



Development of the Danish LRAIC model for fixed networks

R model manual

[Version after 3rd consultation]

Axon Partners Group

December 2020





Contents

Contents	1
1. Introduction	5
2. Model architecture.....	6
3. Inputs	8
3.1. Address database.....	9
3.2. Road database.....	10
3.3. Coverage database	11
3.4. Location of nodes.....	12
3.4.1 TDC's network nodes	12
3.4.2 Norlys' network nodes	14
4. Geographical analysis	15
4.1. Association between the central offices and the buildings	15
4.2. Location of additional network nodes	17
4.2.1 Copper Access Network	18
4.2.2 Coax Access Network	23
4.2.3 Fibre Access Network.....	25
4.3. Definition of the routes between the nodes	26
4.3.1 Access Networks	27
4.3.2 Transmission and Core Networks	29
4.4. Characterization of the coverage input.....	33
4.4.1 Fibre rollout algorithm for TDC.....	34



4.4.2	Fibre rollout algorithm for Norlys	37
5.	Calculation of the KPIs	41
5.1.	Copper Access Network	41
5.1.1	Step 1. Calculation of homes connected directly to a Central Office	42
5.1.2	Step 2. Calculation of km pairs of copper cable.....	42
5.1.3	Step 3. Calculation of km of fibre cable for copper networks	44
5.1.4	Step 4. Calculation of SCs, DPs and MDFs.....	45
5.1.5	Step 5. Calculation of copper joints	47
5.2.	Fibre Access Network	48
5.2.1	Step 1. Calculation of km of fibre cable	49
5.2.2	Step 2. Calculation of equipment configurations for fibre networks	51
5.2.3	Step 3. Calculation of fibre joints.....	53
5.2.4	Step 4. Calculation of additional ducts	54
5.3.	Coax Access Network	55
5.3.1	Step 1. Calculation of km of coax cable	55
5.3.2	Step 2. Calculation of km of fibre cable for coax networks	56
5.3.3	Step 3. Calculation of equipment for coax networks.....	57
5.4.	Other common civil infrastructure elements	58
5.5.	Core and Transmission networks.....	60
5.5.1	Step 1. Calculation of % of traffic managed per chain.....	61
5.5.2	Step 2. Calculation of trenches of core and transmission networks	61
6.	Network elements	62
6.1.	Levels of disaggregation	62



6.2. Disaggregation of the results into the detailed geotypes	65
Annex A. R model guidelines	69
A.1. Getting started	69
A.2. Understanding the control panel	73
A.2.1. Import control panel	73
A.2.2. Execution control panel	74
A.2.3. Export control panel.....	77
A.2.4. Save control panel.....	78
A.3. Expert access to execution scripts	78
A.3.1. Interface scripts	78
A.3.2. Link scripts.....	79
A.3.3. Backend scripts	80
Annex B. R Model technical documentation.....	82
B.1. Interface block	83
B.2. Block 0: Functions.....	83
B.3. Block 1: Loading Inputs	96
B.4. Block 2: Route generation	96
B.5. Block 3: Sections	100
B.6. Block 4: Node Splitting	100
B.7. Block 5: Buildings	101
B.8. Block 6: KPIs.....	104



B.9. Same script as the 6K_CALC_COVERAGE.R but adapted to the alternative operators fibre model.	Block 7: Export	108
B.10. Block 8: Execution		108
Annex C. List of acronyms		109
Annex D. Change log in the R model.....		110



1. Introduction

Since 2003, the DBA has annually regulated the wholesale prices for several fixed-network services through a Long Run Average Incremental Cost (LRAIC) model. As presented in the Model Reference Paper (hereinafter, 'the MRP') from October 2019¹, the relevant changes that occurred in the fixed Danish market since the last major update of the model in 2013, merited a new update of the fixed LRAIC model (hereinafter, 'the model') to make sure it is representative of the current situation and can fulfil DBA's regulatory needs.

The model developed by DBA incorporates the feedback received from the industry in the three consultations processes that have taken place in the context of the modelling phase of the project throughout the year 2020. This model has been developed following the methodological principles laid out in the MRP from October 2019, which was subject to consultation with the industry between 1st July to 30th August 2019.

In the context of the main project, two different Fixed LRAIC model have been developed: One model based on TDC and one model based on Norlys (combining Eniig and SE). Each Fixed LRAIC model consists in two parts:

- ▶ **R model:** Used to perform a geographical assessment of the access and transmission networks in Denmark.
- ▶ **Excel model:** Used to calculate services' costs, following a set of inputs and calculations. In this case, while the calculations considered for both, TDC and Norlys are identical, the inputs included in the model differ, as each is based on the data that operators have been able to provide.

This document represents the R model manual. This manual aims to describe the modelling approach, structure and calculation process followed in the development of the R model. 'Annex A - R model guidelines' presents a simple set of guidelines to follow when using the R model And 'Annex B - R Model technical documentation' presents an explanation to each one of the scripts provided in the R model.

¹ Link: <https://erhvervsstyrelsen.dk/sites/default/files/2019-10/Final%20MRP.pdf>



2. Model architecture

The architecture of the R model is divided into blocks, to ease its conceptualisation. A high-level diagram of the blocks considered in the model and their relationship is presented in the figure below.

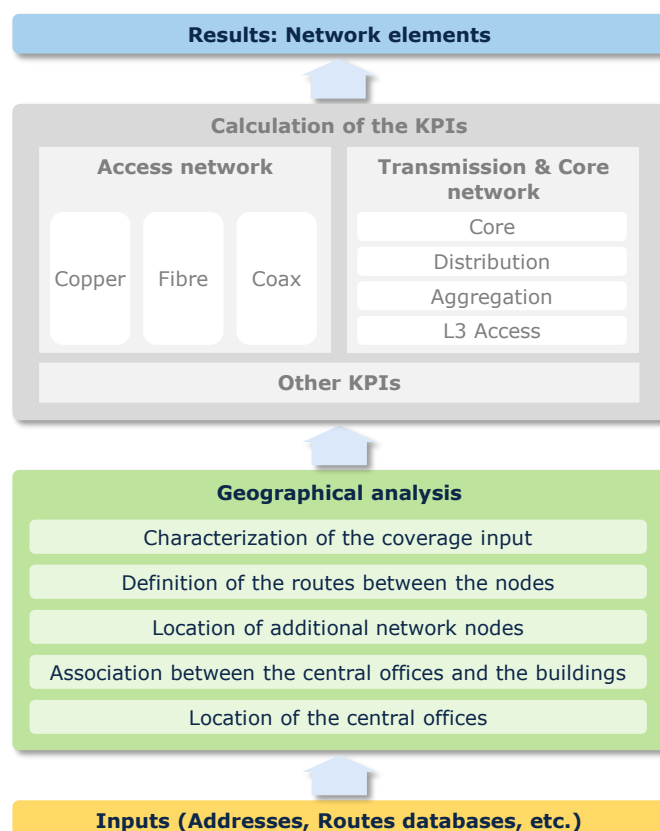


Exhibit 2.1: Structure of the R model [Source: Axon]

As seen in this figure, several main function blocks are relevant to the model. These are:

- ▶ **Inputs:** These inputs represent the databases (e.g. addresses) and all the network information (coverage and regulated areas) that the model uses in to perform its calculations (section '3 - Inputs').
- ▶ **Geographical analysis:** Analysis of the geographical characteristics of Denmark, based on the road network, the existing building and access nodes of the modelled operators.



- ▶ **Calculation of the KPIs** (number of network elements): for each one of the networks. These calculations include copper, coax, fibre, and the transmission/core networks.
- ▶ **Network elements:** Generating the final result tables to be exported to the Excel model.

Each of these modules is presented with a high level of detail in the sections below.



3. Inputs

The geographical analysis developed is data-intensive and has been populated with the information requested from the operators from multiple data requests processes that have taken place in the context of this project, as well as additional publicly available information and other data available to DBA.

For the development of geographical analysis, several databases have been used. The main files used for the geographical analysis are:





Data	Description	Source	Link
Address database	CSV file collecting all the addresses in Denmark	 Agency for Data Supply and Efficiency	https://eng.sdfе.dk/product-and-services/the-danish-address-register/
Road database	GIS file containing all the information related to the national road network	 Agency for Data Supply and Efficiency	https://eng.sdfе.dk/about-us/organisation/geodata/
Coverage database	A file containing the coverage of each address.	 DANISH BUSINESS AUTHORITY TDC NORLYS	Not publicly available
Node location	Location of the main nodes of the access and transmission network	 TDC NORLYS	Not publicly available

Exhibit 3.1 Databases and sources used in the geographical analysis [Source: Axon]

The starting databases available represent the main inputs of the geographical analysis and, thus their content will have a direct impact on the network's characterisation and model results.

Hence, the first step of the geographical analysis is related to the verification of the legitimacy and robustness of the source databases. For this, a data cleaning process has been set to:

- ▶ Adjust inaccurate information
- ▶ Remove unnecessary information



- ▶ Aggregate or disaggregate data depending on the needs of the analysis

presented procedure to clean the databases has been applied to the following four databases:

- ▶ Address database
- ▶ Road database
- ▶ Coverage database
- ▶ Location of nodes

The following sections further develop the content and activities performed in each of the databases considered in the analysis.

3.1. Address database

The address database contains a list of all the Danish households, giving mainly the detail of:

- ▶ Address (including Street, Street number and Zip code)
- ▶ Coordinates (in ETRS89/UTM zone 32N format)
- ▶ AAID (the unique identifier of the address access/Building)

This information is essential to extract relevant information about the buildings in Denmark, which is used to evaluate the footprint of the modelled operator.

First, in order to ensure the robustness of the database, several checks were performed to test the data. These checks included a visual analysis of the projection of the data on a map to verify that a sample of those addresses was consistent with the location of buildings according to satellite imagery and ensuring that all addresses fell inside Denmark borders.

Based on these checks, we observed that the database provided was clean and ready to be used for analysis purposes. Thus, no adjustments were required.

As a final step, given that the database is presented at the household level, the information is aggregated at building level (this is, households within the same building are aggregated together). This aggregation is done based on the AAID field of the database. It should be



noted that information regarding the number of households in each building is stored in this process.

3.2. Road database

The road database contains a list of all the roads in Denmark. This information is relevant as often fixed telecom networks are deployed along with road networks.

The roads are disaggregated in sections, which represent the roads between two intersections.

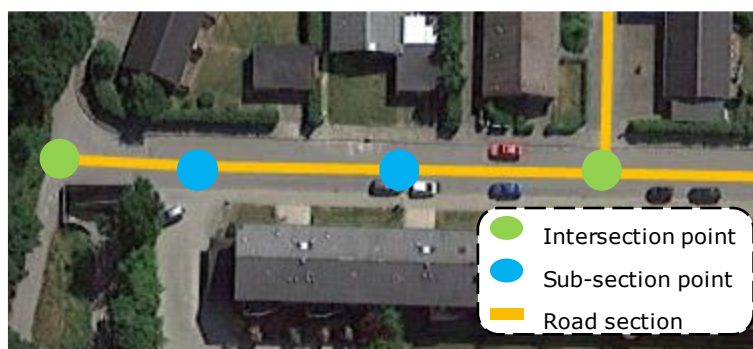


Exhibit 3.2: Extract of the sections available in the roads database [Source: Axon]

The distance between two intersections is calculated as the sum of the distances of all subsections between them.

In order to validate this database, we performed a number of checks, including:

- ▶ Visual analysis of the projection of the data on a map, to verify that a sample of the roads included was consistent with the roads and patch observed in satellite imagery and ensuring that all the database was comprehensive of all of the relevant roads in Denmark.
- ▶ Quantitative analysis of the distance between a sample of points between intersections within the country, in order to evaluate potential missing roads.

Based on these checks, we identified that a set of additional roads had to be included in the dataset. This aspect is key to ensure the connectivity of the road network. With this objective, we added “virtual” sections to the database. This has been ensured through the development of an algorithm that verifies the connectivity of all the sections. Additionally,



a check has been performed to ensure the reasonability of the added virtual sections. 2.642 virtual sections have been included in the database as a part of the data cleaning process (these represent 0,19% of the total sections in the database).

Also, we added to the road database the sections related to submarine cables that were not initially included in the database. We added 31 new sections as a result of this adjustment.

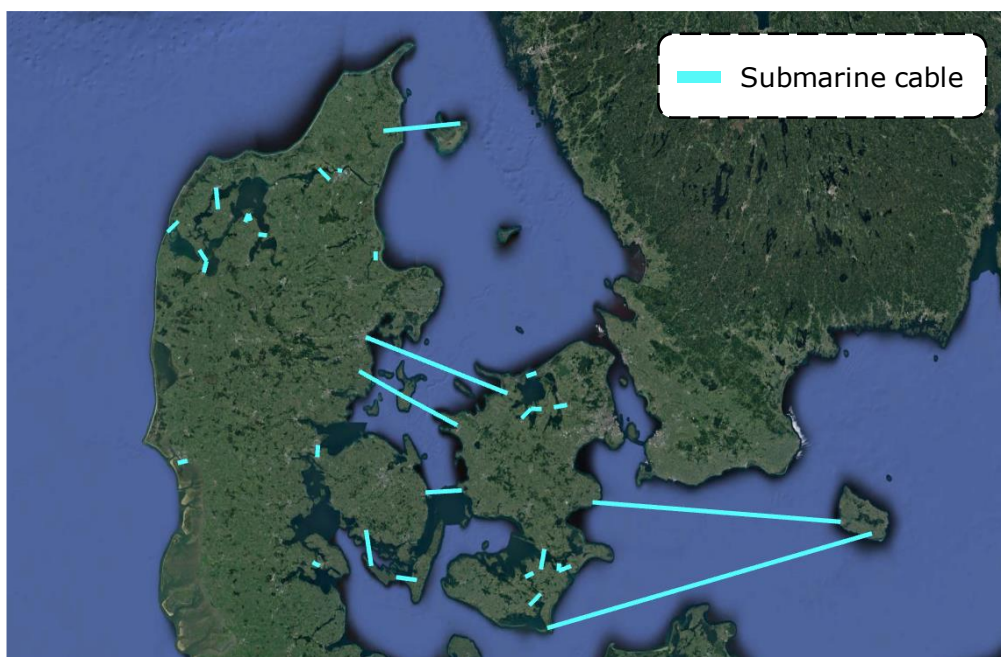


Exhibit 3.3: Addition of the submarine cables [Source: Axon]

Once we ensured the robustness and completeness of the database, we proceeded to consolidate the database, summarising for each section the coordinates of the corresponding intersections and the total distance.

3.3. Coverage database

DBA's coverage database incorporates information about the coverage at household level for copper, fibre, and coaxial technology for the different operators in Denmark. This information is important for the dimensioning of the network, as it is one of the main inputs along with the demand.



Knowing the importance of this database, we proceeded to evaluate its robustness performing a number of preliminary checks:

- ▶ Ascertain that all the addresses included in this database were included in the addresses database
- ▶ Identification of duplicated addresses
- ▶ Calculation of coverage percentages at a national level in order to check if those percentages match with the public data

These checks allowed to verify that the coverage database included robust data ready for analysis purposes. As a result, this information has only been used for copper and coax networks, but not for fibre, due to some inconsistencies with operator data in this case.

Further calculations and algorithms have been performed in order to determine the actual and future coverage for the modelled operators for each technology, please see section 4.4 Characterization of the coverage input for further details.

3.4. Location of nodes

As presented in the MRP, the model follows a modified scorched-node architecture, based on the actual network of the modelled operator. Thus, one of the most important datasets for the definition of the geographical analysis is the location of the network nodes.

During the data request process, we received:

- ▶ A database containing TDC's network nodes.
- ▶ A database containing Norlys' network nodes.

Further details on the data received from each operator are presented below.

3.4.1 TDC's network nodes

The database has information about some of the main nodes of the access network for each technology. The detail of the information provided for each technology is presented below:

- ▶ For the Copper Access Network, TDC's database included the location of:



- The Central Offices (COs)
 - The Main Distribution Frames (MDFs)
 - The Primary Distribution Points (PDPs)
- For fibre access network, based on the information provided by the modelled operator we observed that fibre MSAN (for a PTP architecture) were deployed in the same locations as the copper MDF. Thus, a consistent architecture to the one used for copper networks has been considered. In the case of the fibre OLTs (for a PON architecture) we observe that these are deployed typically in the same location of the copper COs, and thus, the same architecture has been maintained compared to copper.
- For the Coax Access Network, TDC's database included the location of:
- The locations of the coax OLTs, equivalent to those of the Central Offices of the copper network.
 - The locations of the CMCs.

In addition, TDC reported GIS files regarding the areas covered by each of the Central Offices in the network. A sample of these areas are presented in the exhibit below:

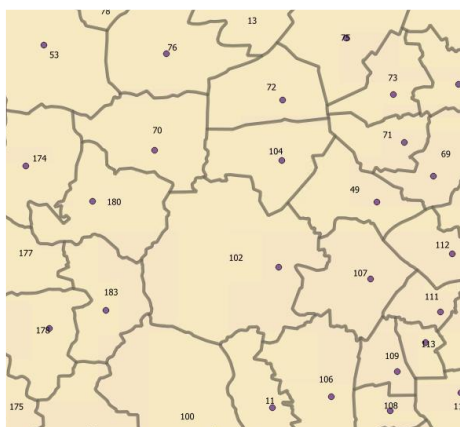


Exhibit 3.4: Snapshot of TDC's Central Office areas in Denmark [Source: Axon]

This information will be used during the determination of the routes in section '4.3 - Definition of the routes between the nodes'. To ensure the robustness of this data, we also proceeded to verify that there was no overlapping between the different areas. In these cases, we reassigned the nodes in the overlapping area to the area of its closest CO.

Finally, the database provided the location of the following transmission network nodes:



- ▶ L3 Access nodes
- ▶ Aggregation nodes
- ▶ Distribution nodes
- ▶ Core nodes

To ensure the robustness of the database, we performed a visual analysis of the projection of the data on a map, to verify that the nodes included were inside the boundaries of the Danish Territory. This check revealed that the information of all the nodes was correct.

3.4.2 Norlys' network nodes

The database has information about some of the main nodes of the network. The detail of the information provided is presented below:

- ▶ For the fibre access network, Norlys has provided the addresses of the locations where OLTs have been deployed. Actual coordinates were not reported as a part of this process, thus we had to determine the actual location of the OLTs from address data. The locations defined for the OLTs have been assumed to be equivalent to the locations of the Central Offices.

In this case, Norlys did not report the boundaries of the areas covered by each OLT, therefore, to determine the homes that are connected to each OLT, we have defined an algorithm based on the concept of the Voronoi diagram (please see section '4.4.Characterization of the coverage input' for further details).

- ▶ For transmission and core networks, the database provided the location of the following nodes:
 - L3 Access nodes
 - Aggregation nodes
 - Core nodes

This information will be used during the determination of the routes in section '4.3 - Definition of the routes between the nodes



4. Geographical analysis

Considering the Central office data provided by operators and explained in previous sections the geographical analysis is performed in the following steps, which have been split according to their nature:

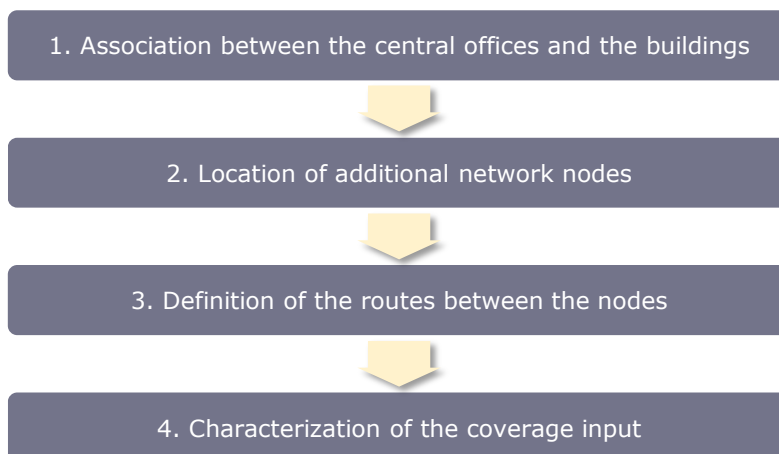


Exhibit 4.1 Steps of the geographical analysis performed [Source: Axon]

The geographical analysis is performed in the R² Model. The detail of the activities carried out in each step is explained below.

4.1. Association between the central offices and the buildings

The association between buildings and nodes has been performed on two different ways depending on the data provided by operators.

For the TDC model, the association between buildings and COs has been performed based on the CO area geometries (actual area covered by the CO) provided by the operator. Therefore, any buildings inside the area covered by each CO have been associated to the CO in the model. Additionally, buildings have been associated to MDFs and PDPs based on the minimum distance route inside the CO geometries.

² Link: <https://www.r-project.org/about.html>



For the Norlys Model the associated buildings have been based on a Voronoi algorithm. This algorithm is explained below:

A Voronoi diagram is a partition of a map into several regions defined through all the closest points to a node included in a set of nodes. For each node there is a corresponding region consisting of all points of the map closer to that node than to any other. These regions are called Voronoi cells.

The algorithm implemented in the R model is an adjusted version of the standard Voronoi, due to the consideration of the road distance instead of a standard metric. The usage of direct distance was avoided as it produced non-reliable results. The algorithm is implemented in script '2D_CALC_AREA.R' (please see further details in the implementation of the scripts in 'Annex B- R Model technical documentation').

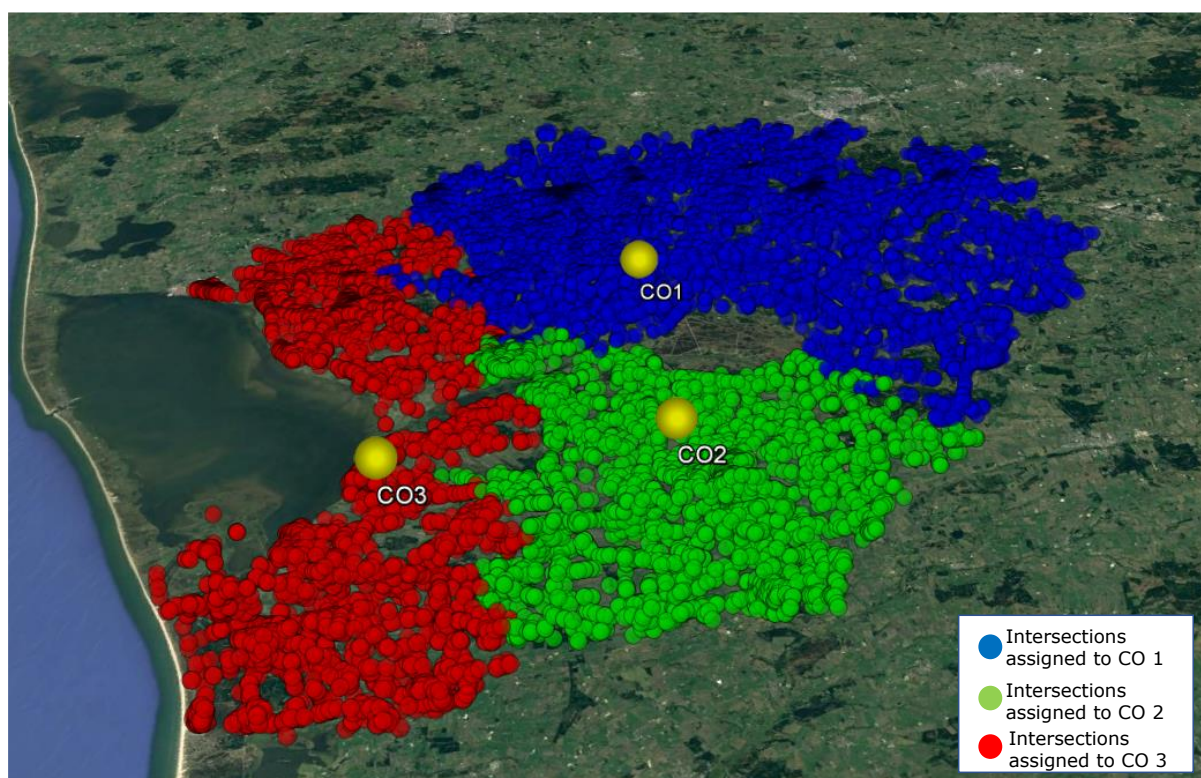


Exhibit 4.2: Illustration of the use of Voronoi diagram [Source: Axon]



4.2. Location of additional network nodes

The objective of the first step of the geographical analysis is the determination of network nodes locations. This information is relevant for the calculation of the routes using a 'shortest path algorithm' between the nodes explained in section '4.3 - Definition of the routes between the nodes'.

As presented above, the locations of the network nodes are primarily based on data from the modelled operator, following a modified scorched node approach (i.e. the locations of the COs are fixed, based on operator data).

Each access network element is defined to have its network node. A high-level overview of the hierarchy of the access nodes for each network is presented in the exhibit below:

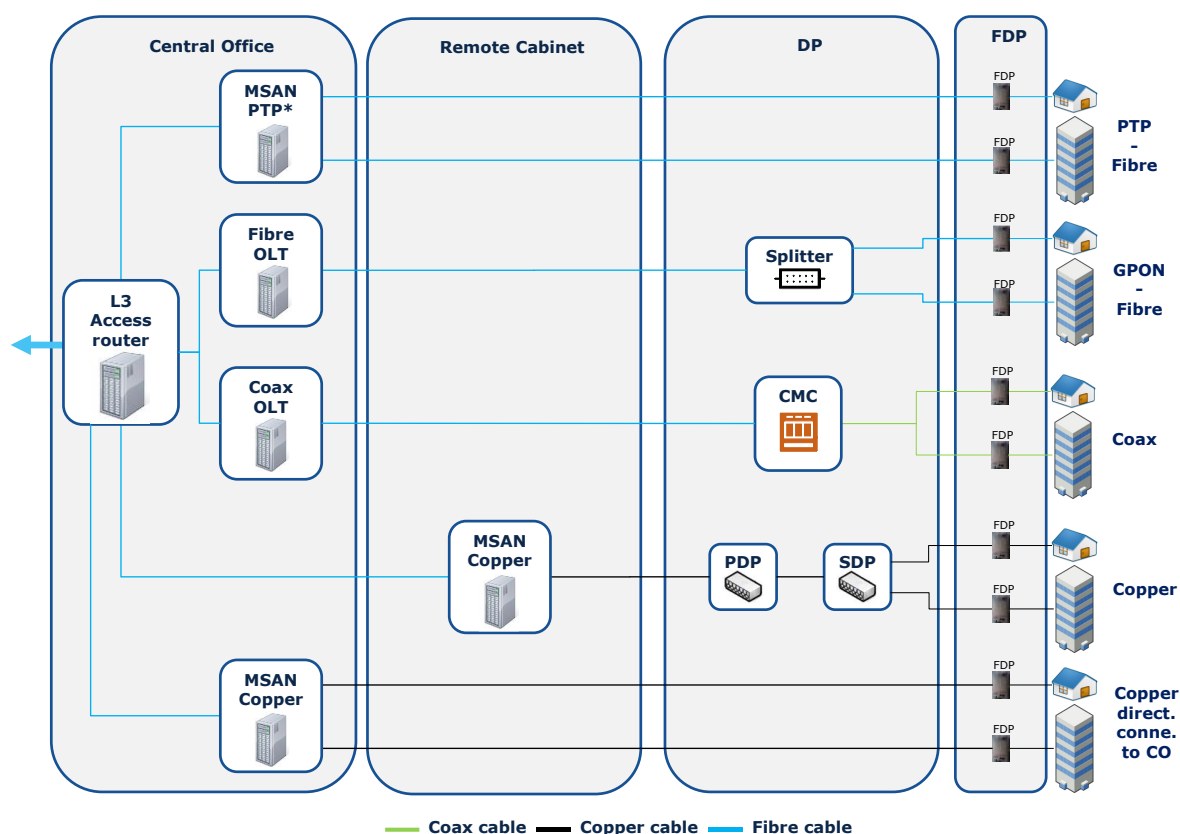


Exhibit 4.3: Architecture of the Danish Network [Source: Axon] Note(*): Fibre MSAN in the case of the TDC's architecture are located in the remote cabinet

The particular description of the network nodes considered under each access network is detailed in the sections below.



