

REDIMadrid: **Estado presente y plan de futuro**

César Sánchez

XX Jornadas REDIMadrid

Madrid, 21 de octubre de 2025

software



César Sánchez
Director

Full Professor en IMDEA-Software, donde lidero un grupo de investigación en:

- Métodos formales
- Lógica computacional, teoría de automátas, juegos y lenguajes formales
- Verificación y síntesis, runtime verification
- Sistemas reactivos



César Sánchez

Director

Full Professor en IMDEA-Sof

- Métodos formales
- Lógica computacional, te
- Verificación y síntesis, m
- Sistemas reactivos

César Sánchez

Temporal Logics for Hyperproperties

Michael R. Clarkson¹, Bernd Finkbeiner², Masoud Koeini¹,
Kristopher K. Micinski³, Markus N. Rabe², and César Sánchez⁴

¹ George Washington University, USA

² Universität des Saarlandes, Germany

³ University of Maryland, College Park, USA

⁴ IMDEA Software Institute, Spain

Abstract. Two new logics for verification of hyperproperties are proposed. Hyperproperties characterize security policies, such as noninterference, as a property of sets of computation paths. Standard temporal logics such as LTL, CTL, and CTL* can refer only to a single path at a time, hence cannot express many hyperproperties of interest. The logics proposed here, HyperLTL and HyperCTL*, add explicit and simultaneous quantification over multiple paths to LTL and to CTL*. This kind of quantification enables expression of hyperproperties. A model checking algorithm for the proposed logics is given. For a fragment of HyperLTL, a prototype model checker has been implemented.

1 Introduction

Trace properties, which developed out of an interest in proving the correctness of programs [32], characterize correct behavior as properties of individual execution traces. Although early verification techniques specialized in proving individual correctness properties of interest, such as mutual exclusion or termination, temporal logics soon emerged as a general, unifying framework for expressing and verifying trace properties. Practical model checking tools [11, 16, 28] based on those logics now enable automated verification of program correctness.

Verification of security is not directly possible with such tools, because some important security policies cannot be characterized as properties of individual execution traces [38]. Rather, they are properties of sets of execution traces, also known as *hyperproperties* [15]. Specialized verification techniques have been developed for particular hyperproperties [5, 27, 41, 43], as well as for *2-safety* properties [52], which are properties of pairs of execution traces. But a unifying program logic for expressing and verifying hyperproperties could enable automated verification of a wide range of security policies.

In this paper, we propose two such logics. Both are based, like hyperproperties, on examining more than one execution trace at a time. Our first logic, *HyperLTL*, generalizes linear-time temporal logic (LTL) [44]. LTL implicitly quantifies over only a single execution trace of a system, but HyperLTL allows explicit quantification over multiple execution traces simultaneously, as well as propositions that stipulate relationships among those traces. For example, HyperLTL can express

M. Abadi and S. Kremer (Eds.): POST 2014, LNCS 8414, pp. 265–284, 2014.
© Springer-Verlag Berlin Heidelberg 2014

ción en:



César Sánchez
Director

Full Professor en IMDEA-Sof

- Métodos formales
- Lógica computacional, te
- Verificación y síntesis, m
- Sistemas reactivos

César Sánchez

Temporal Logics for Hyperproperties

Michael R. Clarkson¹,
Kristopher K. Micinski³

¹ George V

² Universit

³ University o

⁴ IMDEA

Abstract. Two new logics posed. Hyperproperties cha ference, as a property of set logics such as LTL, CTL, au time, hence cannot express proposed here, HyperLTL a ous quantification over mult quantification enables expr algorithm for the proposed a prototype model checker

1 Introduction

Trace properties, which develop programs [32], characterize corre traces. Although early verificati correctness properties of interest poral logics soon emerged as a verifying trace properties. Prac those logics now enable automa

Verification of security is not important security policies can execution traces [38]. Rather, t also known as *hyperproperties* [1 developed for particular hyper properties [52], which are prop: program logic for expressing an mated verification of a wide ran

In this paper, we propose two: on examining more than one exe generalizes linear-time temporal only a single execution trace of a cation over multiple execution tr stipulate relationships among th

LOLA: Runtime Monitoring of Synchronous Systems

Ben D'Angelo * Sriram Sankaranarayanan * César Sánchez * Will Robinson *
Bernd Finkbeiner † Henny B. Sipma * Sandeep Mehrotra ‡ Zohar Manna *

* Computer Science Department, Stanford University, Stanford, CA 94305
{bdangelo,srirams,cesar,sipma,manna}@theory.stanford.edu

† Department of Computer Science, Saarland University finkbeiner@cs.uni-sb.de ‡ Synopsys, Inc.

Abstract—We present a specification language and algorithms for the online and offline monitoring of synchronous systems including circuits and embedded systems. Such monitoring is useful not only for testing, but also under actual deployment. The specification language is simple and expressive; it can describe both correctness/failure assertions along with interesting statistical measures that are useful for system profiling and coverage analysis. The algorithm for online monitoring of queries in this language follows a partial evaluation strategy: it incrementally constructs output streams from input streams, while maintaining a store of partially evaluated expressions for forward references. We identify a class of specifications, characterized syntactically, for which the algorithm's memory requirement is independent of the length of the input streams. Being able to bound memory requirements is especially important in online monitoring of large input streams. We extend the concepts used in the online algorithm to construct an efficient offline monitoring algorithm for large traces.

We have implemented our algorithm and applied it to two industrial systems, the PCI bus protocol and a memory controller. The results demonstrate that our algorithms are practical and that our specification language is sufficiently expressive to handle specifications of interest to industry.

I. INTRODUCTION

Monitoring synchronous programs for safety and liveness properties is an important aspect of ensuring their proper runtime behavior. An offline monitor analyzes traces of a system post-simulation to spot violations of

the specification. Offline monitoring is critical for testing large systems before deployment. An online monitor processes the system trace while it is being generated. Online monitoring is used to detect violations of the specification when the system is in operation so that they can be handled before they translate into observable and cascading failures, and to adaptively optimize system performance.

Runtime monitoring has received growing attention in recent years [1], [2], [3]. While static verification intends to show that every (infinite) run of a system satisfies the specification, runtime monitoring is concerned only with a single (finite) trace. Runtime monitoring can be viewed as an extension of testing with more powerful specification languages.

The offline monitoring problem is known to be easy for purely past or purely future properties. It is well known that for past properties, the online monitoring problem can be solved efficiently using constant space and linear time in the trace size. For future properties, on the other hand, the space requirement generally depends on the length of the trace, which suggests that online monitoring may quickly become intractable in practical applications with traces exceeding 10^6 simulation steps.

In this paper, we present a specification language, intended for industrial use. The language can express properties involving both the past and the future. It is a functional stream computation language like LUSTRE [4] and ESTEREL [5], with features that are relevant to our problem at hand. It is parsimonious in its number of operators (expressions are constructed from three basic operators), but the resulting expressiveness surpasses temporal logics and many other existing formalisms

Proceedings of the 12th International Symposium on Temporal Representation and Reasoning (TIME'05)
1530-1311/05 \$20.00 © 2005 IEEE

M. Abadi and S. Kremer (Eds.): POST, LNCS 3016, 2004. Downloaded on October 20, 2008 at 10:36 from IEEE Xplore. Restrictions apply.
© Springer-Verlag Berlin Heidelberg 2014



César Sánchez

Director

Full Professor en IMDEA-Soft

- Métodos formales
- Lógica computacional, te
- Verificación y síntesis, m
- Sistemas reactivos

César Sánchez

Temporal Logics for Hyperproperties

Michael R. Clarkson¹,
Kristopher K. Micinski³

¹ George V
² Universit
³ University o
⁴ IMDEA

Abstract. Two new logics posed. Hyperproperties cha
ference, as a property of se
logics such as LTL, CTL, a
time, hence cannot express
proposed here, HyperLTL a
ous quantification over mult
quantification enables expr
algorithm for the proposed
a prototype model checker

1 Introduction

Trace properties, which develop
programs [32], characterize corre
traces. Although early verificati
correctness properties of interest
poral logics soon emerged as a
verifying trace properties. Prac
those logics now enable automa

Verification of security is not
important security policies can
execution traces [38]. Rather, t
also known as *hyperproperties* [1
developed for particular hyper
properties [52], which are prop
program logic for expressing an
mated verification of a wide ran

In this paper, we propose two
on examining more than one exe
generalizes linear-time temporal
only a single execution trace of a
cation over multiple execution tr
stipulate relationships among th

LOLA: Runtime

Ben D'Angelo * Sriram Sank
Bernd Finkbeiner † Henny

* Computer Science De
{bdangelo,srirams,c

† Department of Computer
finkbeiner@

Abstract—We present a specification langu
rithms for the online and offline monitoring o
systems including circuits and embedded s
monitoring is useful not only for testing, b
actual deployment. The specification langu
and expressive; it can describe both con
assertions along with interesting statistica
are useful for system profiling and cover
The algorithm for online monitoring of q
language follows a partial evaluation strat
mentally constructs output streams from i
while maintaining a store of partially evaluat
for forward references. We identify a clas
tions, characterized syntactically, for which t
memory requirement is independent of the
input streams. Being able to bound memory
is especially important in online monitoring
streams. We extend the concepts used in t
rithm to construct an efficient offline monitor
for large traces.

