Network Modeling and Planning for Fixed Wireless Access Applications

Brecht De Beelde, Mike Vantorre

***Abstract*—The large bandwidths that are available at mmWave frequencies enable fixed wireless access (FWA) applications, in which fixed point-to-point wireless links provide internet connectivity. In FWA networks, a wireless mesh is created and data is routed towards the interface with the wired internet infrastructure. Typically, one or more fiber point of presence (POP) nodes connect the wireless mesh network to the wired infrastructure. Next to the POP device, two other types of devices are present in the mesh network. Customer premises equipment (CPE) devices are client nodes that connect a household or enterprise to the FWA network, i.e., they are the gateway from the FWA network towards the (wireless) local area network. A second type of devices are EDGE nodes which are part of the FWA network, but do not directly connect a client to the network. Instead, they acts as routers. In this work, we analyze network characteristics of FWA networks, which are modeled as undirected weighted graphs with the POP, CPE, and EDGE nodes being represented by vertices. An edge between two vertices indicates that there is a Line-of-Sight path between the two nodes, using the distance between the nodes as weight. An algorithm is presented for routing internet traffic from a CPE towards the POP, consideriing the available bandwidths of the wireless links.**

***Index Terms*– Fixed Wireless Access, FWA, mmWave, routing algorithm, graph theory, network design.**

I. Introduction

During the past decade, the need for broadband connectivity has increased. Not only do end-users require more data volumes and higher data rates, e.g., for online gaming and video-on-demand streaming services, also the data volumes of enterprises have risen, e.g., due to digitalization, video conferencing, and telework. Network operators are required to update their access networks to enable broadband networks, as data rates of digital subscriber line (DSL) technology are generally limited to 35 Mbps, and the throughput of cable networks is limited to a maximum of 500 Mbps. Fiber optic cables offer download speeds up to 1 Gbps but have a high installation cost. In fixed wireless access (FWA) applications, the last mile of the access network is replaced by a wireless link using a point-to-point radio network. The benefit of FWA networks is that it is easier to deploy, as no digging cost is required to set up the network. Work on standardization of broadband wireless access systems started in 2001, and resulted in the IEEE 802.16 standard that specifies the air interface, including the medium access control layer (MAC) and physical layer (PHY) of fixed and mobile point-to-multipoint broadband wireless access systems [1].

Three types of nodes are present in an FWA network. The point of presence (POP) connects the wireless network to the wired infrastructure, and acts as a gateway. Customer premises equipment (CPE) are the wireless devices that the customer installs and bridge the local area network (LAN) to the FWA network. CPE devices are typically connected at the building facade, using an antenna array to allow beamforming. They are connected via Ethernet to a WiFi access point that is installed inside the building. EDGE nodes form a third type of device and are similar to CPE nodes but do not connect any customer. Instead, they are used mainly for routing and to create an interconnected wireless mesh. These EDGE nodes are typically installed at street furniture, including lampposts, billboards, and street signs.

The IEEE 802.16 standard considers frequencies ranging from 10 to 66 GHz, using channel bandwidths of 25 MHz, which is 5 times more than the bandwidth of long term evolution (LTE) channels. Following the Shannon capacity theorem, the resulting throughput is expected to be 5 times the throughput of LTE. For a single-input single-output (SISO) LTE system with a channel bandwidth of 5 MHz and 256 quadrature amplitude modulation (QAM), the maximum throughput is 24.4 Mbps, which is indeed one fifth of the reported raw data rate of 120 Mbps reported in [1]. Recent advancements in radio technology have enabled mmWave radio communication for frequencies up to 100 GHz [2], [3], [4], [5], [6]. The large bandwidths that are available in the mmWave spectrum enable high-throughput wireless communication. The IEEE 802.11ad standard specifies PHY and MAC layer interfaces for short-range high-throughput wireless systems with carrier frequencies in the V-band (50-75 GHz) and channel bandwidths of 2 GHz, allowing SISO data rates up to 2.5 Gbps [7]. Its successor IEEE 802.11ay supports multiple-input multiple-output (MIMO) and allows data rates up to 40 Gbps [8]. The installation and maintenance cost of FWA links is lower than rolling out optical fiber, while providing data rates comparable to fiber.

In [9], a channel model is provided for suburban FWA networks at 28 GHz with 90% coverage. Beam alignment at mmWave frequencies is discussed in [10]. As rainfall introduces time-varying attenuation on mmWave wireless links, rain attenuation models are required. These are available in [11], [12]. In [13], link budget calculations are presented for FWA links using carrier frequencies ranging from 75 GHz to 400 GHz.

A. Project goal

The project goal is to characterize FWA networks via network analysis and to create a network routing algorithm. The routing algorithm defines to which nodes the CPEs have to connect, and routes traffic from a CPE towards the POP.

B. Outline

The outline of this work is as follows. In Section II, we first provide an overview how we obtain the input data for the FWA network, and then we discuss the various methods how the FWA network data can be represented in a graph structure. We use the graph representation to characterize the network via Python's igraph package, which is discussed in Section III. Section IV presents the network routing algorithm and Section V concludes this report. An overview how our code is structured is provided in the appendix.

II. FWA NETWORK DATA AND GRAPH REPRESENTATION

A. FWA data acquisition

We obtain FWA network data from the Green Radio Access Network Design (GRAND) tool [14] which is a deployment tool for wireless radio access networks. The tool defines base station locations of cellular networks and has been adjusted to allow network performance simulations for FWA networks.

The starting point of the tool is a floor plan of the considered deployment environment. The floor plan consists of three parts: the building locations, the street locations, and the area limits. All the buildings get a unique building identifier. The Cartesian coordinates of a configurable number of nodes are randomly defined via a uniform distribution using the area limits from the floor plan. For each node, new coordinates are generated as long as the coordinates do not correspond with a building that does not already have a node. Once the coordinates correspond to a building, they are modified so that the node is located at the street-level facade of the building. Each building can have at most one node. The result of the node placement algorithm is that a predefined number of nodes are allocated to different buildings on the floor plan, at the facade closest to the nearest street.

The GRAND tool distinguishes between the number of base stations and the number of users, and both are configurable configuration parameters. In the modifications to model FWA networks, base stations are mapped on EDGE nodes, whereas, users represent CPE nodes. The user locations are also randomly defined, with an added constraint that they correspond to building facades at street level.

A link between two nodes exists when there is a Line-of-Sight path, i.e., the direct path between the two nodes is not obstructed. Link characteristics include the link distance (in meter), the signal attenuation from the wireless signal, and the maximum throughput, amongst others. The signal attenuation, also called path loss, depends on the distance, as well as the carrier frequency of the wireless technology. Path loss (in dB) in a free space environment is calculated via (1), with f the frequency (in Hz), d the distance (in meter), and c the speed of light in air, i.e., c = 3e10 m/s.

$$\mathrm{PL}(f,d) = 20\ \log_{10}\left(4\pi d\frac{f}{\mathrm{c}}\right) \tag{1}$$

The GRAND tool generates link data formatted as a comma-separated values (CSV) file, where each row contains a link between two nodes. An input CSV file example is provided in Table. I. In the `Data` directory of the software deliverable, input data is provided for two environments and different configurations.