4.2.1 Copper Access Network

The copper network is mainly composed of the following nodes:

- ▶ **Central Office (CO)**, which represents the last concentration node of the copper access network, establishing the boundaries between the access and the transmission network.
- ▶ **Main Distribution Frame (MDF)**, which can be in CO nodes or remote.
- ▶ **Primary Distribution Point (PDP)**, node that aggregates several SDPs. It is formed by larger distribution point as a second aggregation node.
- ▶ **Secondary Distribution Point (SDP)**, that represent the first aggregation node of premises in the copper network
- ▶ **Final Distribution Point (FDP)**, located just outside the end user's premises

The exhibit below shows the architecture of the copper access network in Denmark.

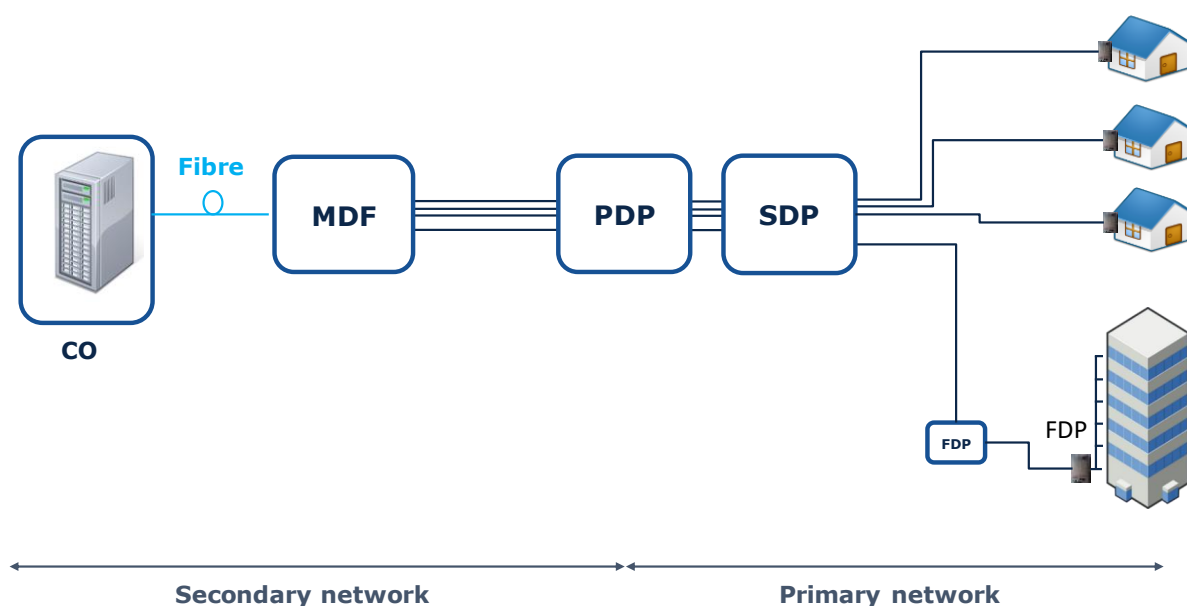


Exhibit 4.4: Architecture of the access copper network [Source: Axon]

The database available outlined the information regarding the location of MDFs and PDPs. However, the location of the FDPs and SDPs is calculated on a bottom-up basis, as presented in the MRP.



Thus, there are a number of tasks to complete in order to fully characterize the copper access network:

- ▶ Definition of the locations of the FDPs
- ▶ Association of the homes to the corresponding PDPs
- ▶ Definition of the locations of the SDPs

Each of these tasks is presented below.

Definition of the locations of the FDPs

FDPs represent the final point of the network, located just outside the end user's premises.



Exhibit 4.5: Location of FDPs [Source: Axon]

For the calculation of FDP's locations, two main steps were followed.

- ▶ Step 1: Identification of the closest section
- ▶ Step 2: Calculation of FDP locations

Each of these steps is described below



Step 1. Identification of the closest section

First, all the sections close to the building are identified as a combination of two intersections. Then, we proceed to the calculation of the distance between the building and the intersections points of each section, selecting the section with the minimum distance, which is chosen as the closest section.

Step 2. Calculation of FDP locations

Finally, it is assumed that the FDP will be located in the same road section as the buildings it connects (as an orthogonal projection of the building towards the section). As a consequence, the “FDP-building” path does not require the shortest path calculation to be performed but rather an assumption on the distance of this link.



Exhibit 4.6: Projections of the buildings on the road sections [Source: Axon]

Association of the homes to the corresponding PDPs

One of the key areas to characterize the copper access network is to determine the area covered by each primary distribution point (PDP).

To do so, the following steps have been applied:

1. Identification of all the buildings and PDP inside the same CO coverage areas. This is based on the boundaries provided by the reference operator.



2. Each building inside a CO area is associated to its closest PDP through the implementation of the shortest algorithm, applied over the roads of Denmark (please see section '4.3.1 - Access Networks' for more details in this matter). As shown in the exhibit below, this allowed us to identify the buildings covered by each PDP, defining a coverage area for the PDPs.



Exhibit 4.7: Example of PDP areas, where each colour represents a different PDP area [Source: Axon]

The output of this process has been mainly used in the definition of the routes between the nodes, facilitating the determination of optimal routes.

Definition of the locations of the SDPs

Once FDPs locations are known we proceeded to the determination of SDP locations. These SDPs aggregate access lines together in order to minimise the costs of deployment.

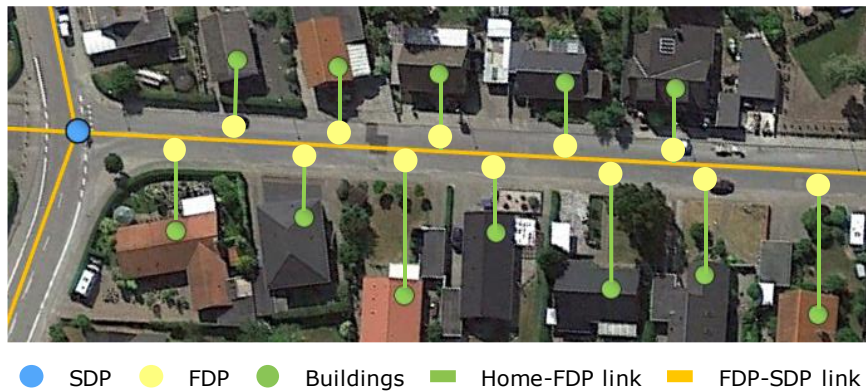


Exhibit 4.8: Example of SDP location [Source: Axon]

In order to determine the location of the SDP, we consider that all SDPs are located in road intersection. Therefore, this section aims to present the hypothesis and calculations performed in order to determine which intersections need to have an SDP.

- ▶ Step 1: Calculation of the number of homes that require crossing each intersection to reach the corresponding PDP.
- ▶ Step 2: For each intersection, we observe if more than 7 homes require it to reach the PDP. Then, for one of these intersections with more than 7 homes associated, an SDP is placed. The minimum number of 7 homes comes from the fact that when designing the applied algorithm, we identified that TDC aggregated roughly 20 homes per SDP on average. So, in order to reflect such scenario, we determined that the minimum number of households per SDP shall be set at 7. Thus, the minimum constraint of 7 is aimed at reflecting the number of SDPs deployed by TDC.
- ▶ Step 3. We recalculate the number of homes that require crossing each intersection, removing those that are already aggregated by the SDP set in Step 2. This process is repeated until there are no more intersections with more routes than the threshold of 7 homes.
- ▶ Step 4. Finally, we identify the routes that have not been assigned to an SDP during this process, and we install an SDP in the COs managing those routes.

As shown in the exhibit below, the result is a list of intersections that include an SDP. Those intersections represent the SDP locations.

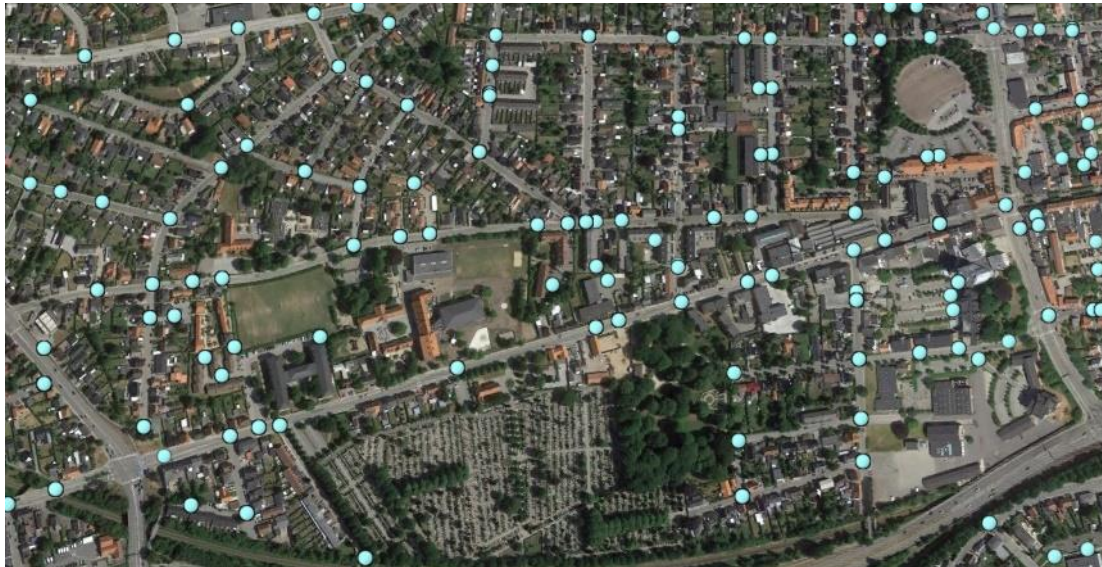


Exhibit 4.9: Illustrative example of SDP locations over a given area [Source: Axon]

4.2.2 Coax Access Network

The coax access network is made of three main nodes:

- ▶ Coaxial OLT, which are installed in CO nodes
- ▶ CMC, Aggregation node of the coaxial network
- ▶ TAP, as for the FDP in the copper, it is located just outside the end user's premises

The exhibit below shows the architecture of the coax access network in Denmark.

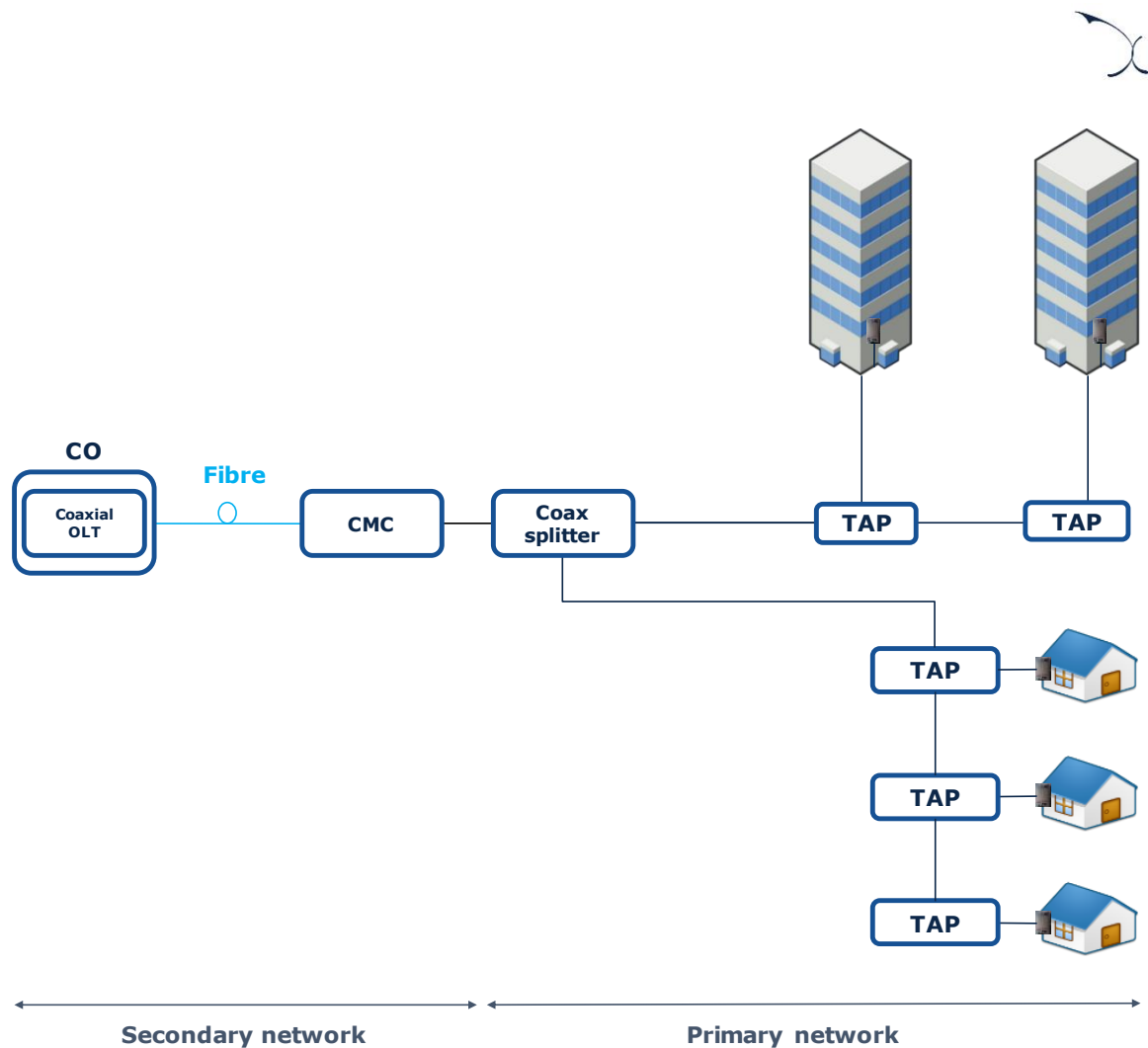


Exhibit 4.10: Architecture of the access coax network [Source: Axon]

The database available outlined the information regarding the location of CMCs and COs. However, during the assessment of the CMC database, we observed some inconsistencies in the number of existing CMCs. Thus, we completed the CMC database based on a Bottom-up approach by adding additional CMCs that would result from splitting the node due to a high capacity constraint.

It should be noted, that in the coax network we consider that there is no PDP, with the fibre cables going directly from the OLT placed in the Central Office to the CMC. Thus, in the case of coax, we calculated the associated CMC for each home in a similar manner as explained in section '4.2.1-Copper Access Network' for the association between PDPs and homes.



On the other hand, the location of the TAPs is calculated on a bottom-up basis, as presented in the MRP. Notably, the locations of the TAPs are calculated following the same approach used for FDP in the copper access network, using the database of the buildings covered by the coax network.

4.2.3 Fibre Access Network

The FTTC network in Denmark is following the same architecture as the copper network, the only difference being a fibre cable is rolled out on the part between the ODF and the FDP of the network instead of a copper cable.

Therefore, the fibre network is composed of the following nodes:

- ▶ **Central Office (CO)**, which represents the last concentration node of the copper access network, establishing the boundaries between the access and the transmission
- ▶ **Optical Distribution Frame (ODF)**, main aggregation point in the fibre access networks
- ▶ **Splitters**, for PON Architecture, which are used to aggregate cables. Their location is in a distribution point between the Central Office and the users. The location of this splitter is slightly different depending on the modelled operator:
 - In the case of TDC, they are located in a location equivalent to those of PDP nodes of the copper network. This is because we observed that the location of these nodes resembled better the number of homes aggregated in each splitter location reported by the operator compared to, for instance, the SDPs from the copper network.
 - In the case of Norlys, the locations of the splitter nodes are located in the SDP nodes.
- ▶ **Final distribution Point (FDP)**, located just outside the end user's premises

In addition, in the case of Norlys, the PDPs, ODFs and CO are located in the same locations. For the sake of clarity, SDPs are not considered for the PTP network of any operator.

The exhibit below shows the overall architecture of the fibre access network.

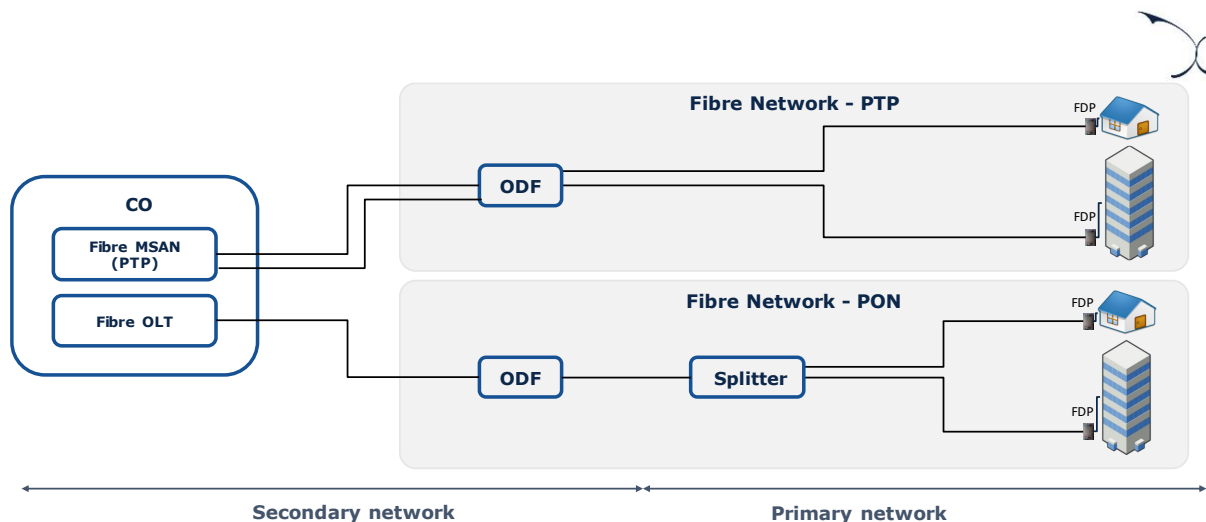


Exhibit 4.11: Architecture of the access fibre network [Source: Axon]

It should be noted that, as mentioned earlier, the modelling for TDC is slightly different, as the PTP MSANs and ODFs been both placed at the MDFs level and not at the CO level. For all other operators, the image applies.

Based on the data reported by the modelled operator, we have considered the location of the nodes of the fibre network to be equivalent to those used in the copper network. The only differences in terms of locations are related to the differences in coverage between networks. The treatment of coverage in fibre networks is described in section '4.4- Characterization of the coverage input' of this document.

4.3. Definition of the routes between the nodes

Once the nodes' locations are determined, the next phase is related to the calculation of the routes connecting each node of the network. As the routes of the network are mainly used to determine the total distance needed for cables, trenches, etc, the results extracted from this analysis are very relevant for network dimensioning.

The steps to calculate the routes between the network nodes are similar for the access technologies, however, the process is slightly different in the case of transmission networks. Hence, the next sections introduce the calculations for each set of networks separately:

- ▶ Access Networks
- ▶ Transmission and Core Networks



4.3.1 Access Networks

The routes that are calculated for the access networks allow connecting the homes to the Central Offices, considering all intermediate levels between both. The process to calculate the routes is based on the calculation of the shortest path (based on Dijkstra's algorithm) between the nodes, considering the different possible paths through the roads in Denmark.

Dijkstra's algorithm is an algorithm used to find the shortest paths between nodes in a graph, which may represent road networks. To find for example the shortest path from a starting point "A" to an end node "B", the algorithm checks all the possible routes and calculates their accumulated distance and chooses the route with the shortest distance. The Dijkstra's algorithm used is therefore a standard implementation of it. More precisely it has been implemented through the use of the R function "*shortest.paths*"³, which is included in the "*igraph*"⁴ package. The following figure illustrates the result of using the Dijkstra's algorithm.

³ <https://www.rdocumentation.org/packages/igraph/versions/0.1.1/topics/shortest.paths>

⁴ <https://cran.r-project.org/web/packages/igraph/igraph.pdf>

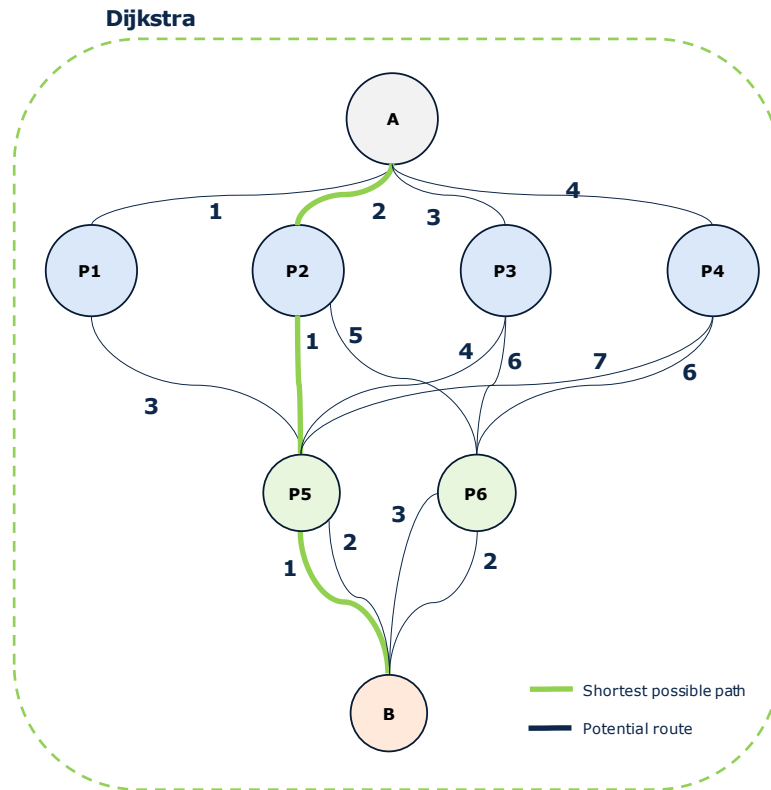


Exhibit 4.12: Illustrative example of Dijkstra algorithm [Source: Axon]

This process is followed for every segment for each access network. For example, in the exhibit below, we present the routes calculated connecting the PDP to MDFs in the copper network.



Exhibit 4.13: Illustrative example of the calculation of routes between network nodes [Source: Axon]

4.3.2 Transmission and Core Networks

Similarly, to the access network, the different nodes from the transmission network need to be interconnected to one another. As previously discussed, the model disaggregates the transmission network into four different layers:

- ▶ L3 Access network
- ▶ Aggregation network
- ▶ Distribution network
- ▶ Core network

A summary of the topology considered is presented in the exhibit below:

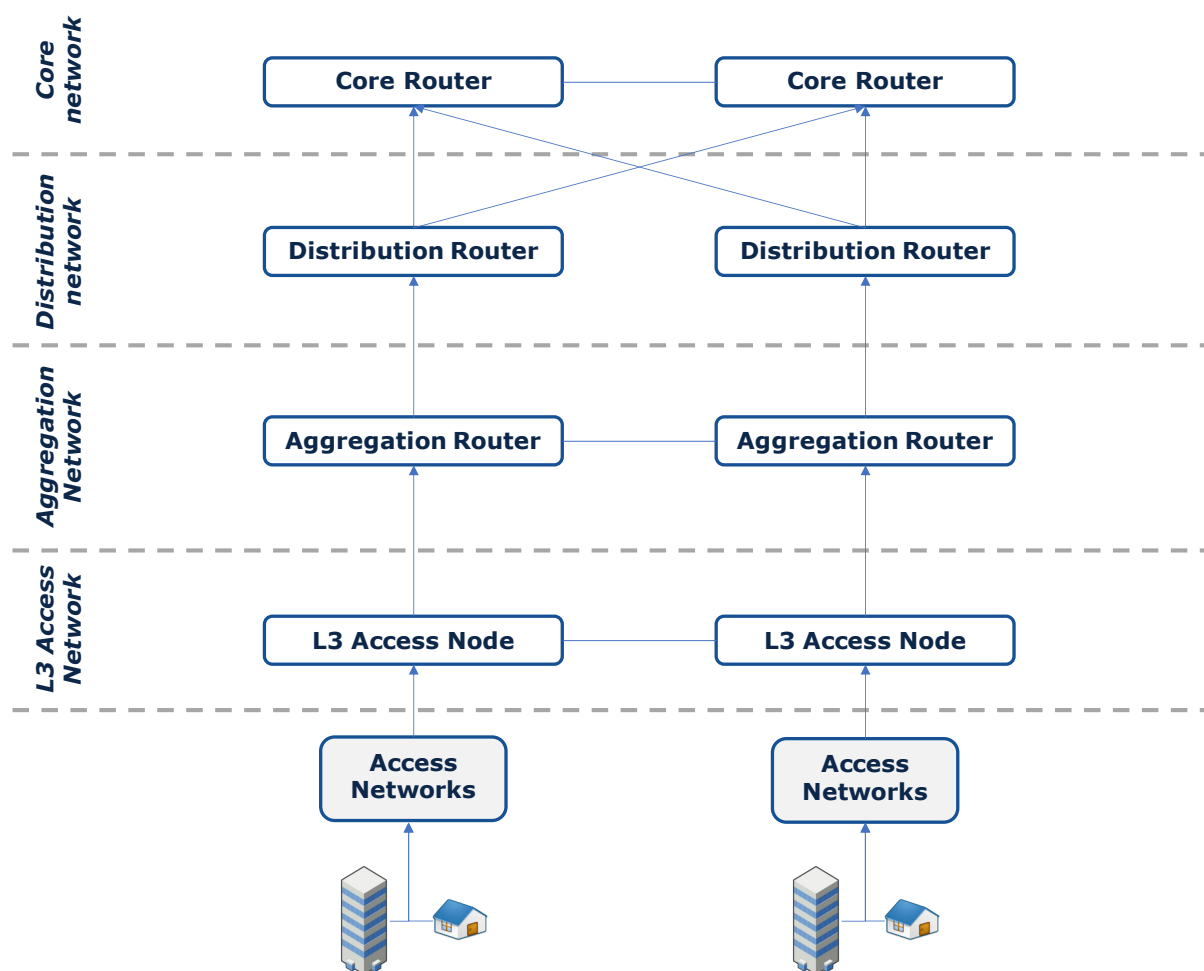


Exhibit 4.14: Architecture of the transmission networks [Source: DBA]

The locations of the transmission nodes, as well as the distribution of the nodes into chains/rings are, for the most part, known based on data reported by the modelled operator. For clarification purposes, a chain/ring is defined as a set of nodes of the same layer interconnected to one another, and which ends are connected to nodes from a higher hierarchy.

