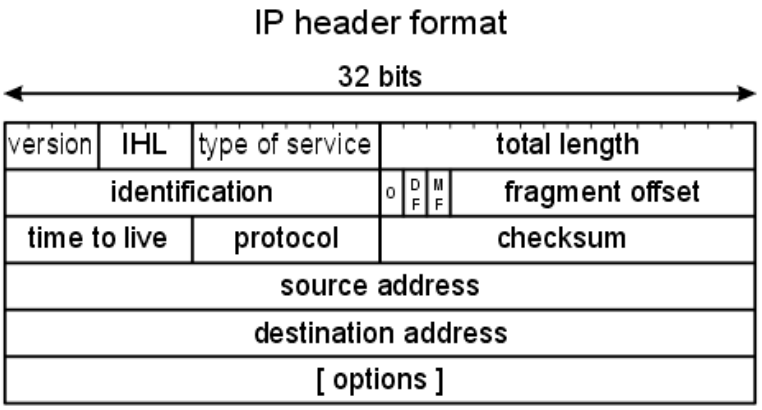

Fig. 4 - IPV4 Packet header.

4. **Transport Layer (L4)** : The transport layer is responsible for ensuring the reliable delivery of data between applications on devices connected to the network. It provides reliable data communication services, congestion control, and other services that facilitate communication between applications.

Compare TCP / UDP

TCP	UDP
Sequence	Unsequence
Reliable	Unreliable
Connection - oriented	Connection less
virtual	low overhead
Acknowledgment	No Acknowledgment
Windowing Flow Control	No windowing or flow control

Fig. 5 - TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are both transport layer protocols in the OSI model. TCP offers reliable, ordered communication, suitable for applications like emails and file transfers. UDP, on the other hand, prioritizes speed and efficiency, often used in video streaming and online gaming.

These layers interact with each other to provide efficient and reliable communication between devices connected to the Internet. For example, when an application on a device sends data to another device, the data is passed from the application layer to the transport layer, which breaks it down into packets and passes it to the network layer, which routes it to the correct destination. The network layer, in turn, passes it to the physical layer, which transmits it through the physical communication media.

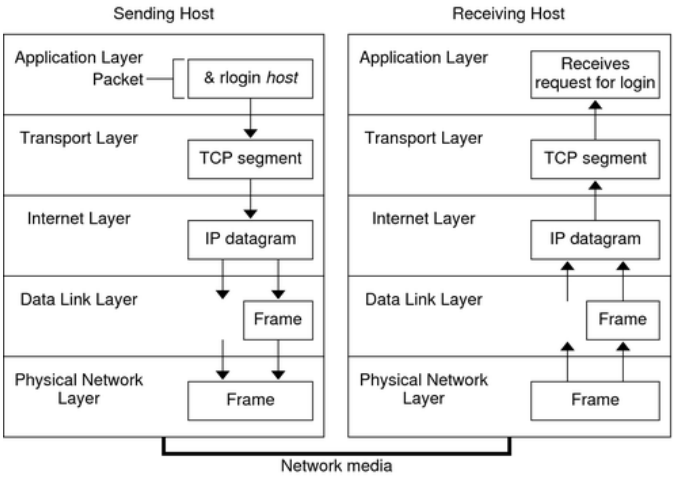

Fig. 6 - A packet´s path through the layers since de sending host to the receiving host.

2.2. Reliable Delivery

The emphasis on reliability in the design of the internet infrastructure, differing greatly from some previously designed global-scale communication infrastructure like the telephone network, can be seen as a conscious choice driven by the diverse range of applications that it is meant to support. By giving this weight to reliability, the internet becomes more suitable for applications that require constant, stable connectivity, such as financial transactions or critical data transfers.

In the article, the authors highlight (and praise) the delegation of reliability to the transport layer and even to the applications themselves, rather than guaranteeing it at the lower layers described previously.

