January 10, 2019

Embedded Ready To Use Hybrid Mesh Network For IoT Applications

Jérôme ELLEOUET
ALE International

Emmanuel HELBERT
ALE International

Docket Number	**FR82018003**
Title	**“Embedded ready to use hybrid mesh network for IoT applications”**
Contributors	**Jérôme ELLEOUET, Emmanuel HELBERT**
Company	**ALE International**

Description of the technical solution:

Gathering data from IoT (**I**nternet **o**f **T**hings) sensors requires a communication infrastructure. This infrastructure could rely on WiFi, LAN or other proprietary networks as well as on mesh networks using any kind of radio technology. Thus, these networks are a prerequisite to conveys this data. In addition, some IoT applications such as asset or human tracking requires a dedicated network of sensors as well. That means that the overall solution requires device provisioning, configuration, installation and maintenance, as well as devices and network planning, leading to huge cost due to manual installation, ad-hoc configuration and maintenance operations.

If we consider the specific case of asset tracking solutions, the common technology is to have devices spread in a building and broadcasting radio waves whose power is measured by a receiver. It is then possible to locate this receiver knowing where the devices are situated in the building. But this requires deploying these devices in the building, changing their battery regularly, that the emitting power is well configured to ensure a good radio coverage of the whole building and that each device is registered on the building map at the right location.

Today, to our knowledge, all of these actions are performed manually and at the very end of an office installation. There might be a configurator tool which is able to calculate how many devices should be used and deployed per floor in a building but it is then necessary to have operators installing these devices as proposed by the tool, confirming the device position in the building, configuring each device (transmission power and beacon) and calibrating the system with a site survey.

This solution aims at removing all the drawbacks of the current solutions by limiting as much as possible the manual operations. Three basic principles are at the core of this solution:

- Devices are embedded in office partitions or in office ceiling (lamp bulbs for instance)
- Device are interconnected in mesh networks and connected to the backbone through some IoT gateways.
- The configuration tool for office partition (Autocad for instance) is also able to calculate the number of devices per floor and their location based on the radio attenuation of the office partitions and doors, the radio parameters of the IoT devices and the type of IoT application.

Put together these three principles solve the four following kind of issues:

Deployment issues: Easy deployment thanks to an early calculation of the number of devices and their location based on the office configuration.

Installation issues: Reduced manpower to deploy and install devices in building. No need for cable deployment

Device configuration issues: Devices are interconnected through a mesh network enabling to push remotely to each device its configuration from a maintenance platform. This interconnection provides also the ability to calibrate the solution and self-healing capabilities as the network can reconfigure itself in case one device is out of order or in maintenance.

Maintenance issues: no need for battery change as the devices could be connected to the main within the office partition.

Looking closely to the solution, itis a system which enables to provision, deploy, install, configure and maintain networks of IoT devices in a building without the need of huge and costly manual operations. The solution is characterized by four mains aspects:

- The devices are embedded in office partitions, floor or ceiling elements. That means that the devices are deployed in the building when installing and configuring the office space. This avoids specific installation afterwards.
- The devices are interconnected in mesh network and linked to the internet through gateways which may also for some of them be embedded in office elements.
- The device locations and radio parameters of each device are calculated when drawing the map of the office (for instance with Autocad tool).
- The network is able to calibrate itself and heals itself if one of the device is out of order or in maintenance

Figure 1 describes a mesh network of IoT devices embedded in office partitions on one floor. The devices of this floor are connected to the internet through two WiFi access point located at two different corners.

