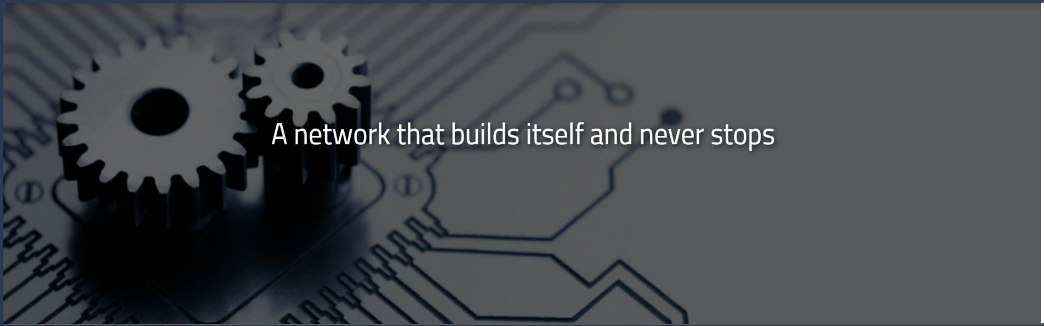
Sebyone S.r.l

DaaS-IoT: The New Frontier of Intelligent Distributed Networks



Sebyone Srl
Cosenza, Roma, ITA

www.sebyone.it
www.daasiot.com

info@sebyone.it

About us

Sebyone is an Italian startup focused on the DaaS project, the goal of our project is to explore a **new network models**.

At Sebyone, we firmly believe that the overlay network paradigm represents a forefront for the **modern digital network theory**. The overlay technology offers a straightforward way to evolve the current infrastructures into more resilient and versatile networks.

The networks we use today emerged between 1950 and 1960 when the computational power and communication technologies were highly limited. At that time, engineers did not have the possibility to choose between the media transmission methods available today, so they focused on achieving the best transmission efficiency with the basic physical channel types.

Preamble

The current networking scenery is populated with many technologies and protocols like high efficiently, for low range communication using energy-saving, and more secure and expensive technologies for long-distance interconnection among multiple nodes.

That is why multiple paradigms and standards emerged to meet various communication needs.

All of these technologies require cooperation to create the same networking environment.

Only their integration ensures true continuity and efficiency.

Sebyone introduces the **Device as a Service Overlay Mesh** (DaaS) paradigm, leveraging its protocol and software stack to enable self-organized, data-oriented infrastructures.

Device as a Service

DaaS technology emerged from **concrete design experience** and the in-depth analysis of specific issues and proposes an **innovative hypothesis** for the development of communication infrastructures for Nb-IoT and wide-band services.

Device-as-a-Service **summarises the concept of reachability of a device inside of a distributed system.**

DaaS Project

The Sebyone's DaaS Project deploys a **new network paradigm** with many features.
In this presentation we'll focus on some of those only:

- **Overlay network**: a logical interconnection of nodes, independent of the underlying transport protocols and media — effectively a 'network within the network' with its own features.

- **Protocol Channelling**: allows nodes to leverage local interfaces (such as Wi-Fi, Ethernet, or Bluetooth) to establish multiple logical links, which are evaluated in terms of availability, performance, and security.

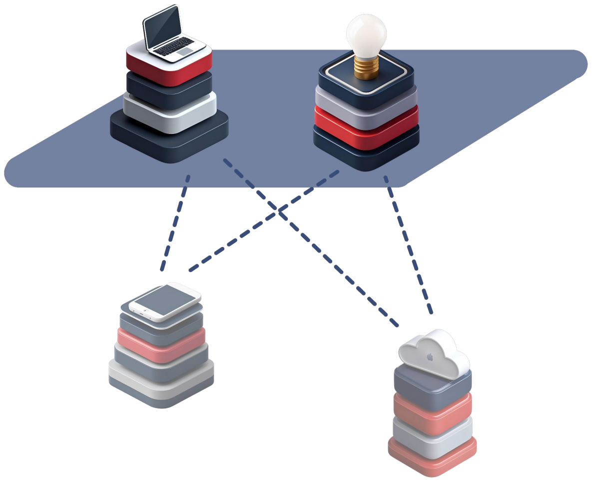
- **Self-Synchronize**: ensures that nodes share a common time reference by dynamically correcting misalignments through mutual consensus with neighboring nodes

Overlay network

An **overlay network** abstracts the physical infrastructure, enabling multiple virtual networks to coexist on the same hardware.