The concept of reliable delivery is then explored within the transport layer, where protocols like **TCP** provide a reliable byte-stream abstraction, where data is transmitted in ordered packets, and any packet losses are recovered by the transport protocol through mechanisms like retransmission. The reliable byte-stream abstraction is implemented by host software, which distinguishes packets, assigns sequence numbers for proper ordering, and retransmits packets until successfully delivered. The text argues that for reliable transport, acknowledgment **(ACK)** messages are necessary and sufficient, while non-acknowledgment **(NACK)** messages are neither necessary nor sufficient. This argument is based on the premise that a reliable transport protocol can only consider a communication successful when it receives ACKs for all packets, as the absence of a NACK, which itself might be lost, does not guarantee packet receipt. Notably, TCP uses explicit ACKs for reliability and incorporates implicit NACKs through mechanisms like timeouts when expected ACKs do not arrive.

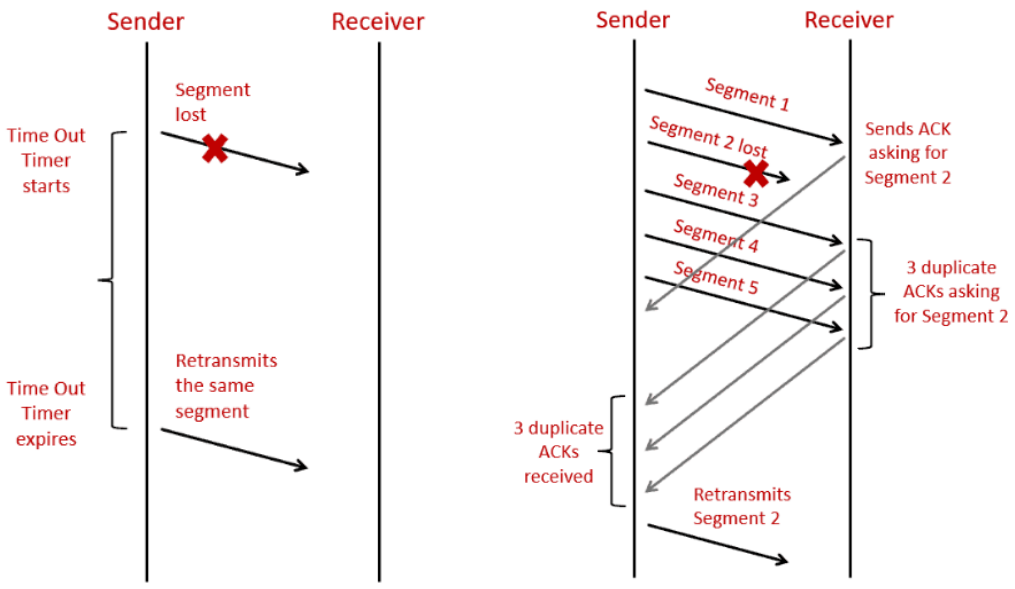

Fig. 7 - Examples of packet retransmission in TCP after timeout (left) or 3 duplicated ACKs (right)

Understanding this design choice can help us clarify how it impacts the internet's longevity. We believe reliability is strongly tied to the concept of "Assuming failure is the normal case" which the authors point as one of the secrets to this longevity. It is expected that, as the internet scales, failures in some components become increasingly likely to occur, and this design treats retransmissions of lost packets as frequent occurrences which helps handling failures in an efficient manner.

3. The "Why's" that gave us better insights on the Internet

3.1. The best-effort service model

The article explains that the Internet adopted a **modest service model** that makes no guarantees about packet delivery, instead of trying to support all possible application requirements. This choice allowed the Internet to accommodate multiple sorts of packet networks, enabling rapid growth and innovation, by leveraging the intelligence of the end hosts to adapt to network conditions. This insight allowed us to appreciate the **trade-offs between performance and flexibility** in network design and how the Internet's own simplicity was responsible for its meteoric rise.