For the Norlys, the architecture is equivalent of the one presented above but eliminating the distribution layer which is not use in the Norlys topology.

However, in some instances (especially with regards to the L3 access network), the actual links within each chain are unknown. Then, to determine the links and routes needed to



connect all the nodes with their corresponding higher hierarchy node, we applied a TSP⁵ algorithm for each chain. The objective of this algorithm is to order the different nodes (e.g. Central Offices) in a way that the following conditions are met: i) all nodes are connected ii) each node is connected only once and iii) the total distance of the links is minimized. The algorithm is implemented in R through functions included in the “tsp”⁶ package. The TSP algorithm outputs how to interconnect the nodes among themselves to minimise the total distance of the chain.

Based on this information, we finalised the logical interconnection between each node in the different layers of the transmission network:

- **L3 Access nodes:** This network allows the connection of the COs and aggregation nodes in Denmark. For TDC is composed of 1.105 nodes, distributed across 156 chains. For Norlys is composed of 342 nodes, distributed across 93 chains.

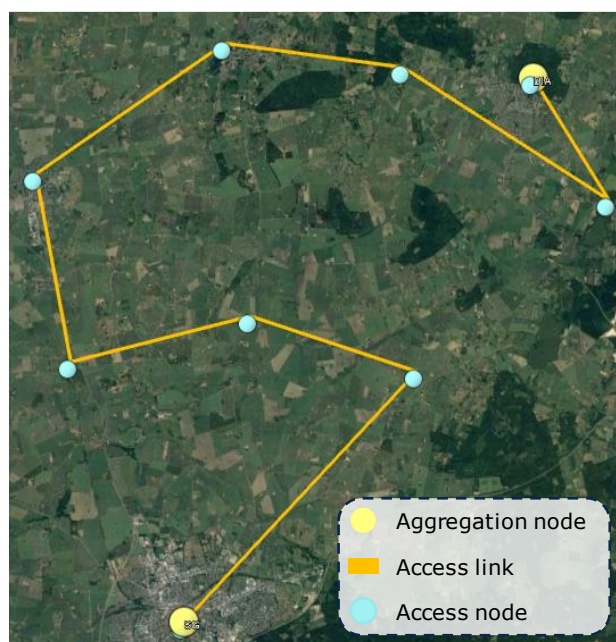


Exhibit 4.15: Illustrative example of the nodes and links in the L3 Access network [Source: Axon]

- **Aggregation nodes:** The aggregation network represents the interconnection of the L3 Access network with the distribution and core nodes. For TDC this layer has 88

⁵ Travelling Salesman Problem.

⁶ <https://cran.r-project.org/web/packages/TSP/index.html>



nodes, distributed over 21 chains. For Norlys this layer has 36 nodes, distributed over 19 chains.

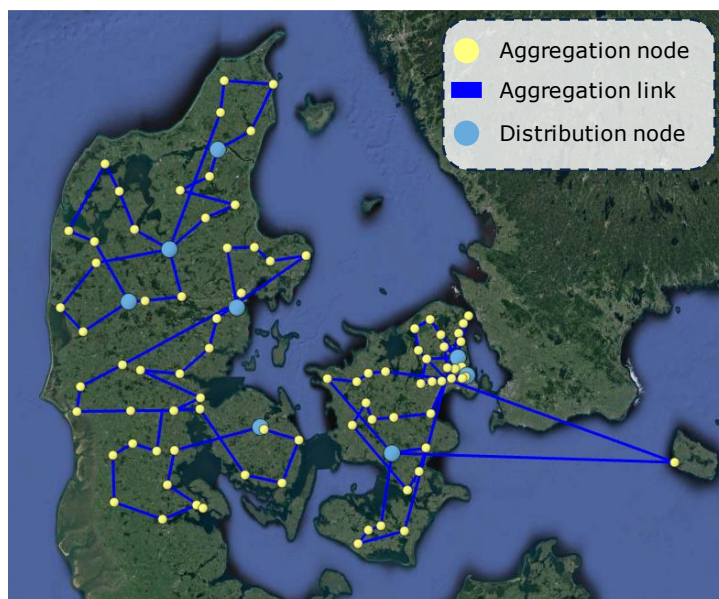


Exhibit 4.16: Illustration of the nodes and links in the aggregation network [Source: Axon]

- **Distribution nodes:** The distribution network has 8 nodes for TDC, responsible to connect the aggregation network with the core network. Each of these nodes is connected to two separate core nodes (dual-homing solution). For Norlys, this has not been implemented.

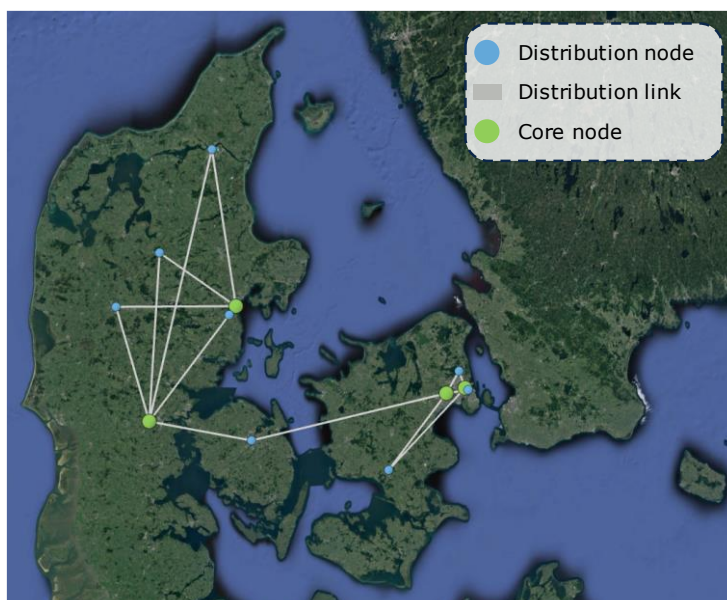




Exhibit 4.17: Illustration of the nodes and links in the distribution network [Source: Axon]

- Core nodes: The core network has 4 nodes for TDC, all interconnected to each other in a ring topology. This layer has 6 Nodes for Norlys, which supplement the lack of distribution nodes for this operator.

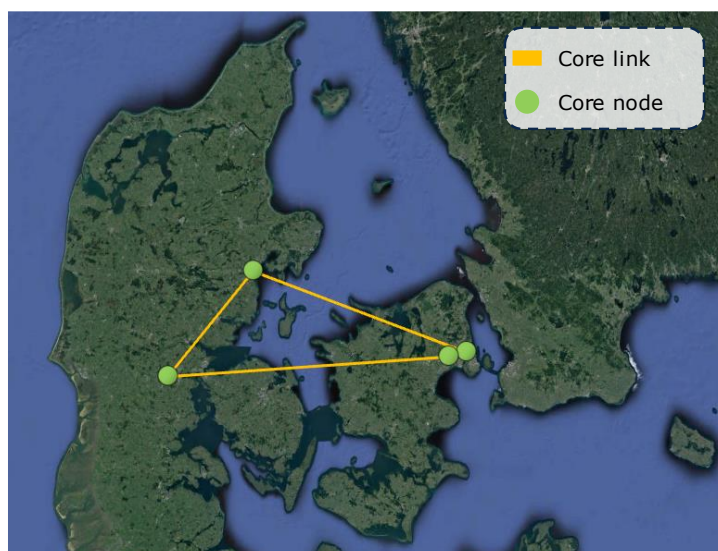


Exhibit 4.18: Illustration of the nodes and links in the distribution network [Source: Axon]

Once, all the chains/rings have been determined we proceed to the calculation of the road distance between the nodes, which is applied based on the same algorithm presented in section '4.3.1 - Access Networks', by means of the road graph calculating the minimum distance.

4.4. Characterization of the coverage input

One of the key results of the geographical analysis is the determination of the homes that are covered for each modelled year. In the context of the R model, a covered home means that the access network reaches the point of the closest road where the drop would be installed to connect the user.

These inputs were defined following different approaches for the copper, coax and fibre networks, as shown below:



- ▶ **Copper networks⁷:** The 2018 coverage levels (actual buildings covered by the operator) were extracted from the data available within DBA (see section '3.3 - Coverage database' for further information about this database). Coverage levels for the other years (before and after 2018) were set at the same level as for 2018 based on the indications received from the modelled operator.
- ▶ **Coax networks:** Coax networks have been modelled differently for TDC and Norlys:
 - **TDC:** Similar to copper, the 2018 coverage levels (actual buildings covered by the operator) were extracted from the data available within DBA and all other years were set at the same level, based on the indications received from the modelled operator.
 - **Norlys:** The coverage levels for Norlys, have been extracted from a list of covered houses delivered by the operator.
- ▶ **Fibre coverage:** In this case, the fibre footprint is defined dynamically, with each year considering the coverage levels reported by the modelled operators. PON and PTP fibre networks are treated separately for this input. Given that the information reported by each operator is different, the following sections present the approaches followed in the definition of this input for each operator. The fibre coverage data can be included either at Household level (Norlys) or at Central office level (TDC). In addition the model is able to work also considering only national information, in the absence of better information.

4.4.1 Fibre rollout algorithm for TDC

The fibre rollout algorithm defined for TDC has been tailored to the information reported by this operator. Notably, the information items reported by the operator includes:

- ▶ **Item A:** Total number of homes covered for each of the years modelled (2005-2038).
- ▶ **Item B:** Number of homes passed in each Central Office region for the year 2019, separately for PTP and PON networks.
- ▶ **Item C:** Number of homes passed in each Central Office region for the year 2025, separately for PTP and PON networks.

⁷ Only relevant for TDC



In addition to this information, the model considers DBA's current coverage database to exclude any buildings already passed with fibre by alternative operators.

Based on this data, the algorithm implemented consists in the following steps:

- ▶ Step 1: Disaggregating the COs into PON and into PTP
- ▶ Step 2: Determining the coverage levels for each CO and for each modelled year
- ▶ Step 3: Defining which buildings to cover for each year and CO

Each of these steps is presented with further detail below.

Step 1: Disaggregating the COs into PON and into PTP.

Based on the data reported by TDC in Item B above, we were able to disaggregate each CO and determine whether the deployment was either PON or PTP for each of the COs where TDC has already deployed fibre.

This selection was performed based on the data reported, also considering the fact that, as reported by TDC, the PTP deployment has taken place mostly in the area denominated as the "DONG area".

Step 2: Determining the coverage levels for each CO and for each modelled year.

As presented above, Item A of the information reported by TDC included data on the number of homes currently passed for fibre networks. In addition, we considered this information as the starting point for 2019, and then proceeded to estimate the number of homes passed for historical periods and for forecasted periods separately. This process takes place separately for PON and PTP networks, however, the process for both is completely equivalent.

Using the national level data and items B and C (information at CO level for years 2019 and 2025, respectively). We have extrapolated the number of homes to cover in each year ensuring that all the sum of coverage at CO level is compliant with the national level coverage reported in item A.

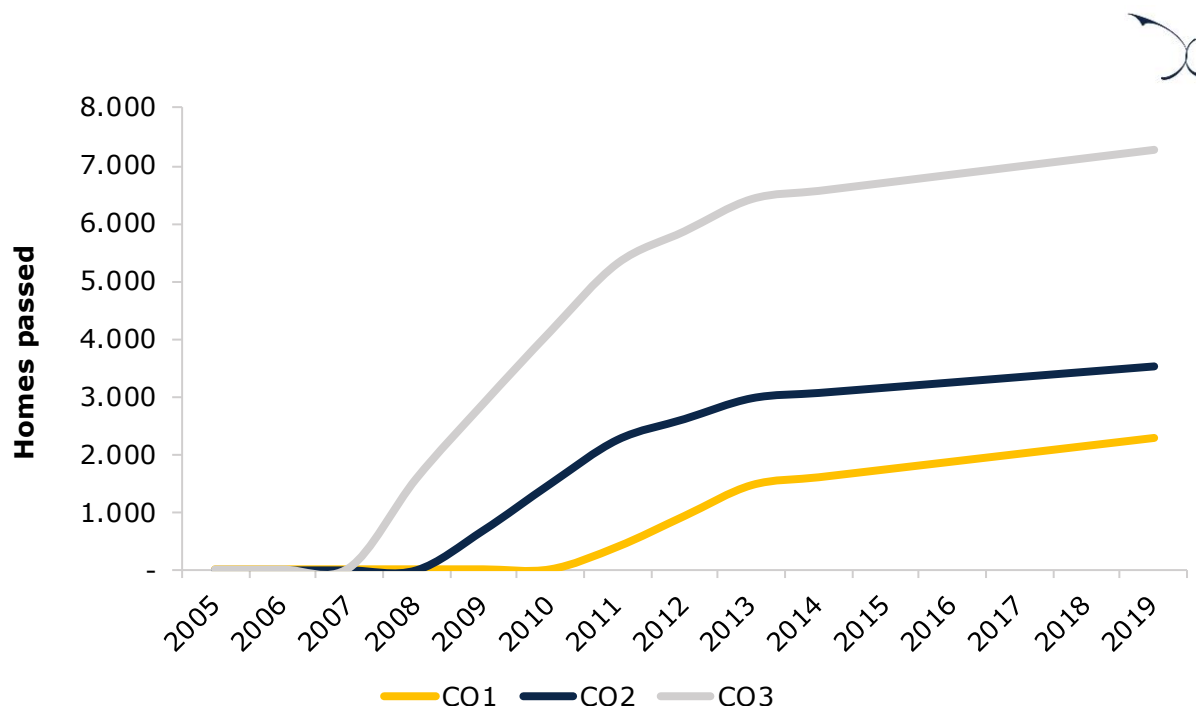


Exhibit 4.19: Illustrative example of the treatment of the fibre rollout algorithm for the historical period [Source: Axon]

In addition to this, we observe that TDC presented a set of additional homes that would be covered, which represent “greenfield deployments” (i.e. areas where the homes have not been built yet). To include these homes, we allocated the number of greenfield deployments expected each year based on the number of homes available (i.e. not covered by fibre or coax) in each CO.

Step 3: Defining which buildings to cover for each year and CO.

The previous step determines the number of homes that will be covered in any given year for the fibre network. This is an input of the R model, as steps 1 and 2 are performed outside the model in an ad-hoc analysis.

In this step, the R model defines which are the buildings that shall be covered each given year. This is performed by ordering the buildings by their road distance to the Central Office (please see section ‘4.3 - Definition of the routes between the nodes’ for further details on the approach to calculate road distances in the R model). Buildings are covered following the groups explained in the previous step. The final results for an illustrative Central office are presented in the exhibit below.

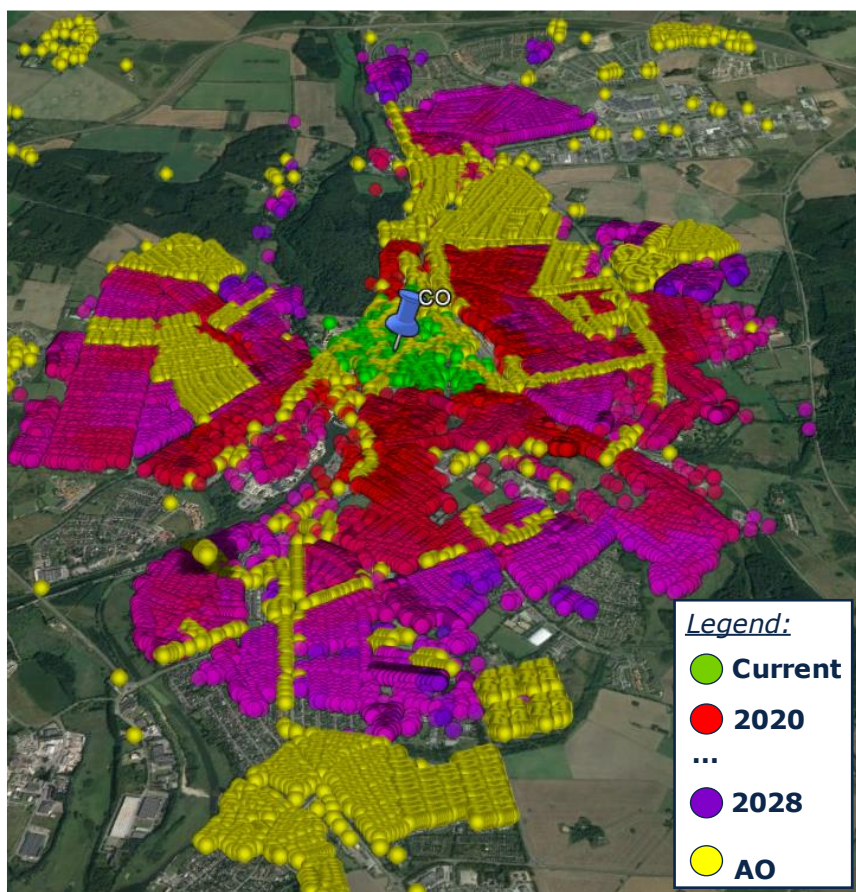


Exhibit 4.20: Illustrative example of the results of the rollout algorithm for a CO [Source: Axon]

In this step, the model avoids, to the maximum extent possible, covering buildings which are covered with fibre from an alternative operator.

4.4.2 Fibre rollout algorithm for Norlys

The fibre rollout algorithm defined for Norlys has been tailored to the information reported by this operator. Notably, the information items reported by the operator includes:

- ▶ **Item A:** Total number of homes covered for each of the years modelled (2005-2038) for each technology (PTP or PON).
- ▶ **Item B:** Current household coverage, at household level.
- ▶ **Item C:** Expected household coverage by 2038, at household level.
- ▶ **Item D:** Technology (PON or PTP) in each CO.

Based on this data, the algorithm implemented consists in the following steps:



- ▶ Step 1: Disaggregating the COs into PON and into PTP
- ▶ Step 2: Separating between current and future coverage
- ▶ Step 3: Defining which buildings to cover for each year

Step 1: Disaggregating the COs into PON and into PTP.

The disaggregation between PON and PTP was automatic due to the item D reported by the operator. The following map shows a rough idea of the coverage zone on each technology.



Exhibit 4.21: Area covered by Norlys' fibre network [Source: Axon]

Step 2: Separating between current and future coverage

The information reported on item B and item C allowed us to divide the building data between two differentiated sets, one for current coverage and another for future coverage. The following figure illustrates this process at CO level.



Exhibit 4.22: Illustrative example of buildings of a CO being divided in the two data sets [Source: Axon]

Step 3: Defining which buildings to cover for each year

As seen above, the data has already been divided in two deployment periods, but it was necessary to set the exact years each building should be covered.

In the **historical period**, the buildings considered in the “current” coverage dataset are ordered from the ones closer to their corresponding CO to the ones further away. Then for each year, the model observed from data item A the number of buildings to cover for any given year and selects this number of buildings from the ordered list (e.g. if there are 100 buildings to cover any given year, the model takes the closest 100 buildings from the dataset). At all times, only the buildings that have been reported by Norlys are considered in the calculations.

In the **forecasted period** a similar approach is followed. First, all buildings currently covered by Norlys are considered. Then, from the dataset of the future buildings to cover,



the buildings are once again, ordered based on their distance to their corresponding CO. Buildings that are closer to the CO are passed first, ensuring that all buildings are covered.



Exhibit 4.23: Illustrative example of buildings the future deployment in a CO [Source: Axon]



5. Calculation of the KPIs

The geographical analysis aims to calculate the number of passive network elements (e.g. cables, trenches, etc.) required to meet the coverage levels defined for each network element and for each year (for further indications on the calculation of the coverage levels defined for each network, please see section '4.4-Characterization of the coverage input').

As the modelled operator manages different access and transmission networks that involve passive civil infrastructure element. For each of these elements, the model calculates two results:

- ▶ The number of assets required, for each modelled year and geotype considered in the model.
- ▶ The disaggregation of the asset in terms of the available modularities (e.g. number of strands in fibre cables). This disaggregation is calculated considering only the year with the highest rollout requirements, in order to ensure the deployment is forward-looking.

The calculation of these KPIs is divided as follows:

- ▶ Copper Access Network
- ▶ Fibre Access Network
- ▶ Coax Access Network
- ▶ Other common civil infrastructure elements
- ▶ Core and Transmission networks

Each of these blocks is presented separately below.

5.1. Copper Access Network

The algorithm for copper access networks is organised into five steps, as shown in the chart below:

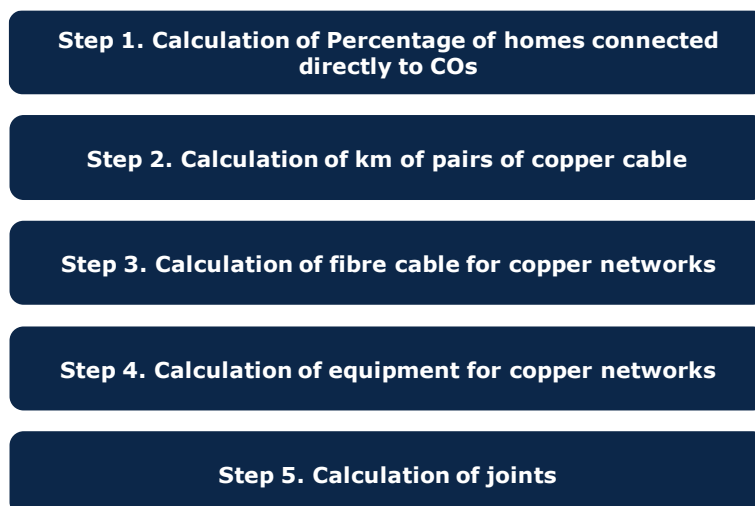


Exhibit 5.1: Steps followed for the calculation of copper network elements [Source: Axon]

The following subsections provide further detail about each specific step of the algorithm.

5.1.1 Step 1. Calculation of homes connected directly to a Central Office

The objective of this step is to disaggregate the homes that are directly connected to an MDF located in the Central Office (instead of a remote MDF). This step is relevant to ensure that the calculation of fibre elements is done only for those homes that are connected to the CO through a remote MDF.

Therefore, in this step, we proceed to the calculation of the percentage of homes that are directly connected to a CO, outputting the number of homes connected in each Central Office that do not pass through a remote MDF.

5.1.2 Step 2. Calculation of km pairs of copper cable

This step focuses on the calculation of the length and typology of the copper cables used in the access copper network.

Copper cables are used in the network to connect the segments from the FDP to MDFs (FDP-SDP, SDP-PDP, PDP-MDF). The final drop of copper that connects the homes to the FDP is calculated in the Excel model as it depends on the demand.



We consider that a 2-pair cable is deployed for each home passed. Notably, a catalogue of cables with a variable number of pairs per cable is available in the model based on typologies handled by the modelled operator. Therefore, the calculations of the model have to ensure that the typology of the cables used in the network as the cables aggregate represents the most efficient alternative, with the constraints posed by the levels of coverage in the network and the available cable sizes from the catalogue.