We have implemented our algorithm and
two industrial systems, the PCI bus protocol
controller. The results demonstrate that our
practical and that our specification language
expressive to handle specifications of interes

I. INTRODUCTION

Monitoring synchronous programs for sa
ness properties is an important aspect of c
proper runtime behavior. An offline mon
traces of a system post-simulation to spot

This research was supported in part by NSF
21403, CCR-02-20134, CCR-02-09237, CNS-0411
0430102, by ARO grant DAAD19-01-1-0723, by N
tract N00014-03-1-0939, by the Siebel Graduate Fel
the BMBF grant 01 IS C38 B as part of the Verif

Proceedings of the 12th International Symposium on Temporal Representation and Reasoning (TIME'05)
1530-1311/05 \$20.00 © 2005 IEEE

M. Abadi and S. Kremer (Eds.): POST, LNCS 3025, pp. 200–214, 2004.
© Springer-Verlag Berlin Heidelberg 2014

Boolean Abstractions for Realizability Modulo Theories

Andoni Rodríguez^{1,2} and César Sánchez¹

¹ IMDEA Software Institute, Madrid, Spain
{andoni.rodriguez,cesar.sanchez}@imdea.org
² Universidad Politécnica de Madrid, Madrid, Spain

Abstract. In this paper, we address the problem of the (reactive) realizability of specifications of theories richer than Booleans, including arithmetic theories. Our approach transforms theory specifications into purely Boolean specifications by (1) substituting theory literals by Boolean variables, and (2) computing an additional Boolean requirement that captures the dependencies between the new variables imposed by the literals. The resulting specification can be passed to existing Boolean off-the-shelf realizability tools, and is realizable if and only if the original specification is realizable. The first contribution is a brute-force version of our method, which requires a number of SMT queries that is doubly exponential in the number of input literals. Then, we present a faster method that exploits a nested encoding of the search for the extra requirement and uses SAT solving for faster traversing the search space and uses SMT queries internally. Another contribution is a prototype in Z3-Python. Finally, we report an empirical evaluation using specifications inspired in real industrial cases. To the best of our knowledge, this is the first method that succeeds in non-Boolean LTL realizability.

1 Introduction

Reactive synthesis [30,31] is the problem of automatically producing a system that is guaranteed to model a given temporal specification, where the Boolean variables (i.e., atomic propositions) are split into variables controlled by the environment and variables controlled by the system. Realizability is the related decision problem of deciding whether such a system exists. These problems have been widely studied [17,21], specially in the domain of Linear Temporal Logic (LTL) [29]. Realizability corresponds to infinite games where players alternatively choose the valuations of the Boolean variables they control. The winning condition is extracted from the temporal specification and determines which player wins a given play. A system is realizable if and only if the system player

This work was funded in part by the Madrid Regional Gov. Project “S2018/TCS-4339 (BLOQUES-CM)”, by PRODIGY Project (TED2021-132464B-I00) funded by MCIN/AEI/10.13039/501100011033/ and the European Union Next Generation EU/PRTR, and by a research grant from Nomadic Labs and the Tezos Foundation.

© The Author(s) 2023
C. Enes and A. Lal (Eds.): CAV 2023, LNCS 13966, pp. 305–328, 2023.
https://doi.org/10.1007/978-3-031-37709-9_15



César Sánchez
Director

Full Professor en IM[

- Métodos formale
- Lógica computaci
- Verificación y sínt
- Sistemas reactivos

César Sánchez



Cesar Sanchez

Professor of Computer Science, [IMDEA Software Institute](#)

Verified email at imdea.org - [Homepage](#)

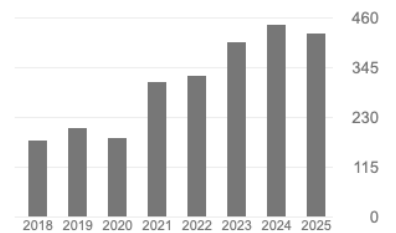
[Formal Methods](#) [Logic](#) [Reactive Systems](#) [Computer Science](#)

[FOLLOW](#)

<input type="checkbox"/>	TITLE	CITED BY	YEAR
<input type="checkbox"/>	Temporal logics for hyperproperties MR Clarkson, B Finkbeiner, M Koeini, KK Micinski, MN Rabe, C Sánchez International Conference on Principles of Security and Trust, 265-284	408	2014
<input type="checkbox"/>	Lola: Runtime monitoring of synchronous systems B d'Angelo, S Sankaranarayanan, C Sanchez, W Robinson, B Finkbeiner, ... Temporal Representation and Reasoning, 2005. TIME 2005. 12th International ...	405	2005
<input type="checkbox"/>	Algorithms for model checking HyperLTL and HyperCTL B Finkbeiner, MN Rabe, C Sánchez International Conference on Computer Aided Verification, 30-48	219	2015
<input type="checkbox"/>	A survey of challenges for runtime verification from advanced application domains (beyond software) C Sánchez, G Schneider, W Ahrendt, E Bartocci, D Bianculli, C Colombo, ... Formal Methods in System Design 54 (3), 279-335	145	2019
<input type="checkbox"/>	Runtime verification for decentralised and distributed systems A Francalanza, JA Pérez, C Sánchez Lectures on Runtime Verification: Introductory and Advanced Topics, 176-210	105	2018
<input type="checkbox"/>	Verifying hyperliveness N Coenen, B Finkbeiner, C Sánchez, L Tentrup International Conference on Computer Aided Verification, 121-139	96	2019
<input type="checkbox"/>	Regular linear temporal logic M Leucker, C Sánchez Theoretical Aspects of Computing-ICTAC 2007, 291-305	92	2007
<input type="checkbox"/>	Striver: Stream runtime verification for real-time event-streams F Gorostiaga, C Sánchez International Conference on Runtime Verification, 282-298	88	2018

Cited by [VIEW ALL](#)

	All	Since 2020
Citations	3096	2101
h-index	24	22
i10-index	60	39





Public access [VIEW ALL](#)

5 articles	53 articles
not available	available

Based on funding mandates

Co-authors [EDIT](#)

	Felipe Gorostiaga IMDEA Software Institute	>
	Bernd Finkbeiner Professor of Computer Science, ...	>
	Martin Leucker Professor of Computer Science, ...	>



César Sánchez
Director



David Rincón
Coordinador Técnico



Alejandro Recio
Técnico de soporte



Carlos de Higes
Técnico de red

+ Apoyo de IMDEA-Software:

- gestión y soporte administrativo (María Alcaraz y su equipo, Manuel Carro, Antonio Fernández-Anta)
- técnico (Juan Céspedes, Roberto Lumbreras, Tomas Kriukelis, David Colera)

Jerarquía de **redes telemáticas de investigación**:

- GEANT: Pan-Europea
- RedIRIS: España
- REDIMadrid: Comunidad de Madrid

Regulado por acuerdo marco

Sábado 1 junio 2002

BOE núm. 131

pa e Iberoa-
31, de 13 de
i de España

l y orgánica-
ecretaría de
ica, sin per-
erio de Edu-
: juzgue más
n, conforme

inisterios de
arán, anual-
a capacidad
resupuestos

MINISTERIO DE CIENCIA Y TECNOLOGÍA

10663 *RESOLUCIÓN de 6 de mayo de 2002, de la Secretaría de Estado de Política Científica y Tecnológica, por la que se dispone la publicación del Protocolo General por el que se establece el Acuerdo Marco entre el Ministerio de Ciencia y Tecnología y la Comunidad Autónoma de Madrid para la coordinación de actuaciones en materia de investigación científica, desarrollo e innovación tecnológica.*