It introduces two layers, the **underlay** which is the physical network and the **overlay**, built on top through software.

In particular to the software technologies and logical models adopted to virtualize access to the resources of the layer below.



The overlay network offers the possibility of distributing services based on the connectivity and mobility needs of endpoints and applications, isolating the latter from the real consistency of the physical network infrastructures used.

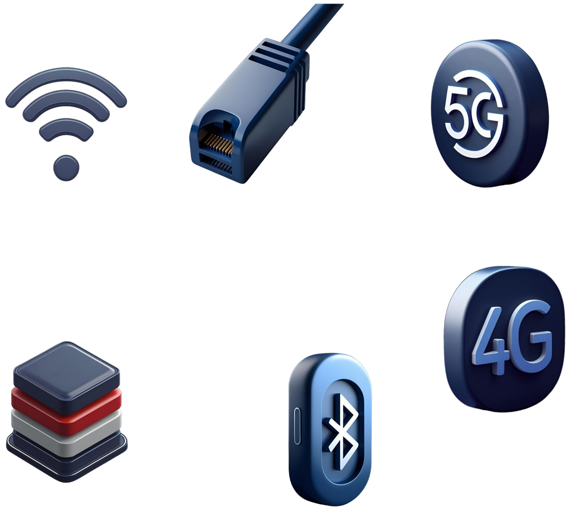
Channeling features

Nodes use its **local communication interfaces** (Wi-Fi, Ethernet, Bluetooth, etc.etc.) to activate multiple logical connections, which are then evaluated in terms of ***availability***, ***performance***, and ***security***.

This protocol uses the best communication channel available.

Devices can pass through channels, ensuring **signal availability** and **transmit data** to their destinations.

Drivers are involved in managing these interface-level operations.

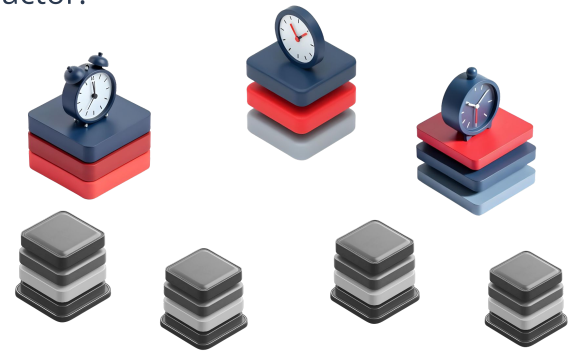


Self-Synchronize features

DaaS uses the **dATS** algorithm and its sub-protocol developed to enable autonomous and precise time synchronization of devices active on the network. The main features of **dATS** are:

- **Server-less**: alignment to the time reference without the need to reach a specific node that centralizes the synchronization service.
- **Resilient**: synchronization even in changing network conditions, maintaining high reliability.
- **Robust**: intolerant to latency and noise typical of networks that use shared channels.
- **Scalable**: effective synchronization even in multi-hop and large-scale distributed networks.
- **Precision**: effective time alignment up to an estimable approximation factor.

Nodes act as receivers and transmitters to exchange data useful for performing drift and offset compensation of the clocks.



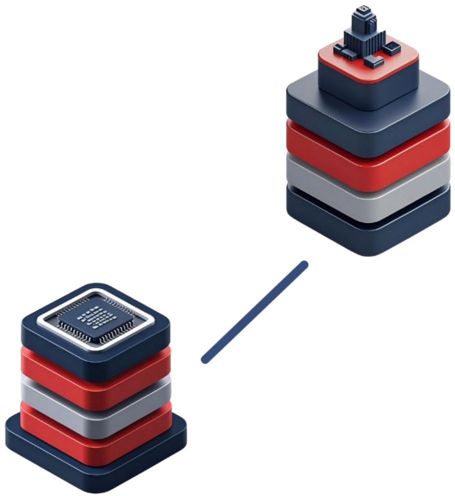
dATS enables precise synchronization of clocks across multiple devices, despite the presence of substantial communication delays

How does DaaS work?