3.2. The DNS decisions

The Domain Name System (DNS) is a critical part of the internet, converting user-friendly domain names into IP addresses. This system is hierarchical, addressing three main challenges: administrative control, handling a high volume of requests, and managing billions of names. Understanding the "why" behind DNS decisions offers insights into the internet's structure and function.

As mentioned in the article, DNS assigns **administrative control** over each domain name to a unique authority. This decentralization ensures that different entities can manage their own naming spaces without central control, fostering innovation and diversity online. In addition to that, DNS is designed to handle a **high volume of resolution requests**. Its hierarchical structure allows for parallel name resolution, improving performance and scalability as the internet grows.

Finally, the DNS is designed to provide these properties at a **scale of billions** of application-level names. By adopting an hierarchical structure, DNS guarantees the efficient management of this large number of names, ensuring that the system remains scalable and robust.
By understanding the rationale behind these decisions, we can appreciate the importance of decentralization, scalability, and performance in the design of the system. This knowledge is crucial for maintaining the internet's decentralized nature, and performance.

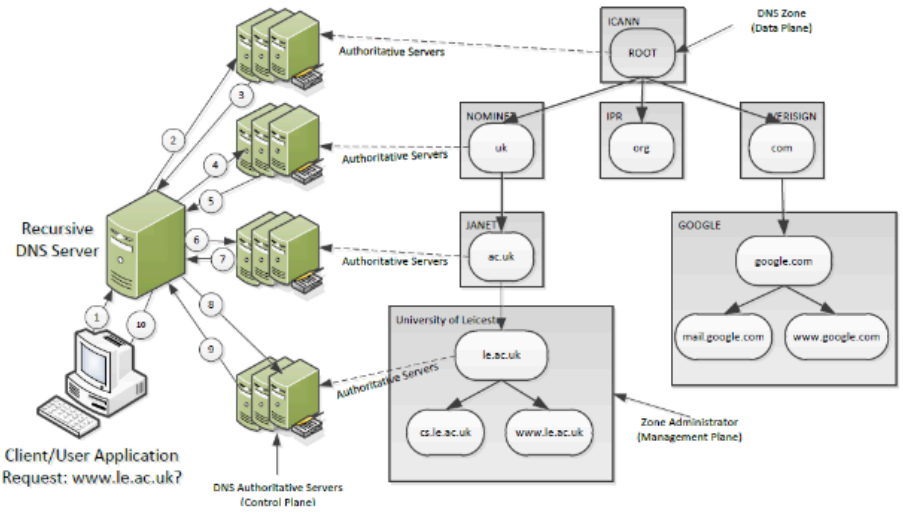

Fig. 8 - An illustration of the DNS resolution process. Radwan, Marwan & Heckel, Reiko, (2015)

4. Unpacking the Title

The title of the article, **"Extracting The Essential Simplicity of the Internet"** is quite a thought-provoker. At first glance, it might seem a bit contradictory. The Internet, as we all know it, is huge and complex. But here's the catch – the article isn't saying that the Internet is simple. Instead, it's about finding the simplicity at its core, the basic principles that make it work so well.

4.1. Historical Context

The Internet's design roots go back to the 1970s, a time when computers were just starting to talk to each other. It was the Wild West of digital communication – no standard rules, just a lot of experimentation. The designers of the Internet had a major task: to **create a system that could link different computer networks**, each with its own language and rules. It was like getting people who speak different languages to have a smooth conversation without a translator.

But here's where the brilliance comes in. Instead of creating a complex system with lots of rules to cover every possible scenario, the Internet's architects went for something more straightforward. They focused on basic, flexible principles that could work with any type of network. This choice for **simplicity and adaptability** is what the title is alluding to. It's about how these simple core ideas have allowed the Internet to grow and adapt over the years, handling everything from email to streaming videos without needing a complete overhaul.