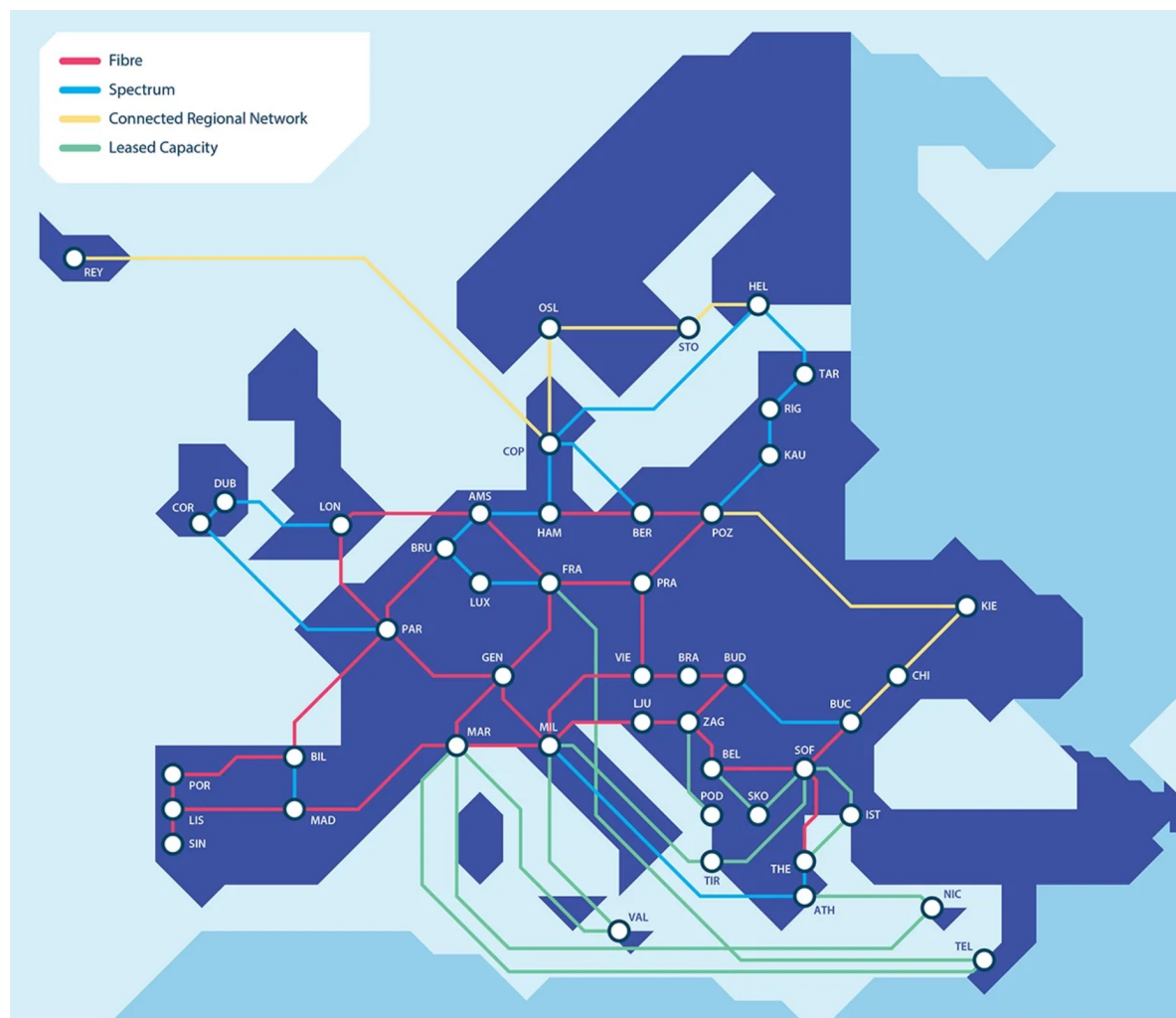
Madrid, 6 de mayo de 2002.—El Secretario de Estado, RAMÓN MARIMÓN Suñol.

ANEXO

Protocolo General por el que se establece el Acuerdo Marco entre el Ministerio de Ciencia y Tecnología y la Comunidad Autónoma de Madrid para la coordinación de actuaciones en materia de investigación científica, desarrollo e innovación tecnológica

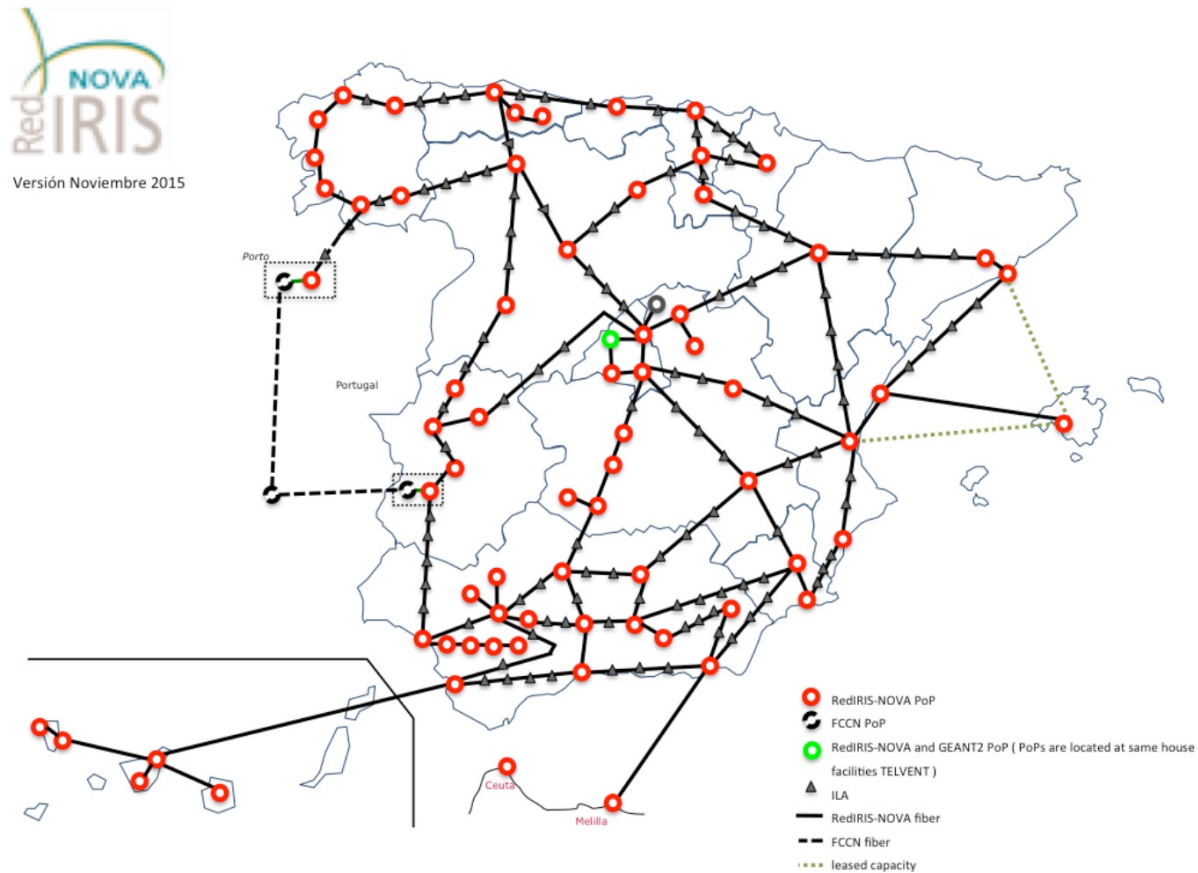
EL MINISTERIO DE CIENCIA Y TECNOLOGÍA

Geant (Red Europea)



(Financiada por la UE)