A DaaS network is made up of at least two nodes that are interconnected.

The node is a device that hosts the **DaaS stack**, which uses the communication resources available locally to reach other nodes.

A DaaS node is considered **connected** if there is at least one communication channel through which it is possible to reach another node in the network.



In a DaaS network the node is defined by a tuple of identifiers: the ***Daas IdeNtifier (DIN)*** that identifies the instance and the ***System IDentifier (SID)*** that identifies the distributed system to which the node belongs.

Demo - Introduction

The main goal is to demonstrate the validity of DaaS technology in building distributed measurement systems.
The evaluation includes tests that will be performed using the DaaS-IoT (daasiot.com) software product and **dsperf** (https://github.com/sebyone/dsperf).

DaaS-IoT implements the Sebyone DaaS protocol and makes it available to **quickly activate multi-platform node networks**.
Some of the unique characteristics of DaaS which will be evaluated are:

Overlay and channelling
The node uses specific drivers and different hardware communication resources.

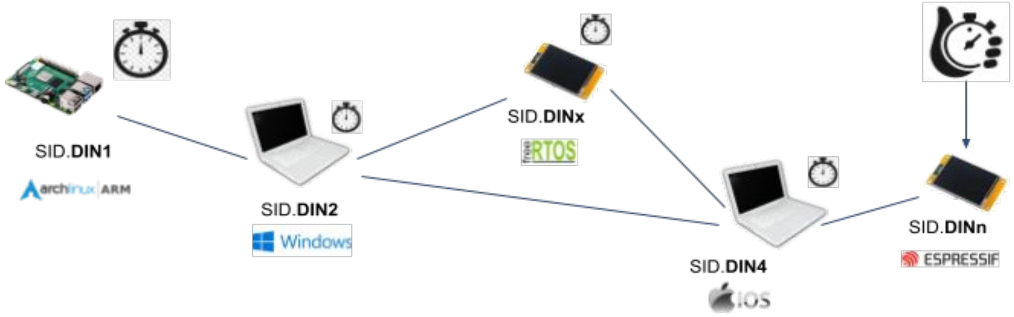
Automatic synchronization of all network's nodes
The nodes in the network synchronize with each other with a proprietary server-less algorithm.

Test Environment

The test environment is defined to a **minimal network infrastructure**, specifically designed to allow the performances measured to **different underlay protocols and DaaS overlay**.

To perform accurate testing, we rely on standard devices such as:

- Raspberry Pi 3 Model A+ (Wi-Fi, Bluetooth)
- Raspberry 3 Model B+ (Wi-Fi, Bluetooth, Ethernet)
- ESP32 DevKitC (Wi-Fi, Bluetooth)
- 4G LTE Router (Wi-Fi, Ethernet)
- Switch 10/100Mb (Ethernet)



Case 1

Direct link between two nodes in DaaS network

An overlay network where all nodes communicate with the same underlying technology.
In this scenario we have set up a **WiFi LAN on IPV4 protocol**.

We measured the overhead introduced by DaaS Overlay Technology on a transferring process resolved between **two nodes directly connected:**

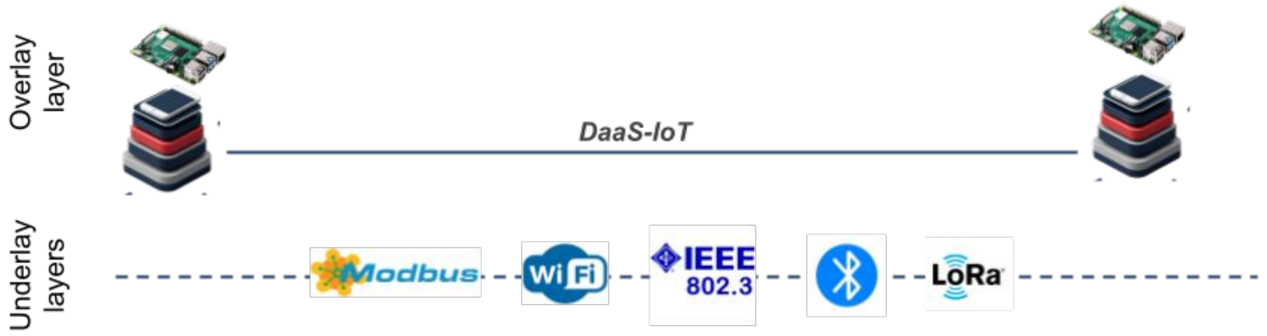


Fig.1 - "Case 1: link between two nodes"