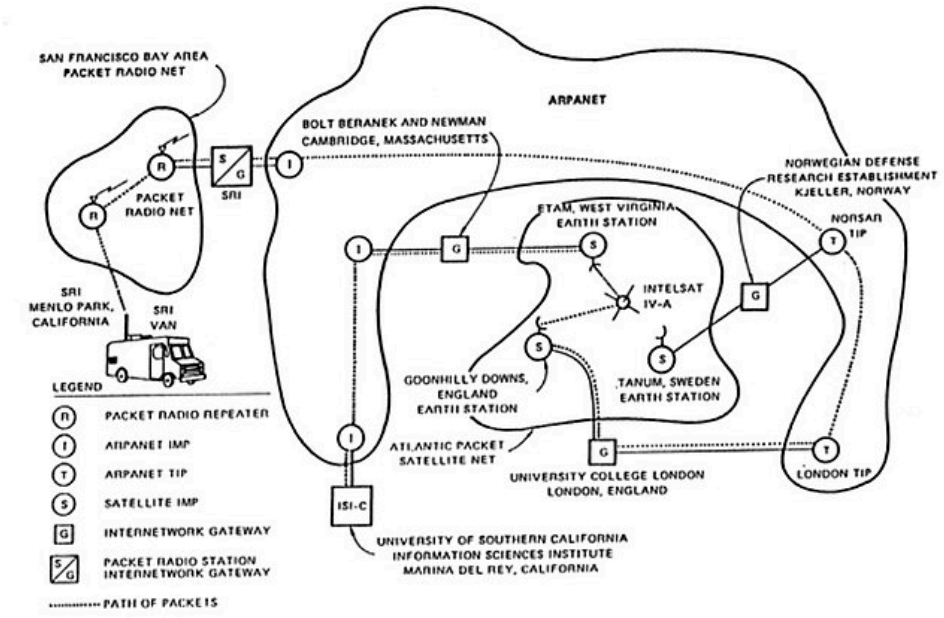

Fig. 9 - **First Internet Demonstration, 1977**. Early version of the Internet Protocol linking the ARPANET, Packet Radio Network (PRNET), and Atlantic Satellite Network (SATNET).

The compatibility offered by **standardized protocols** like TCP/IP was a game-changer. It allowed for the expansion and scaling of the network without the need to modify the underlying infrastructure significantly. This flexibility meant that as new networks emerged, they could join the, at the time, ARPANET ecosystem without requiring extensive custom integration work. This ability to accommodate a vast and growing patchwork of networks was a foundational principle that enabled the Internet to evolve into the global communication tool it is today.

5. What we learned about our Internet design

Reading the article, we gained some insights into our own design of the Internet. In the sections that follow, we'll describe them in more detail.

5.1. Lack of different routing schemes

In the article, we learned about the different routing schemes that the internet uses. These schemes ("path-vector" routing, "distance-vector" routing, "link-state" routing, among others) were designed and optimized to fulfill routing needs in specific parts of the Internet. For instance, although distance-vector routing could be used to route packets between ASes, it is not a suitable routing scheme, as more control is needed over which routes to use and announce.

Our design did not take this into account and used a universal routing scheme, based on "path-vector" routing, for every device on the Internet. Although this approach would work, having the different routing schemes that were described helps solve some issues with our design:

- **high latency** - In our system, the number of hops (number of routers a packet needs to go through to reach its destination) of a packet could be very high. This tends to mean higher latency. In the real Internet, the different routing schemes help ensure optimal performance. For instance, with "distance-vector" routing, a route with a minimal number of hops is used, which contributes to lower latency overall.
- **slow growth** - without the concept of ASes, the Internet growth would not have been as explosive. Profit-driven ASes would not have a good way to compete and there would be little incentives for them to build major Internet infrastructure.
- **low availability and hard to scale** - in our design, there is nothing that differentiates a household router and a data-center that is built to route millions of packets per hour. As such, a household router could become overwhelmed if too many routers were trying to route traffic through it, causing an outage at that router. To avoid these problems, household routers would need to become more performant, in order to not get overwhelmed. In the real Internet design, this doesn't happen because packets tend to be routed towards the border routers and, as such, household routers don't need to be very powerful. Instead, the border routers can be made more powerful, since it is expected that they will receive more traffic.