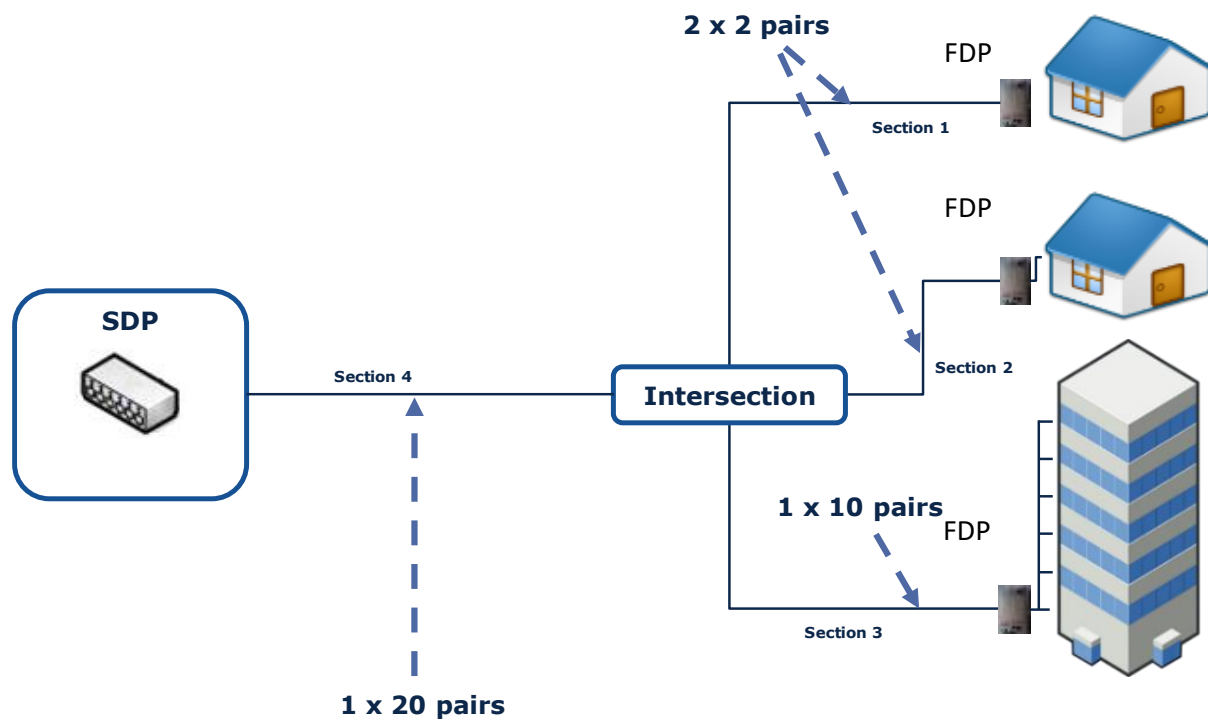


Exhibit 5.2: Illustrative example of the aggregation of copper cables in FDP-SDP segments [Source: Axon]

In order to accomplish this task, the model carries out the following algorithm for each section where a copper cable should be installed:

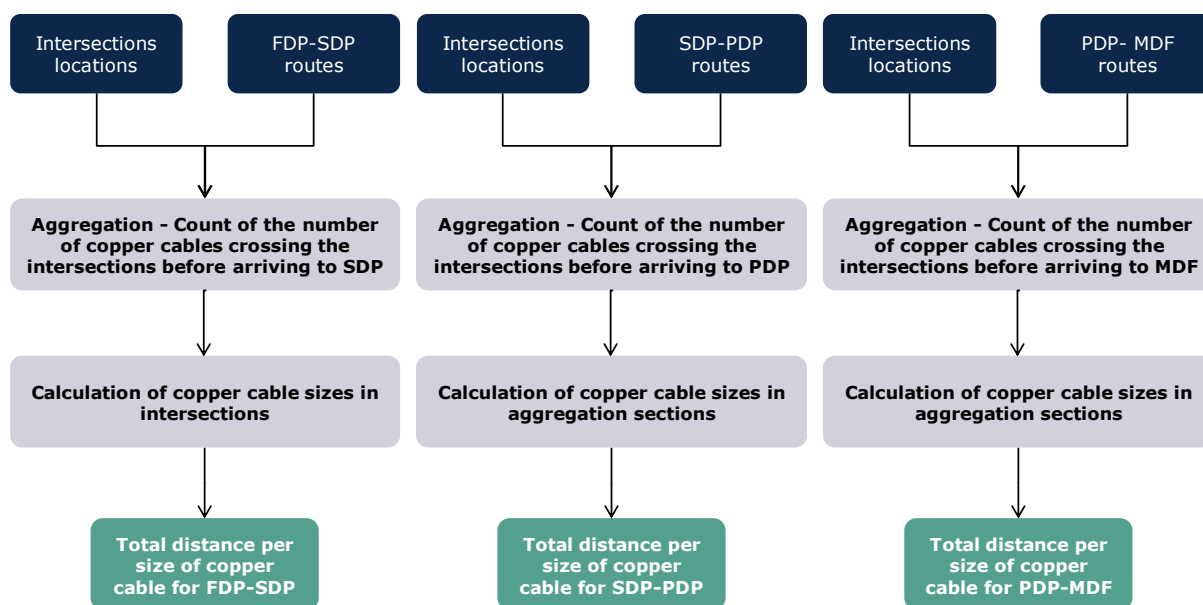


Exhibit 5.3: Algorithm used for the calculation of the km of copper cables per network layer in copper networks [Source: Axon]

This algorithm allows the calculation of the length of copper cable to install in each section as well as the efficient size of the cable.

The output of this algorithm is the length of the copper cable per size required to reach the next higher level in the network.

5.1.3 Step 3. Calculation of km of fibre cable for copper networks

Fibre cables are used to interconnect the remote MDFs to the Central Offices. The model considers that all remote MDFs are connected through optic fibre. We consider that there is a single fibre cable connecting the MDF to the Central office, which is equivalent to the one used in higher-layer transmission networks (12 strands).

To calculate the km of fibre needed for this segment of the network, the following algorithm is performed:

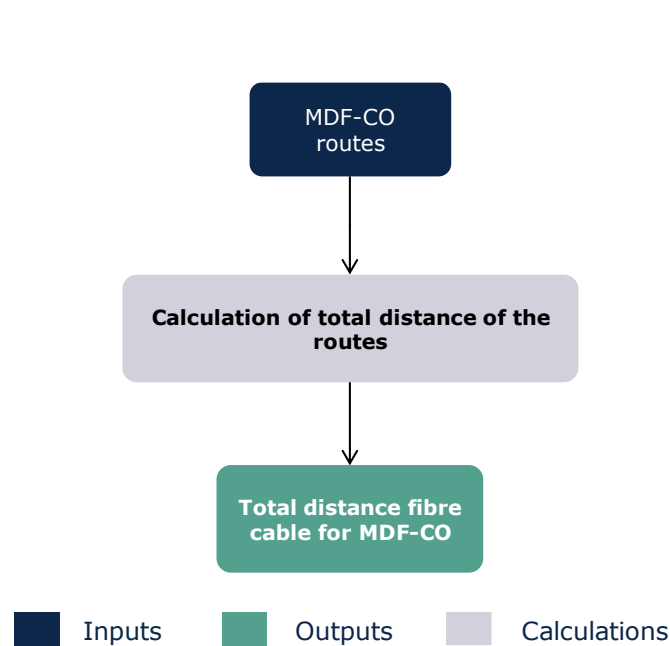


Exhibit 5.4: Algorithm used for the calculation of the fibre cable distance used in the copper network [Source: Axon]

The output of this algorithm is, for each remote MDF, the length of the fibre required to reach the Central Office.

5.1.4 Step 4. Calculation of SCs, DPs and MDFs

The third step involved the calculation of the number of street cabinets (SC), Distribution Points (DP) and Main Distribution Frame (MDF) needed for each scenario of the copper access network. Each of these is detailed below separately.

Calculation of the number of street cabinets

The street cabinets are installed to hold remote MDFs nodes. The model defines different sizes of cabinets, based on the number of homes passed it can hold.

Hence, in order to determine the number of street cabinets needed the following algorithm has been applied to each MDF location:

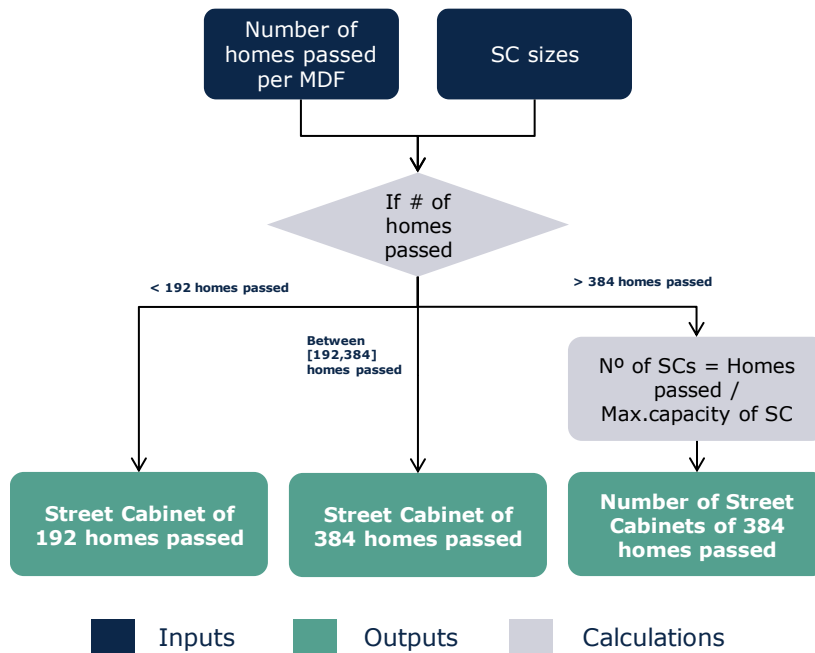


Exhibit 5.5: Number of street cabinets algorithm [Source: Axon]

The output of the algorithm is the final number of street cabinets and its typology.

Calculation of number of DPs

The distribution points aggregate the cables coming from the homes and are installed in SDP and/or PDP nodes. Similarly, to other network elements, the model includes different DPs depending on the number of homes passed. Hence, in order to determine the number of DPs needed the following algorithm has been applied to each SDP/PDP location:

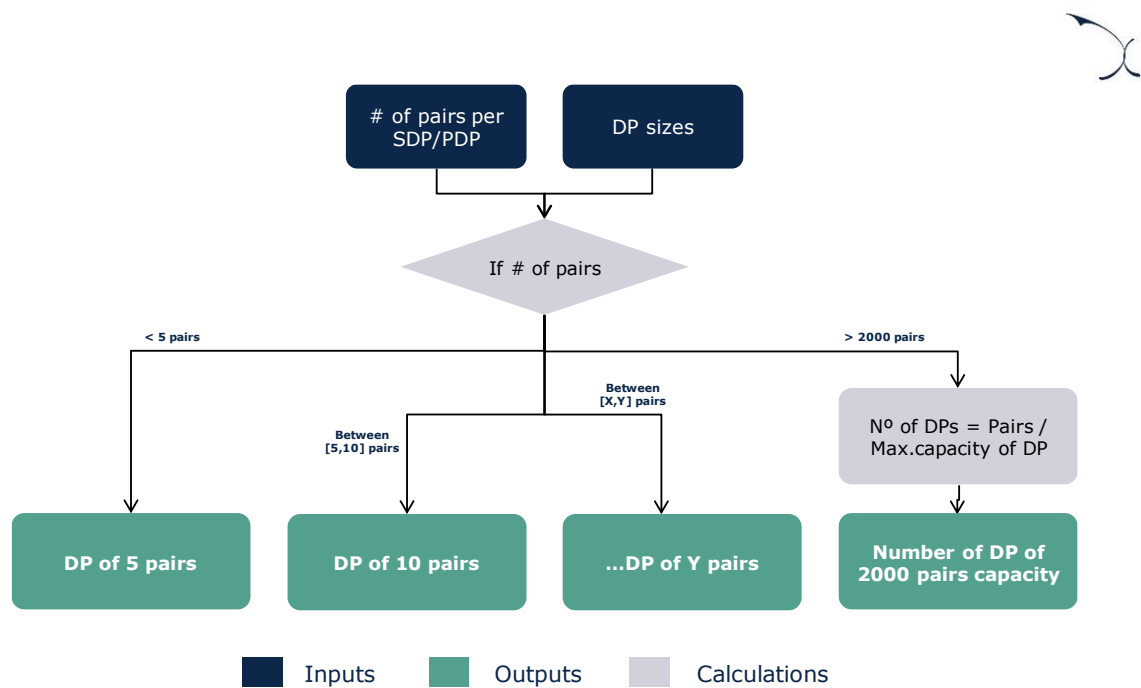


Exhibit 5.6: Number of DPs algorithm [Source: Axon]

The output of the algorithm is the final number of distribution points in the copper network and its typology.

Calculation of number of MDFs

The MDF is an equipment that is installed at the points of the network where copper cables end. This means it can be in remote locations (cabinet) or in the Central Office. Hence, in order to determine the number of MDFs needed and its size, the model considers for each potential location the number of homes passed.

5.1.5 Step 5. Calculation of copper joints

The joints are used in the copper network to merge cables, facilitate the management of their direction as well as to reduce costs in its installation and operation. Therefore, it is assumed that the joints are needed in every intersection where there are copper cables.

Therefore, the total number of copper joints are calculated as the sum of the installed joints in each intersection of the country.

However, similarly to other network elements, the model considers different sizes for the joints depending on the number of lines aggregated. Therefore, in order to calculate the number of joints per intersection per size, the following algorithm is followed:

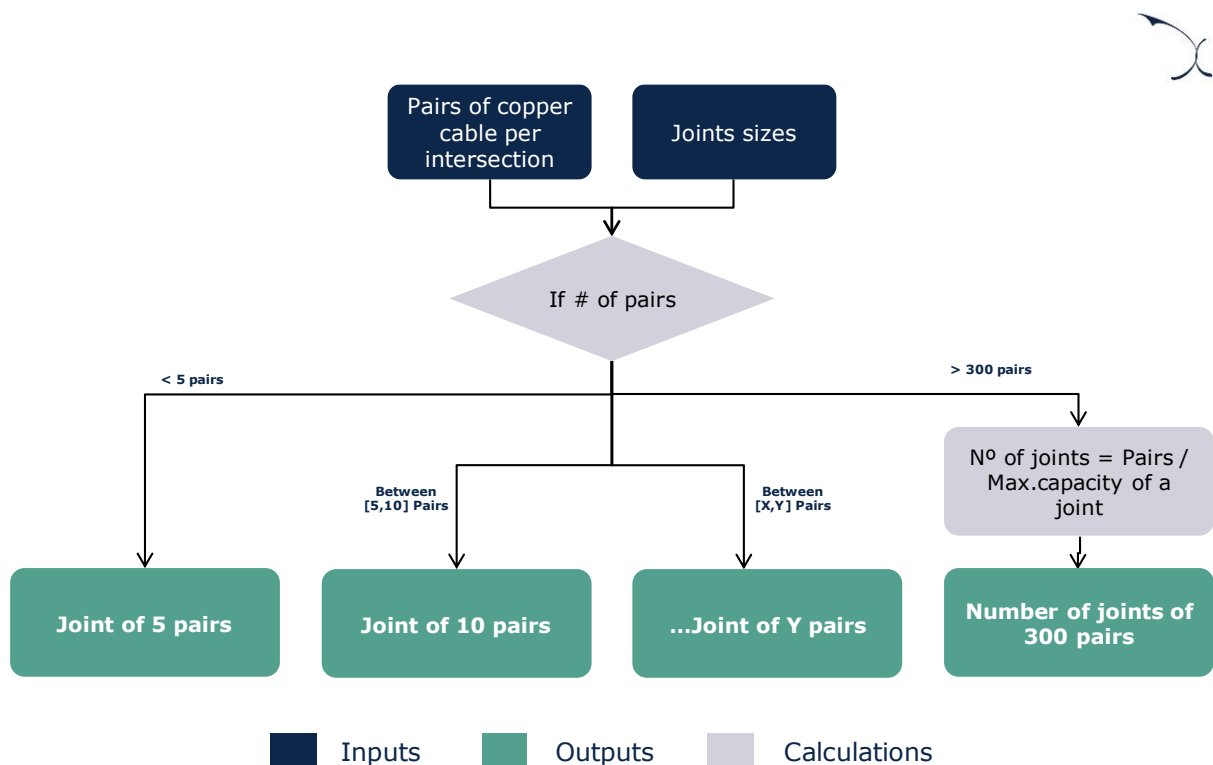


Exhibit 5.7: Algorithm for copper joints [Source: Axon]

The output of the algorithm is the final number of joints in the copper network and its typology.

5.2. Fibre Access Network

The algorithm for fibre access networks is organised into four steps, as shown in the chart below:

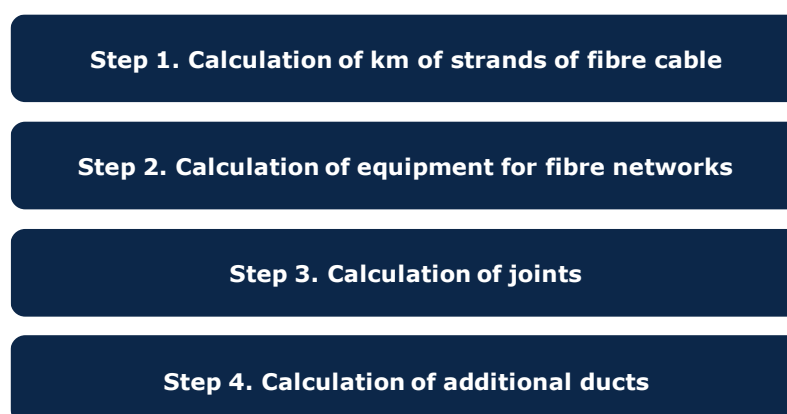


Exhibit 5.8: Steps followed for the calculation of fibre network elements [Source: Axon]



After analysing the data provided by TDC, the average number of households aggregated by splitter were similar to the ones aggregated in the copper PDP. Therefore, the SDPs are not used for TDC fibre network and the PDPs are the ones handling the splitters. On the other hand, the splitters of Norlys were placed in the SDPs.

The algorithms used for PTP and PON are almost identical except for a few differences explained in this section. Therefore, although the algorithms are separated in the R model, the explanations on this section applies to both PTP and PON networks.

The following subsections provide further detail about each specific step of the algorithm.

5.2.1 Step 1. Calculation of km of fibre cable

The first step focuses on the calculation of the total kilometres of fibre cable used in the access fibre network.

Fibre cables are used in the network to connect the home to the Central Offices (FTTH). The final drop of fibre that connects the homes to the FDP is calculated in the Excel model using the average final drop distance calculated in the R model (see sections '5.4-Other common civil infrastructure elements') as it depends on the demand.

We consider that a 2-strand cable is deployed for each home passed. Notably, a catalogue of cables with a variable number of strands per cable is available in the model based on typologies handled by the modelled operator. Therefore, the calculations of the model have to ensure that, as cables start to aggregate, the typology of the cables used in the network represent the most efficient alternative, with the constraints posed by the levels of coverage in the network and the available cable sizes from the catalogue.

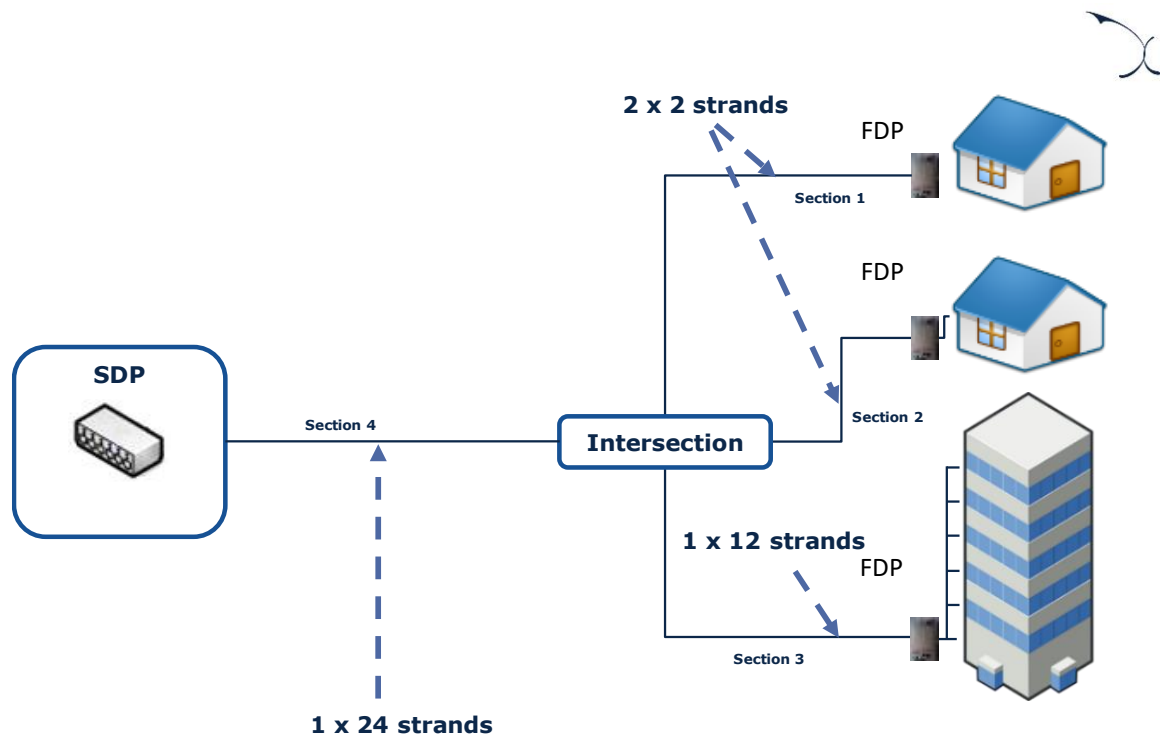


Exhibit 5.9: Aggregation of fibre cables in FDP-SDP segments [Source: Axon]

In order to accomplish this task, the model carries out the following algorithm for each section where a fibre cable should be installed:

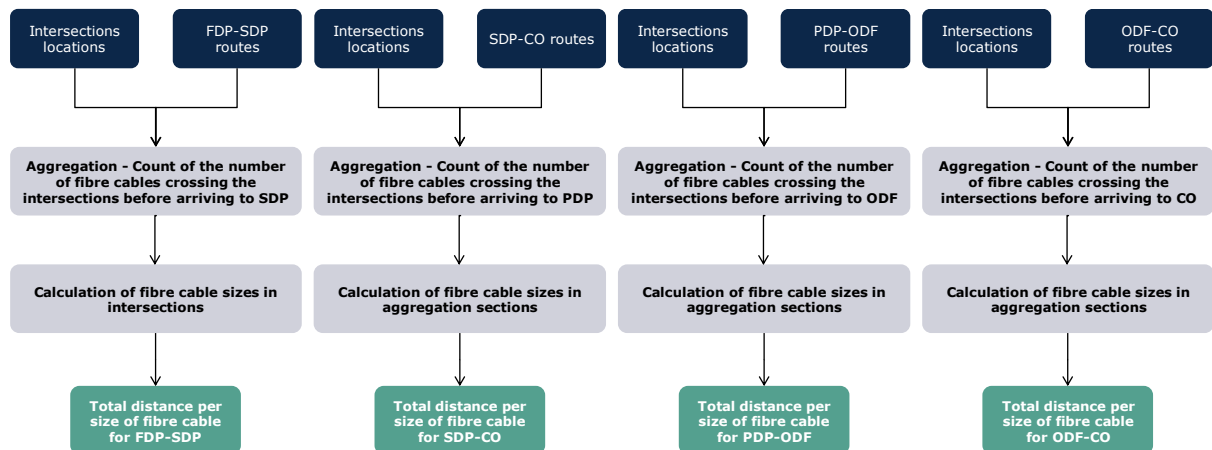


Exhibit 5.10: Algorithm used for the calculation of the km of fibre cables per network layer in fibre networks [Source: Axon]

It is important to note that links may be sequentially aggregated to save digging, ducting, and cabling costs. Therefore, the algorithm allows the calculation of the length of fibre cable to install in each section as well as the efficient size of the cable. The output of this



algorithm is the length of the fibre cable per size required to reach the next higher level in the network.

In the PON architecture, it has been included splitters (1:32). This reduces the number of strands of fibre cables required at higher levels.

The PTP architecture is based on the characteristics of the modelled operators' network, the model considers the need of only one cable of 2 strands between the ODF and the CO.

The output of this algorithm is the length of the fibre cable per size required to reach the next higher level in the network.

5.2.2 Step 2. Calculation of equipment configurations for fibre networks

The following step is related to the calculation of the number of equipment needed for each scenario of the fibre access network. The main equipment is:

- ▶ Calculation of number of splitters
- ▶ Calculation of number of DPs
- ▶ Calculation of number of ODF

Calculation of number of splitters

The splitters are used in GPON architecture to split the fibre for its distribution towards the households. In order to calculate the number of splitters needed in the network the following algorithm has been applied:

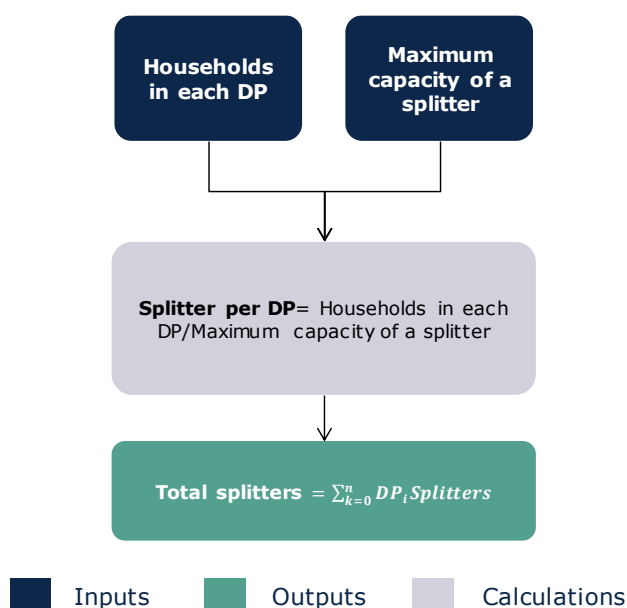


Exhibit 5.11: Algorithm for the calculation of the number of splitters [Source: Axon]

Calculation of number of DPs

The distribution points are installed in SDPs/PDPs nodes (SDPs for Norlys and PDPs for TDC). In addition, DPs are only considered in the PON network and not in the PTP network. Hence, in order to determine the number of DPs needed the following algorithm has been applied to each SDP and PDP location:

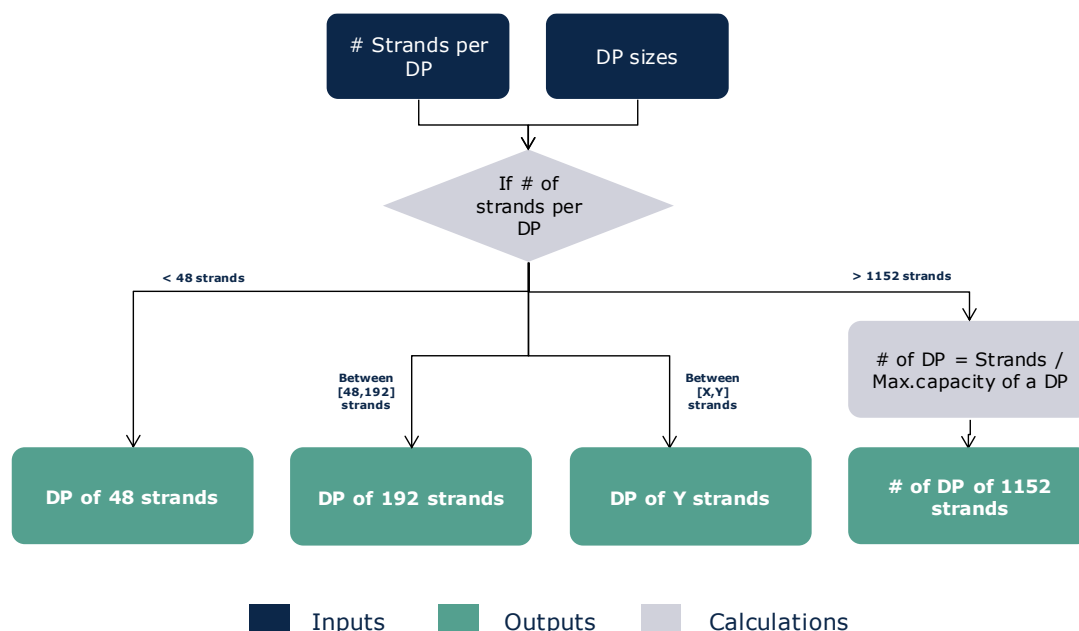




Exhibit 5.12: Algorithm used for the calculation of the number of DPs [Source: Axon]

The output of the algorithm is the final number of distribution points in the fibre network and its typology.