The nodes are characterized by node identifier and node type. Another file gives specific information about the nodes, including their locations, which is not relevant for this project. For each link, the following characteristics are provided: link distance (based on the node locations), PL, throughput (bitrate and maximum bitrate), uplink and downlink specific absorption rate (which relates to electromagnetic exposure). In the CSV file containing the link data, the node identifiers and node types are listed. Unlike initial assumptions, the nodeID between EDGE and CPE nodes is not shared, i.e., the combination (nodeID, nodeType) is unique.

The data from Table I is obtained from a simulation with 600 base stations and 100 users as configuration parameters. This results in a higher number of EDGE nodes, compared to the smaller number of CPE nodes. We believe that such a high number of EDGE nodes, i.e., infrastructure nodes that do not connect users but are only used to form the mesh network and providing network connectivity to CPE nodes, is not representative for real FWA networks. Therefore, we do not use the node types for the network characterization. Instead, we perform simulations with different numbers of base stations and without users. All nodes are located at building facades at street level and are considered to be CPE nodes, creating a mesh network.

From the link data file, several metrics can be used to create a weighted graph. One option is to use the maximum bit rate, i.e., the combined throughput of all data streams that are passed on the link. The network routing algorithm would then be a heuristic that defines a route to the POP with shortest-hop count for each CPE, and a maximum number of streams for each link that depends on the edge weight. However, the maximum bit rate depends on the modulation and coding scheme (MCS) that can be used for a certain link. Advanced MCSs allow higher data rates, but we need a "better" signal quality, i.e., with a lower noise (or higher signal-to-noise ratio), in order to correctly decode the signal. As such, the MCS that can be used for a certain link depends on the path loss. The higher the path loss, the lower the received signal power, and the more difficult it gets to decode the signal, and therefore, the more robust the MCS should be (resulting in lower data rates). Using the path loss as weight would therefore be another option. The network routing algorithm could then consist of finding the path with lowest path loss.

In this work, we select the distance as weight. As path loss is related to distance, e.g., via Friis formula, the link quality and maximum bit rates depend on the distance. Using the distance as metric allows us to easily extend the work, e.g., for using different path loss models, or using different wireless technologies (that might use other MCSs and other frequency bands).

A link budget calculation allows to determine throughput based on configuration parameters, such as the wireless technology that is used and distance between antennas. The received signal power is found by subtracting PL from the sum of the antenna gains and transmit power. With a higher received power, more complex modulation and coding schemes

Table I
INPUT COMMA-SEPARATED VALUE (CSV) FILE EXAMPLE

NodeAid	NodeAType	NodeBid	NodeBType	distance	isLOS	pathLoss	maxPathLoss	bitrate	maxbitrate	isAssignable	sarDL	sarUL	status
85	CPE	211	EDGE	0.07117801545	true	56.8881033415	97.480272768	46.0	7508.0	true	0.8767150427	3.26545253E-6	CONNECTED
85	CPE	588	EDGE	19.09430567	true	100.11643999	100.480261768	46.0	5775.0	true	4.1689217896E-5	8.76005491E-4	INIT
85	CPE	227	EDGE	22.24593196	true	101.297410793	103.480262768	46.0	5620.0	true	3.1763384893E-5	0.001161731547	INIT
85	CPE	407	EDGE	34.15389034	true	104.611585823	105.18026277	46.0	5005.0	true	1.480843363E-5	0.002466440385	INIT
85	CPE	191	EDGE	55.314146155	true	108.338832131	109.480272768	46.0	3080.0	true	6.277482775E-6	0.0058645868	INIT
85	CPE	275	EDGE	60.655054316	true	109.051383027	109.480272768	46.0	3080.0	true	5.327500934E-6	0.00681936411	INIT
85	CPE	103	EDGE	75.46402201	true	110.74012186	112.480261768	46.0	2695.0	true	3.6112330944E-6	0.01011218202	INIT
85	CPE	333	EDGE	85.22263991	true	111.680228578	112.480262768	46.0	2695.0	true	2.9083372325E-6	0.01250332549	INIT
85	CPE	302	EDGE	91.60930083	true	112.238874759	112.480262768	46.0	2695.0	true	2.557295967E-6	0.01428302327	INIT
85	CPE	594	EDGE	97.36875739	true	112.710219414	113.480262768	46.0	2503.0	true	2.2942705176E-6	0.0159088345	INIT
29	CPE	166	EDGE	0.10151535674	true	59.632619784	97.480262768	40.0	7508.0	true	0.4660246872	3.26545233E-6	INIT
29	CPE	57	EDGE	11.212744718	true	96.001222763	97.480261768	40.0	7508.0	true	1.0753390219E-4	3.396137765E-4	INIT
29	CPE	174	EDGE	19.099431885	true	100.118515339	100.480272768	40.0	5775.0	true	4.1669446606E-5	8.76475679E-4	INIT
29	CPE	51	EDGE	22.269330107	true	101.305537037	103.480262768	40.0	5620.0	true	3.1703939804E-5	0.001158501312	INIT
29	CPE	75	EDGE	40.78786502	true	105.983801646	106.18026177	40.0	4620.0	true	1.0796612538E-5	0.00338227618	INIT
29	CPE	29	EDGE	51.8640575	true	107.840973832	109.480261768	40.0	3080.0	true	7.0399616495E-6	0.005187983236	CONNECTED
29	CPE	217	EDGE	59.181635484	true	108.861273756	109.480272768	40.0	3080.0	true	5.565978461E-6	0.00650845343	INIT
29	CPE	277	EDGE	61.870650516	true	109.204778922	109.480272768	40.0	3080.0	true	5.142642722E-6	0.00712224435	INIT
29	CPE	571	EDGE	62.93495013	true	109.336626491	109.480262768	40.0	3080.0	true	4.988903969E-6	0.00732649499	INIT
29	CPE	562	EDGE	69.31410403	true	110.082975348	112.480262768	40.0	2695.0	true	4.201162895E-6	0.00869512603	INIT
2	CPE	300	EDGE	0.07726111779	true	57.522055197	97.480261768	46.0	7508.0	true	0.7576385471	3.26545236E-6	INIT
2	CPE	82	EDGE	6.427939545	true	91.700028054	97.480261794	46.0	7508.0	true	2.895108674E-4	1.261437611E-4	INIT
2	CPE	557	EDGE	15.956427461	true	98.728605079	100.480261768	46.0	5775.0	true	5.738616238E-5	6.3639018E-4	INIT

can be used, resulting in a higher throughput.

B. Graph data structure

Based on the output from the GRAND tool we described in the previous section, a graph is constructed. We have investigated different approaches to represent the data in a graph.

1) Using tuples with node type and node identifier: A first implementation is to extract all unique tuples (nodeId, nodeType) for nodeA and nodeB in the link data file, and adding these as vertices in the graph. Each vertex has several attributes, including an "id" attribute that corresponds to the nodeId and a "type" attribute that corresponds to its type. We then construct the edges by iterating over each link in the data file, and looking up the identifiers of the two corresponding tuples. The edge weight is taken from the link data file.

2) Duplicating CPE *nodes:* Under the assumption that the nodeId is unique, i.e., CPE and EDGE nodes with the same nodeId link to the same (physical) node and act as routing node as well as connecting users, we propose to duplicate all CPE nodes. We first add the number of EDGE nodes that is present in the data file, with "id" and "type" attributes equal to the nodeId and node type. We then add all CPE nodes with the nodeId as "id" attribute. Next, we add edges with zero weight between the CPE and EDGE vertices having the same "id" attribute. Finally, edges are added according to the data file, using the distance as weight.