5.2. Lack of NACKs for efficiency transmissions

Regarding the transport layer, our design used ACKs in the Internet packets to ensure that their transmission is reliable. An ACK is used by the receiver to signal to the sender that a packet was successfully received. In the real Internet design, NACKs are also used to signal that a packet was not successfully received. Although NACKs are not necessary for reliable transmission, **they improve the efficiency of the transmission**, as the sender can retry the transmission of the packet as soon as it receives a NACK, instead of waiting for a timeout.

5.3. Lack of Port Numbers

In our design, Internet packets were made up of 5 fields: source, destination, acknowledgment, checksum and optional data. As described in our previous work, the source and destination fields are Internet identifiers (IP addresses, in the article), so routers know how to route packets and the receiver knows who sent the packet. The acknowledgement is used to acknowledge the successful reception of previously transmitted packets. The checksum field is used to guarantee the integrity of the packet. The optional data field is used to exchange messages of the layers above between the sender and the receiver. In essence, the packets in our design are a simplified combination of the IP (L3) packets with the TCP (L4) packets in the real Internet design. Having multiple ports offers efficient communication, security, scalability flexibility and fault tolerance in network communication.

Fig. 10 - Packet structure in our design of the Internet

With this in mind, we understood that a **source port number** and a **destination port number** are also necessary. Port numbers are used by TCP to uniquely identify an application running on the host operating system that is sending or receiving Internet packets. Without a port number, a device would only be able to establish a single connection to another device at a time or, if that were not the case, packets meant to be sent and received by an application could instead be sent or received by another application, severely limiting the capabilities of the Internet.

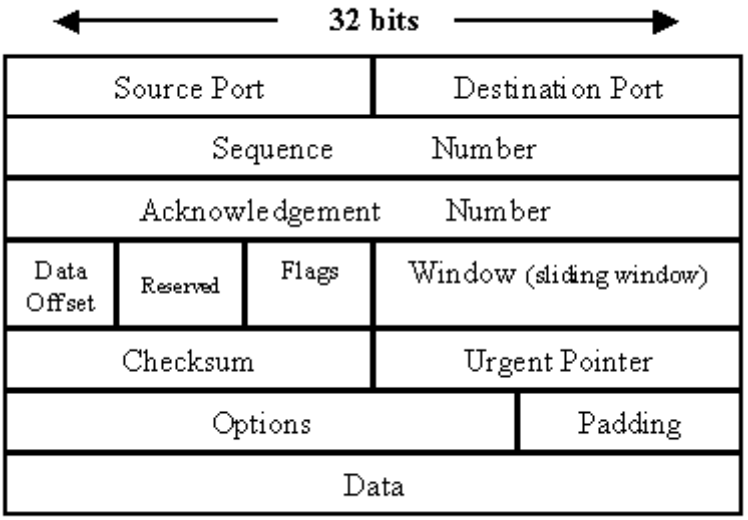

Fig. 11 - Transport Layer packet including ports and control flow mechanisms

5.4. Lack of Security in communications

Our model did not guarantee some security properties that are crucial today and that prevent numerous attacks that over time have been used to exploit this lack of security or at least lack of secrecy.
The importance of security in internet communications, including encryption, cannot be overstated. Here are some key reasons why security is crucial:

Confidentiality: Encryption ensures that only authorized parties can access and understand the data being transmitted. This is critical for protecting sensitive information such as personal data, financial details, and business secrets.
Integrity: Encryption helps prevent data tampering and ensures that the information received is the same as the information sent. This is essential for maintaining the accuracy and reliability of data.
Authentication: Encryption can be used to verify the identity of the sender and receiver, ensuring that the communication is legitimate and not intercepted or altered by malicious actors.
Non-repudiation: Encryption can provide proof of the origin and integrity of a message, making it difficult for the sender to deny sending the message or for the receiver to deny receiving it.