Case 1

Measurement

To run the test, we used the dsperf tool in two phases:

1. Measuring the underlay network throughput (using socket.h)
2. Measuring the overlay network throughput (using libdaas.h)

Transfer Time [s] - dsperf (v.0.34) - Underlay			
Block size	802.3ab	802.11b	802.15 (LE2M)
10Kb	0,0130	0,0140	0,0655
100Kb	0,0562	0,0614	1,1442
5Mb	0,0701	0,0980	25,5714

Throughput [MBps] - dsperf (v.0.34) - Underlay			
Block size	802.3ab	802.11b	802.15 (LE2M)
10Kb	0,045	0,997	0,000196
100Kb	54,976	1,207	0,003770
5Mb	637,535	109,920	0,009604

We performed multiple measurements by varying the size of the transferred data blocks, and consequently, the number of packets transmitted and received.

Case 1

Measurement

After that we performed the same test using DaaS Overlay with respect to same underlay technologies:

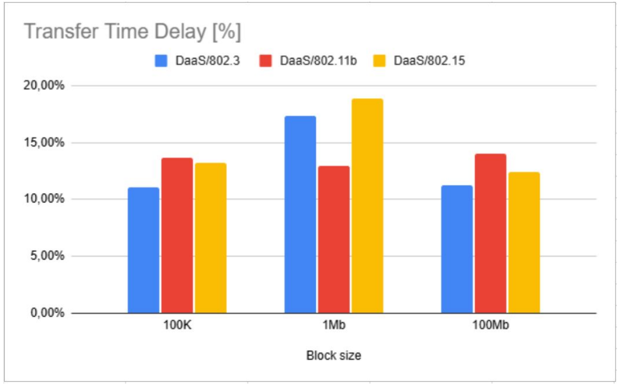
Transfer Time [s] - dsperf (v.0.34) - Overlay			
Block size	DaaS/802.3	DaaS/802.11	DaaS/802.15
100K	0,0145	0,0159	0,0742
1Mb	0,0660	0,0694	1,3597
10Mb	0,0780	0,1117	28,7514

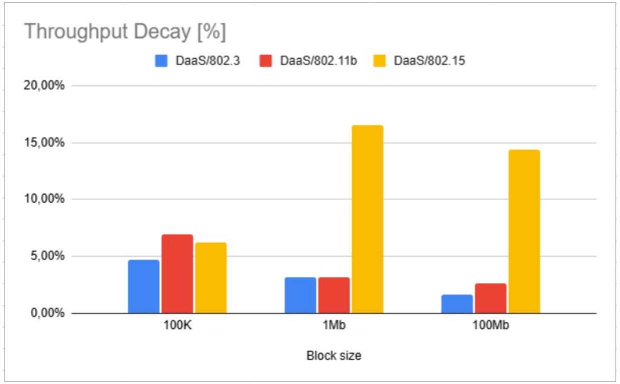
Throughput [MBps] - dsperf (v.0.34) - Overlay			
Block size	DaaS/802.3	DaaS/802.11	DaaS/802.15
100K	0,043	0,928	0,000184
1Mb	53,250	1,168	0,003147
10Mb	626,852	107,041	0,008226

Case 1

Measurement

Transfer Time Delay [%]			
Block size	DaaS/802.3	DaaS/802.11b	DaaS/802.15
100K	11,04%	13,65%	13,24%
1Mb	17,35%	12,96%	18,84%
100Mb	11,25%	14,00%	12,44%
average	13,21%	13,54%	14,84%