3) Getting CPE *nodes, manually defining* EDGE *nodes :* Under the assumption that a mesh network exists without considering EDGE nodes, i.e., all CPE nodes have a path towards the POP without requiring an EDGE node that does not serve any users, we can create a graph with only two node types, i.e., POP and CPE.

In our final implementation, we get a predefined number of CPE nodes from the GRAND tool, i.e., nodes that are randomly placed on the floor plan with a constraint that there is at most one node for a single building, and that the node is located at the building facade facing the street. From the GRAND tool we obtain CPE nodes that are randomly placed on building facades. Edges indicate whether a wireless link is possible. However, it is possible that the graph is not connected, depending on the building and CPE node locations. In this case, we will manually add EDGE nodes to the graph. The POP, CPE and EDGE nodes are represented by vertices, with a "type" attribute and with a "throughput" attribute. An "id" attribute is assigned that corresponds to the NodeId from the input data. The "throughput" attribute represents what the required download data rate is that should be available for the user. In our implementation, we give all users the same throughput, but it is possible to differentiate between different users, e.g., depending on the subscription plan. As the location of CPE nodes is random, running simulations for the same environment with identical configuration settings will always result in another graph. We only use the nodeId data and distance between different nodes.

Notice the similarity with a real FWA network deployment. A network operator doesn't know in advance which users will subscribe for internet connectivity, which is in line with the randomness of the CPE node locations. The number of CPE nodes represents the number of subscribers to the network and is related to the penetration rate. An operator will manually add EDGE nodes in order to provide quality of service (QoS) and to serve users that are otherwise not connected. For the placement of these EDGE nodes, it is necessary to check regulations, e.g., where are we allowed to place them. The nodes will typically be deployed at street furniture such as lampposts, or at public buildings.

III. NETWORK CHARACTERIZATION

In this section, we first present the methodology how the network is characterized, using the graph representation from Section II-B3. For the validation of our analysis scripts, we use a simplified small FWA network for which we can easily calculate graph statistics manually, which is presented in Section III-B. The network analysis of a realistic network follows in Section III-C.

A. Methodology

For investigating the graph metrics, we use the Python package igraph. After verification that the definitions used

Table II
LINK DATA USED FOR THE VALIDATION OF THE IMPLEMENTATION

NodeAid	NodeAType	NodeBid	NodeBType	distance	isLOS	pathLoss	maxPathLoss	bitrate	maxbitrate
1	EDGE	2	EDGE	1	100	100	100	100	100
1	EDGE	5	EDGE	3	100	100	100	100	100
1	EDGE	7	EDGE	2	100	100	100	100	100
2	EDGE	3	EDGE	5	100	100	100	100	100
3	EDGE	6	EDGE	2	100	100	100	100	100
3	EDGE	7	EDGE	3	100	100	100	100	100
5	EDGE	7	EDGE	1	100	100	100	100	100
6	EDGE	7	EDGE	5	100	100	100	100	100
4	EDGE	2	EDGE	2	100	100	100	100	100

by igraph match the definitions used in the course notes, we analyze the following metrics.

1) Vertex degree: the number of adjacent edges of a vertex
2) Vertex betweenness: the number of shortest paths from all vertex pairs that pass through the vertex
3) Eccentricity: maximum of the shortest distance from a vertex to all other vertices in the graph
4) Graph Radius: the smallest eccentricity in the graph
5) Diameter: the length of the longest shortest path between two vertices
6) Average path length: average of all shortest path lengths
7) Characteristic path length: median of the length of all shortest paths
8) Average hop count: average of the number of hops on the shortest path between two vertices in the graph

For each distance metric, the analysis is repeated for a weighted graph, using the distance between nodes as a weight, and without weights, using the hop count to calculate the different metrics.

B. Validation

Next to the verification that the metric definitions match the definitions used in the course notes, we use a simple input file and confirm that the analysis code has the same results as a manual calculation of the metrics. Table II presents the link data that is used for the validation of the implementation, using the same format as the input file that we obtain from the GRAND tool.

Figure 1 shows the graph representation of the link data. For each vertex, the vertex degree, eccentricity (using hop count) and vertex betweenness are provided. No vertex has eccentricity 0 as the input graph is connected. From the graph visualization, it is clear that the graph diameter is 3, i.e., the longest shortest path connects vertex 4 to vertices 5, 6, or 7. The average path length is 4.3 hops, and the characteristic path length is 4. The average hop count is 1.7.

C. Network analysis

We analyze three different data sets representing different FWA networks. The configuration settings to generate the data are presented in Table III, and the CPE locations of the different data sets are shown on a floor plan in Figure 2. The graph representation is shown in Fig. 3 for two scenarios, and the results of the graph characterization are presented in Table IV.

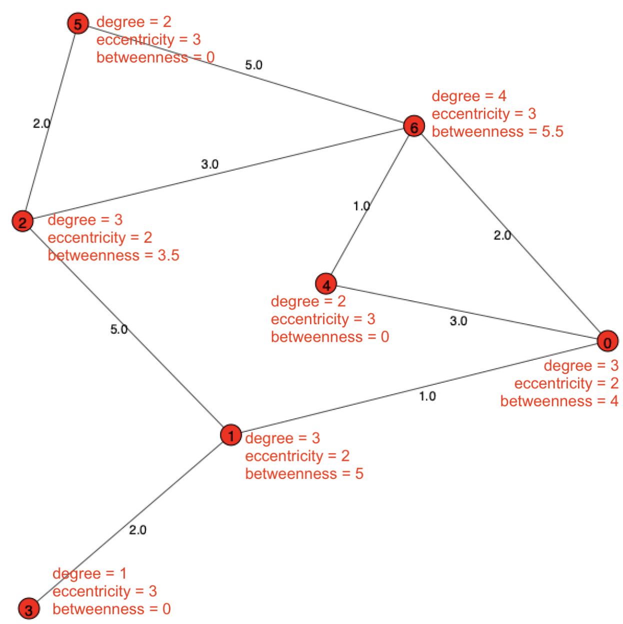


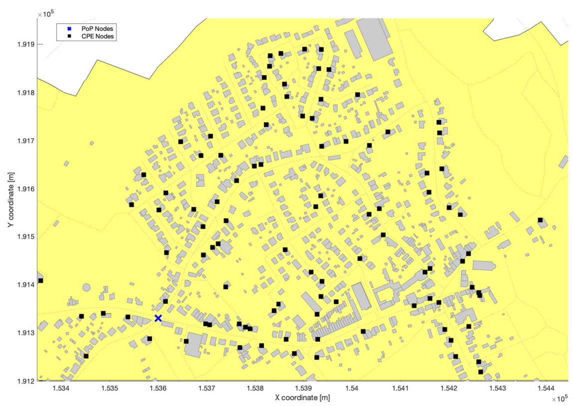
Figure 1. Graph representation of validation data.