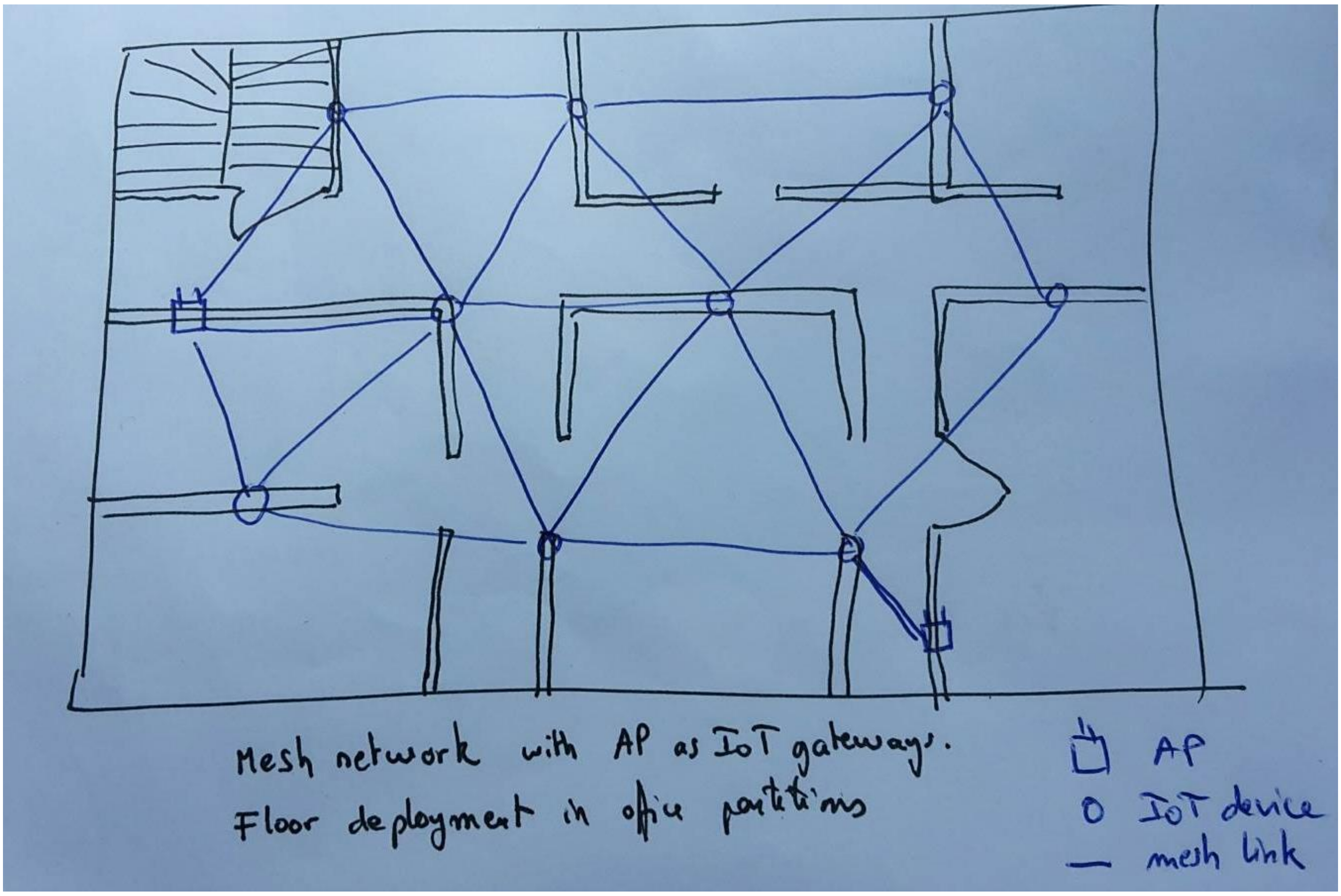
Mesh network with AP as IoT gateways.
Floor deployment in office partitions
AP
IoT device
mesh link

Figure 1

1) Office configuration :

One of the main concern when deploying an IoT network is to manage precisely the location of each device on a map. This is particularly true for location based services such as asset tracking or geofencing. The usual procedure is to indicate on the office map where the devices are after each installation. The solution avoids this issue as the IoT network is defined on the office map prior to the installation at the same time as the office configuration.