Calculation of number of ODF

The ODFs are installed in COs nodes for the PON topology and in the copper MDFs for the PTP topology. The algorithm is the same for TDC and Norlys but on the Norlys MDFs and COs are positioned in the same exact position therefore ODFs are positioned in the COs. Hence, in order to determine the number of ODFs needed the following calculation flow is applied:

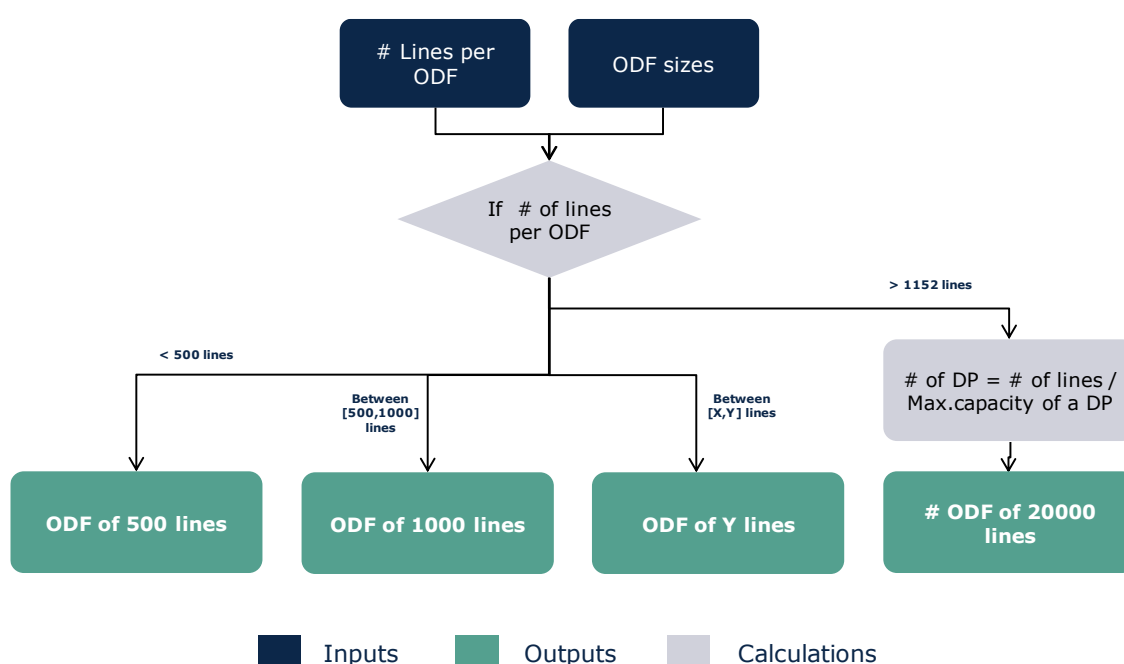


Exhibit 5.13: Algorithm used for the calculation of the number of ODFs [Source: Axon]

The output of the algorithm is the final number of optical distribution frames in the PON fibre network and its typology.

5.2.3 Step 3. Calculation of fibre joints

The joints are used in the fibre network along with distribution points. Therefore, it is assumed that the joints are needed in every FDP. Additionally, it is considered that one joint is needed in each SDP of the fibre PON Network.



Therefore, the total number of fibre joints are calculated as the sum of the installed joints in each FDP/SDP of the country. In order to calculate the number of joints per FDP/SDP and size, the following algorithm is followed:

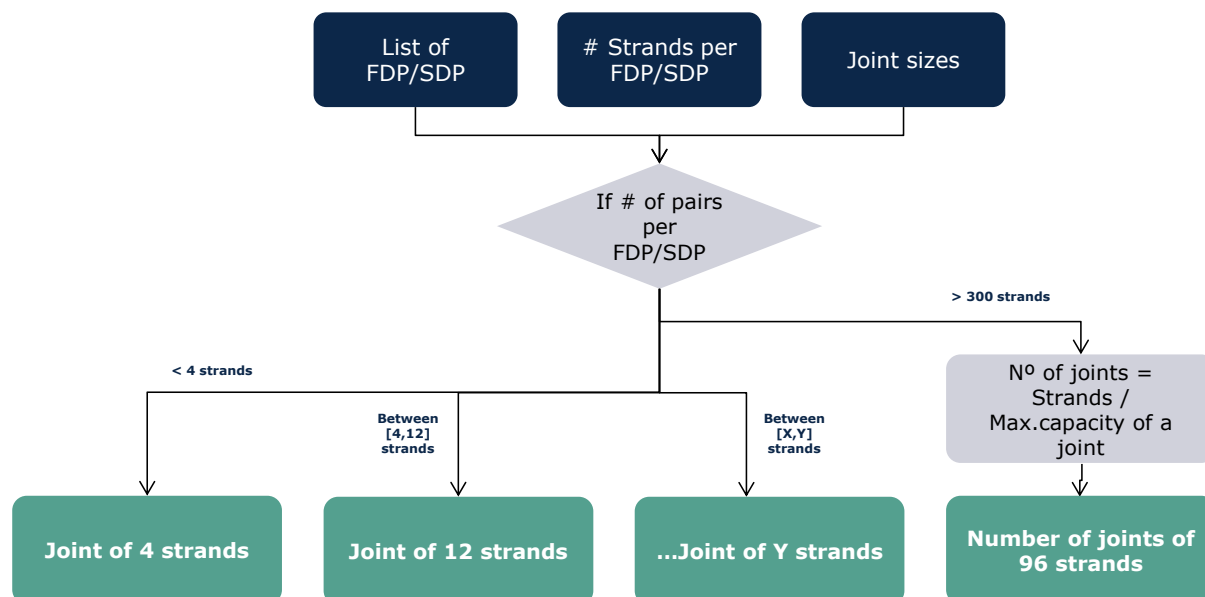


Exhibit 5.14: Algorithm used for the calculation of fibre joints [Source: Axon]

After the total number of joints per size is determined as the sum of all the joints in each FDP/SDP.

5.2.4 Step 4. Calculation of additional ducts

The ducts are used for the installation of fibre cables inside the trenches to ensure their protection. Thus, it is assumed the ducts are needed in every section with fibre cables.

The R Model calculates the total kilometres of additional ducts needed in the network. Indeed, the model already assumes that one km of duct is installed for each km of trenches in the network. However, in most of the cases, more than one duct will be required per section, as the ducts have maximum size and load.

Therefore, the total length of additional ducts is calculated as the sum of the additional ducts required for each section, which are calculated based on the following algorithm:

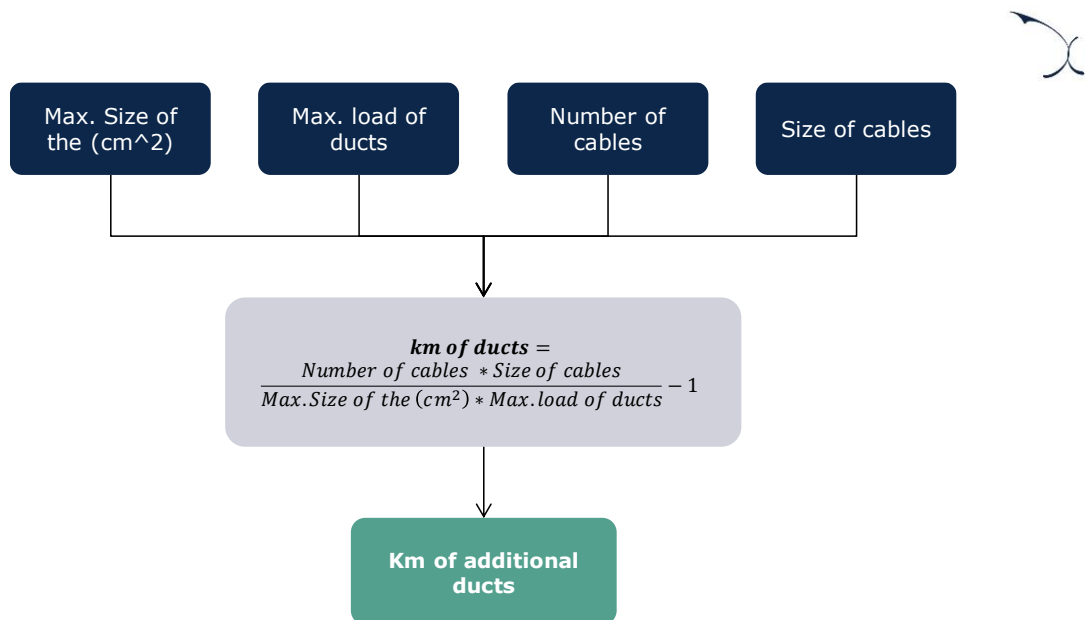


Exhibit 5.15: Calculation of additional ducts in PON Fibre Network [Source: Axon]

5.3. Coax Access Network

The algorithm for coax access networks is organised into three steps, as shown in the chart below:

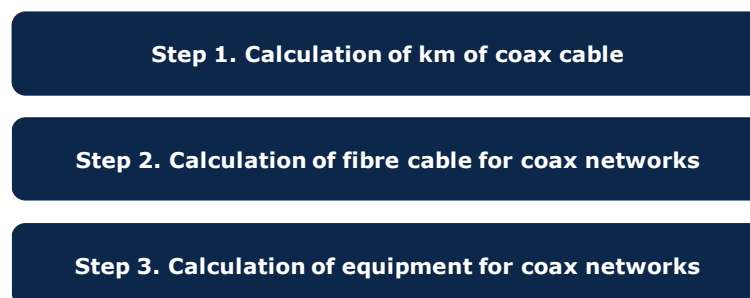


Exhibit 5.16: Steps followed for the calculation of coax network elements [Source: Axon]

The following subsections provide further detail about each specific step of the algorithm.

5.3.1 Step 1. Calculation of km of coax cable

The first step focuses on the calculation of the total kilometres of coax cable used in the access coax network. Coax cables are used in the network to connect the segments from the TAPs to CMCs. The final drop of fibre that connects the homes to the FDP is calculated in the Excel model using the average final drop distance calculated in the R model (see sections '5.4 - Other common civil infrastructure elements') as it depends on the demand



For this calculation, the following algorithm is implemented:

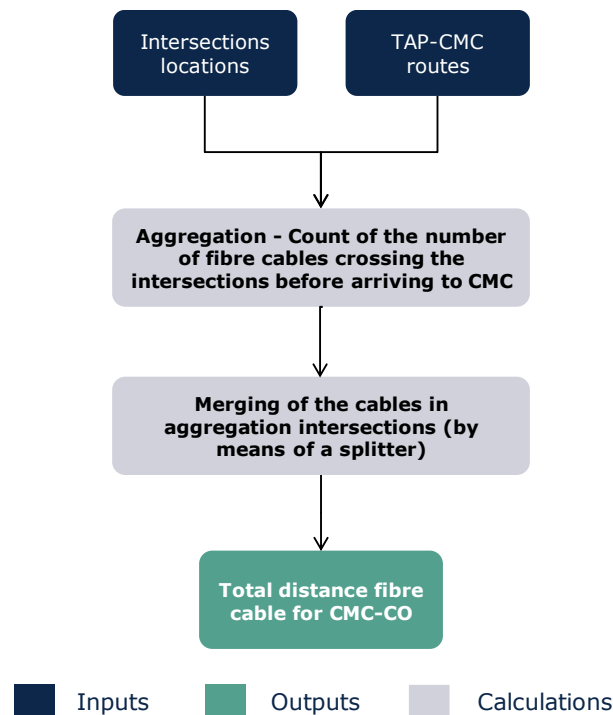


Exhibit 5.17: Algorithm used for the calculation of coax cable for coax networks [Source: Axon]

The output of this algorithm is the length of the coax cable required to reach the next higher level in the network.

5.3.2 Step 2. Calculation of km of fibre cable for coax networks

Fibre cables are used to interconnect the remote CMCs to the Central Offices. The model considers that all CMCs are connected through optic fibre. We consider that there is a single fibre cable connecting the CMC to the Central office, which is equivalent to the one used in higher-layer transmission networks (12 strands).

To calculate the km of fibre needed for this segment of the network, the following algorithm is performed:

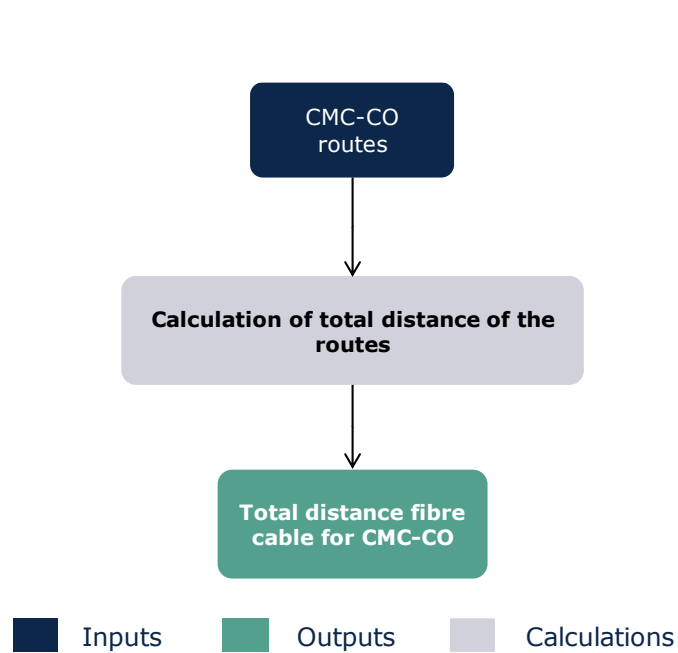


Exhibit 5.18: Algorithm used for the calculation of the fibre cable distance used in the coax network
[Source: Axon]

The output of this algorithm is, for each CMC node, the length of the fibre required to reach the Central Office.

5.3.3 Step 3. Calculation of equipment for coax networks

The third step involved in the:

- ▶ Calculation of number of CMCs, OLTs and TAPs
- ▶ Calculation of number of coaxial splitters

Calculation of number of CMCs, OLTs and TAPs

The TAPs, OLTs and CMCs represent the main equipment used in the coax network. The number of locations with this equipment is determined based on the coverage information provided. Hence:

- ▶ The number of TAPs is equal to the number of buildings covered.
- ▶ The number of CMCs is equivalent to the number of CMC nodes determined in section '4.2-Location of additional network nodes'.
- ▶ The number of OLTs is equal to the number of Central Offices where there is coverage for coaxial network



Calculation of number of coaxial splitters

Finally, the splitters are used in the coax network in order to split high-level cables to cover households. Hence, knowing the routes of the coaxial cables, the geographical model identifies all the intersections that are crossed by a coaxial cable, then it assigns one splitter per intersection.

5.4. Other common civil infrastructure elements

This step is related to the calculation of the km of trenches in the access network. the trenches needed for each network are calculated separately for the three access networks (copper, fibre, and coax).

The trenches are calculated considering all the routes calculated in section '4.3 - Definition of the routes between the nodes' for the three technologies. The process followed for every section is the following:

1. For every section, a trench is installed in the side with more buildings. As we can see in the figures below, the first trench is placed on the upper side of the section, for this reason. In the second exhibit, the trench is placed on the lower side, where there are more buildings.

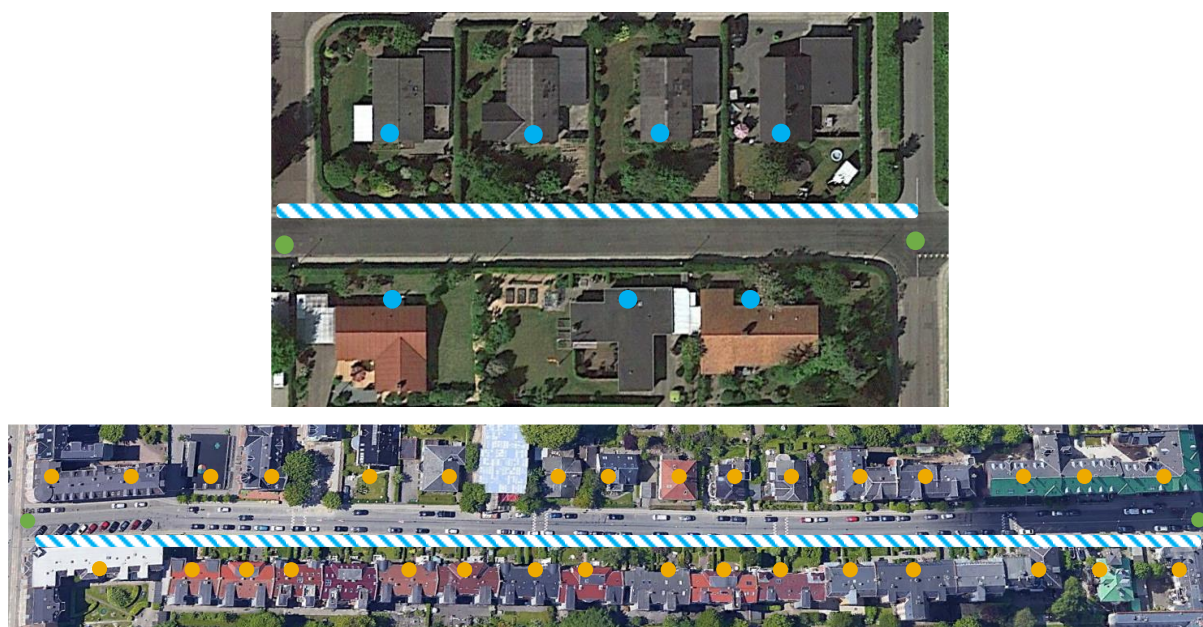


Exhibit 5.19: Identification of the side to install the main trench in each section [Source: Axon]



2. The next step considers that the building on the other side of the road must be able to either have their own trench or to have a road crossing to reach the existing trench. For this, in each section two main options have been considered:
 - a. We have considered that in the case the distances between the houses of the secondary side of the road (the side of the road with the lower number of homes) is greater than the road width⁸, then all houses on the secondary side are covered through road crossings



Exhibit 5.20: First option for the second trench in each section [Source: Axon]

In this case, the total length of the trenches to be dug out for each section would be:

$$d = \text{Length of the section}$$

And the number of road crossings:

$$C = \text{Number of buildings in the side with less buildings} \cdot \text{width of the road}$$

- b. If the average distance between buildings lower than 8 meters, we consider that it is more cost-efficient to dig out an additional trench on the other side of the road.

⁸ This comparison is weighted by the cost of a normal trench and a road crossing, to ensure a more realistic result.



Exhibit 5.21: Second option for the second trench in each section [Source: Axon]

In this case, the total length of the trenches to be dug out for each section would be:

$$d = 2 \cdot \text{Length of the section}$$

The length of the road crossings is extracted as a different KPI to the trenches due to the comparable high cost of said road crossings compared to the regular trenches along the road.

Additional KPIs

In addition to trenches, some general KPIs are calculated:

- ▶ The total number of buildings per geotype
- ▶ The average number of houses per building per geotype

5.5. Core and Transmission networks

The algorithm for core and transmission networks is organised into two steps, as shown in the chart below:

Step 1. Calculation of % of traffic managed per chain

Step 2. Calculation of trenches of core and transmission networks

Exhibit 5.22: Steps followed for the calculation of core and transmission network elements [Source: Axon]

The following subsections provide further detail about each specific step of the algorithm.



5.5.1 Step 1. Calculation of % of traffic managed per chain

The first step focused on the calculation of the percentage of traffic that is being managed for each of the chains of each of and core networks.

In order to do that, the following formula has been applied for each chain:

$$\% \text{ of traffic} = \frac{\sum \text{Homes managed by the nodes of the chain}}{\text{Total homes}}$$

This traffic is used in the Excel model in order to characterize the equipment of the core and transmission network.

5.5.2 Step 2. Calculation of trenches of core and transmission networks

In this step, we proceeded to the calculation of the trenches required for the transmission network. This step is divided into two different substeps:

- ▶ The total transmission trench distance is calculated using the shortest path distance (already explained in previous sections) between each node of the transmission links.
- ▶ Using the sections where a trench of the transmission should be used and the sections where a trench of the access network should be used two KPIs are calculated:
 - The percentage of transmission network not shared with access network nationwide
 - The percentage of the access network that is shared with the core network at geotype level

Each one of the results of the substeps is extracted as a KPI for the excel model.



6. Network elements

This section presents the main outputs of the R Model, and how these results are aggregated before being exported to Excel.

6.1. Levels of disaggregation

The outputs of the R Model are aggregated at a “geotype” level. The geotypes are defined based on four different sets of classifications:

- ▶ **Region:** One of the five administrative regions in Denmark (“Hovedstaden”, “Midtjylland”, “Nordjylland”, “Sjælland” and “Syddanmark”).
- ▶ **Degree of urbanisation:** Depending on the building density, the geotypes are divided into “Urban”, “Suburban” or “Rural”.
- ▶ **Type of dwelling:** Depending on the type of building considered (“Single-dwelling” or “Multi-dwelling”)
- ▶ **Regulatory status:** Depending on the regulatory status of the buildings, geotypes are divided in to “Regulated or “Not-regulated”.

The characterisation of each of the aforementioned classifications is further detailed below.

Region

The first geotype classification is used related to the administrative regions in Denmark. There are 5 regions in Denmark:

- ▶ Hovedstaden
- ▶ Midtjylland
- ▶ Nordjylland
- ▶ Sjælland
- ▶ Syddanmark

In this case, each home is assigned to its corresponding region based on its location. If there is a home that is located in a given region but the Central Office to which it is



assigned is located in a different region, the home adopts the region from the associated Central Office.

Degree of urbanisation

Based on the available information at sector level across the country, we have proceeded to classify all CO coverage areas of TDC into geotypes. The number of geotypes has been set to 3 in order to represent three different types of areas: Urban, Suburban and Rural.

The geotype definition has been performed through cluster analysis. This cluster exercise is carried out using a “k-means” algorithm, considering two main variables that have been selected to characterize the geotypes from TDC’s COs:

- ▶ Number of households per CO (households/per CO).
- ▶ Area per CO coverage area (km²).

Based on this, we observed the following characterisation of the different existing Central Offices:

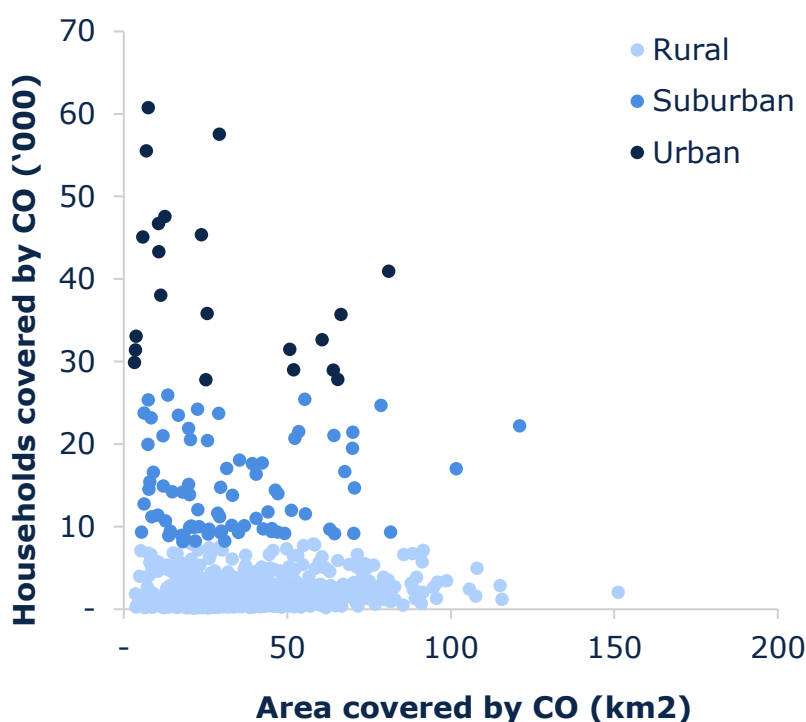


Exhibit 6.1: Characterisation of TDC’s central offices into geotypes [Source: Axon]



Based on this characterisation, we obtained the following distribution of geotypes across Denmark:

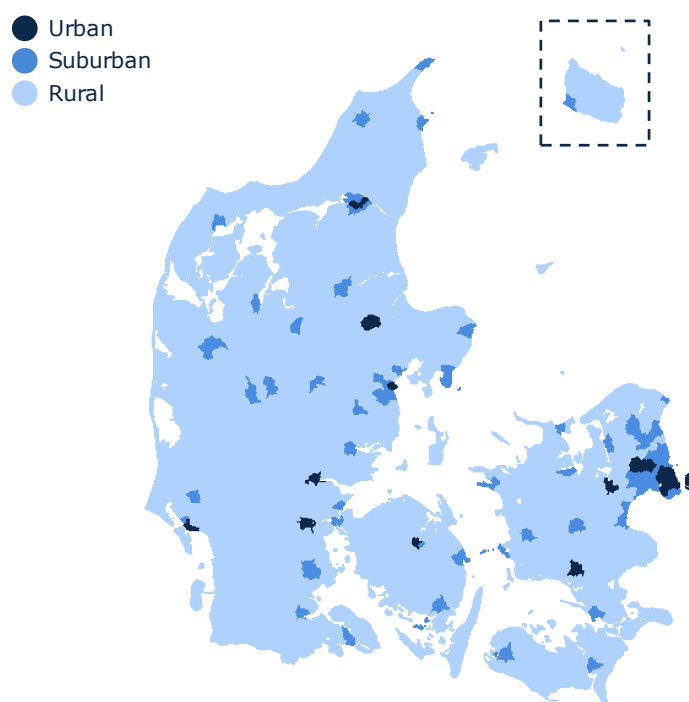


Exhibit 6.2: Classification into geotypes of TDC's CO areas for the geographical analysis [Source: Axon]

As it can be extracted from the exhibit, more dense areas in the country are classified as urban geotype whereas less populated sectors are identified as rural.

In order to keep consistency with the results presented for TDC, alternative operators are modelled after these results. Notably, we assign for each building in the network of the alternative operator the same degree of urbanisation as defined for TDC. Then, we look at all the buildings for each CO for the alternative operator and associate to each of these COs the degree of urbanisation that is representative of most buildings covered by the CO. For instance, if a CO area has 100 buildings which 95 were urban in TDC's dataset and were 5 suburban, that CO is defined as "Urban". We note that in the case of Norlys data, this process results in more than 95% of the buildings matching the same degree of urbanisation defined for TDC.