Table III
CONFIGURATION SETTINGS FOR DIFFERENT FWA SCENARIOS

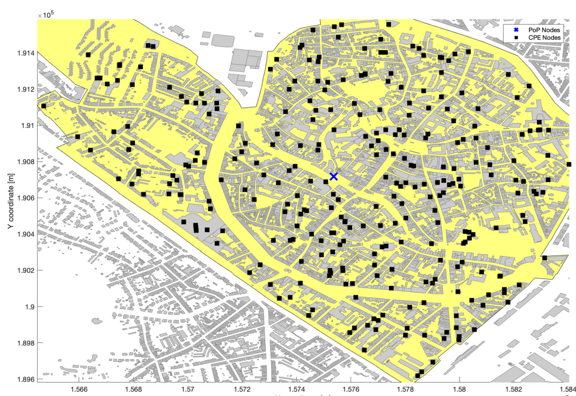
FWA type	Environment	Number of nodes
Dense urban	Urban village (Leest)	600
Urban pilot	Urban village (Leest)	100
City deployment	City center (Mechelen)	300

It can easily be seen that the dense urban deployment also reflects a dense behavior in the metrics. The dense network has a higher average node degree compared to a less dense network. As expected, distance metrics, including eccentricity, radius, diameter, and path length metrics, are lower for the dense urban network than for the urban pilot network, for metrics based on hop count, and based on edge weights. The large average vertex betweenness indicates a high clustering, so a graph simplification is possible by combining cliques, i.e., full connected subgraphs, into a single vertex. The graph diameter and average hop count will also influence network performance on a higher level. As radio propagation in free space travels at the speed of light, propagation delays are minimal. However, with a higher number of hops on the path, the packet latency will increase due to an increased processing

(a) Dense urban



(b) Urban pilot



(c) City center

Figure 2. Floorplan and cpe node locations for different scenarios. The POP node is visualized by a blue cross and the CPE nodes are represented by black squares.

time on the hops. Therefore, the latency in the urban pilot network is expected to be higher than for the dense urban and city deployment.

As we want to route the network traffic towards the POP, which connects the wireless mesh to the wired infrastructure, eccentricity and graph diameter are less important. However,

Table IV
NETWORK CHARACTERIZATION FOR DIFFERENT FWA SCENARIOS

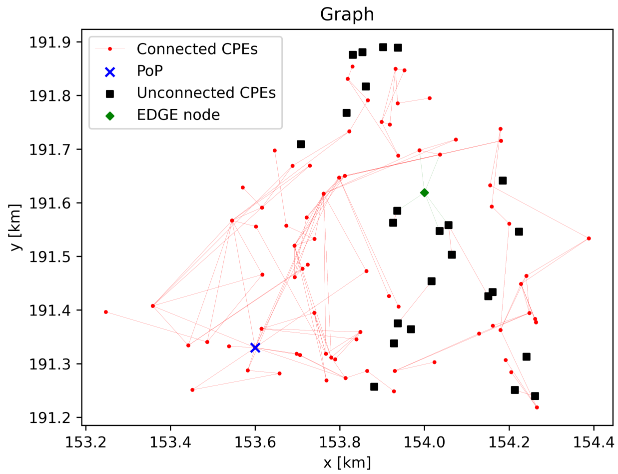
Metric	Dense urban	Urban pilot	City
Avg. graph degree	12.9	3.19	13.0
Avg. vertex betweenness	1357	200	370
Diameter (hop count)	8	18	7
Diameter (weight)	1369.7	2976.1	3021.7
Radius (hop count)	5	9	4
Radius (weight)	815.7	1535.9	1539.0
Avg. eccentricity (hop count)	6.4	12.6	5.3
Avg. eccentricity (weight)	1030.6	2137.5	2126.1
Avg. path length (hop count)	3.85	5.91	3.00
Avg. path length (weight)	474.4	985.8	951.3
Char. path length (hop count)	3.79	5.32	2.91
Char. path length (weight)	453.7	892.7	917.1

Table V
NETWORK CHARACTERIZATION METRICS ON POP NODE FOR DIFFERENT FWA SCENARIOS

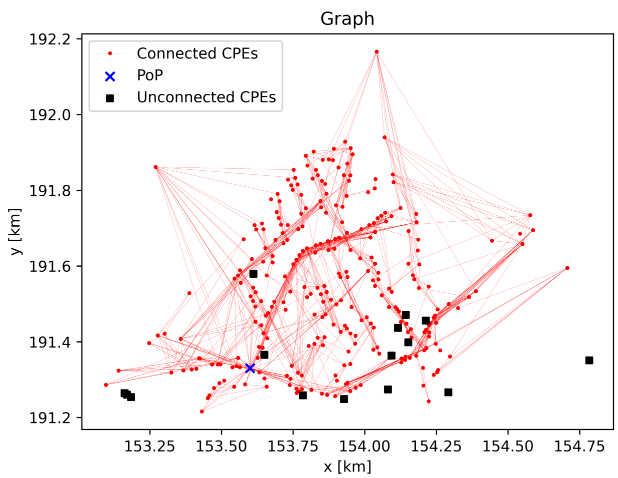
Metric	Dense urban	Urban pilot	City
Degree	29	10	6
Vertex betweenness	36287	952	4
Eccentricity (hop count)	6	9	5
Avg. path length (hop count)	3.17	3.72	2.73
Eccentricity (weight)	907.3	1769.4	1716.7
Avg. path length (weight)	375.8	702.2	870.3

the vertex characteristics of the POP are critical to the success of the network deployment. Table V presents some characteristics of the POP node. The eccentricity is an important parameter for the latency, whereas the vertex degree influences the total network capacity, as all the FWA network traffic is carried over one of the edges of the POP node. The average path length, using the link distances as weights, is important as it defines the maximum throughput of the data that gets transmitted over the link. In the urban pilot and city deployment, the averaged path length is larger and, therefore, the capacity of the links will be lower. The low value for the POP betweenness in the city deployment can be explained by the fact that the network is more spread out over the area. The POP lies central in the physical deployment area, but does not necessarily lie central in a graph context. There are multiple shortest paths from one node in the network to another node that do not include the POP, leading to a low POP betweenness value. This is in contrast to the urban deployments, where it is less likely to have a shortest path which does not contain the POP.

A further analysis for the city deployment shows that the average graph degree significantly increases, to 27, when the number of CPEs increases from 300 to 900. However, the graph diameter is 8 and does not significantly increase, as the area is kept identical. Even though the number of CPEs is tripled, the average hop count stays the same. The average hop count of all vertices towards the POP also stays the same, so an increase in number of CPEs will increase the network capacity, but not the latencies. The vertex degree of the POP increases to 22, but its eccentricity stays at 5. It should be noted that the location of the POP is important. If the location of the POP for the configuration with 900 CPEs moves north (location x=157.4 km, y=191.3 km), the vertex degree of the POP decreases from 22 to 9, which will be a huge bottleneck,



(a) Urban pilot with 100 CPEs



(b) Dense urban with 300 CPEs

Figure 3. Graph model for different configurations. The POP node is visualized by a blue cross, the connected CPE nodes are visualized by red circles, and the CPE nodes that are not connected to the POP node are visualized by black squares. By adding EDGE nodes, visualized by a green diamond symbol, these unconnected nodes can get connected.

its eccentricity increases from 1.2 km to 1.9 km, and the average path length to all other vertices increases from 2.7 to 3.1 hops.

IV. Network Routing

The goal of the network routing algorithm is to define a route from each CPE towards the POP, given a predefined quality of service (QoS) requirement, i.e., we need to allocate a certain throughput to each CPE.

We assume that IEEE 802.11ad wireless technology is used, with a carrier frequency of 60 GHz and data rates up to 4.6 Gbps for a single-carrier physical layer (PHY). Furthermore, we assume that the required maximum download speed is 300 Mbps for each CPE, which corresponds to the download speed of the Telenet WIGO subscription.