The configuration tool (Autocad for instance) is equipped with advanced calculation capabilities. Knowing the radio attenuation of office elements such as ceiling, partition, metallic doors, etc., knowing the radio standard used by the IoT devices and knowing the IoT application to be deployed in the building, we are able to calculate the location and the number of IoT devices in the building. The algorithm is the following and based on two criteria:

- Each device needs to be able to communicate with at least one other device (mesh network). That means the balance of Power should be positive for at least one link.
- A mobile radio receiver or transmitter should detect or be detected (by) a minimum of three devices.

These two criteria depend on the device radio sensitivity, radio emission power and global attenuation between both emitter and receiver. All these parameters depend on the IoT radio standard, the distance between the devices and the office configuration.

As a rule of thumb, one can consider that each device, when emitting at maximum power shall be able to establish a radio link with at least three other devices. That is to say:

Pmax > Ad + Ao + Pr where Pmax is the maximum emitting radio power allowed by the standard, Ad is the attenuation due to the distance between the devices, Ao is the attenuation due to the office elements (partition, doors, furniture…) and Pr is the sensitivity of the receiving device.

2) Embedding the devices :

Embedding an IoT device in an office element leads to several requirements:

- It should be easily reachable for maintenance
- It should be powered to avoid battery change
- WiFi access point should be connected to the wired network
- Antenna should be placed so as not to have the element attenuate the emitted power.

Figure 2 proposes a solution to embed a device (BT Beacon or AP) in an office partition. The partition is pre-wired to connect it to the main network through a dedicated socket. It embeds also a low voltage transformer to feed the IoT device and other devices such as mobile phones. The partition encloses a small closet at the top behind a removable cover where the IoT device (PCB) is plugged. The antenna is printed on the top of the partition and connected to the IoT device through the socket. In addition to the IoT, the partition provides main plugs to connect PC, office lamp, etc.

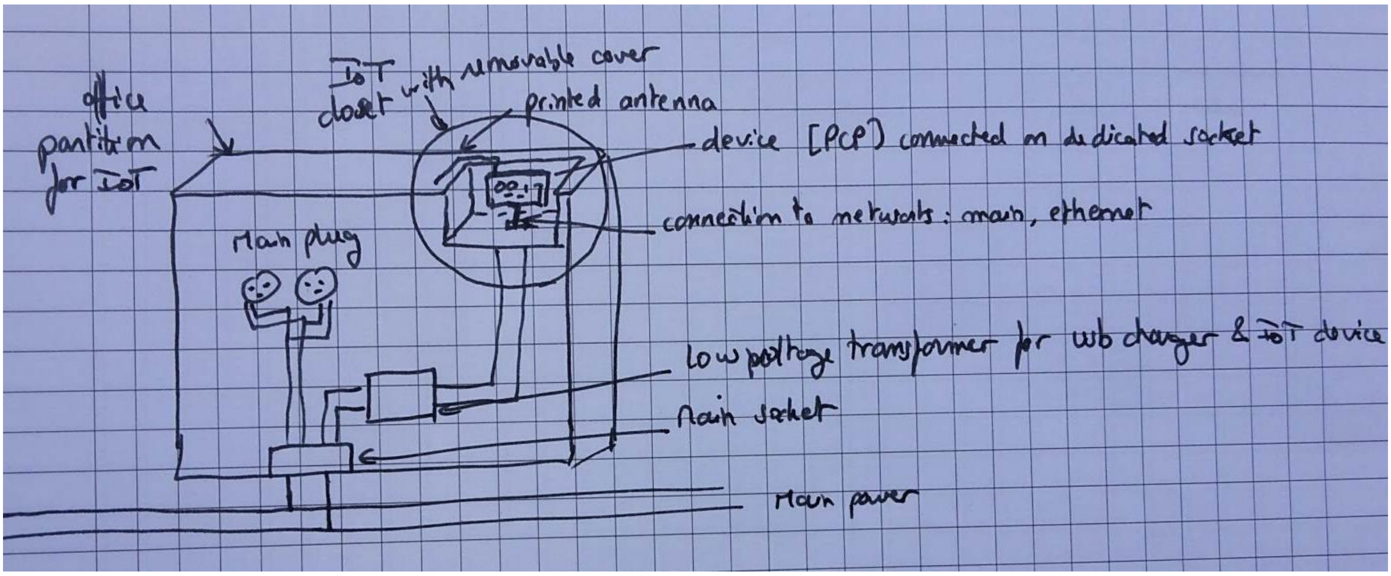
office partition for IoT
IoT closet with removable cover
printed antenna
device [PCP] connected on dedicated socket
connection to networks : main, ethernet
main plug
low voltage transformer for usb charger & IoT device
main socket
main power