Type of dwelling

An additional disaggregation performed is the definition of whether a building is single dwelling (only one home in the building) or multi-dwelling (four-or-more homes per building). This information is extracted from each building based on the data available from the address database presented in section '3.1 - Address database'.

Regulatory status

Finally, disaggregation between regulated and non-regulated areas is also considered.

This disaggregation is based on the latest analysis of markets 3a and 3b, where DBA lifted the price regulation obligations applicable to TDC for fibre-related services in some specific areas of Denmark, introducing a geographical disaggregation of the regulatory obligations imposed. As per the latest market decision the regulation in the model is introduced considering the area where the operator holds SMP.

Therefore, the model is able to calculate the costs stemming from regulated and non-regulated areas separately. It should be noted that results based on this disaggregation are only applicable to fibre services.

6.2. Disaggregation of the results into the detailed geotypes

In order to proceed to the allocation of the results (KPIs) of the geographical analysis into the different geotypes⁹ the steps below are followed:

1. Considering that the region and degree of urbanisation are defined at CO level, all the KPIs can be easily assigned to the region and degree of urbanisation without further adjustments.

⁹ It is important to note that in some cases the final numbers to be introduced in the Excel may not be integers. This is mainly due to the fact that the KPIs are in first place calculated per region and the degree of urbanisation (those number are integers) and then split depending on the type of dwelling and the regulatory status, which may alter the numbers, generating decimals.



2. In order to disaggregate the results in all the different geotypes we used two methodologies to further split the KPIs based on the regulatory status and the type of dwelling:

- **Number of houses:** The first methodology, mainly used to disaggregate KPIs not related to distance (DP, joints, MDF, ODF, etc), is based on the number of houses per each geotype in the CO. The final KPIs per geotype are calculated multiplying the KPIs per CO by the percentage obtained dividing the number of houses per geotype and the total number of houses
- **Distance:** The second methodology is used for KPIs related to distance (trenches and cables). In order to account for the distance cost factor on these KPIs, we calculated a weight which is the result of multiplying the number of houses per building by the distance house-CO. The final KPIs per geotype are calculated multiplying the KPIs per CO by the percentage obtained dividing the sum of the weights of the buildings per each geotype and sum of the weight of all the buildings of the CO.

The main outputs of the R model and the driver used to aggregate them into the different geotypes are presented in the table below:

KPI	Driver
Civil Infrastructure	
Access - Coax - Trenches (CMC-FDP)	Distance
Access - Copper - Trenches (MDF-FDP)	Distance
Access - Fibre (PON) - Trenches (SDP-FDP)	Distance
Access - Fibre (PTP) - Trenches (SDP-FDP)	Distance
Access - Coax - Trenches (CO-CMC)	Distance
Access - Copper - Trenches (CO-MDF)	Distance
Access - Fibre (PON) - Trenches (CO-SDP)	Distance
Access - Fibre (PTP) - Trenches (CO-SDP)	Distance
Access - Coax - Trenches (Road crossings)	Distance
Access - Copper - Trenches (Road crossings)	Distance
Access - Fibre (PON) - Trenches (Road crossings)	Distance
Access - Fibre (PTP) - Trenches (Road crossings)	Distance
Access - Fibre (PON) - Ducts	Distance
Access - Fibre (PTP) - Ducts	Distance
Copper Access Network	
Access - Copper - Cables (pairs)	Distance
Access - Copper - Fibre cable	Distance
Access - Copper - Street Cabinet	Number of houses



KPI	Driver
Access - Copper – DP	Number of houses
Access - Copper – MDF	Number of houses
Access - Copper – CO	Number of houses
Access - Copper - Joints	Number of houses
Access - Copper - Homes connected directly to CO	Number of houses
Fibre Access Network	
Access - Fibre (PON) - Strands (PON) (SDP-FDP)	Distance
Access - Fibre (PTP) - Strands (PTP) (SDP-FDP)	Distance
Access - Fibre (PON) - Strands (PON) (CO-SDP)	Distance
Access - Fibre (PTP) - Strands (PTP) (CO-SDP)	Distance
Access - Fibre (PON) - PON OLT	Number of houses
Access - Fibre (PON) - Joints (PON)	Number of houses
Access - Fibre (PTP) - Joints (PTP)	Number of houses
Access - Fibre (PON) - PON Splitter	Number of houses
Access - Fibre (PON) - DP	Number of houses
Access - Fibre (PTP) - PTP MSAN	Number of houses
Access - Fibre (PON) - ODF	Number of houses
Access - Fibre (PTP) - ODF	Number of houses
Access- Fibre (PON)- Trenches shared with coax	Distance
Access- Fibre (PTP)- Trenches shared with coax	Distance
Coax Access Network	
Access - Coax - Cables	Distance
Access - Coax - Fibre cable	Distance
Access - Coax - CMC	Number of houses
Access - Coax - OLT	Number of houses
Access - Coax - Tap	Number of houses
Access - Coax - Coax splitter	Number of houses
Access - Coax - DP	Number of houses
Transmission and Core Networks	
Access chains	N/A
Aggregation chains	N/A
Distribution chains	N/A
Core nodes	N/A
Submarine links	N/A
Other KPIs	
Average dwellings per building	N/A
Number of buildings per geotype	N/A
Share of allocation of core costs	N/A
Share of the core transmission network shared with the access network	N/A
Percentage of the network not shared with the access networks	N/A



KPI	Driver
Share of the submarine transmission network shared with the access network	N/A
Percentage of Copper COs with other technologies (fibre or coax)	N/A
Single dwelling share from SDUs	N/A

Exhibit 6.3: Outputs of the R Model [Source: Axon]

Finally, these KPIs have to be exported to Excel. For more detail about the Export, please refer to section A.2.3 - Export control panel’.



Annex A. R model guidelines

This section, which provides guidelines on how to use the R model, is divided among the following sub-sections:

- ▶ **Getting started**, detailing the main considerations and specifications to run the model.
- ▶ **Understanding the control panel**, describes the Control Panel of the model, which is the main user interface where the main options and scenarios can be selected.
- ▶ **Expert access to execution scripts**, describes the main structure of the app and the folders to find the main scripts of the model.

A.1. Getting started

Important note 1: The public model submitted for consultation includes only a sample dataset. This sample dataset includes only two randomised central offices. Due to the nature of this model, the results are not relevant for the Excel model and the model does not provide results for the core network. The purpose of the public model is to illustrate the calculations and algorithms that are performed in the R-model.

The complete datasets have only been shared with the operators to whom the data belongs to, for confidentiality reasons. Nonetheless, please note that the algorithms and calculations included in the model shared for consultation are identical to the ones considered in the internal version with the complete dataset handled by DBA to ensure a full review is possible.

Important note 2: The R model is strictly formed by the files in the folder APP_FOLDER/DBA_App/app/shiny. Any file outside this location is only delivered so the R model may work independently of having R installed in the computer. These files, packages and subroutines have not been developed by Axon or DBA.

Setting up the R model

The R model includes two ways to execute: i) via the standalone app and ii) via RStudio. We recommend that the model is ran with operating system "Windows 10".



The standalone app is completely self-contained program that has everything necessary to be executed. The first step that the user should follow is to un-zip the app into a local folder. Then, to start the app, the user must click on the file named "LRAIC_Rmodel.bat" on the app folder. Note that the opening process of the app takes between 1-5 minutes in the "sample" model (we do not recommend executing the models with the larger confidential datasets with the standalone app, R Studio should be used instead), as it loads the dataset. Upon complete loading, the R model pops-up a window letting the user know that the process has been completed.

The execution through RStudio provides the user with more visibility, however, it requires a longer set-up process. To set up the model via RStudio to access and review the scripts involves the following steps:

- ▶ **Step 1. Installing RStudio.** To install RStudio, please download 'RStudio desktop' from the following link: <https://rstudio.com/products/rstudio/>. After the file has been downloaded, please follow the set-up guide to install the program. Please, note that the model has been developed using RStudio version 1.2.5033. We recommend using this version to ensure full compatibility with the model.
- ▶ **Step 2. Installing R.** After installing RStudio you will be required to install R. You may download R from the official source¹⁰. Please, note that the model has been developed using R version 3.6.3. We recommend using this version to ensure full compatibility with the model.
- ▶ **Step 3. Installing the required packages.** After installing R, and before opening the app, you will be required to install the packages included in the scripts. To ensure compatibility, we recommend installing the same packages and versions utilised when building the model. To do so, please follow the following steps:
 - **Substep 1.** Look up where the libraries are stored in the computer. For this purpose, include the following command in RStudio's console: `.libPaths()`. In turn, RStudio will return the address (or addresses) where the libraries are stored in the computer.
 - **Substep 2.** Close RStudio and do not open it again until requested.

¹⁰ Source: R version 3.6.3 - <https://cran.r-project.org/bin/windows/base/old/3.6.3/>



- **Substep 3.** *[Only necessary if not performing a fresh installation of R]* Uninstall R. This may be done through the control panel of the computer. Once uninstalled, verify all parts of R the library path that has extracted in substep 1 is empty and does not include any file. If any file found in any of this location, please manually delete these files. After this procedure has been complete, please re-install R (see step 2 above).
 - **Substep 4.** Install the required packages. Go to the model folder “\LRAIC_Rmodel\app\library” and copy all the contents of the folder. Then paste these files in the computer library path found in Step 1.
 - **Substep 5.** Open R Studio and execute the same code as in step 1: `.libPaths()`. Ensure that the path returned by this command is the same as the one used in substep 4.
- **Step 4. Create a new project.** In order to ensure the scripts work properly, a new project shall be created in RStudio. To do so, click on the ‘File’ tab in RStudio, then on ‘New Project...’. After this, in the new window will pop-up, click on the ‘Existing Directory’ option.

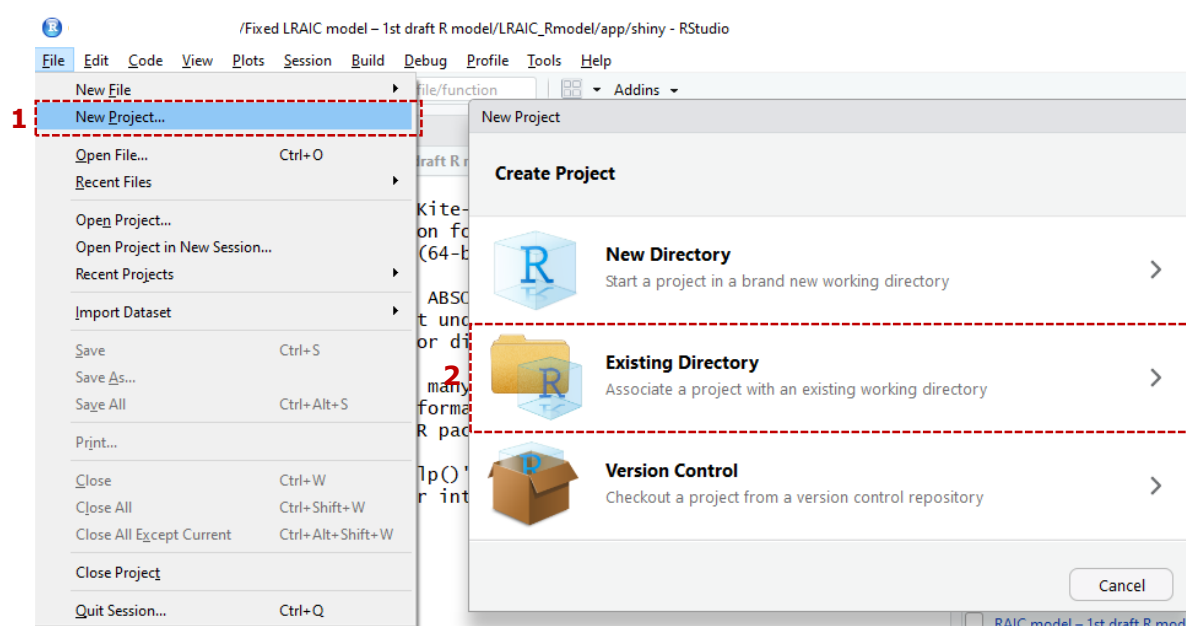


Exhibit 6.4: Overview of the process to create a new project [Source: Axon]

After this, a new window will request the existing address of the project. In this case, the user shall select in the ‘Project working directory’ the following address:

`APP_FOLDER/LRAIC_Rmodel/app/shiny`



Where 'APP_FOLDER' represents the location of the folder where the .zip file of the app has been unzipped. After this folder has been selected, click on the 'Create Project' button.

- ▶ **Step 4. Opening the app (optional).** After the project has been created, the user may interact with the scripts included (please see section 'Annex B - R Model technical documentation' for further details on the scripts). In addition, the user may open the app within RStudio to interact with it. To do so, the user shall open the 'global.R' script in the main working folder (double click). Then the user shall press the 'Run App' included in the main RStudio screen.

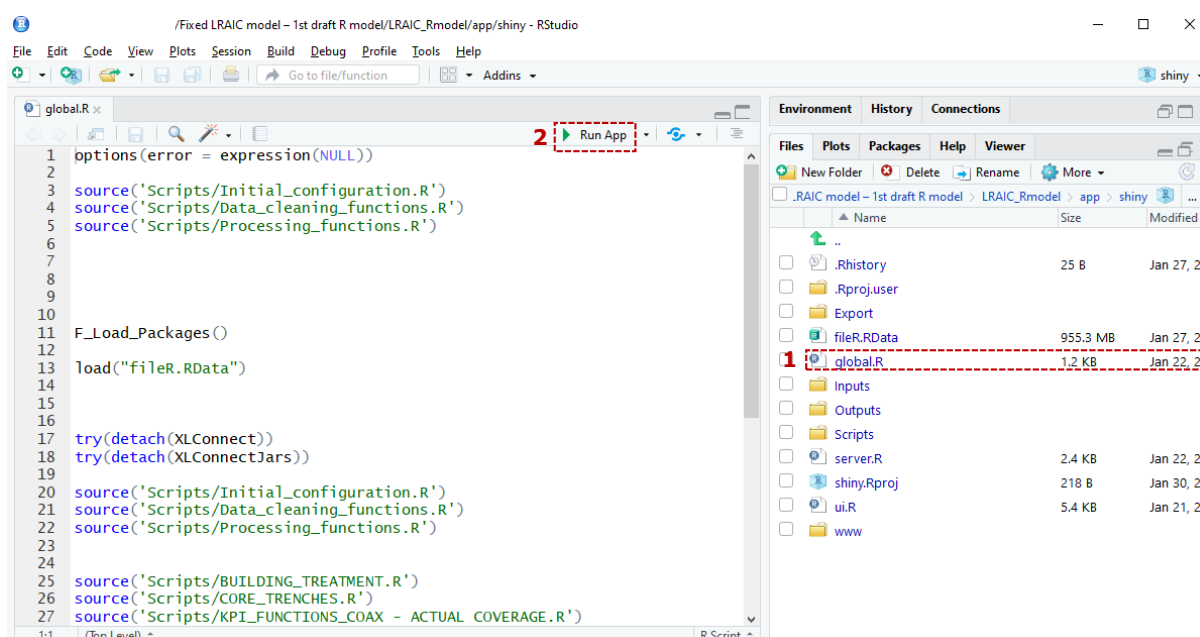


Exhibit 6.5: Opening the app from RStudio [Source: Axon]

Computer requirements

The R model is computationally intensive and requires a reasonably powerful computer in order to run smoothly. To run the model, we recommend the following specifications:

Specification	Sample model	Confidential models
RAM	▶ Recommended: 32 GB	▶ Recommended: 64 GB
Free space in the hard drive	▶ 5 GB	▶ 20 GB

Exhibit 6.6: Minimum and recommended specifications to run the R model [Source: Axon]



A.2. Understanding the control panel

The application represents the main interface user-model. The application uses a web interface that opens in the predefined web browser of the user. The interface is used to select the model's main available options, configure the execution mode, and run the model. The following figure shows a snapshot of the control panel.

AXON **DANISH BUSINESS AUTHORITY**

Fixed LRAIC cost model

Import Control Panel

Regulation mode
Zipcodes

IMPORT REGULATION

Execution Control Panel

Execution mode
4 KPIs

Networks
☒ Civil Infrastructure
☒ Copper
☒ Fibre
☒ Coax
☒ Tx & Core

Operator
TDC

RUN

Export Control Panel

Export mode
Export KPIs

EXPORT

Exhibit 6.7: Control Panel of the R Model [Source: Axon]

As the exhibit shows the control panel consists of 4 main blocks:

- ▶ Import control
- ▶ Execution control
- ▶ Export control
- ▶ Save control

A.2.1. Import control panel

The objective of the import control panel is to allow the user to update some parameters and inputs relevant to the calculations performed in the model. The model allows the update of the regulatory status of the homes included in the exercise.



To do that, the model has different options that shall be selected by the user. There are two different modes to load this input:

- ▶ By postal code
- ▶ By home
- ▶ By electricity company

Once the option of loading selected, the "IMPORT REGULATION" button must be pressed to import the data of the option selected. If no option is selected, the default option is "by electricity company".

The app comes with built in templates that shall be used for the purpose of updating the regulatory status of the postal codes or the homes. These templates are located in the input app folder with the following names:

- ▶ 'ZIPCODE_REGULATION.xlsx', to be used to load the regulatory status at postal code level.
- ▶ 'HOUSES_REGULATION.xlsx', to be used to load the regulatory status at home level.
- ▶ 'Electricity Regulation.xlsx', to be used to load the regulatory status at home level.

Please note that the templates at zipcode and building provided are filled only with sample data and do not contain an exhaustive list of the regulated houses or zip codes. The regulatory status of the buildings included in the current version of the model are based on the proposed approach in DBA's draft market review.

A.2.2. Execution control panel

While the model comes complete pre-executed, the user can decide to update part of the calculations. This should only be done in the case that some of the inputs have been changed (for instance, to update figures in a yearly update or to model a new operator).

Notably, the Execution Control Panel allows to run the execution of the calculations selecting several parameters:

- ▶ Execution Mode
- ▶ Networks



The options under each alternative are presented below.

Execution Mode

The panel displays five options of execution:

Execution Mode	Description	Execution Time	Execution need
KPIs	<p>This mode calculates the KPIs associated to each network (cables, ducts, trenches, etc.). The different modelled networks can be selected.</p> <p>More than one option can be selected at the same time.</p>	<p>The execution time depends on the selected network options and the source dataset:</p> <ul style="list-style-type: none">▶ Full dataset:<ul style="list-style-type: none">• Civil Infrastructure (1-5 mins)• Copper (5-10 mins)• Fibre (15-30 mins)• Coax (1-5 mins)• Transmission (40-70 mins)▶ Sample:<ul style="list-style-type: none">• Civil Infrastructure (<30 seconds)• Copper (<30 seconds)• Fibre (2-3 mins)• Coax (<30 seconds)• Transmission (4-6 mins)	<p>KPIs should be executed in the event that any of the inputs of the R model are updated.</p> <p><i>(e.g. to be performed upon annual update of pricing decisions or when modelling a new operator)</i></p>
Building treatment	<p>This mode prepares the address database to easily extract the KPIs. Some of the calculations performed in this mode involve the calculation of coverage, geotype definition and assigning SDP information to buildings</p>	<ul style="list-style-type: none">▶ Full dataset: 0,5-2h▶ Sample: <30 seconds	<p>This mode should be executed in the case that new inputs are loaded into the model</p> <p><i>(e.g. when modelling a new operator or if updating the source coverage database in the annual update process)</i></p>



Execution Mode	Description	Execution Time	Execution need
Node Splitting	This mode performs the node splitting and the location of SDPs in the different networks.	<ul style="list-style-type: none">▶ Full dataset: 4-10h▶ Sample: <30 seconds	This mode should be used only in the case that different network nodes are updated in the model. <i>(e.g. to be performed when modelling a new operator)</i>
Section treatment	This execution mode converts the generated routes and divides it into sections so the KPIs can calculate the results using a section level of detail.	<ul style="list-style-type: none">▶ Full dataset: 1-3h▶ Sample: <30 seconds	This mode should be used only in the case that different network nodes are updated in the model. <i>(e.g. to be performed when modelling a new operator)</i>
Generate routes	This execution mode recalculates all routes nationwide.	<ul style="list-style-type: none">▶ Full dataset: >24 hours▶ Sample: 4-5 mins <i>In this execution model, the model prints out logs regarding the status of the model's execution</i>	This mode should be used only in the case that different network nodes are updated in the model. <i>(e.g. to be performed when modelling a new operator)</i>
Full Execution	This execution mode recalculates all the model.	<ul style="list-style-type: none">▶ Full dataset: >24 hours▶ Sample: 4-5 mins	This mode should be used only in the case that different network nodes are updated in the model. <i>(e.g. to be performed when modelling a new operator)</i>

Exhibit 6.8: Execution modes in the R Model [Source: Axon]

All times indicated in the table above have been measured under a controlled environment (112 GB of RAM, octa-core processor with a frequency of 3.2 GHz).

We have observed a variance up to $\pm 50\%$ while rerunning the same calculations in the aforementioned system. Additionally, we have observed that closing open instances running in the background can greatly improve the execution times.

Finally, we have observed a significant reduction of time when the code is executed through the console and not through the interface.**Networks**

This part of the panel is only to be used when execution mode "KPIs" is selected. The following networks can be chosen:



- ▶ Civil Infrastructure
- ▶ Access Copper Network
- ▶ Access Fibre Network
- ▶ Access Coax Network
- ▶ Core and Transmission Network

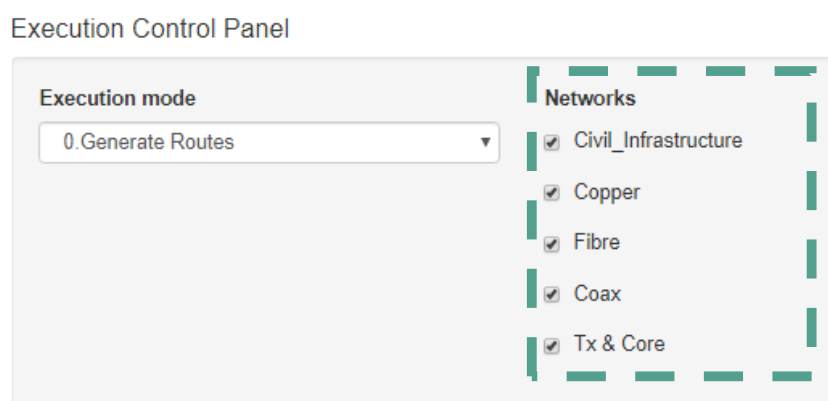


Exhibit 6.9: Execution Control Panel, selection of networks [Source: Axon]

A.2.3. Export control panel

The Export Control Panel creates Excel files containing the latest results of the calculations performed in R. Results are exported to the following folder: APP_FOLDER/DBA_App/app/shiny/Export

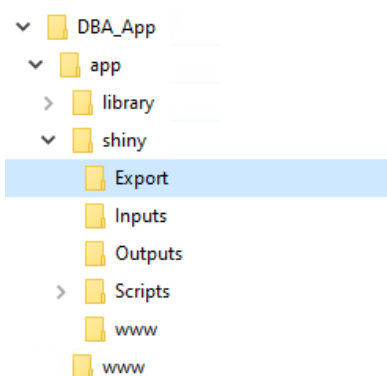


Exhibit 6.10: Folder Tree where to find the exported data [Source: Axon]

The results of this export can be loaded in the Excel model.



A.2.4. Save control panel

Finally, this panel allows to save the results of the latest execution into a dataset pressing on the "SAVE" button. This action is only required when executing some of the algorithms presented in section 'A.2.2 - Execution control panel'. Results can be exported without saving the dataset.

Please note that this action takes roughly 30 minutes for the full set and around 10 minutes for the sample set and that it cannot either be cancelled or reversed.

Warning: if this option is stopped through command line, task administrator or other administration tools, it may corrupt the existing dataset and cause loss of data.

A.3. Expert access to execution scripts

The scripts included in the R model are distributed into three main groups:

- ▶ Interface scripts
- ▶ Link scripts
- ▶ Backend scripts

The functions and scripts included in each group are presented below.

A.3.1. Interface scripts

These scripts are responsible of building the app interface. The scripts are included in the following folder:

APP_FOLDER/LRAIC_Rmodel/app/shiny

Where 'APP_FOLDER' represents the location of the folder where the .zip file of the app has been unzipped.

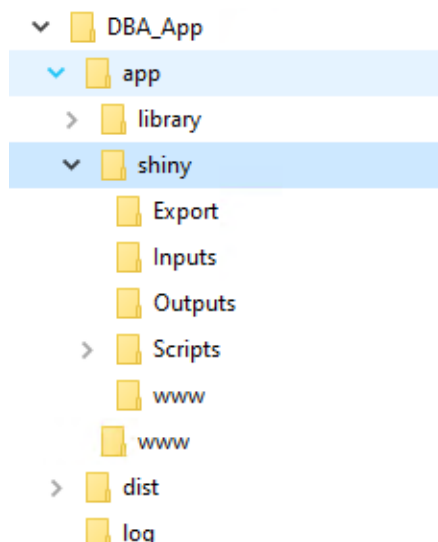


Exhibit 6.11: Folder tree where to find the interface scripts [Source: Axon]

The scripts included in this group are described in Annex B section B.1 Interface block.

A.3.2. Link scripts

The scripts included in this group are responsible for grouping the main functions executed when using the interface and linking them with the underlying functions they involve (please, see 'A.3.3– Backend scripts', for further details). Each of the scripts in this is related to one of the execution options of the interface.

The scripts are included in the following folder:

APP_FOLDER/LRAIC_Rmodel/app/shiny/Scripts/Blocks

Where 'APP_FOLDER' represents the location of the folder where the .zip file of the app has been unzipped.

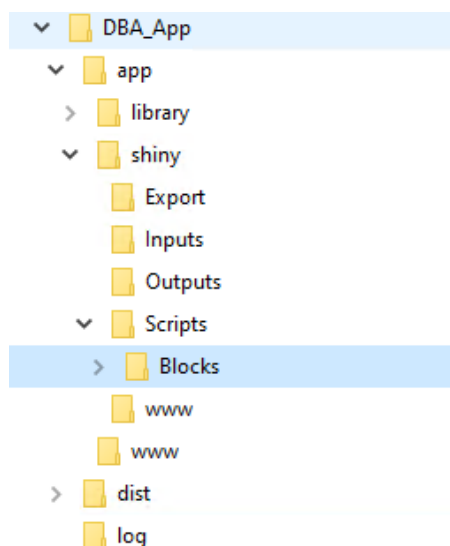


Exhibit 6.12: Folder tree where to find the link scripts [Source: Axon]

The scripts included in this group are described in Annex B.