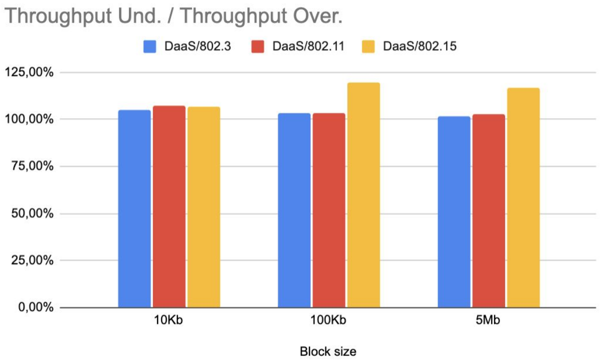
Throughput Decay [%]			
Block size	DaaS/802.3	DaaS/802.11b	DaaS/802.15
100K	4,66%	6,98%	6,26%
1Mb	3,14%	3,21%	16,52%
100Mb	1,68%	2,62%	14,35%
average	3,16%	4,27%	12,38%

Case 1

Considerations

By analyzing the measurement data, we can compare the throughput values obtained from transfers using only the underlay layer with those measurement using the overlay APIs.

The overhead is expressed either as a percentage or as a ratio between the two values: **trh_und / trh_over**.



Finally, considering the advanced features introduced by DaaS — such as self-configuration, security, and scalability mechanisms — and the resulting ability to optimize the distributed measurement network, **the introduced delay can be considered negligible compared to the overall benefits**.

Case 2

All nodes follow the same time reference

In this case, we analyze the network’s ability to synchronize the timers of all active nodes.

In a DaaS network, an “active” node is always synchronized to a time reference known as the "network time", with a negligible margin of error referred to as the "**network period"**.

Network time and network period are independent of transfer times and network latencies.

In short, for a node to become active, it must synchronize its timer with that of the first already active node it contacts.

From that moment on, the node aligns with the reference time provided by the oldest active node (the time-master), regardless of the underlying communication technologies.

Case 2

Measurement

To run the test we used node implementations that display the device's internal timer time. The first activated node assumes the time-master condition, i.e. it defines the reference time for the entire network.

The evaluation focuses on:

1. Time taken to achieve synchronization
2. Timer discrepancies between nodes

Nodes send log messages with their local timer values before and after synchronization, the applied time delta, and the total synchronizations performed. Logs are stored using a custom logging application.

The experiment included network perturbations such as channel switching, timer changes on non-master nodes, and adding new nodes to verify synchronization (e.g., powering on an ESP-TFT).

Case 2

Measurement

#	t.master [ms]	Remote node start timestamp	1st→t.mast time to sync	Remote node synced timestamp (after ats)	network period (error between the two timestamp after dATS)
	1749599228646	1749599229029	383 ms	1749599228646	+/- 27 ms
	1749599240656	1749577983986	322 ms	1749599240656	+/- 14 ms
	1749599252661	1749574735436	664 ms	1749599252661	+/- 7 ms
	1749599264663	1749566715891	337 ms	1749599264663	+/- 3ms
	1749599300689	1749550745559	344 ms	1749599300689	+/- 1ms
	….	….	…	…	…

Case 2

Considerations

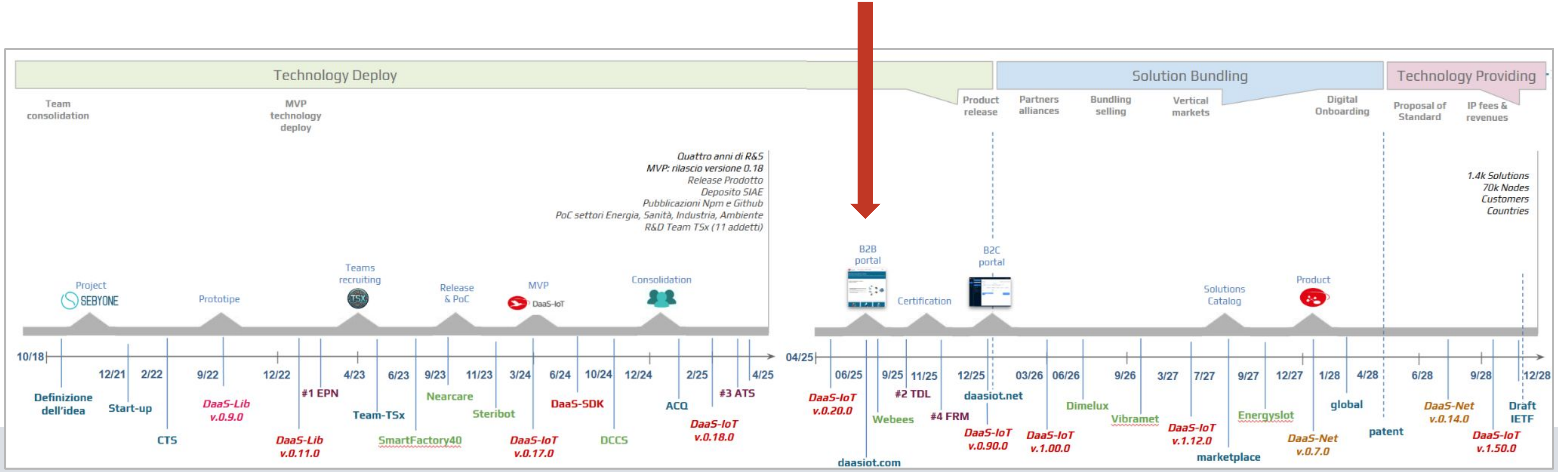
Even when remote nodes start with different clocks, early tests of the dATS protocol have shown it can reliably synchronize each node to the network's master clock.