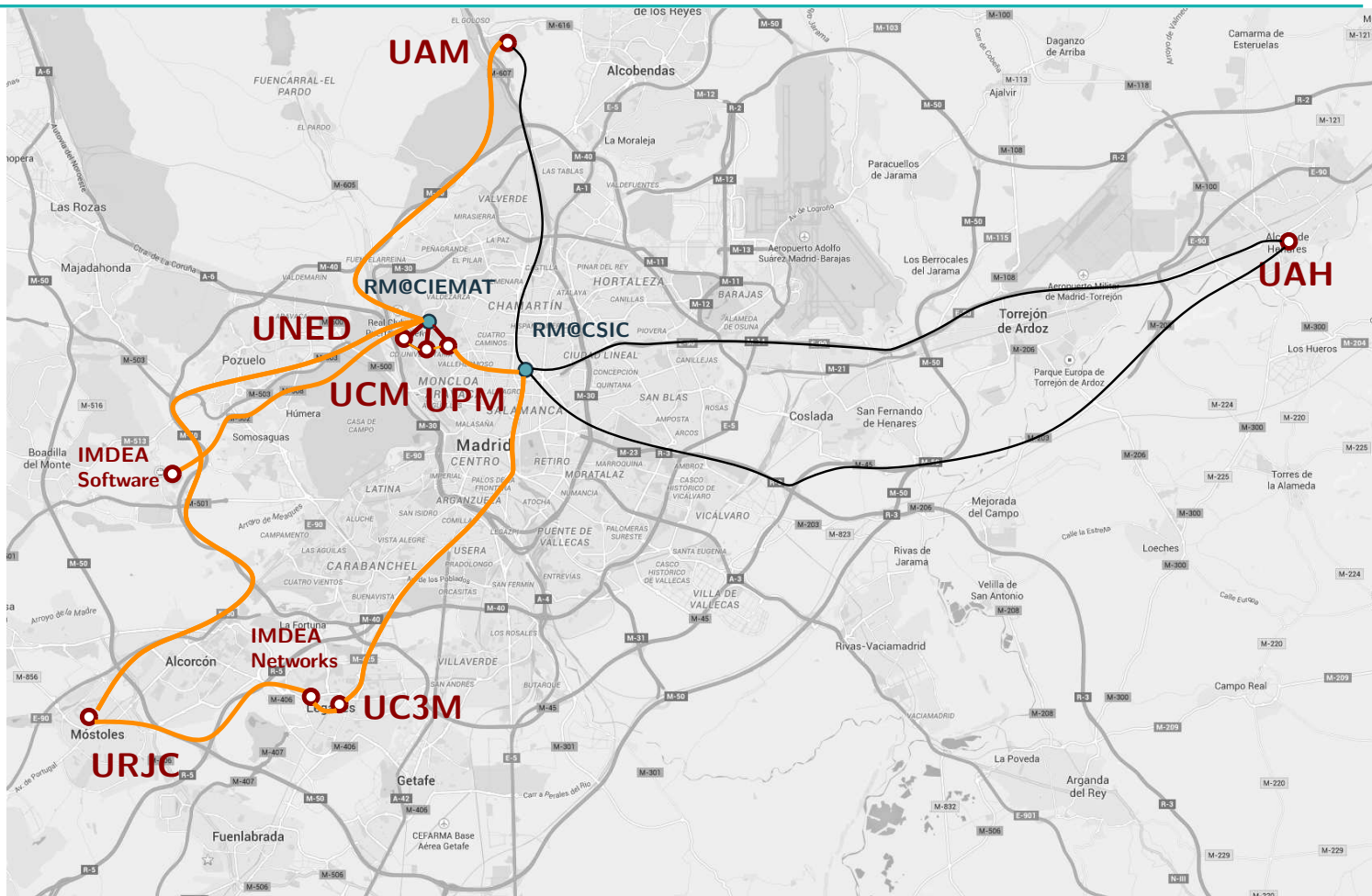
Huella de fibra de RedIRIS-NOVA



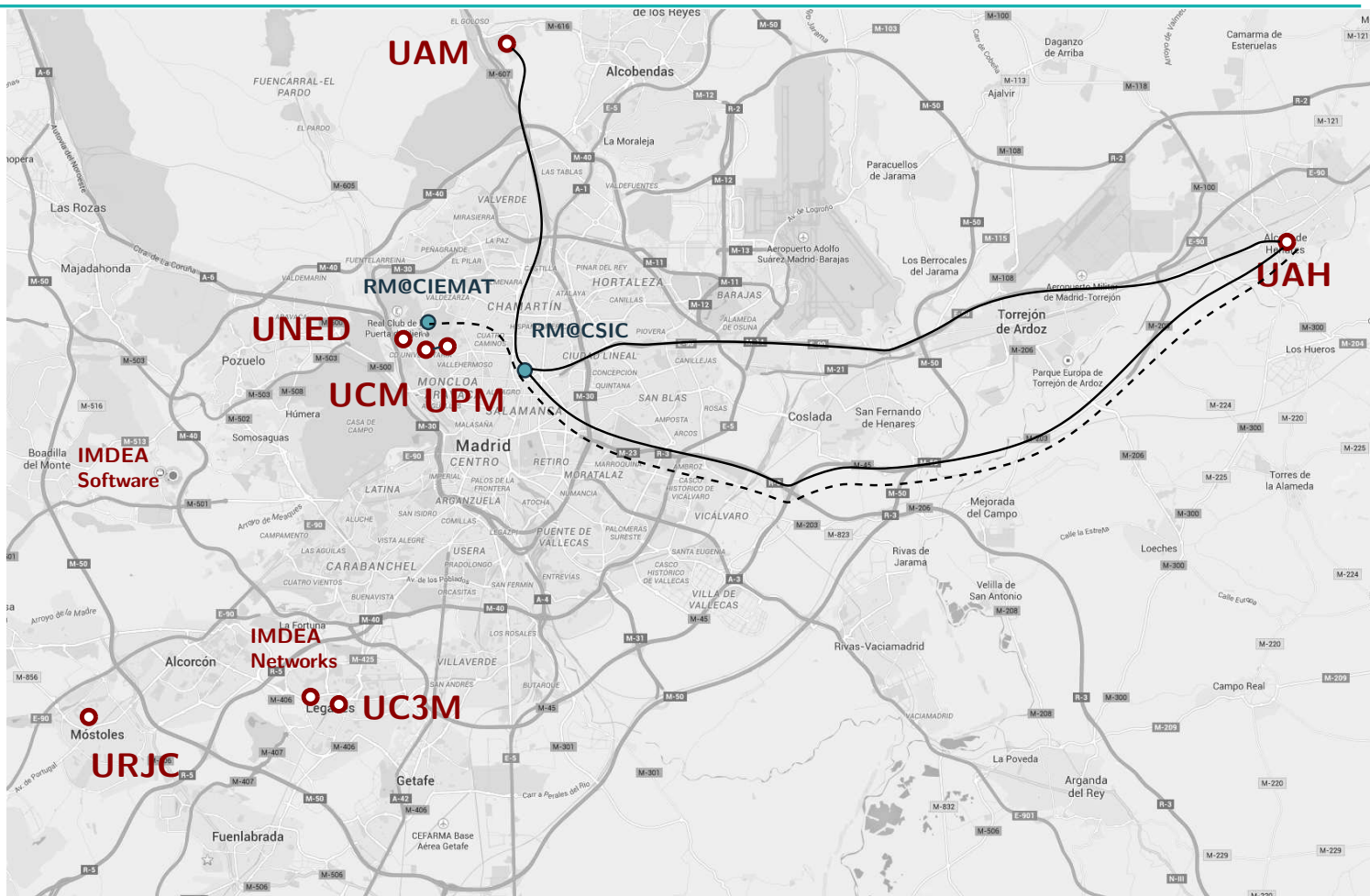
REDIMadrid (la Red de Madrid)



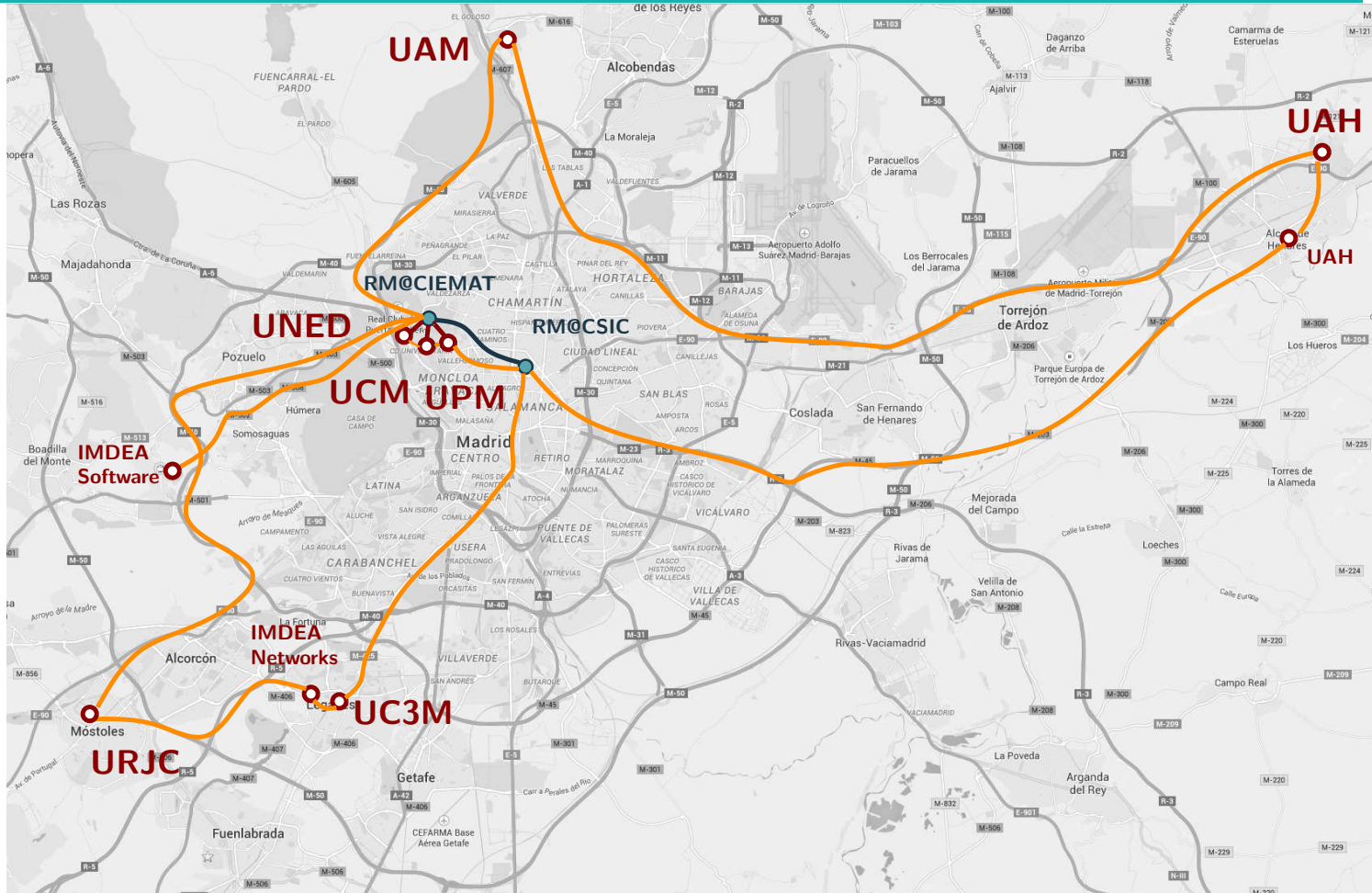
Topología Física Actual CORE (fin 2025)