A. Algorithm

Before performing network routing, we validate whether the FWA network allows the required QoS to all CPEs. Furthermore, we calculate the available throughputs of the wireless links based on the weights of the edges, i.e., the distance metric. Therefore, we split the algorithm in two parts: we start with the graph preparation algorithm which is presented in Algorithm 1, and then perform the network routing algorithm presented in Algorithm 2.

In the graph preparation algorithm, we first verify whether the graph is connected. If not, it is impossible to server the users with a CPE that does not have a path towards the POP. In this case, we return a warning message which triggers manual interaction, and run the algorithm on the largest connected subgraph containing the POP. We transform the weight of the edges from a distance to a throughput. From the distance, we calculate path loss via (1), and use the link budget equation from (2) to obtain received power $\mathrm{P_{RX}}$ in dBm, with $\mathrm{P_{TX}}$ the transmit power in dBm, $\mathrm{G_{TX}}$ and $\mathrm{G_{RX}}$ the TX and RX antenna gains in dBi and L_F the feeder loss in dB.

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} - L_F - PL \tag{2}$$

For IEEE 802.11ad and 60 GHz, we use the following values. Transmit power $\mathrm{P_{TX}}$ = 15 dBm, antenna gains $\mathrm{G_{TX}} = \mathrm{G_{RX}}$ = 23 dBi, and feeder loss L_F = 2.5 dB. A mismatch of 3 dB is used. Depending on the received power, a certain MCS value can be used. For each link, we relate the received power to a throughput, using the MCS values from Table VI. By using other values in the link budget equation (2), we can perform network routing for other wireless technologies.

When we have the available throughputs for all the links in the graph, we first verify that the total throughput of edges connected to the POP is equal to or larger than the number of CPEs multiplied with the throughput/download speed requirement for each CPE. If this is not the case, not all CPEs will have the required throughput. Another scenario where not all CPEs are able to connect to the POP is when the graph is not connected, i.e., one or more vertices do not have any edges.

If the graph preparation has failed, e.g., due to the graph non-connectivity or due to a throughput bottleneck on the edges connecting the POP, a manual intervention is required, in which we place additional EDGE nodes in the network. In another CSV file, we list the manually added EDGE nodes with their locations, and the links towards CPE nodes. We add the EDGE nodes as vertices to the graph, with EDGE as "type" attribute and zero "throughput" attribute. Edges are added to connect the EDGE vertices to CPE vertices. Based on the CPE and EDGE node locations, the Euclidean distance is calculated and added as edge weight. An example is provided in Fig. 3a, where the nodes around coordinates (x=154.1 km, y=191.5 km) are not connected. By adding an EDGE node at location (x=154.0 km, y=191.6 km), seven additional CPE nodes get connected to the network. The location of the EDGE node is based on an analysis of the floor plan and using

Table VI
RECEIVER SENSITIVITY (P_{RS}), MODULATION, AND DATA RATE (DR) USED IN IEEE 802.11AD FOR THE DIFFERENT MODULATION AND CODING SCHEMES (MCS) [7]

MCS	0	1	2	3	4	5	6	7	8	9	10	11	12
P_{RS} **[dBm]**	-78	-68	-66	-64	-64	-62	-63	-62	-61	-59	-55	-54	-53
Modulation	DBPSK	$\pi/2$-BPSK	$\pi/2$-BPSK	$\pi/2$-BPSK	$\pi/2$-BPSK	$\pi/2$-BPSK	$\pi/2$-QPSK	$\pi/2$-QPSK	$\pi/2$-QPSK	$\pi/2$-QPSK	$\pi/2$-16QAM	$\pi/2$-16QAM	$\pi/2$-16QAM
DR [Mbps]	27.5	385	770	962.5	1155	1251	1540	1925	2310	2502	3080	3850	4620

Data: FWA network graph g and QoS requirement t
Result: Updated graph g' with throughput attribute
if g *is connected* **then**
  for e *in edgelist* g **do**
    Calculate throughput via link budget;
    Add throughput as attribute to e;
  end
  Total network throughput T = # CPE x t ;
  if $T < \sum_{e \in edgelist \text{ POP}}$ *throughput* e **then**
    $g' = g$;
  else
    g' = [] ;
  end
else
  g' = largest connected subgraph ;
end
Return g';

Algorithm 1: Graph preparation

Google Maps[1]. If the graph is not connected, we return the largest connected subgraph and validate whether this subgraph contains the POP.

The output from the graph preparation algorithm, i.e., a connected graph with available throughput and distances attributes for each edge, is used as input data for the network routing algorithm (Algorithm 2). The required throughput is identical for all CPEs. We add the throughput requirement as attribute to the CPE vertices in the graph.

Before starting the actual routing algorithm, we simplify the graph by combining fully connected clusters (cliques) into a single vertex with a throughput requirement equal to the sum of the requirements of the vertices it replaces. The number of nodes within a replaced clique is used as a configurable parameter which can be changed to optimize the routing algorithm. Only replacing cliques with a very high number of nodes will not lead to a noticeably simplified graph. Replacing cliques with a low number of nodes can lead to too much simplification, which can lead to problems in the routing algorithm, as we discuss later. The possible cliques are sorted by the number of nodes within the clique. Fully connected subgraphs with a high number of nodes will be replaced first, as those cliques will lead to a large simplification. If the considered clique has nodes common to another, not yet replaced, clique (such as node 6 in Figure 1), only one fully connected subgraph is replaced. The other possible clique is ignored. The sum of all throughput requirements within a clique is taken as throughput requirement of the replacement node.

After replacing each possible clique, the equivalent nodes are connected with the neighbors from the cliques. For this, the distance between the replaced clique node and the neighbor is taken as the distance from a random node to the neighbor, as it is assumed that the clique nodes are located near each other. The throughput between the replaced node and a neighbor is equal to the sum of all throughputs from a clique node to the neighbor. This whole procedure is illustrated for a simple graph in Figure 4, where the edge labels are the throughputs on a certain link. The clique $(3, 4, 5)$ is replaced by a single vertex with vertex id 1000. The edge between the equivalent node has weight equal to 5390 Mbps, being the sum of 4620 Mbps and 770 Mbps.

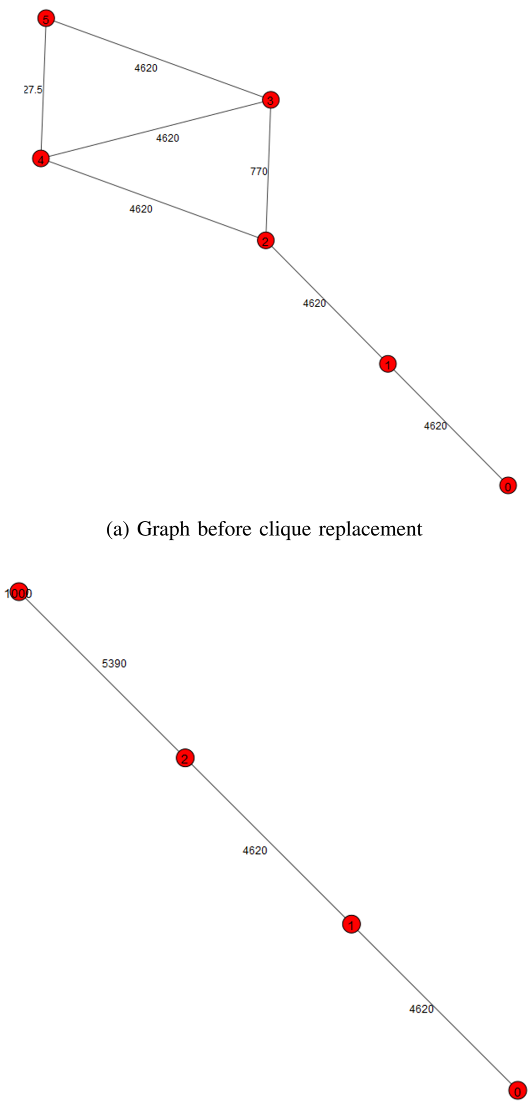
Now that we have a simplified graph, we perform the actual routing. For all vertices with a CPE type attribute, the shortest paths towards the POP is first calculated via Dijkstra's algorithm, using link distances as weights. We also investigated other weights, or no weights at all, but this resulted in a suboptimal solution. The vertices are then sorted in the following order