Figure 2

3) Running the network :

<u>Initialization:</u>

What we know once the office has been built is that there are IoT devices in office elements and their location. It is then necessary to associate the device Id with these locations. Several solutions could be foreseen.

- The first one is to make a correlation between the IoT unique identifier and the office element serial number at manufacturing. When building the office, the location of this element is recorded in the map.
- The second solution which is more convenient is to detect automatically the location of each device thanks to a simple algorithm. This is what we describe in this solution below.

The second thing to do is to configure the radio parameters of each device and the application profile (mode - emitter or receiver, Tx power, latency). This is done upon the IoT monitoring application request, relayed by the IoT controllers (typically a WIFI Access Point with built-in BLE chipset acts as a BT mesh controller) and gateways and transferred to each device through the mesh network. **Figure 3** describes the overall architecture supporting this initial configuration.

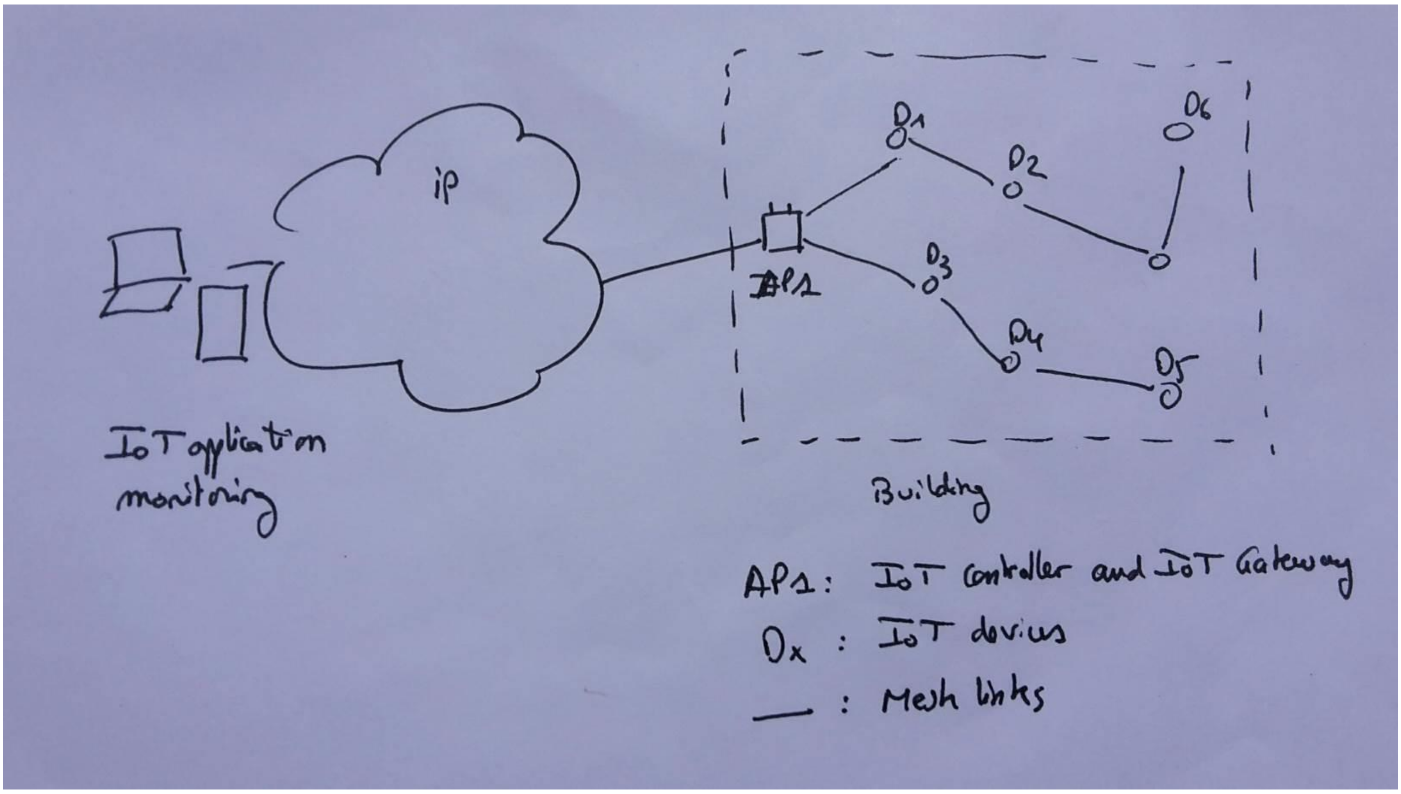
ip
D1
D2
D3
D4
D5
D6
AP1
IoT application monitoring
Building
AP1: IoT controller and IoT Gateway
Dx : IoT devices
: Mesh links