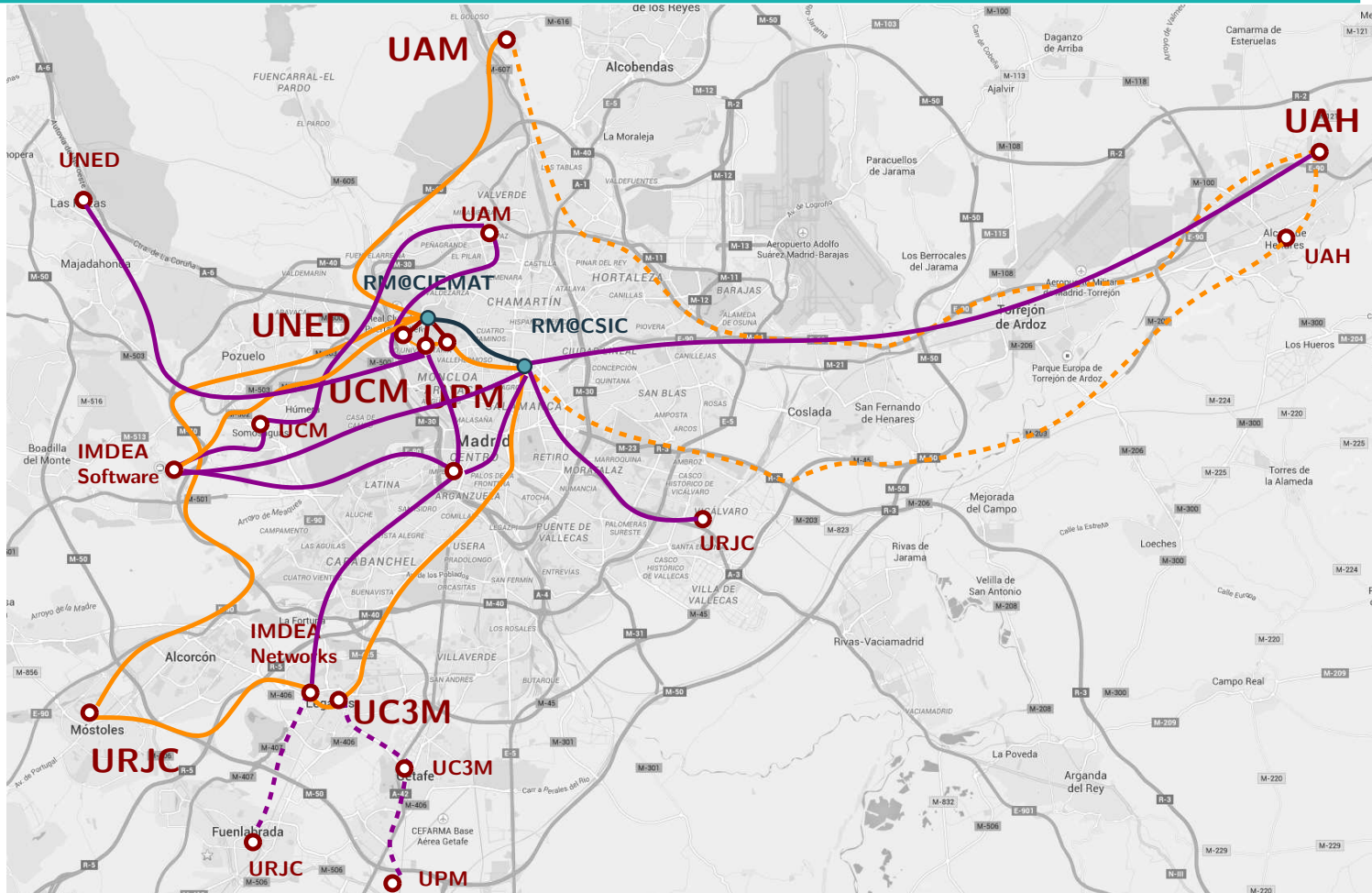


Topología Anillos Alquiler Remanente (2025-)

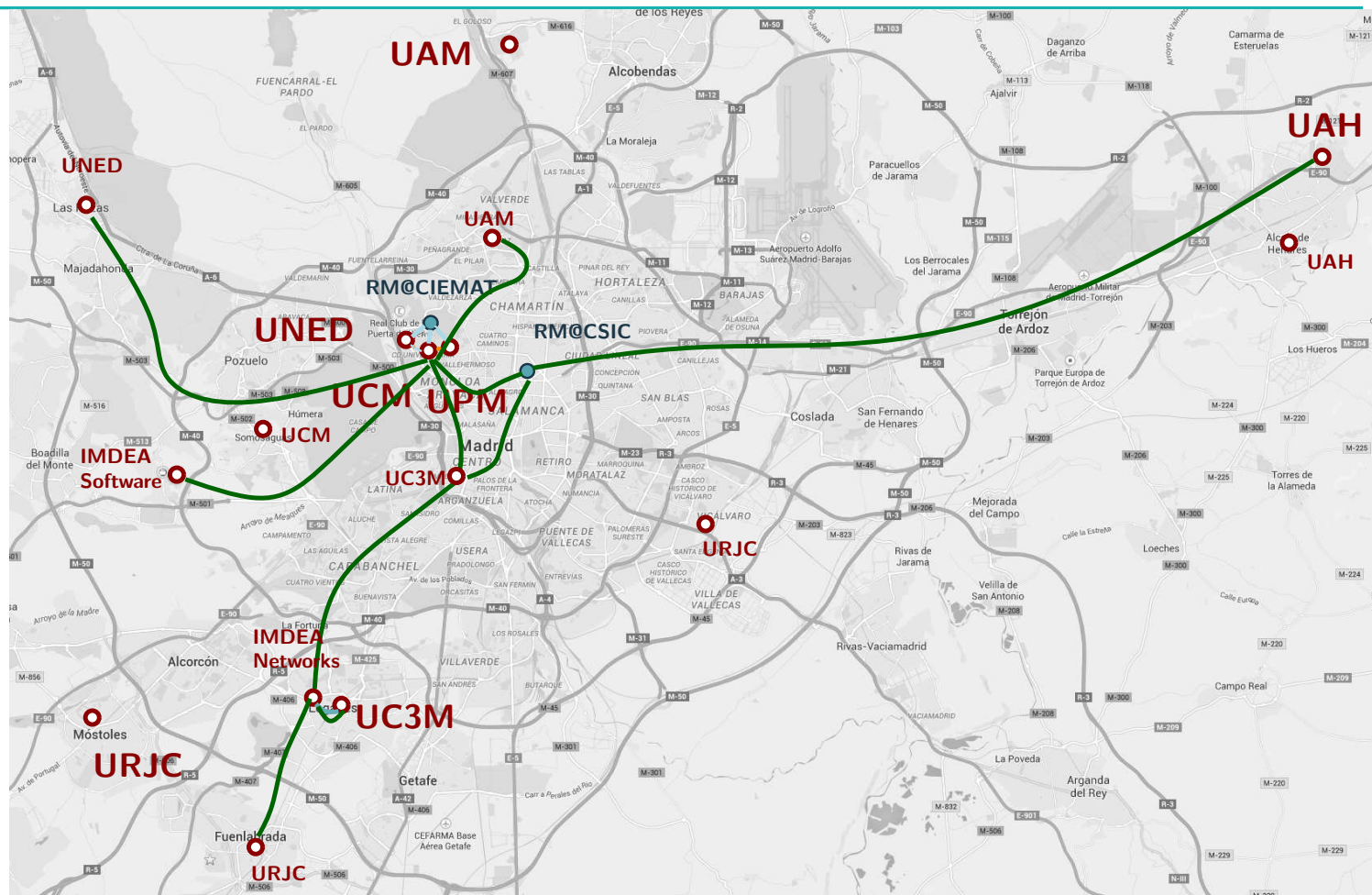


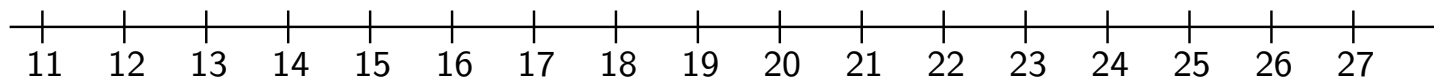
Topología Física Objetivo (CORE)