1) The required throughput: perform routing first or vertices with highest throughput
2) The number of shortest paths: for vertices with equal throughput requirement, perform routing first for vertices with the lowest number of shortest paths
3) The number of hops on the shortest path: perform routing first for the vertices with the highest number of hops on the shortest path

Nodes with a high throughput requirement, e.g., due to differentiated users or due to the clustering algorithm, need to be routed first. Then, the nodes that have a low number of shortest paths should be routed, to prevent that we first optimize routes of other CPEs and then have no more bandwidth available for these nodes. Lastly, we take into account the number of hops on the shortest path. Preferably, the vertices with a large hop count are routed first, e.g., to get minimum latency.

Now that we have a sorted list of vertices for which we need to define routing towards the POP, we again use a shortest path algorithm to define the routing. We add an attribute to the vertex with the path that needs to be followed, and then update the available throughput attribute on all edges along that path, i.e., the "available" remaining throughput is lowered by the throughput requirement of the vertex. If the available throughput on an edge is lower than the minimum throughput requirement of a single vertex, we remove the edge from the graph, to prevent that this edge will be used for routing traffic of other nodes. Therefore, the graph gets updated and the

[1]In a real network deployment, there are restrictions where antennas can be installed, so determining the antenna locations is always a manual step.



(a) Graph before clique replacement

(b) Graph after clique replacement

Figure 4. Overview of the clique replacement procedure.

shortest path algorithm will result in other paths compared to the first time we ran the algorithm.

The last step of the algorithm consists of defining the paths within a clustered subgraph of the original graph, i.e., the one that was replaced by a single vertex. Two cases are considered: the clique node is a starting node of the path, or the clique node is an intermediate, traversing node in the path. In the first case, a path from each node in the clique is found towards the next node in the total path. After this, the rest of the path is found using the method above with throughput requirement equal to the total requirement of the clique. When constructing the path, a clique is considered as one single large ”packet” that gets routed towards the POP. This leads to a simplification when calculating paths as there are less shortest paths to calculate. However, when many cliques are replaced, e.g. when a relatively small number of nodes within a clique are present, many large ”packets” are routed, which could lead to bottlenecks.

Data: Graph g and QoS requirement t
Result: A route to the POP for each CPE node
Assign attribute t to each vertex v in g;
Create new graph g';
for *all full subgraphs s in g'* **do**
  Replace s by a single vertex v;
  Assign attribute t to v with throughput size(s) x t;
end
for *all vertices v in g'* **do**
  Define number of shortest paths n towards POP;
  Define number of hops h on shortest path ;
end
Sort all vertices based on t, n, and h ;
for *all vertices v in sorted g'* **do**
  Define shortest path from v to POP (Dijkstra);
  Add attribute to v with path info;
  Subtract t from all edge throughput attributes;
  if *available throughput attribute of edge $e < t$* **then**
    Remove e ;
  end
end

Algorithm 2: Network routing for predefined throughput requirements

The second case is when the replaced node is intermediate. Then, a path from the previous node in the path is found towards the next node in the path, passing through the clique. The bandwidths inside the cliques are adapted using the same method as routing a ”normal” node.

In the graph preparation algorithm, we only performed some sanity checks whether a solution could exist. It is possible that during network routing, a throughput bottleneck on one of the edges appears. In this case, we return an error but continue with network routing of the remaining vertices. To overcome the bottleneck, more EDGE nodes can be manually placed to increase the network capacity.

B. Network routing results

We performed network routing simulations for the different configurations from Table III, with and without the simplifications of cliques. Results of the network routing algorithms are discussed for both a modified version of the urban dense network, and the urban pilot network (100 CPEs), and can be found in Table VII. It is found that the urban dense network (600 CPEs) cannot support the throughput requirement of 300 Mbps for each CPE without adding few EDGE nodes. For simplicity to test the network routing algorithm, the network is modified such that it only contains 300 CPEs. When replacing cliques, a configuration parameter indicating the number of nodes within a clique needs to be chosen. The urban pilot will replace cliques consisting of 3 nodes, whilst the 300 CPE network will replace cliques consisting of 5 nodes. These values were empirically obtained.

The first step of Algorithm 1 consists of taking the largest connected subgraph. It can be seen that the urban pilot of 100 CPEs has a comparably low number of nodes in the largest connected subgraph compared to the network with

Table VII
RESULTS OF THE ROUTING ALGORITHMS FOR URBAN NETWORKS. A MODIFIED VERSION CONSISTING OF 300 CPES IS USED INSTEAD THE DENSE NETWORK OF 600 CPES.

Result	100 CPEs	300 CPEs
Nodes in largest connected subgraph	76	285
Total throughput requirement (Mbps)	23100	85500
Available throughput at POP (Mbps)	26564.5	97400
Node improvement (%)	0.7662	0.7902
Edge improvement (%)	0.6667	0.6691
Number of skipped nodes	21	5

300 CPEs: only 76% of the nodes are present in the largest connected subgraph whereas for the 300 CPE network, 95% of the nodes are connected with each other. This can easily be explained by the fact that a less dense network (spread over the same physical area) has less nodes that could possibly merge connected subgraphs.

The second step of Algorithm 1 is to check if the POP can support the total network throughput requirement. Table VII shows the total network requirement (for a requirement per node of 300 Mbps), and the available throughput at the PoP. For the considered networks of 100 CPEs and 300 CPEs, it is found that the PoP supports these networks. This concludes Algorithm 1.

Algorithm 2 will first start by simplifying the graph by replacing cliques into a single vertex. By doing this, edges internal to the replaced clique are omitted. Next to this, if multiple nodes of a clique have an edge towards a certain neighboring vertex, the algorithm will replace these edges by a single link. The results of the simplification are also presented in Table VII. The number of vertices in the simplified graph compared to the number of vertices in the original graph, i.e., node improvement, is 76% for the urban trial and 79% for an urban scenario with 300 CPEs. The number of edges in the graph compared to the number of edges in the original graph is 66% for both scenarios. We conclude that both networks have a similar degree of simplification.

Finally, paths towards the POP are constructed. Each node that does not have a valid path to the POP will throw an error and will be skipped in the algorithm. It is found that for the urban pilot of 100 CPEs, a relativily high number of nodes have no valid path. This is in contrast to the network with 300 CPEs, where only 5 nodes have no valid path. This can be explained by the fact that the algorithm will first route the nodes with a high throughput requirement, which in this case are the replaced clique nodes. As these nodes are considered as one "packet", large throughputs are send over the links, rapidly decreasing the throughput over a certain link. As there are less paths towards the POP, at some point all the possible edges will be operating at their maximum throughput.