A.3.3. Backend scripts

This group of scripts gather all other functions and scripts used in the R model.

The scripts are included in the following folder:

APP_FOLDER/LRAIC_Rmodel/app/shiny/Scripts

Where 'APP_FOLDER' represents the location of the folder where the .zip file of the app has been unzipped.

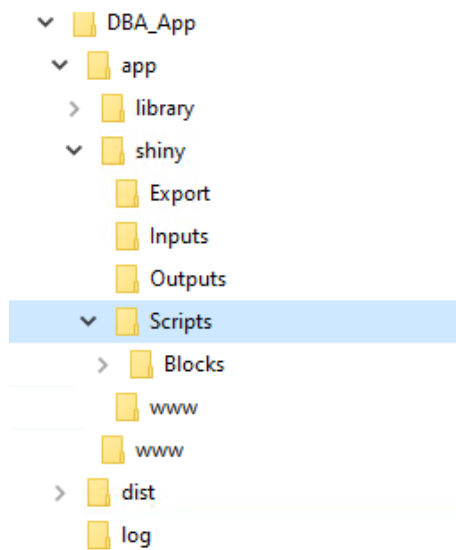


Exhibit 6.13: Folder tree where to find the backend scripts [Source: Axon]

The scripts included in this group are described in Annex B.



Annex B. R Model technical documentation

The R model is structured in different blocks, each one is used to calculate information that is needed to finally export the KPIs to the excel model. The following diagram presents the workflow of the R model and how the blocks are interconnected.

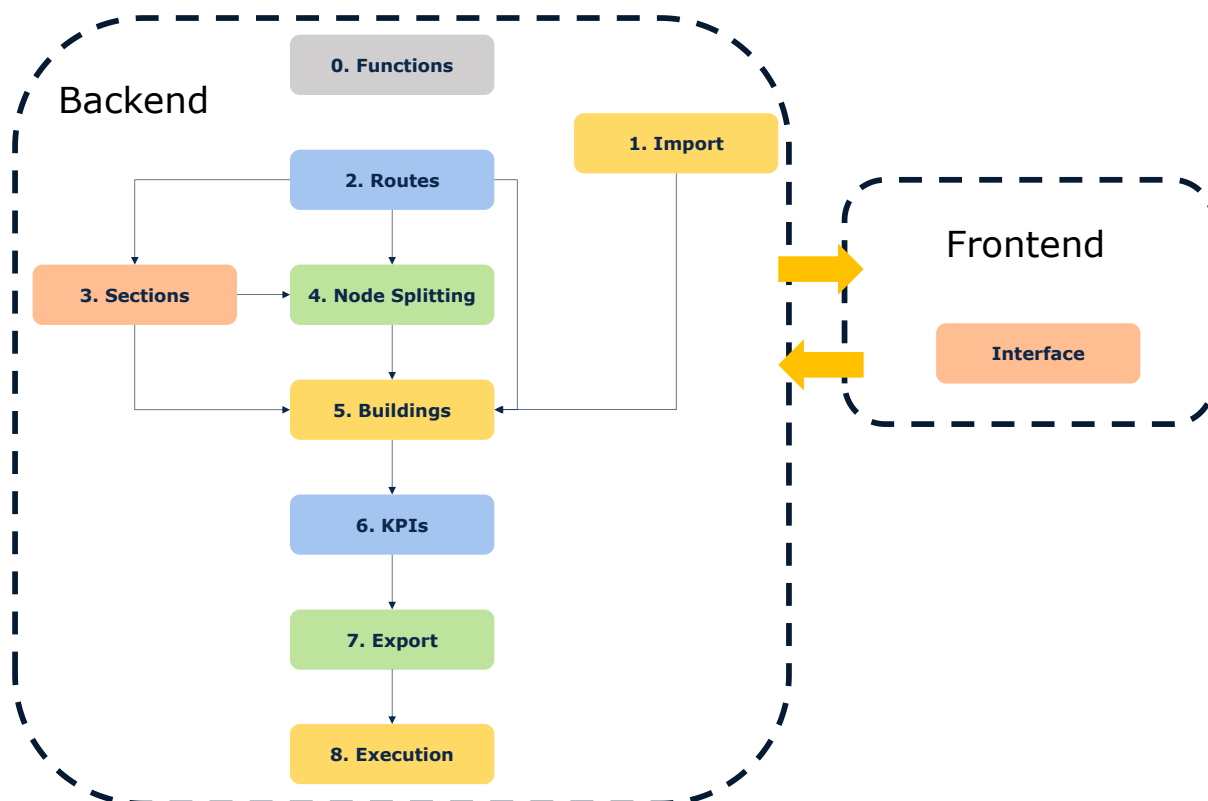


Exhibit 6.14: R Model Structure [Source: Axon]

As shown in the diagram above the R model is divided in 9 different blocks. These blocks are:

- ▶ Interface block: block in charge of the interface of the model.
- ▶ Block 0: Functions: This block gathers functions uses in the rest of blocks.
- ▶ Block 1: Loading Inputs: This block is used to load external inputs into the model through the interface.
- ▶ Block 2: Route generation: this block is used to calculate the routes between the different network locations



- ▶ Block 3: Sections: This is a support block that treats the data of the routes, so it is useable for the KPI calculation
- ▶ Block 4: Node Splitting: This block is in-charge of adding nodes to the network based on a series of rules
- ▶ Block 5: Buildings: This block gets all the data calculated in previous blocks and prepares the buildings, so they are ready for the KPI calculation
- ▶ Block 6: KPIs: This block is the one in charge of calculating the KPIs.
- ▶ Block 7: Export: This block is in-charge of exporting the results of the KPIs into the excel
- ▶ Block 8: Execution: This block is used to perform some minor adjustments in the calculation flow utilised to run the R model for Norlys (which includes two sub-models, one for the fibre network and one for the coax network).

The following sections explain each one of the blocks, the scripts that are contained in them.

B.1. Interface block

The interface block is constituted by three scripts which do not perform any calculation and are only set to execute the interface. The three scripts are:

- ▶ global.R: this script loads the functions (B.1.Interface block) and loads the dataset.
- ▶ server.R: This script commands the actions performed on the interface. It links the front-end with the back-end.
- ▶ ui.R: This script defines the aspect of the interface.

B.2. Block 0: Functions

This block is mainly formed by functions that will be used in other blocks. No calculation is performed in this block.

0A FUN DATA CLEANING.R

This script contains several functions used for loading and prepare data to be used during the calculation of routes. Particularly, it contains the following five functions:



- ▶ **F_Load_Nodes:** This function loads the different nodes of the network into the dataset. This function uses the nodes files loaded into the inputs folder, the EPSG¹¹ code of the dataset and a value to determine if the loaded coordinates should be converted to the EPSG code of the system.
- ▶ **F_Load_Regions:** This function loads the geometric regions of the nodes (if available) into the dataset. As an input, it takes the source file folder and the EPSG code.
- ▶ **F_Load_Regions_Boundaries:** This function takes the regions of the nodes and the EPSG code of the previous function and then loads the region boundaries, as a spatial polygon. Additionally, this function repairs any region boundary which is not a closed polygon.
- ▶ **F_Load_Roads:** This function is in charge of loading the roads into the dataset.
- ▶ **F_Generate_Roads_Unique:** This function uses the roads table and extracts the real number of unique roads.

0B_FUN_INITIAL_CONFIGURATION.R

This script contains several functions that prepare the environment to properly execute the rest of the functions.

- ▶ **F_Install_Packages:** This function installs the required packages into the R environment. This function does not take any input.
- ▶ **F_Load_Packages:** Once the packages are installed, this function loads the required packages into the environment.
- ▶ **F_Load_Parameters:** This function loads the required information in terms of folder location into the R environment.
- ▶ **F_Load_Initial_Configuration:** This function prepares the environment prior to execution. It sets the visualization mode, the usage of exponential numbers and the default folder location.

¹¹ The code extracted from the EPSG Geodetic Parameter Dataset (also EPSG registry) is a public registry of spatial reference systems, Earth ellipsoids, coordinate transformations and related units of measurement. This registry enables the model to easily establish associations between different coordinate systems.



0C FUN PROCESSING FUNCTION.R

This script contains several functions that are used throughout the R model but more specifically focused on functions used for the route calculation. The function used are:

- ▶ **F_Identifies_Nearest_Point_Generic:** this function returns the closest point to a set of origins from a set of destinations.
- ▶ **F_Add_Clusters:** this function returns a set of clusters from a set of points and a maximum size of the cluster.
- ▶ **F_Add_Clusters_per_Specific_Region:** similar function to **F_Add_Clusters** but limiting it by region.
- ▶ **F_Calc_Distance_to_Road:** This function uses the set of coordinates, the roads, the intersections, minisections and buildings to calculate the distance from the said origin to the road by extrapolating that point.
- ▶ **F_Calc_Distance_to_Road_by_Area:** Similar function to the **F_Calc_Distance_to_Road** but limiting the calculation per CO area instead of nationwide.
- ▶ **F_Identifies_Nearest_Roads_Generic:** This function uses a set of origins and the road coordinates to look for the closest road point and returns it.
- ▶ **F_Calc_Nodes_per_Type:** This function is used to disaggregate all the nodes into the different categories (CO, MDF, PDP or CMC) and calculate several parameters. In addition, this function assigns nodes to different geometries and calculates the area of the said geometry.
- ▶ **F_Identifies_Nearest_Top_Points_Generic:** This function is similar to the function **F_Identifies_Nearest_Point_Generic** but instead of returning one destination per origin it returns the desired amount of closest destinations.
- ▶ **F_Generate_Grafo:** This function creates a graph using a set of connected sections as edges.
- ▶ **F_Convert_Long_Lat_to_UTM_by_EPSG_to_EPSG** This function converts coordinates from one coordinate system to another.
- ▶ **F_Convert_Long_Lat_to_UTM_by_CRS_to_EPSG:** This function returns a CRS object from a set of coordinates with a specific coordinate system.
- ▶ **F_ProjectPoint:** This function returns a projection point given a point and a line.



- ▶ **F_Calculates_Way_to_Go_Generic:** This function returns the route given a graph, a set of origins and a destination.
- ▶ **F_Verify_Intersections_in_Grapho:** This function validates that the vertices of the graph are correct.
- ▶ **F_Calc_Nearest_Intersections:** This function calculates the N (number to be specified) number of nearest intersections to one coordinate.
- ▶ **F_Get_Random_Points:** This function returns a number of random sample from a set of points.
- ▶ **F_Valid_Intersections_from-Origin_to-Destination:** This function verifies that it exists a valid route between one origin one destination in a graph.
- ▶ **F_Generates_Cluster:** This function given a number of points, a maximum number of points and the centre, generates clusters of the said points
- ▶ **F_Calc_Distance_to_Node:** This function calculates the distance of several intersection origins to the node intersection destination. This is done through a graph and using other functions in this same script.
- ▶ **F_Calc_Distance_to_Node_by_Area:** This function is equivalent to the **F_Calc_Distance_to_Node** but performs the calculation at area level and then joins all the results from the different areas.
- ▶ **F_Calculates_New_Area_for_Nodes:** This function is a support function. It basically provides a bigger area to each one of the nodes in the case a route cannot be find inside the pre-defined area.
- ▶ **F_Calculates_New_Area_for_Sections:** This function is a support function. It basically provides a bigger area to each one of the nodes in the case a route cannot be found inside the pre-defined area. This function is specifically to temporary modify the assignation of sections to achieve the desired result.
- ▶ **F_Add_Distance_by_Road_Considering_Actual_Cluster:** This function returns the distance by road given a number of origins. The difference is that it gets the area of calculation and enlarges it based on the number of origins and destination in order to ensure that the calculation may be performed.
- ▶ **F_Add_Distance_Road_Interpolated_to_Intersection:** This function uses the minisections and previous functions to calculate the distance of the interpolated point to the closest road point and then to calculate the distance between that road point to the closest intersection.



- ▶ **F_Add_Distance_Road_Interpolated_to_Intersection_by_Area:** Same function as the **F_Add_Distance_Road_Interpolated_to_Intersection** but performs the calculations separately by each CO area in order to minimize computational utilization.
- ▶ **F_Calculates_Building_Coax_to_Co_Identif:** This function takes the information calculated by each building, CMC and CO to build the final **T_Buildings_Coax** table with all the needed information.
- ▶ **F_Calculates_Building_to_Co_Identif:** This function takes the information calculated by each building, PDP, MDF and CO to build the final **T_Buildings** table with all the needed information.
- ▶ **F_Calculates_PDP_to_CO_Identif:** This function takes the information calculated by each PDP, MDF and CO to build the final **T_Nodes_PDP** table with all the needed information.
- ▶ **F_Calculates_CMC_to_CO_Identif:** This function takes the information calculated by each CMC and CO to build the final **T_Nodes_CMC** table with all the needed information.
- ▶ **F_Calculates_MDF_to_CO_Identif:** This function takes the information calculated by each MDF and CO to build the final **T_MDF_CMC** table with all the needed information.
- ▶ **F_Add_Geometry_ID:** This function uses spatial objects and boundaries to assign a spatial point into a region. This function is used to assign buildings, intersections, nodes to their CO area.
- ▶ **F_Add_Geometry_ID_to_Section_Considering_Int:** This function uses spatial objects and boundaries to assign a spatial point into a region. This function is used to assign sections to their CO area.
- ▶ **F_Add_Information_of_CO:** This function is used to assign the CO to a table which contains the CO area. For example, a CO area may have several regions (islands) so the same CO has to be assigned to all the areas which are used by the said CO.
- ▶ **F_Improve_Geometry_ID:** This function is used after the geometry is assigned to ensure that the points used are correctly positioned. In addition, it uses the intersections correctly placed to improve the assignation of the geometry
- ▶ **F_Improve_Zipcode:** Similar to the function **F_Improve_Geometry_ID**. This function is used on those points that does not have a postal code associated and therefore it is assigned by distance to the closes point which already has a postal code.
- ▶ **F_Add_Distance_Road_Interpolated_to_Intersection_by_Area_AO:** This is the same function as the **F_Add_Distance_Road_Interpolated_to_Intersection_by_Area** with two



minor changes. The changes are it has been improved the amount of calculation per internal loop and it has been added the possibility that the distance is look on all the graph a not only I part of it in the case the Voronoi area has produced a not connected graph.

- ▶ **F_Identifies_Nearest_Roads_Generic_2:** It performs the same calculations as the **F_Identifies_Nearest_Roads_Generic** but with an improved algorithm for smaller datasets.
- ▶ **F_Calc_Distance_to_Node_by_Area_AO:** This function is the same as the one **F_Calc_Distance_to_Node_by_Area** with some improvements on the calculation for small datasets.
- ▶ **F_Calc_Distance_to_Node_by_Area_AO2:** This function is the same as the one **F_Calc_Distance_to_Node_by_Area** with the added functionality of ensuring that the calculation is performed in a larger area in case the geometry created by the Voronoi algorithm is a not connected graph.
- ▶ **F_Calc_Distance_to_Road_by_Area_2:** this function is the same as **F_Calc_Distance_to_Road_by_Area** but where the function **F_Identifies_Nearest_Roads_Generic** has been changed to use the more efficient for smaller datasets **F_Identifies_Nearest_Roads_Generic_2**.

0D FUN NODE ASSIGNATION SAMPLE.R

This script contains the function **F_TX_node_Treatment_sample** which is a similar function to the **F_TX_node_Treatment** and is only used when executing the sample (only two COs) model. This function treats the data relative to the core and transmission nodes and prepares it to be processed in other function. It changes the column names and assigns to each node its closes intersection. The main difference between the function on this script and the other one is that the assignation of intersection is merely done as a direct distance instead of considering the road. This is due to the smaller dataset included in the sample model.

0E FUN CMC TREATMENT.R

This script contains two functions related to the calculation of the CMC location and treatment of the dataset generated by these CMCs and coaxial network.



- ▶ **F_CMC_Generation_Route2:** This function is one of the functions in charge of modifying the routes calculated in the script `F_SectionNode_Generation_coax` to accommodate the information of the CMCs calculated in the previous function. This function is called in the script `4C_CALC_CMC_TREATMENT.R` and takes as an input the Buildings and sections (`T_Buildings` and `T_Sections`), as well as, the results of the previous function. The output of this function is an intermediate table which contains the routes from the building to the CMC for each building.
- ▶ **F_CMC_Generation_Route4:** This function converts the information obtained from the function `F_SDP_Generation_Route2` and converts it to the same format as the table `T_Sections_Total` so the said table may be modified to contain CMC information.

OF FUN SDP TREATMENT.R

This script contains four functions related to the calculation of the SDP location and dataset treatment of the data generated by these SDPs in the copper and coaxial network.

- ▶ **F_SDP_Generation:** This function is in charge of calculating the actual locations of the SDPs. The inputs of this function are, the buildings (`T_Buildings` containing the routes Building to PDP), the sections (`T_Sections`) and a value to determine where to put the SDP (`p_value`). The function performs two main calculations. First, it disaggregates the routes into the different intersections assigning to each intersection which building routes pass through it. Once the routes it disaggregates it performs the SPD calculation for each PDP. For each PDP it searches the furthest intersection from the PDP which complies with the requisite set by the `p_value`. Once that intersection is found, an SDP is located in that position and the buildings associated to it are removed from all the intersections further up the road. The process is performed iteratively until all the SDPs have been located. This function is executed during the pre-KPI stage and it is called by the `4B_CALC_SDP_CALCULATION.R` script.
- ▶ **F_SDP_Generation_Route2:** This function is one of the functions in charge of modifying the routes calculated in the script `3A_CALC_SECTIONS.R` to accommodate the information of the SDPs calculated in the previous function. This function is called in the scripts `5B_CALC_BUILDINGS_PREPARATION.R` and `5F_CALC_BUILDINGS_PREPARATION_AO.R` takes as an input the Buildings and sections (`T_Buildings` and `T_Sections`) as well as, the results of the previous function. The output of this function is an intermediate table which contains the routes from the building to the SDP for each building.



- ▶ **F_SDP_Generation_Route4**: This function basically converts the information obtained from the function **F_SDP_Generation_Route2** and converts it to the same format as the table **T_Sections_Total** so the said table may be modified to contain SDP information.
- ▶ **F_SDP_Generation_Route4_AO**: This function performs the same calculations as the **F_SDP_Generation_Route4** but has been optimized for smaller datasets.

0G FUN BUILDINGS CALCULATION.R

This script contains the functions in charge of assigning the degree of urbanization (called geotype in the R Model) and the type of dwelling units to the different buildings (**T_Buildings**). This function is called from script **5B_CALC_BUILDINGS_PREPARATION.R** and **5F_CALC_BUILDINGS_PREPARATION_AO.R**

Both functions **F_Build_Treatment** and **F_Build_Treatment_AO**, have the same inputs: **CO Nodes (T_Nodes_CO)** and buildings (**T_Buildings**). The main difference is that the function **F_Build_Treatment_AO** uses predefined limits based on number of households and area, The first function performs a uses the K-means method to cluster the Cos in function of number of households and area without setting any limit. This is done to ensure that all operators modelled uses the same limits. The functions first assign the geotype to the nodes and then to the buildings that are assigned to. The dwelling type (MDU/SDU) is directly done by setting the value depending on the number of houses per building.

0H CALC EXECUTION CLEANER.R

This script is only used to erase intermediate data tables no longer needed and cleans the RAM memory. This script is executed after the calculation of routes.

0I FUN CORE TRENCHES.R

This script contains five functions in charge of calculating different inputs of routes and trenches of the core and transmission networks.

- ▶ **F_CALC_CORE_PERC**: This function calculates the unique sections covered by any of the four access networks. This function mainly uses the buildings and route sections of each one of the access networks



- ▶ **F_CALC_CORE_PERC2:** This function uses the result of the previous function and the core trenches to calculate the percentage of core trenches that are shared with the access network
- ▶ **F_CALC_CORE_PERC3:** This function is the invers of the **F_CALC_CORE_PERC2**. Us using the core sections and the access sections calculates what percentage per geotype are the access trenches shared with the core trenches.
- ▶ **F_CALC_TX_SHARE:** This function calculates the core cost allocation of core networks between geotypes. In particular, it calculates the said allocation based on the kilometers of trenches.
- ▶ **F_CALC_TX_SHARE_house:** This function calculates the core cost allocation of core networks between geotypes. In particular, it calculates the said allocation based on the number of houses connected to each ring.

0J FUN COAX.R

This script contains the function **F_CALC_KPI_COAX_Scenarios**. This function is in charge of calculating all coax KPIs explained in section 5.3 Coax Access Network. The inputs of this function are mainly the Buildings (**T_Buildings_Coax**), sections (**T_Sections**) and the route sections (**T_Sections_Total_Coax**) .

0K FUN COAX PASSIVE.R

This script contains the function **F_CALC_KPI_Passive_Coax_Scenarios**. This function is in charge of calculating the civil infrastructure for coaxial networks KPIs explained in section 5.4 Other common civil infrastructure elements. The inputs of this function are mainly the Buildings (**T_Buildings_Coax**), sections (**T_Sections**), the route sections (**T_Sections_Total_Coax**) the road width (**V_Road_With**) and road crossing threshold (**p_Threshold_trenches**).

0L FUN COPPER.R

This script contains the function **F_CALC_KPI_COPPER_Scenarios**. This function is in charge of calculating all copper KPIs explained in section 5.1 Copper Access Network. The inputs of this function are mainly the Buildings (**T_Buildings**), sections (**T_Sections**) and the route sections (**T_Sections_Total**), cable assets (**I_Cables_Assets_Copper**), and the splitting factors on different network levels (**V_FDP_Splitting_Copper**, **V_SDP_Splitting_Copper**, **V_PDP_Splitting_Copper**).



0M FUN COPPER PASSIVE.R

This script contains the function `F_CALC_KPI_Passive_Scenarios_Copper`. This function is in charge of calculating the civil infrastructure for copper networks KPIs explained in section 5.4 Other common civil infrastructure elements. The inputs of this function are mainly the Buildings (`T_Buildings`), sections (`T_Sections`), the route sections (`T_Sections_Total`) the road width (`V_Road_With`) and road crossing threshold (`p_Threshold_trenches`).

0N FUN FIBRE PON.R

This script contains two functions `F_CALC_KPI_FIBRE_PON` and `F_CALC_KPI_FIBRE_PON_AO`. This function are in charge of calculating all fibre KPIs explained in section 5.2.Fibre Access Network. The inputs of this function are mainly the Buildings (`T_Buildings`), sections (`T_Sections`) and the route sections (`T_Sections_Total`), cable assets (`I_Cables_Assets_Fibre`), the splitting factors on different network levels (`V_FDP_Splitting_Fibre`, `V_SDP_Splitting_Fibre`, `V_PDP_Splitting_Fibre`) and the duct and cables crossection (`V_area_ducts` and `I_area_cable_Fibre`). The difference between two functions is due to the consideration of SDPs as explained ins section 5.2.Fibre Access Network.

0O FUN FIBRE PASSIVE PON.R

This script contains two functions `F_CALC_KPI_Passive_Fibre_PON` and `F_CALC_KPI_Passive_Fibre_PON_AO`. These functions are in charge of calculating the civil infrastructure for fibre networks KPIs explained in section 5.4 Other common civil infrastructure elements. The inputs of this function are mainly the Buildings (`T_Buildings`), sections (`T_Sections`), the route sections (`T_Sections_Total`) the road width (`V_Road_With`) and road crossing threshold (`p_Threshold_trenches`).

0P FUN KPIS SUBFUNCTIONS.R

This script contains several functions to be used during the calculation of the access network KPIS. The functions are:

- ▶ `Cable_FIBRE_SDP_CO`: This calculates the fibre cables between the SDP and the CO. For this calculation, the function needs as an inputs the different sections per network layer, the different access nodes, the type of cables that may be used and the splitting



factors at each network layer. Basically, this function gets all the sections per network layer and counts the amount of cables needed in the said section. After that all information is then treated by obtaining the number of cables per CO and per geotype and network layer. The final output of this function is the sum of the cables for the different network layers per geotype from the SDP to the CO. as explained in section 5.2 Fibre Access Network for the fibre network the PDPs have been considered as the only SDPs

- ▶ Cable_FIBRE_house_SDP4: This function is equivalent to the function Cable_FIBRE_SDP_CO but considering different network layers
- ▶ cable_Copper_last: This function is a similar function to the one used for fibre (Cable_FIBRE_house_SDP) but in this case the cables to be used are the copper ones. Additionally it calculates the cables from the houses until the MDF.
- ▶ cable_Copper_last: Equivalent function to the cable_Copper_last but it calculates the fibre cable in the copper network from the MDF to the CO
- ▶ SDP_FIBRE_COPPER: This function mainly assigns to each SDP in the network the DP size from the available Distribution point sizes. This function is used for copper and fibre with the difference of the distribution sizes used.
- ▶ PDP_FIBRE_COPPER: same function that the SDP_FIBRE_COPPER function but with the PDP.
- ▶ CO_FIBRE_COPPER: Function to assign CO assets to the CO depending on the available CO assets for the network
- ▶ EXPORT_ASSETS: this function is to put all the KPIs in the same column structure.
- ▶ EXPORT_DISTRIBUTION : this function uses the KPIs calculated to calculate the asset distribution.
- ▶ joints_FIBRE_COPPER: this function uses the cables and the asset joints to calculate the needed joints in different network parts
- ▶ F_over_KPIs: This function is a way to create virtual assets to be used when assigning any assets. For instance, this function if the maximum cable in a network is the one of 100pairs. This function will create virtual assets to ensure that in the case a bigger assets than the biggest assets is needed the function will correctly assign the asset.
- ▶ cable_FIBRE_SDP_CO_PTP: This function is the same function as Cable_FIBRE_SDP_CO with the difference that it ensures that only one cable goes from the MDF to the CO as the MSAN is positioned in the MDF.



- ▶ `cable_FIBRE_SDP_CO_AO` This function is the same function as the `cable_FIBRE_SDP_CO` with the difference that all the SDPS are considered as such.
- ▶ `cable_FIBRE_SDP_CO_PTP_AO`: This function is the same function as `Cable_FIBRE_SDP_CO_PTP` with the difference that all the SDPS are considered as such.
- ▶ `cable_FIBRE_SDP_CO_AO_2`: This function is the same function as the `cable_FIBRE_SDP_CO` with the difference that all the SDPS are considered as such.