Figure 3

Application embodiment :

Figure 4 below describes the system in two modes:

Geo fencing or location based services: A user is moving within the building with a device detecting BT beacons emitted by the IoT devices. The local application installed on the mobile device measures the radio power of the closest IoT devices and is able to calculate locally its position in the building and trigger the relevant service. Stellar LBS manages today such features. For this service, the IoT device transmit periodically bursts of radio power with their Id and their tx power. As per reference to **figure 4**, the mobile device measures the Tx power Px of devices D5, D8 and D9 and calculates its position. Positioning information of IoT devices (typically BLE beacons) are known by the application and coming from the IoT Monitoring application.

Asset tracking: An asset is equipped with a BT beacon emitting regularly its Id and its Tx power. This power is measured at each IoT device under the asset radio coverage. The IoT gateway subscribe to notifications/frames from a given IoT device. The power measured and the IoT device Id are then transmitted through the mesh network towards the IoT gateway (again, typically a WLAN AP) which forwards the whole information, if issued by authorized devices, to a central system that will process the precise location of the asset and trigger the relevant service if necessary. As per reference to **figure 4**, the beacon tied to the asset broadcast its Identifier AssetId with a radio power of P_{asset}. Devices D1, D2 and D4 measure the radio power $P_{received}$ at device reception and forward the couple (AssedId,$P_{received}$) towards the IoT gateway AP1. Data issued by D4 and D1 are sent directly to AP1 whereas data issued by D2 is transmitted hop by hop through the mesh network using D1 as forwarding node.