This behavior is somewhat expected. The act of simplifying the graph by replacing cliques is only useful for large graphs with a lot of possible paths (which is typical for mesh networks). These graphs will have additional redundant paths towards the POP, which can be used if other links can not be used anymore. It is then less likely to not find a valid path. For smaller networks, it can be more useful to ignore the replacement of cliques and do the algorithm on the full graph.

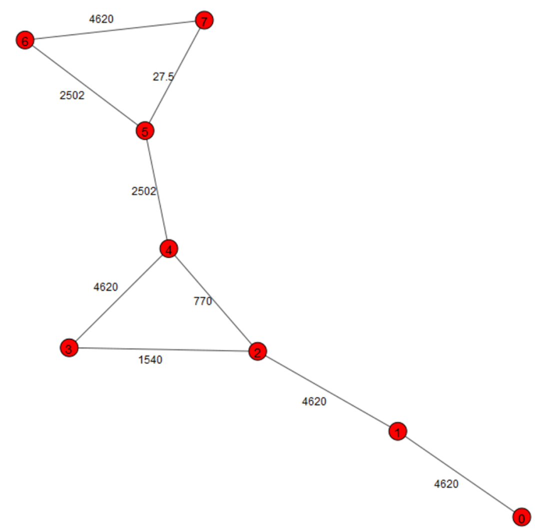


Figure 5. Graph used to show results of the routing algorithm

Table VIII
RESULT OF ROUTING ALGORITHM ON THE GRAPH OF FIGURE 5.

Node	Path
1	(1)
2	$(2, 1)$
3	$(3, 2, 1)$
4	$(4, 3, 2, 1)$
5	$(5, 4, 3, 2, 1)$
6	$(6, 5, 4, 3, 2, 1)$
7	$(7, 6, 5, 4, 3, 2, 1)$

For the urban trial scenario, a path towards the POP is found for all connected CPEs if clique simplification is omitted. A visual inspection of the different paths shows that they correspond to what we intuitively expect.

In order to illustrate how the routing algorithm generates acceptable paths, the graph of Figure 5 is used, with available throughput values on the edge labels. First, shortest paths are found in the graph with replaced cliques, resulting in paths for node 1, and cliques $(2, 3, 4)$ and $(5, 6, 7)$. The next step is to route the nodes inside cliques. The final paths are found in VIII. It is clear that the paths follow edges with highest throughputs.

The clique simplification is effective for the dense urban and city deployment, whereas, for the urban trial, grouping cliques results in high throughput requirements that result in throughput bottlenecks that are not present without the clique simplification.

C. Rain influence

The wavelengths in mmWave frequencies range from 11 mm at 26 GHz to 1 mm at 300 GHz, while the diameter of raindrops is on the order of 1 to 10 mm. In addition to the permittivity of water which also differs from free space, electromagnetic waves incident on rain drops will suffer from attenuation and scattering, causing a lower received signal strength [15]. Models are available to predict attenuation due to rain [16], defining a specific attenuation γ in dB/km via (3) as a function of the rain rate R in mm/h and coefficients k and

Table IX
INFLUENCE OF RAIN ATTENUATION ON TOTAL AND AVERAGE NETWORK THROUGHPUT.

Scenario	Weather	Total throughput	AVG edge throughput
100 CPEs	Sun	411.940 Gbps	3.349 Gbps
100 CPEs	Rain	360.500 Gbps	2.930 Gbps
600 CPEs	Sun	13 386.844 Gbps	3.502 Gbps
600 CPEs	Rain	12 326.428 Gbps	3.225 Gbps

α that depend on frequency and are derived via a scattering analysis.

$$\gamma = kR^{\alpha} \tag{3}$$

For a signal using a carrier frequency of 60 GHz, the specific attenuation for a rain rate intensity of 15 mm/h is 6.8 dB/km. For a carrier frequency of 120 GHz, the specific attenuation increases to 9.0 dB/km. For heavy rainfall with an intensity of 25 mm/h, the specific attenuation increases to 10.1 dB/km at 60 GHz and to 12.6 dB/km for 120 GHz.

In the graph preparation algorithm, we add the specific attenuation as input parameter to calculate path loss. The specific attenuation gets multiplied with the distance in km, and added to the previously calculated path loss.

Table IX presents the influence of rain on the total available network throughput and average edge throughput that is available on the links for two environment scenarios, considering the IEEE 802.11ad standard operational at 60 GHz, and comparing sunny weather with no additional attenuation to heavy rainfall with a specific attenuation of 10 dB/km. Investigation of the path loss for the different wireless links shows that additional attenuation due to rain does not seem to be substantial, as the average link distances are small. The additional attenuation ranges from 0.1 dB for the smaller links up to 3.7 dB for a link with distance 370 m. However, an additional attenuation of 3 dB has a considerable impact on the throughput that is available, as the receiver sensitivities from Table VI are close to one another. For the urban pilot scenario, the total network throughput, i.e., the sum of the throughputs of all wireless links, decreases by 12.5% and the average edge throughput decreases by 419 Mbps in the event of heavy rain. For the dense urban scenario, the total network throughput decreases by 8% and the average edge throughput decreases by 277 Mbps. We conclude that the impact of rain on a dense scenario is lower than for a scattered scenario. This is caused by the smaller link distances for a denser scenario.

However, for both scenarios, the available throughputs on the edges connecting the POP to the CPEs is not sufficient to route the data from all CPEs in the network in case of heavy rain.

V. CONCLUSIONS AND FUTURE WORK

In this work, we have used graph theory to analyze the architecture of FWA networks, and presented a routing algorithm to define how CPE nodes can route their internet data traffic towards the POP node that connects to the wired infrastructure. While some graph metrics, including average vertex degree and average path length, influence the capacity of the network (e.g., what is the average available link throughput), whereas, others influence other wireless system characteristics. An example of the latter is the average hop count that will be related to the packet latency.

In this report, we have presented FWA network examples using IEEE 802.11ad technology for different deployment types. However, as we calculate path loss based on the link distances and derive available throughputs based on a link budget calculation, it is possible to perform network analysis and routing for other technologies and frequency bands as well.

A. Future work

Future work includes the inclusion of more realistic channel models, considering atmospheric loss, vegetation loss, and non-Line-of-Sight channel models. If vegetation obstructs the direct path, the additional attenuation results in a lower received signal power which will decrease the available throughput of the wireless link.

Other future work is to consider different user throughput requirements, e.g., use three internet connection subscription plans with different throughputs, and use a user distribution on which plan is used. As en example, we could have a cheap low-data rate plan where users get a throughput of 30 Mbps, a standard plan with a throughput of 250 Mbps, and an expensive high-data rate plan with a throughput of 600 Mbps. In this scenario, we could assign these plans to the different users via a user distribution. The chance that a user selects the low-data rate plan is 25%, the standard plan is 45%, and the high data-rate plan is 30%.

Finally, more advanced techniques can be used in the routing algorithm, for instance taking centrality measures into account when selecting possibly replaced cliques and replacing different kind of subgraphs (e.g. subgraphs with a combination of strong and weak links).