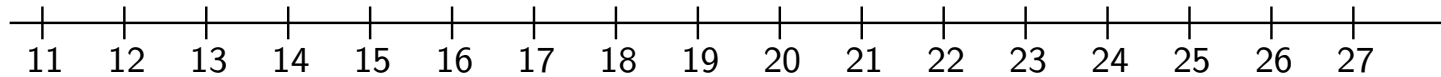




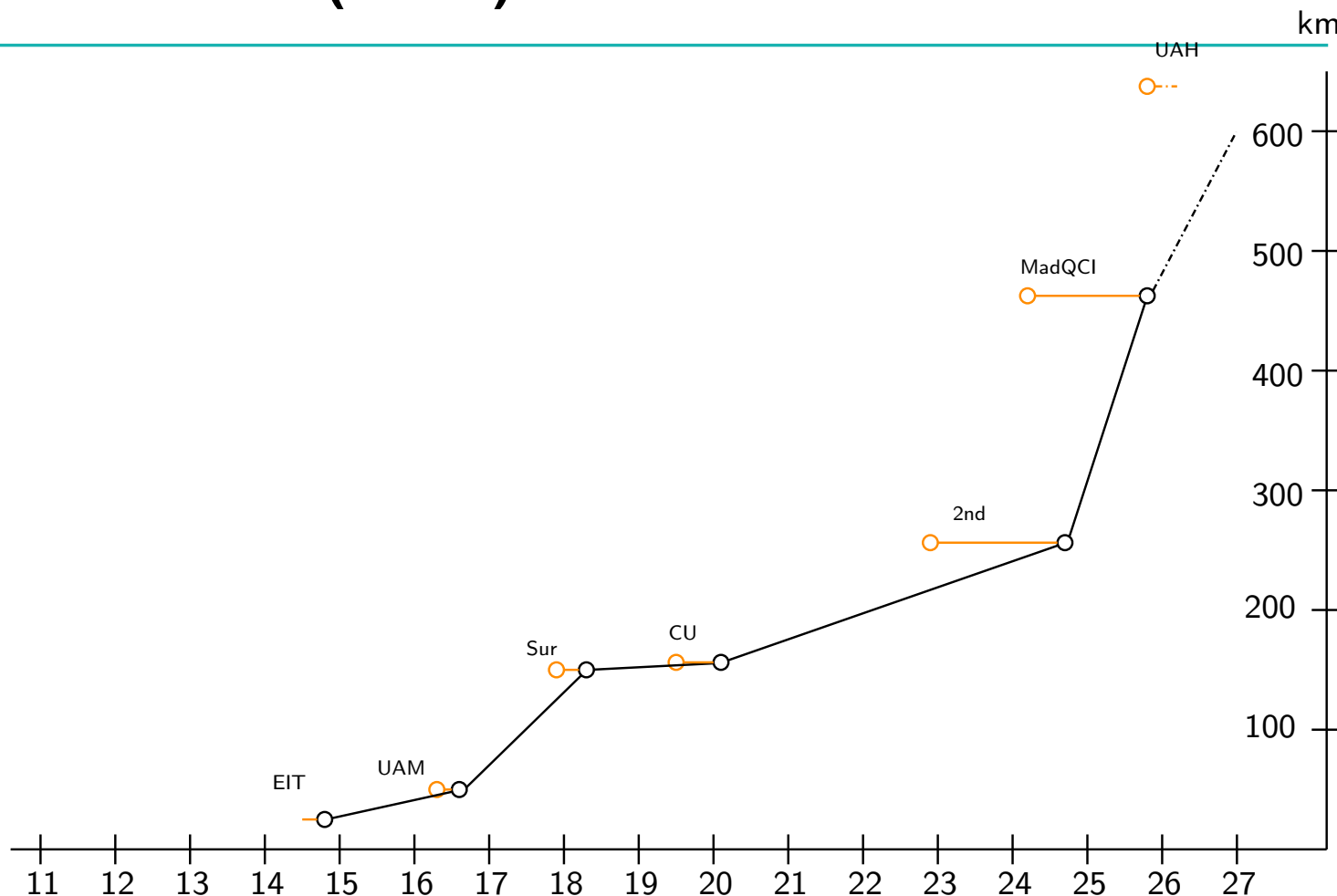
Red Cuántica (MadQCI)