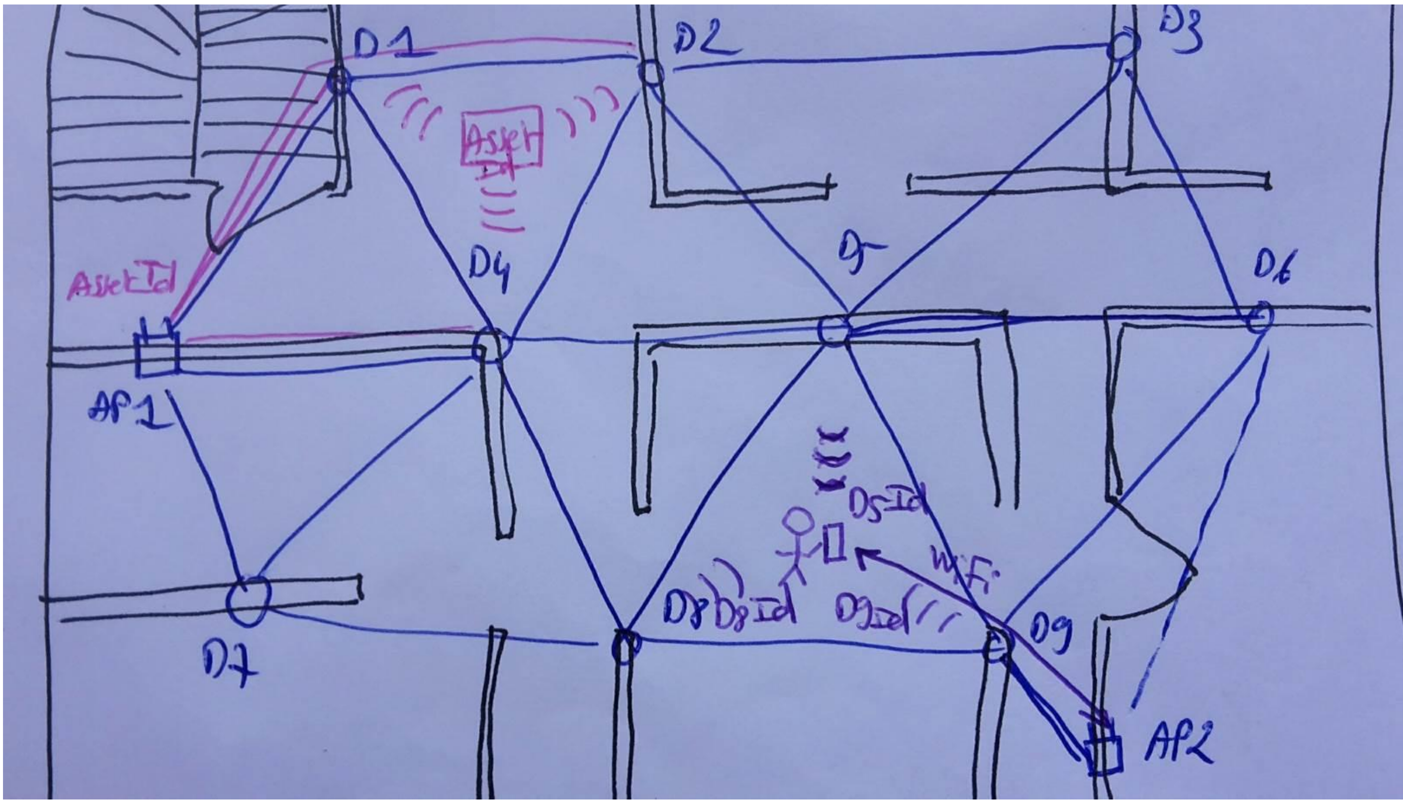

Figure 4 : Location based service and IoT mesh networks

Self-healing:

In case on device gets out of order, the neighbor devices will report the event to the system by detecting that the device has stopped emitting radio. It is then possible to heal the network by increasing the Tx power emitted by the remaining devices to maintain a functional mesh network and ensure enough power at mobile device reception (location based service mode). The system will also report the incident for maintenance and precise which device has to be changed or fixed.

Automatic map provisioning:

This process aims at associating automatically the IoT device location and its identifier so as to be able to run location based services. The algorithm is based on two elements:

- We know where devices are located, but we don't know which device is where. Let's call these locations x1, x2, x3…ap1 and ap2 for WiFi access points.
- Each device is able to report the id of the other devices located in its close area. We can set radio power threshold to discriminate and keep the closest devices in the area. Let's write these Ids D1, D2, D3,…AP1 and AP2 for WiFi access points as noted in figure 4. But what we don't know is where are D1, D2, …

The purpose of the algorithm is to calculate the association between x_i and D_i. That is to say D1 is at x1, D2 is at x2, etc.