APPENDIX

To code for graph creation, analysis, and network routing is attached to this report, and structured as follows. There are four main directories in the deliverable. The Documentation directory provides documentation. The Data directory contains the input data to our software framework, including the data generated by the GRAND tool and the manually added information on EDGE nodes. There is a subdirectory for each simulation scenario. The link files used to test our code can be found in “AnalysisTest” (cf. Section III-B) and “RoutingTest” (cf. Section IV). In “InitialTest”, data is provided that was used for first testing, generated from GRAND with 300 EDGE nodes (i.e., base stations) and 100 CPE nodes (i.e., users). There are three data directories for an urban village, i.e., Leest which is located close to Mechelen, with respectively 100, 300, and 600 CPEs. Three data directories for an urban city, i.e., Mechelen. Two simulations are with 900 CPEs, and the difference is the location of the POP node. One simulation uses 300 CPEs. The Code directory contains the software framework, consisting of python scripts, and the Results directory contains generated figures. Upon running the code, the main information is printed to the terminal, whereas, more logging information is appended to the file graph_analysis.log.

```
/
├── README.md: general overview
├── Documentation
│   ├── Project_proposal.pdf
│   └── Report.pdf: this document
├── Data: input data
│   ├── Leest_100BS_1U: 100 CPEs
│   │   ├── basestations.csv: CPE node info
│   │   ├── links.csv: links between CPEs
│   │   └── edge_nodes.csv: manually added
│   │       file with location and links
│   │       from EDGE nodes
│   ├── Leest_300BS_1U: 300 CPEs
│   │   ├── basestations.csv: CPE node info
│   │   └── links.csv: links between CPEs
│   └── Leest_600BS_1U: 600 CPEs
│       ├── basestations.csv: CPE node info
│       └── links.csv: links between CPEs
├── Code
│   ├── graph_creation.py: create a a graph
│   │   based on data in links.csv file
│   ├── graph_analysis.py: analysis of a
│   │   graph (presented in Sect. III)
│   ├── graph_extension.py: code to extend
│   │   a graph with manual added EDGE
│   │   nodes (via edge_nodes.csv)
│   ├── graph_plannning.py: functionality
│   │   to perform network routing
│   ├── util_planning.py: actual
│   │   implementation of routing algorithm
│   ├── --------------------------
│   ├── plot_map.py: plotting functionality
│   ├── util_plot_graph.py: visualize graph
│   ├── --------------------------
│   ├── test_analysis.py: code to test the
│   │   analysis code, by using a graph
│   │   that is identical to the one from
│   │   the course notes, results are
│   │   presented in Sect. III-B
│   ├── test_routing.py: code to test the
│   │   routing algorithm
│   ├── network_algorithm_Leest_300_CPEs.py:
│   │   run network routing algorithm for
│   │   village with 100 CPEs
│   ├── networkalgorithm_Leest_300_CPEs.py:
│   │   run network routing algorithm for
│   │   village with 300 CPEs
│   └── rain_influence_analysis.py: example
│       to analyze influence of rain on two
│       scenarios
└── Results
    ├── Input_CPE_locations: CPE locations
    │   provided via GRAND
    ├── Graph_representation: graph
    │   representation of input data
    │   ├── Leest_300BS_1U: 300 CPEs
    │   └── Leest_600BS_1U: 600 CPEs
    └── Test_graph_analysis: result of
        graph analysis validation
```

Three types of scripts are present in the Code directory. The first types are the core building blocks of the software framework, to create and analyze graphs, and to perform routing. A second type of scripts contain helper functionality, e.g., for plotting. A third type of scripts is for testing (sub)functionality and to analyze input data.

In the Results directory, we have the visualized input from the GRAND tool, including the network with 900 CPEs that is discussed at the end of Section III-C with two POP locations.

References

[1] "Ieee standard for local and metropolitan area networks part 16: Air interface for broadband wireless access systems," *IEEE Std 802.16-2009 (Revision of IEEE Std 802.16-2004)*, pp. 1–2080, 2009.

[2] A. Al-Saman, M. Cheffena, O. Elijah, Y. A. Al-Gumaei, S. K. Abdul Rahim, and T. Al-Hadhrami, "Survey of millimeter-wave propagation measurements and models in indoor environments," *Electronics*, vol. 10, no. 14, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/14/1653

[3] S. Hur, S. Baek, B. Kim, Y. Chang, A. F. Molisch, T. S. Rappaport, K. Haneda, and J. Park, "Proposal on millimeter-wave channel modeling for 5g cellular system," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 454–469, 2016.

[4] X. Wu, C. Wang, J. Sun, J. Huang, R. Feng, Y. Yang, and X. Ge, "60-ghz millimeter-wave channel measurements and modeling for indoor office environments," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 4, pp. 1912–1924, 2017.

[5] I. A. Hemadeh, K. Satyanarayana, M. El-Hajjar, and L. Hanzo, "Millimeter-wave communications: Physical channel models, design considerations, antenna constructions, and link-budget," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 870–913, 2018.

[6] S. A. Busari, K. M. S. Huq, S. Mumtaz, L. Dai, and J. Rodriguez, "Millimeter-wave massive mimo communication for future wireless systems: A survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 836–869, 2018.

[7] *IEEE 802.11ad - IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band.* IEEE Computer Society, 2012.

[8] *IEEE 802.11ay-2021 - IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Enhanced Throughput for Operation in License-exempt Bands above 45 GHz.* IEEE Computer Society, 2021.

[9] J. Du, D. Chizhik, R. Feick, M. Rodríguez, G. Castro, and R. A. Valenzuela, "Suburban fixed wireless access channel measurements and models at 28 ghz for 90% outdoor coverage," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 1, pp. 411–420, 2020.

[10] J. Zhang and C. Masouros, "Learning-based predictive transmitter-receiver beam alignment in millimeter wave fixed wireless access links," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3268–3282, 2021.

[11] Z. A. Shamsan, "Rainfall and diffraction modeling for millimeter-wave wireless fixed systems," *IEEE Access*, vol. 8, pp. 212 961–212 978, 2020.

[12] B. De Beelde, D. Plets, E. Tanghe, C. Li, and W. Joseph, "V-band rain attenuation measurement setup," in *2022 3rd URSI AT-AP-RASC, Gran Canaria, 29 May – 3 June 2022*, 2022, pp. 1–4.

[13] Z.-K. Weng, A. Kanno, P. T. Dat, K. Inagaki, K. Tanabe, E. Sasaki, T. Kürner, B. K. Jung, and T. Kawanishi, "Millimeter-wave and terahertz fixed wireless link budget evaluation for extreme weather conditions," *IEEE Access*, vol. 9, pp. 163 476–163 491, 2021.

[14] M. Deruyck, E. Tanghe, D. Plets, L. Martens, and W. Joseph, "Optimizing LTE wireless access networks towards power consumption and electromagnetic exposure of human beings," *COMPUTER NETWORKS*, vol. 94, pp. 29–40, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2015.11.023

[15] O. Zahid and S. Salous, "Long-term rain attenuation measurement for short-range mmwave fixed link using dsd and itu-r prediction models," *Radio Science*, vol. 57, no. 4, p. e2021RS007307, 2022, e2021RS007307 2021RS007307. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2021RS007307

[16] ITU-R-P.838-3, "Specific attenuation model for rain for use in prediction methods," International Telecommunication Union, Tech. Rep., 2005.