0Q FUN TEMPLATE.R

This script contains the function `Geotype_to_template` which uses the template provided in the inputs to convert raw data into data that fits into the excel template

0R FUN CORE.R

This script contains several functions used for the calculation of core KPIS.

- ▶ `F_TX_node_Treatment`: Function that treats the Core/Distribution/aggregation/L3 access nodes in order to be able to correctly work with them. It basically changes column names and perform minor operations
- ▶ `F_TX_link_Treatment`: similar to the previous function. This function performs adjustments to the core links uploaded into the dataset.
- ▶ `F_TX_link_access_Treatment`: This function uses the nodes per ring and the connection of those ring to other hierarchy levels to calculate the links of the L3 access links. This algorithm uses the TSP algorithm with a predefined start and end.
- ▶ `F_tx_node_route_seq`: This function is used to calculate the route intersections followed by each of the transmission links.
- ▶ `F_tx_node_route_dist`: This function is used to calculate the route distance of each of the transmission inks.

0S FUN SECTION DISAGGREGATION.R

This function contains two functions in charge of disaggregated the route information present in the Buildings table (`T_Buildings` and `T_Buildings_Coax`) column `Seq_Int_to_go`. The functions input are the buildings and sections tables. The functions perform the calculation of disaggregating once for each network level (Building-PDP, PDP-MDF and



MDF-CO for the Copper network, and Building-CMC and CMC-CO for the coaxial). The difference between the two function is which network is used with which function. The function F_SectionNode_Generation is used for the copper and fibre networks and the functions F_SectionNode_Generation_Coax is used for the coaxial network.

0T_FUN_SECTION_DISAGGREGATION_CORE.R

This script performs the same calculations as the ones in the function F_SectionNode_Generation but for the routes on the Transmission and core networks. The difference is that instead of the buildings the input of this function is the routes calculated and transmission link level.

0X_FUN_FIBRE_PTP.R

This script contains two functions F_CALC_KPI_FIBRE_PTP and F_CALC_KPI_FIBRE_PTP_AO. These function is in charge of calculating all fibre KPIs explained in section 5.2.Fibre Access Network. The inputs of this function are mainly the Buildings (T_Buildings), sections (T_Sections) and the route sections (T_Sections_Total), cable assets (I_Cables_Assets_Fibre), the splitting factors on different network levels (V_FDP_Splitting_Fibre, V_SDP_Splitting_Fibre, V_PDP_Splitting_Fibre) and the duct and cables crossection(V_area_ducts and I_area_cable_Fibre) . The difference between two functions is due to the consideration of SDPs as explained in section 5.2.Fibre Access Network.

0Y_FUN_FIBRE_PASSIVE_PTP.R

This script contains two functions F_CALC_KPI_Passive_Fibre_PTP and F_CALC_KPI_Passive_Fibre_PTP_AO. These functions are in charge of calculating the civil infrastructure for fibre networks KPIs explained in section 5.4 Other common civil infrastructure elements. The inputs of this function are mainly the Buildings (T_Buildings), sections (T_Sections), the route sections (T_Sections_Total) the road width (V_Road_Width) and road crossing threshold (p_Threshold_trenches).



B.3. Block 1: Loading Inputs

This block is formed by three scripts used to import new data into the dataset.

1A IMP COVERAGE FIBRE.R

This script opens a file containing the number of houses covered by fibre (PON and PTP) per year and loads it into the dataset.

1B INP REGULATION HOUSES.R

This script opens a file containing houses which are regulated or non-regulated and loads it into the dataset.

1C INP REGULATION ZIPCODES.R

This script opens a file containing zip codes which are regulated or non-regulated and loads it into the dataset.

1C INP REGULATION ELECTRICITY.R

This script changes the regulation type of the buildings depending on the electricity company

B.4. Block 2: Route generation

This Block is formed by the scripts in charge of calculating the routes between buildings and networks nodes on the access networks.

2A CALC GENERATE ROUTES.R

This script is in-charge of calculating the routes between the house and the CO. This script is a general script which call different scripts and functions in order to perform all the necessary calculations. The steps followed by this script can be summarized in the following steps:

- ▶ First it resets the building and node information by erasing all the information that have been calculated in previous iterations



- ▶ Then it proceeds to calculate the distance from each network level from building to node and node to node. The functions in charge of this calculation are: `F_Calc_Distance_to_Node_by_Area_AO2` and `F_Add_Distance_Road_Interpolated_to_Intersection_by_Area_AO`.
- ▶ Finally all data is gathered in the buildings and nodes tables (`T_Buildings`, `T_Nodes_PDP`, `T_Nodes_MDF`, `T_Buildings_Coax` and `T_Nodes_CMC`) this is done through the functions: `F_Calculates_Building_to_Co_Identif`, `F_Calculates_PDP_to_CO_Identif`, `F_Calculates_MDF_to_CO_Identif`, `F_Calculates_Building_Coax_to_Co_Identif` and `F_Calculates_CMC_to_CO_Identif`.
- ▶ Finally it removes all information generated which will not be reused.

2B CALC GENERATE ROUTES AO.R

This script performs the same calculations as the script `2A_CALC_GENERATE_ROUTES.R` but with the following minor differences:

- ▶ The nodes are not pre-loaded and can be changed. This is done to ensure that the script may be used for different operators. Additionally, the script calculates the distance node to road. This is done through the functions: `F_Load_Nodes`, `F_Add_Geometry_ID`, `F_Calc_Distance_to_Road_by_Area2` and `F_Calc_Nodes_per_Type2`.
- ▶ Additionally the script calculates with a Voronoi algorithm which intersections, buildings and sections have to be associated in each node. This is done in the script `2D_CALC_AREA.R`.
- ▶ Secondly the script assigns to each building a geometry (CO Area) and calculates the distance from each building to the road. This is done through the functions `F_Calc_Distance_to_Road_by_Area_2`.
- ▶ Some functions have been modified to improve the performance in datasets smaller than the one for the incumbent operator and also to deal with problems caused by the Voronoi approach on the buildings in the frontier of the area. The functions in charge of these calculation are `F_Calc_Distance_to_Node_by_Area_AO2` and `F_Add_Distance_Road_Interpolated_to_Intersection_by_Area_AO`.

2C CALC ADD ROUTE PROCESSING.R

This script contains a way to process sections that due to the Voronoi algorithm have not been assigned a CO area and assigns one to them.



2D_CALC_AREA.R

This script is in charge of performing the Voronoi algorithm described in section 4.4 Characterization of the coverage input. The script is organized in several steps:

- ▶ The script performs the Voronoi algorithm adding each intersection a CO area. This is done as explained in 4.4 Characterization of the coverage input
- ▶ Based on the Voronoi the script calculates the actual area in m² of each CO area

After the information is added to the each one of the tables where needed (e.g. the CO area which each intersection and section pertains)

2E_CALC_INITIAL_PROC_AO.R

This script is in charge of performing a series of pre-processing of data for Norlys. Basically, this script ensures that the number of buildings considered are the ones that will actually be covered by the end of the modelling period. This is done to minimize calculations and improve model efficiency. During this process, an information such as actual and future house deployment is stored for use in steps further down the process.

2F_CALC_OUTSIDE_VORONOI.R

The Voronoi algorithm has been limited to check intersections up to 30km in direct distance from each node. This is done to improve efficiency and minimize errors. This script is to calculate and prepare some calculations for the cases where a building is left out of the boundaries of the areas created by the Voronoi algorithm.

2G_CALC_GENERATE_ROUTES_SAMPLE.R

This script is virtually identical to the script 2A_CALC_GENERATE_ROUTES.R with minor difference on efficiency and redundancy adjustments.

2H_CALC_GENERATE_ROUTES_CALL.R

This script calls either the script 2A_CALC_GENERATE_ROUTES.R, the scripts 2B_CALC_GENERATE_ROUTES_AO.R and 2E_CALC_INITIAL_PROC_AO.R or the script 2G_CALC_GENERATE_ROUTES_SAMPLE.R depending on the operator being modelled.



2I CALC GENERATE ROUTES AO COAX.R

This script performs the same calculations as the script 2A_CALC_GENERATE_ROUTES.R but with the following minor differences:

The nodes are not pre-loaded and can be changed. This is done to ensure that the script may be used for different operators. Additionally, the script calculates the distance node to road. This is done through the functions: F_Load_Nodes, F_Add_Geometry_ID, F_Calc_Distance_to_Road_by_Area2 and F_Calc_Nodes_per_Type2.

Additionally, the script calculates with a Voronoi algorithm which intersections, buildings and sections have to be associated in each node. This is done in the script 2J_CALC_AREA_COAX.R

Secondly the script assigns to each building a geometry (CO Area) and calculates the distance from each building to the road. This is done through the functions F_Calc_Distance_to_Road_by_Area_2.

Some functions have been modified to improve the performance in datasets smaller than the one for the incumbent operator and also to deal with problems caused by the Voronoi approach on the buildings in the border of the area. The functions in charge of these calculation are F_Calc_Distance_to_Node_by_Area_AO2 and F_Add_Distance_Road_Interpolated_to_Intersection_by_Area_AO.

2J CALC AREA COAX.R

This script is in charge of performing the Voronoi algorithm for the coax network described in section 4.4 Characterization of the coverage input. The script is organized in several steps:

- ▶ The script performs the Voronoi algorithm adding each intersection a CO area. This is done as explained in '4.4 Characterization of the coverage input'.
- ▶ Based on the Voronoi the script calculates the actual area in m² of each CO area

After the information is added to the each one of the tables where needed (e.g. the CO area which each intersection and section pertains)



2K CALC INITIAL PROC OTHER.R

This script is in charge of performing a series of pre-processing of data for alternative operators. Basically, this script selects the buildings considering the utility company with a footprint in the area of study. This is done to minimize calculations and improve model efficiency.

B.5. Block 3: Sections

This block is solely formed by one script in charge of converting the data calculated in Block 2: Route generation into data at node/building and section level.

3A CALC SECTIONS.R

This script is in charge of disaggregating the route information generated during the route calculation into a table at building/node and section level. Basically, it is used to have what are the actual sections being used on each network level. This is done for Copper and fibre on one hand and coax on the other. The said calculations are performed by calling functions F_SectionNode_Generation and F_SectionNode_Generation_coax

B.6. Block 4: Node Splitting

This block is mainly formed by scripts in charge of calculating positions of new nodes due to node splitting.

4A CALC NODE SPLITTING.R

This script calls the scripts that calculate the node splitting perform on the different network. The scripts called are 4B_CALC_SDP_CALCULATION.R which calculates the SDP position for the copper and fibre network and the 4C_CALC_CMC_TREATMENT.R which calculates the CMC splitting to be performed on the Coax network.

4B CALC SDP CALCULATION.R

This script calls the function F_SDP_Generation in charge of calculating the new positions of the SDPs.



4C CALC CMC TREATMENT.R

This building is in charge of performing a node splitting in the coaxial network in order to ensure an average of 200 households per CMC. In order to perform the said calculation it sets a maximum limit of 500 households passed per CMC.

The calculation is done as explained in section '4.2 - Location of additional network nodes'. Once the new CMC positions are calculated the script modified the routes calculated previously. It basically changes which buildings are connected to which CMC and the route description. Additional functions used are F_CMC_Generation_Route2 and F_CMC_Generation_Route4.

B.7. Block 5: Buildings

This block is primarily formed by scripts in charge of calculating the remaining information needed for the building in order to calculate the KPIs. The information calculated includes coverage, geotypes or node information

5A CALC BUILDING TREATMENT.R

This script calls either the script 5B_CALC_BUILDINGS_PREPARATION.R or the script 5F_CALC_BUILDINGS_PREPARATION_AO.R depending on the dataset being used.

5B CALC BUILDINGS PREPARATION.R

This script prepares the buildings and the network node tables that will be used in the calculation of the KPIs. This script performs the following action:

- ▶ Updates inputs independent from the route's calculations. This is done by calling the script 5C_INP_ADDITIONAL_INPUTS.R
- ▶ erases several columns and parameters that will be re-calculated in order to avoid problems during the calculations performed
- ▶ Adds to the CO nodes (T_Nodes_CO) the region and the regulation
- ▶ Calculates the total distances from home to CO for each building (T_Buildings column Total_Distance)
- ▶ Applies the information of regulation and region to each building



- ▶ Extracts the information of SDPs calculated. Modifies the table T_Sections_Total from script F_SectionNode_Generation to change which sections from each building pertain on each network level. It also adds the SDP information to each building.
- ▶ Adds the information from Geotype and dwelling type to the buildings table (T_Buildings and T_Buildings_Coax). This is done through function F_Build_Treatment on script 0G_FUN_BUILDINGS_CALCULATION.R.
- ▶ Loads the coverage database and extract the information from the desired operator for each network.
- ▶ Loads which buildings are covered by third party operators, this is done through the script 5E_CALC_FIBRE_COVERAGE_OTHER_OPERATORS.R.
- ▶ Calculates the yearly coverage of fibre. Basically, assigns which buildings will be covered on each year. This is done by calling the script 5D_CALC_FIBRE_COVERAGE.R.
- ▶ Finally, all information is gathered in the main tables and used information is erased from the data set.

5C_INP_ADDITIONAL_INPUTS.R

This script is used to load a number of additional inputs that are required after the calculation of the routes is performed in the scripts from blocks 1 to 4. The following information is loaded:

- ▶ General dimensioning parameters: I_Parameters
- ▶ Region information: I_Region
- ▶ Regulation information: I_Regulation
- ▶ Template for exporting information at geotype level: I_Template
- ▶ Backbone nodes: I_Nodes_TX
- ▶ Backbone links (from Aggregation until Core): I_Links
- ▶ Backbone Information to construct the links of the access network: I_Links_acc1, I_Links_acc2, I_Links_acc3, I_Links_acc4
- ▶ Relationship between distribution and Core networks: I_dist_core
- ▶ Information of the submarine nodes: I_sub_node
- ▶ Information of number of houses to cover per CO and year on fibre: I_CO_ALL



- ▶ COs considered PON: I_CO_PON
- ▶ COs considered PTP: I_CO_PTP
- ▶ Coverage Database: I_Coverage
- ▶ Assets and network parameters considered in the model: I_Cables_Assets_Copper, V_FDP_Splitting_Copper, V_SDP_Splitting_Copper, V_PDP_Splitting_Copper, I_Cables_Assets_Fibre, V_FDP_Splitting_Fibre, V_SDP_Splitting_Fibre, V_PDP_Splitting_Fibre, I_DP_Copper, I_DP_Fibre, I_SC_Copper, I_SC_Fibre, I_Joints_Copper, I_Joints_Fibre, I_MDF_Copper, I_ODF_Fibre, Fibre_OLT, V_Threshold_trenches, V_max_coax_CMC_OLT, V_max_coax_Channels, V_max_ports, I_Coaxial_DP and V_Road_Width

5D CALC FIBRE COVERAGE.R

This script is in charge of calculating the fibre deployment algorithm. This algorithm uses as an input the buildings and the number of houses to be covered per CO and year which is an input loaded from the file CO_ALL. Once the houses covered by other operators are discarded the algorithm selects the houses by distance to the CO according to the inputs already stated.

5E CALC FIBRE COVERAGE OTHER OPERATORS.R

This script calculates which houses are covered by fibre for the Norlys model. This is done using the coverage database. The result of this calculation is used in the coverage deployment algorithm to avoid covering houses which are covered by fibre of third-party operators.

5F CALC BUILDINGS PREPARATION AO.R

The inputs calculations and results are equivalent to those present in 5B_CALC_BUILDINGS_PREPARATION.R, The main differences between scripts is that all reference to coax has been eliminated, some code lines have been adapted to cope with the difference in modelling between operators and finally it implements a different fibre algorithm than the said script.

The coverage algorithm deployed in this script basically considers which houses will be covered by 2019, which houses will be covered by 2038 and the number of houses covered per year and selects each house by its distance to the CO.



5G_CALC_BUILDINGS_PREPARATION_AO_COAX.R

The inputs calculations and results are equivalent to those present in 5B_CALC_BUILDINGS_PREPARATION.R, the main differences between scripts is that all reference to copper and fibre has been eliminated, some code lines have been adapted to cope with the difference in modelling between operators.

5H_CALC_BUILDINGS_PREPARATION_OTHER.R

The inputs calculations and results are equivalent to those present in 5B_CALC_BUILDINGS_PREPARATION.R, the main differences between scripts is that all reference to coax has been eliminated, some code lines have been adapted to cope with the difference in modelling between operators.

5I_CALC_FIBRE_COVERAGE_OTHER.R

This script is in charge of calculating the fibre deployment algorithm for alternative operators. This algorithm is used to calculate the fibre coverage when the data of the alternative operator is reported at node level.

B.8. Block 6: KPIs

This is the block primarily responsible for calculating the final KPIS for the different networks.

6A_KPI_COAX.R

This script calls the functions F_CALC_KPI_COAX_Scenarios and F_CALC_KPI_Passive_Coax_Scenarios which perform the calculations of the coax KPI network.

6B_KPI_COPPER.R

This script calls the functions F_CALC_KPI_COPPER_Scenarios and F_CALC_KPI_Passive_Scenarios_Copper which perform the calculations of the Copper KPI network.



6C KPI FIBRE.R

This script either calls one of the following two scripts depending on the data set being used: 6M_CALC_FIBRE_YEARLY_AO.R or 6L_CALC_FIBRE_YEARLY.R. These scripts are in charge of calculating the yearly KPIs of the fibre network

6D KPI PASSIVE.R

This script call one function and one script. The function (F_CALC_KPI_Passive_Scenarios_general) is in-charge of calculating general KPIs common in all the networks and the script 6K_CALC_COVERAGE.R is in-charge of calculating the yearly coverage for each network.

6E KPI GENERAL.R

This algorithm puts together all the KPIs from the different access networks to be able to export through the script 7A_OUT_EXPORT.R.

6F KPI CORE.R

This script is in-charge of calling one of the following three scripts depending on the dataset being executed: 6G_CALC_CORE.R, 6H_CALC_CORE_SAMPLE.R or 6N_CALC_CORE_AO.R

6G CALC CORE.R

This script is in-charge of calculating the KPIs of the core network. The calculations performed are:

- ▶ Calculation of the trenches used per the core and transmission networks; this is done by calculating the route distance between the different network nodes. For nodes from aggregation until core this the calculation of which nodes are linked is imported as an input. On the other hand, these links are calculated for the access network based on a TSP algorithm and the information of which nodes pertains to the same network ring. In addition, the route generation is converted to section level information similar to the process followed by the function F_SectionNode_Generation. The functions associated to these calculations are F_TX_node_Treatment, F_TX_link_access_Treatment, F_TX_link_Treatment, F_tx_node_route_distance F_SectionNode_Generation_CORE



- ▶ Considering the trenches used by the access network the script calculates the percentage of trench sharing between access and core networks. This is done using the functions F_CALC_CORE_PERC, F_CALC_CORE_PERC2 and F_CALC_CORE_PERC3
- ▶ The third step is to calculate the amount of traffic handled by each network node in terms of houses. This process is mainly done the script considering the node and ring relationships imported from the core network input file. This information is calculated at ring level. Some of the functions used in this part are F_CALC_TX_SHARE and F_CALC_TX_SHARE_house
- ▶ Submarine cables are calculated based on the information loaded. The submarine cable information contains both the submarine cable distance and its attributable shared cost between geotypes.

6H CALC CORE SAMPLE.R

This script is exactly the same as the script 6G_CALC_CORE.R with the difference that it uses the function F_TX_node_Treatment_sample instead of F_TX_node_Treatment. The difference between these functions is that the first one only considers direct distance for assigning the intersection to a node instead of considering the CO area and the road.

6K CALC COVERAGE.R

This script is in charge of calculating the coverage KPI. The algorithm basically calculates the number of passed houses per year, geotype and network and then divided it by its total, obtaining the coverage percentage KPI which is an input of the sheet 2A INP COVERAGE of the excel model.

6L CALC FIBRE YEARLY.R

This script is in charge of calling the appropriate functions to correctly calculate the fibre KPIs for each modelling year. The script performs four loops one for each of the fibre functions in F_CALC_KPI_FIBRE_PON, F_CALC_KPI_Passive_Fibre_PON, F_CALC_KPI_FIBRE_PTP and F_CALC_KPI_Passive_Fibre_PTP. On each loop and using information related to coverage calculated in 5A_CALC_BUILDINGS_PREPARATION.R, the algorithm takes the actual buildings passed the said year and then calculates the KPIs accordingly.



Additionally, of the yearly coverage these scripts calculate an additional KPI which is the percentage of areas where the three is fibre and coax over the number of areas with copper. As explained in section 5.2 Fibre Access Network, the scripts used in this section consider the PDPs as SDPs to account for the reality of the TDC fibre network.

6M CALC FIBRE YEARLY AO.R

This script performs the same calculations as the script 6L_CALC_FIBRE_YEARLY.R but instead it calls the functions used to calculate the alternative operators KPIs which are F_CALC_KPI_FIBRE_PON_AO, F_CALC_KPI_Passive_Fibre_PON_AO, F_CALC_KPI_FIBRE_PTP_AO and F_CALC_KPI_Passive_Fibre_PTP_AO. The main difference between both scripts is that in this script the SDPs are considered as SDPs.

6N CALC CORE AO.R

This script is equivalent to the script 6G_CALC_CORE.R the main difference being that the links between access nodes are also being loaded into the dataset instead of being calculated through a TSP algorithm.

6O CALC COVERAGE COAX.R

Same script as the 6K_CALC_COVERAGE.R but adapted to the Norlys coax model.

6P CALC CORE AO COAX.R

Same script as the 6G_CALC_CORE.R but adapted to Norlys inputs and outputs.

6Q CALC COVERAGE AO.R

Same script as the 6K_CALC_COVERAGE.R but adapted to the Norlys fibre model.

6R CALC FIBRE YEARLY OTHER.R

This script performs the same calculations as the script 6L_CALC_FIBRE_YEARLY.R but instead it calls the functions used to calculate the alternative operators KPIs which are F_CALC_KPI_FIBRE_PON_AO, F_CALC_KPI_Passive_Fibre_PON_AO, F_CALC_KPI_FIBRE_PTP_AO and F_CALC_KPI_Passive_Fibre_PTP_AO. The main difference between both scripts is that in this script the SDPs are considered as SDPs.



6S CALC COVERAGE OTHER.R

B.9. Same script as the 6K_CALC_COVERAGE.R but adapted to the alternative operators fibre model. **Block 7: Export**

This block is formed by the scripts in charge of exporting all the R results into an excel file.

7A OUT_EXPORT.R

This script is in-charge of loading the excel output template load all relevant KPIs and export it to the export folder as explained in Annex A.

7B AO_JOINCTION.R

This script is in-charge of exporting all relevant data from the Norlys fibre model, so the coax model can import the data and export the KPIS jointly.

B.10. Block 8: Execution

This block is formed by the script in charge of sequential executions.

8A FULL_EXECUTION.R

This script is in-charge of execution the code from whole model sequentially and in the right order.

8B LOADING_FIRST.R

This script is in-charge of correctly switching from the coax to the Norlys fibre model to avoid any problems in the execution.

8C LOADING_SECOND.R

This script is in-charge of correctly switching from the fibre to the Norlys coax model to avoid any problems in the execution.



Annex C. List of acronyms

BSA	Bitstream Access
BU	Bottom-Up
BU-LRAIC	Bottom-up Long-run Average Incremental Costs
CAPEX	Capital expenditure
CMC	Coaxial Media Converter
CO	Central Office
CSV	Comma-separated values
DBA	Danish Business Authority
DKK	Danish Krone
DP	Distribution Point
DWDM	Dense wavelength division multiplexing
FDP	Final Distribution Point
FTTC	Fibre to The Cabinet
G&A	General and Administrative
GPON	Gigabyte Passive Optical Networks
IT	Information and Technology
Kbps	Kilobit per second
LRAIC	Long-run average incremental cost
Mbps	Megabit per second
MDF	Main Distribution Frame
MRP	Model Reference Paper
MSAN	Multi-Service Access Nodes
ODF	Optical Distribution Frame
OLT	Optical Line Terminal
OPEX	Operational expenditure
PDP	Primary Distribution Point
PON	Passive Optical Network
PTP	Point to Point
SC	Street Cabinet
SDP	Secondary Distribution Point
SMP	Significant Market Power
VoD	Video on Demand
WACC	Weighted Average Cost of Capital



Annex D. Change log in the R model

The table below presents the most relevant changes implemented in the R model after the 3rd consultation with the industry.

Script modified	Change
INPUTS (No script)	Added a threshold that can be used to limit in the fibre rollout algorithm the distance starting from which homes with fibre from alternative operators can be covered
General (all scripts)	The code has been cleaned to eliminate all commented and unnecessary code. Additionally, all code not actively used in the model has been removed.
General (all scripts)	Now, the model prints to the console several extracts of information to show the process of execution.
0M_FUN_COPPER_PASSIVE.R 0K_FUN_COAX_PASSIVE.R 0O_FUN_FIBRE_PASSIVE_PON.R 0P_FUN_KPIS_SUBFUNCTIONS.R	Adjusted the calculation to ensure that road crossings and parallel trenching are correctly considered based on the feedback provided
6D_KPI_PASSIVE.R 6E_KPI_GENERAL.R	Adjusted the way the results from Norlys' fibre and coaxial submodels are joined to ensure consistency in the geotype distribution
2K_CALC_INITIAL_PROC_OTHER.R 5E_CALC_FIBRE_COVERAGE_OTHER_OPERATOR.R 5H_CALC_BUILDINGS_PREPARATION_OTHER.R 5I_CALC_FIBRE_COVERAGE_OTHER.R 6R_CALC_FIBRE_YEARLY_OTHER.R 6S_CALC_COVERAGE - OTHER.R 6T_CALC_CORE_OTHER.R	Adjusted the R model so now it is capable to model any generic fibre operator. These scripts are variations of available scripts to account for small difference when modelling the different alternative operators.
7A_OUT_EXPORT.R	Adjusted the export calculation to enable the model to work with the modelling of standard models, as well as corrected some inconsistencies in the extraction of the KPIs.

MADRID (HQ)
Sagasta, 18, 3
28004, Madrid
Tel: +34 91 310 2894

ISTANBUL
Buyukdere Cad. No 255, Nuroi
Plaza B 04 Maslak 34450
Tel: +90 212 277 70 47

MEXICO CITY
Torre Mayor Piso 41, Paseo de la
Reforma 505, Cuauhtémoc
Ciudad de México, 06500
Tel: +52 55 68438659

AXON 