Based on the two hypotheses above we can build these two sets of tables:

What x_i are supposed to detect:

ap1 can see	**x1 can see**	**x2 can see**	**x3 can see**	**x4 can see**	**x5 can see**	**x6 can see**	**x7 can see**	**x8 can see**	**x9 can see**	**ap2 can see**
x1	ap1	x1	x2	ap1	x4	x3	ap1	x4	x5	x6
x4	x4	x4	x5	x1	x6	x5	x4	x7	x6	x9
x7	x2	x5	x6	x7	x3	x9	x8	x5	x8	
		x3		x8	x8	ap2		x9	ap2	
				x5	x9					

What D_i actually detect:

AP1 sees	**D1 sees**	**D2 sees**	**D3 sees**	**D4 sees**	**D5 sees**	**D6 sees**	**D7 sees**	**D8 sees**	**D9 sees**	**AP2 sees**
D1	AP1	D1	D2	AP1	D4	D3	AP1	D4	D5	D6
D4	D4	D4	D5	D1	D6	D5	D4	D7	D6	D9
D7	D2	D5	D6	D7	D3	D9	D8	D5	D8	
		D3		D8	D8	AP2		D9	AP2	
				D5	D9					

Let's assume we know where AP1 is located. AP1 = ap1. Now, we consider the different hypotheses for x1, x4 and x7.

base ap1	Hypothese 1	Hypothese 2	Hypothese 3	Hypothese 4
x1	D7	D4	D1	D1
x4	D1	D1	D7	D4
x7	D4	D7	D4	D7

If D7 = x1, D7 should see what x1 can see: ap1, x4, x2. But D7 actually sees AP1, D4 and D8 and with hypothesis 1, ap1, x7 and D8. This is incoherent because x7 is not seen by x1. Let's try hypothesis 2.

If D4 = x1, D4 should see what x1 can see: ap1, x4, x2. But D4 actually sees AP1, D1, D7, D8 and D5 and with hypothesis 2, ap1, x4, x7, D8 and D5. This is incoherent because x7 is not seen by x1 and there are more devices seen than expected. Let's try hypothesis 3.

If D1 = x1, D1 should see what x1 can see: ap1, x4, x2. D1 actually sees AP1, D4 and D2 and with hypothesis 3, ap1, x7 and D2. This is incoherent because x7 is not seen by x1. Let's try hypothesis 4. In that case, D1 sees ap1, x4 and D2 which is coherent if we assume that D2 = x2.

Now, let’s consider new hypotheses for this new position x2. X2 can see two new devices located in x5 and x3. Let’s try to identify who is x5 and x3.

base ap1	Hypothese 5 for x2	Hypothese 6
x1	**D1**	D1
x4	**D4**	D4
x7	**D7**	D7
x2	**D2**	D2
x5	D3	D5
x3	D5	D3

If D5 = x3, D5 should see what x3 can see: x2, x5 and x6. D5 actually sees D4, D6, D3, D8 et D9 and with hypothesis 5, x4, D6, x3, D8 and D9. This is incoherent as x3 can’t see x4 and there are more devices than expected. Let’s try hypothesis 6.

if D3 = x3, D3 should see what x3 can see: x2, x5 et x6. D3 actually sees D2, D5 et D6 and with hypothesis 6, x2, x5, D6 which is coherent if we assume that x6 = D6

Now, let’s consider new hypotheses for this new position x6. x6 can see two new devices located in x9 and ap2. Let’s try to identify who is x9 and ap2.

base ap1	Hypothese 7 for x6	Hypothese 8
x1	D1	D1
x4	D4	D4
x7	D7	D7
x2	D2	D2
x5	D5	D5
x3	D3	D3
x6	D6	D6
x9	AP2	D9
ap2	D9	AP2
x8		D8

If AP2 = x9, AP2 should see what x9 can see: x5, x6, x8 and ap2. AP2 actually sees D6 and D9 and with hypothesis 7, x6 and ap2. This is incoherent as x9 can see more devices than expected. Let’s try hypothesis 8.

if D9 = x9, D9 should see what x9 can see: x5, x6, x8 and ap2. D9 actually sees D5, D6, D8 and AP2 and with hypothesis 8, x5, x6, D8 and ap2 which is coherent if we assume that x8 = D8

All ids have now been located.

We can generalize this algorithm as follow:

```
new_device = first known location;
While (remaining deviceId to locate)
{
    For each (location_hypothesis(new_device))
    {
        Update_with_known_location(topology_seen);
        If (compare (topology seen; topology expected))
        {
            Update Ids with locations;
            Update(new_device);
            break;
    }
}
```