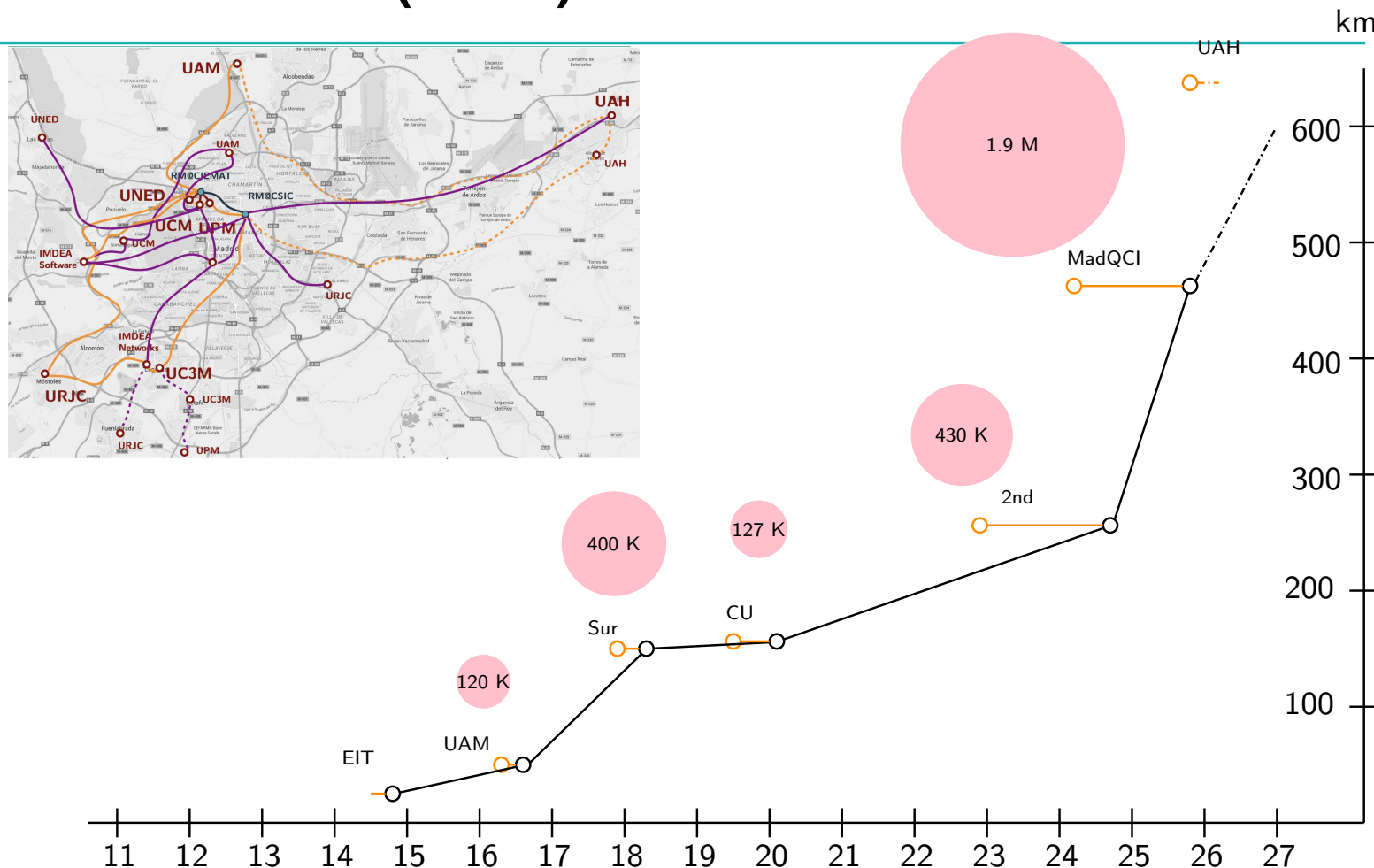




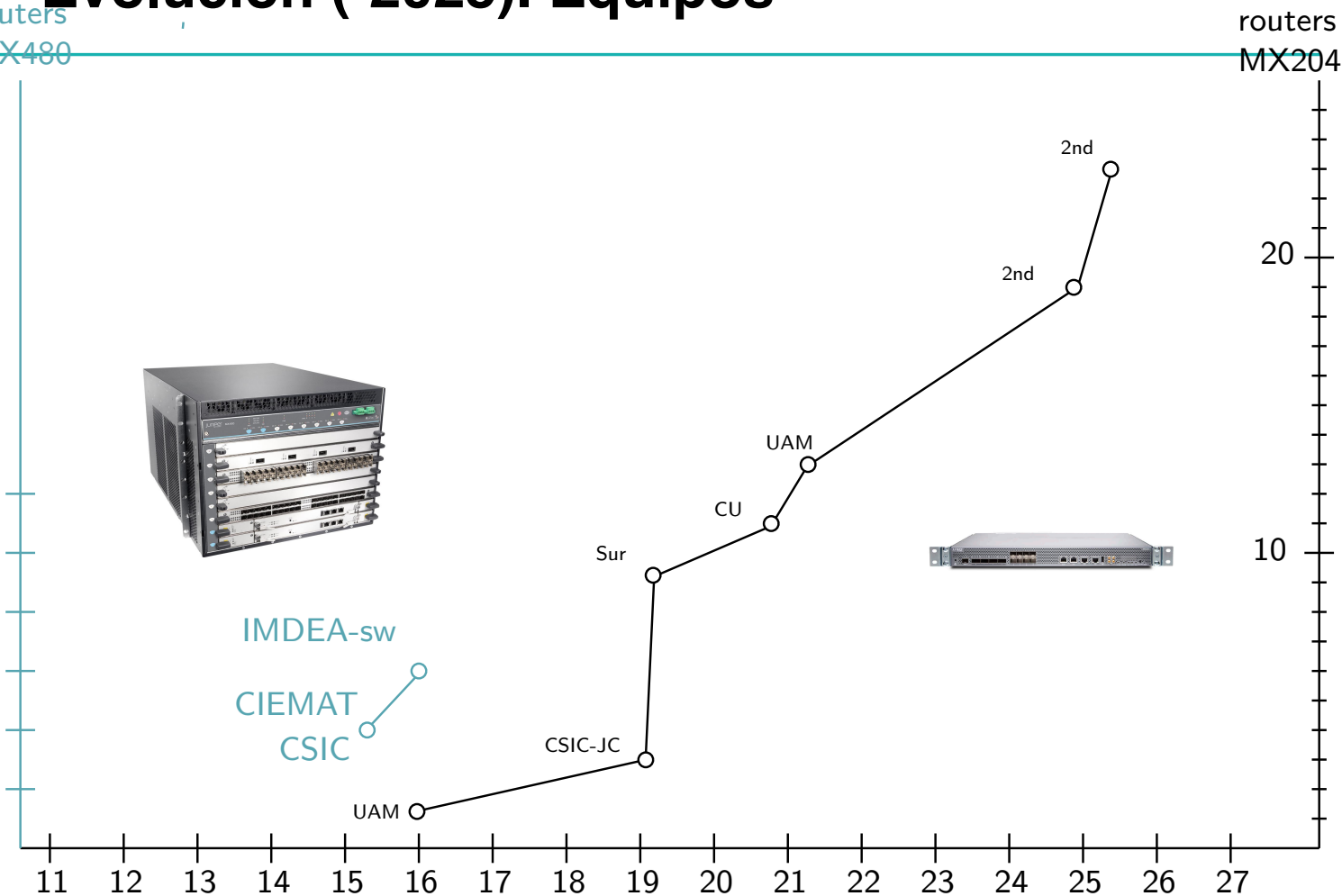
Evolución (-2025). Fibra Oscura



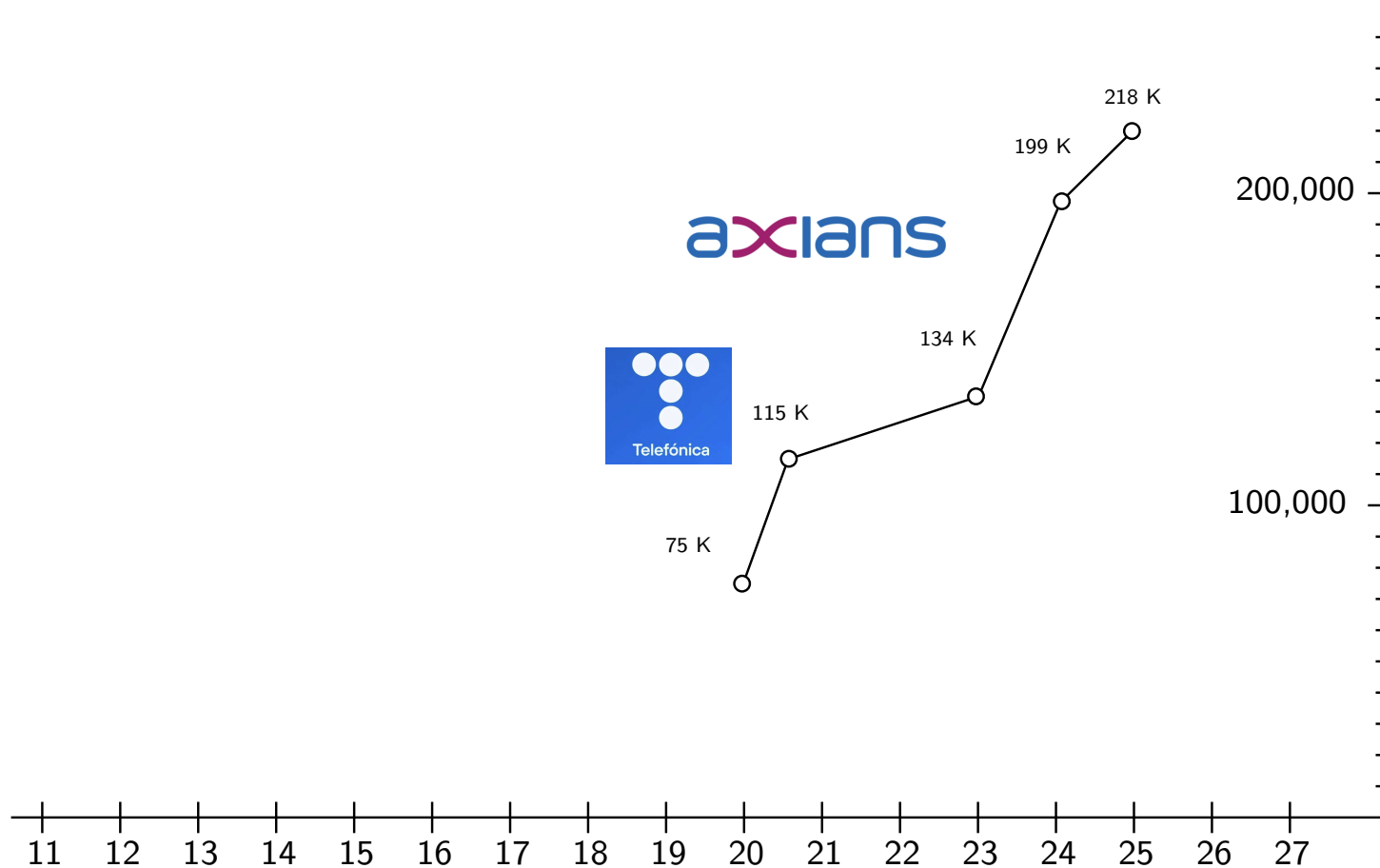
Evolución (-2025). Fibra Oscura



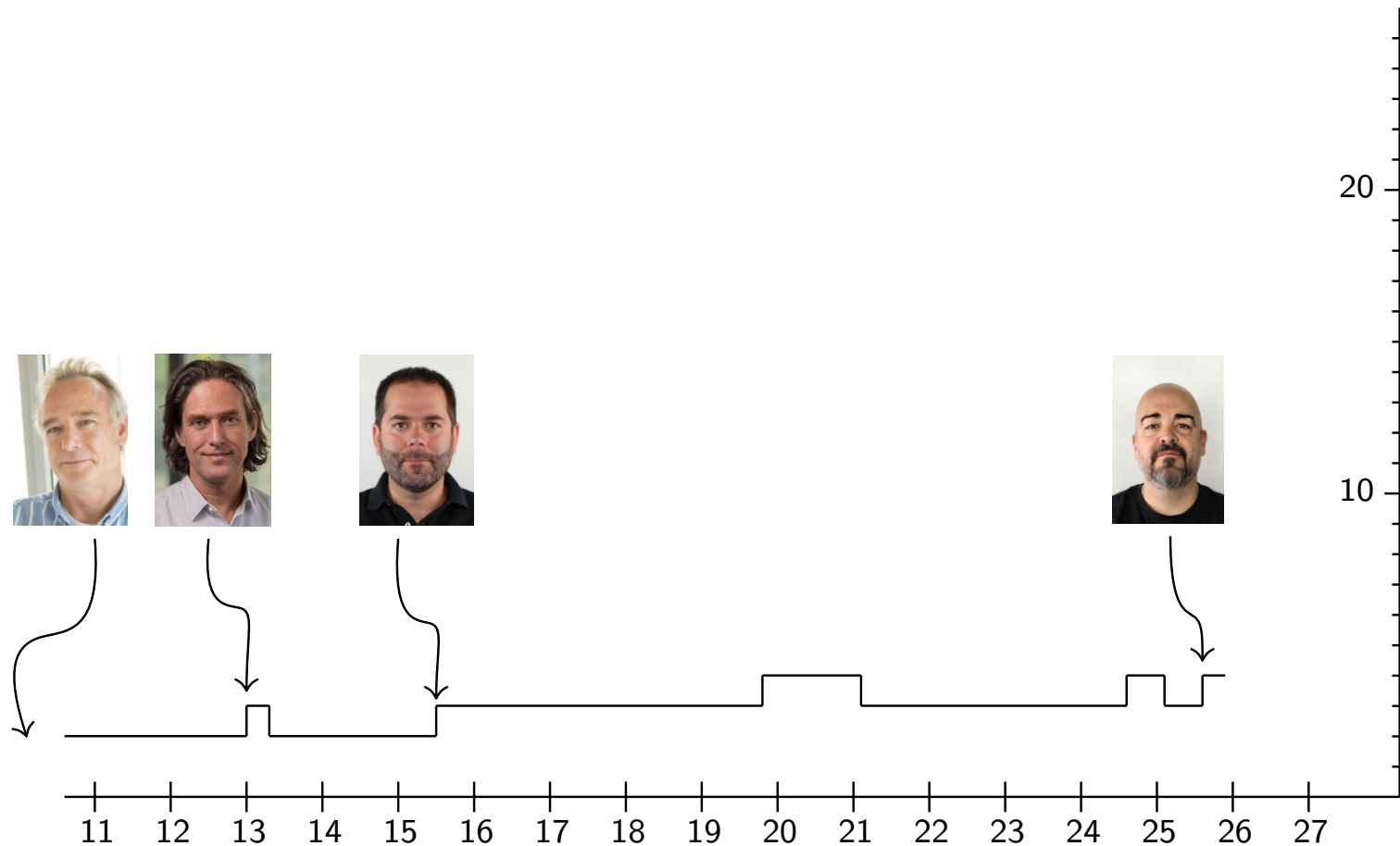
Evolución (-2025). Equipos



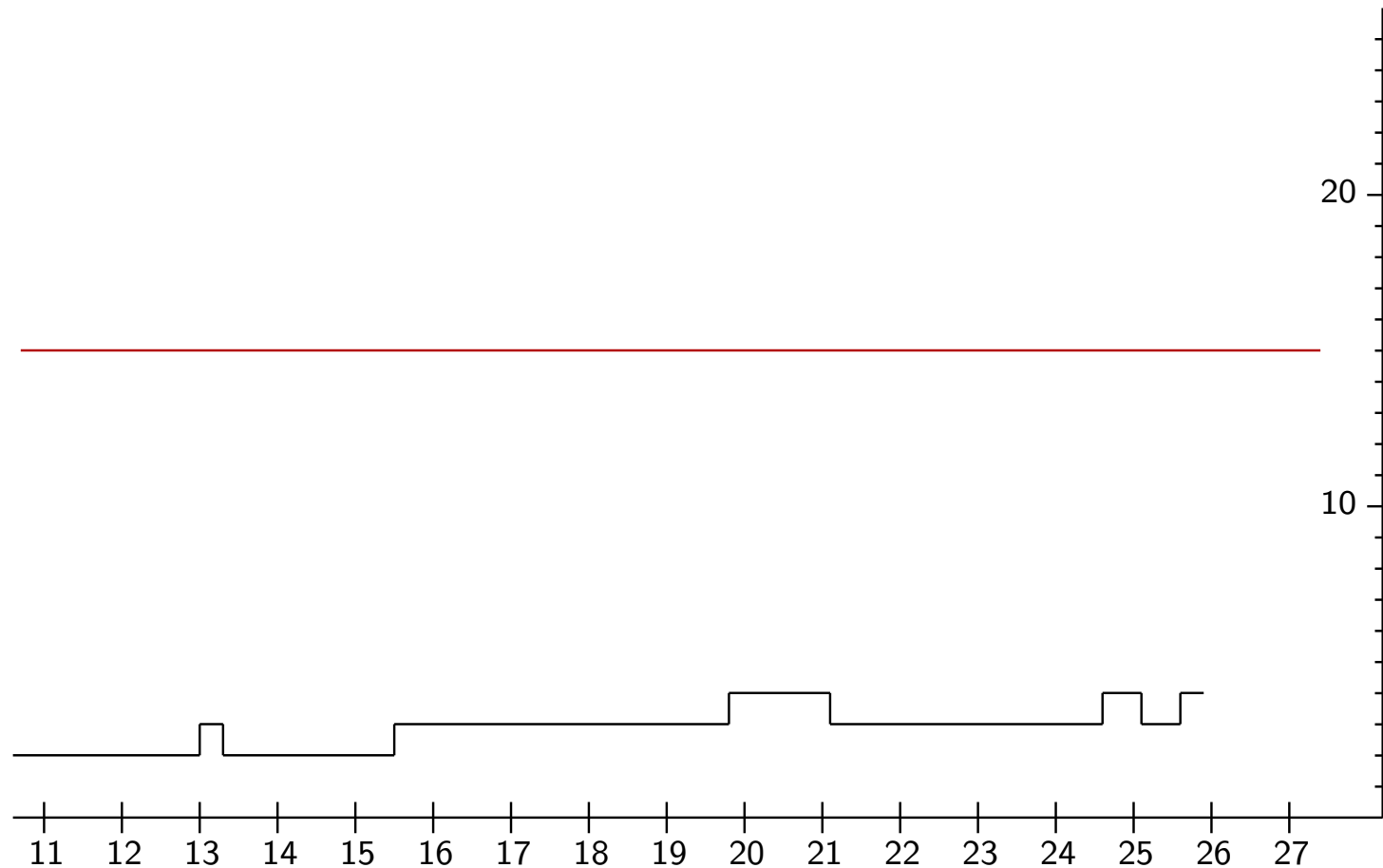
Evolución (-2025). Externalización NOC



Evolución (-2025). Personal



Evolución (-2025). Personal



Fase de despliegue (-2020):

1. Completar la red troncal (UAH) → *en proceso de despliegue (RedIRIS)*

Fase expansión (2024-26):

1. Backbone a 100GB (y transición progresiva a 100GB) → *en proceso (80%)*
2. Nuevas fibras oscuras (expansión) → *en proceso (90%)*
3. Nuevo CPD en IMDEA Software → *operativo*
4. Nueva red de investigación (MadQCI + ...) → *desplegada*

Fase estable (2026-):

1. Alineamiento de IRUs
2. Seguridad
3. Investigación

Fibra oscura REDIMadrid:

Total: \approx 635km:

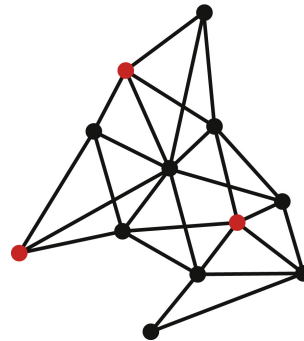


Telefónica

- 355 un par
- 280 dos pares



Correos Telecom



Trafico cursado en REDIMadrid:

Total: \approx Picos de 65Gb/s

Todas las universidades con
posibilidad de conexión a **100G**.

Equipos de comunicaciones:

Router instalados: 20, de los cuales 2 son
P's y 2 son RR.

Número total de equipos ópticos: 9

Usuarios de REDIMadrid:

Número total de usuarios: 354.737