The timestamp compensates also the network delays, ensuring high precision regarding packet signature, which is important for time-sensitive applications.

Avg time used to sync	Avg network period error	Avg ATS packet used for the sync process
380 ms	+/- 6.3 ms	12 pkt

The flexibility of this approach is further demonstrated by the ability to connect a new node and quickly bring it into sync with the network's reference clock.

Conclusion



DaaS Project is **very ambitious** but, as history teaches us, many of the technical solutions to real problems were born and then evolved into forms that we could only dream of.

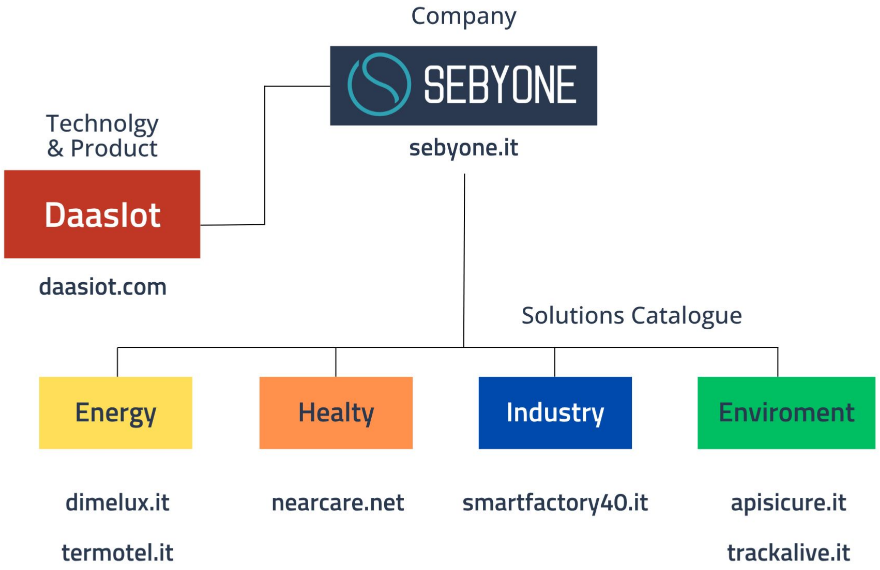
In the same way, I believe that now, more than ever, **we are ready to change the way we think about digital communication and its critical elements**.

Let's imagine **totally decentralized and intelligent networks**, no longer channels on which data can pass but active and distributed systems capable of autonomously re-organizing themselves to **redesign the communications landscape in a not too distant future**.

Greetings

We are working hard to concretize our vision and we understand how challenging research can be without collaborations.
For these reasons and much more, we enjoy sharing our work with anyone, individuals and organizations, who are interested in our project.

We invite you to contact us for any questions. We look forward to hearing from you on our platforms.



Thank you.

Let's build the future of connectivity together.

Sebyone Srl
Cosenza, ITA
Roma, ITA

info@sebyone.it

amministrazione@sebyone.it
developers@sebyone.it
hr@sebyone.it

www.sebyone.com
www.daasiot.com
www.daasiot.net
www.trackalive.it
www.smartfactory40.it
www.dimelux.it