Número total de mujeres: 179.149(50.51%)

AntiDDoS: *Genie networks*



Capacidades Actuales (2025)

Entidades miembros de REDIMadrid	Velocidad de acceso
Conexión de REDIMadrid con RedIRIS en CSIC	100 Gbps (+)
Conexión de REDIMadrid con RedIRIS en CIEMAT	100 Gbps (+)
Conexión de REDIMadrid CSIC con REDIMadrid CIEMAT	200 Gbps
Consejo Superior de investigaciones Científicas (CSIC)	10 Gbps
Instituto CIB (Centro de Investigaciones Biológicas)	10 Gbps
Universidad de Alcalá de Henares (UAH)	10 Gbps / 10Gbps
Universidad Autónoma de Madrid (UAM)	10 Gbps / 10Gbps
LHCONE	10 Gbps (+100GBps)
Universidad Carlos III de Madrid (UC3M)	40 Gbps / 40 Gbps
Universidad Complutense de Madrid (UCM)	40 Gbps/ 40 Gbps
Univesidad de Educación a Distancia (UNED)	10 Gbps/ 10 Gbps
Univesidad Politécnica de Madrid (UPM)	100 Gbps / 100 Gbps / 10 Gbps
Universidad Rey Juan Carlos (URJC)	10 Gbps / 10 Gbps (+)
Instituto IMDEA Software	10 Gbps / 100 Gbps
Instituto IMDEA Networks	10 Gbps / 10 Gbps
Instituto IMDEA Energía	100 Mbps
Instituto IMDEA Nanociencia	1 Gbps
Instituto IMDEA Materiales	100 Mbps
Universidad Europea de Madrid	10 Gbps
Casa de Velázquez	1 Gbps
Universidad San Pablo CEU	10 Gbps
ESIC	1 Gbps
CUNEF	1 Gbps



¿PREGUNTAS?

thank you!

César Sánchez

Cesar.sanchez@imdea.org

software.imdea.org

software