COMPARISON OF BEAM-FORMING & RELAYING IN SPARSE SENSOR NETWORKS

MIKULÁŠ KREBS

This study focuses on the differences in power consumption between beam-forming and relaying data transmission methods in a sparse wireless ad-hoc network. These two methods are observed for the same parameters using an identical network topology in a simulation programme that was developed as a part of this study.

This paper introduces the reader the background of sensor networks and exposes the aims of this study and methods used to simulate these networks. Finally, the results of a simulation are analysed and the two methods are compared, followed by the conclusion of the study and the project itself.

INTRODUCTION

This paper explains and informs the reader about the background, the methods used, and the decision-making process that had to be implemented during this project. It is structured in four chapters. Each chapter focuses on a different aspect of the work while always exposing the research behind it and the decision-making process. The introduction explains the project background and then its general aims. The first chapter explains the project requirements and the way they were verified. This is followed by a chapter on the general design of the project. The third chapter goes into a more detailed description of the processes behind the simulation and the reasoning behind them followed by a chapter on testing the solution. The fourth chapter shows the results of the project including the data generated and its analysis. While the last chapter is focused on reflecting about the whole project, what possibilities there were, which were used, and what could be done with the resulting product in the future.

PROJECT BACKGROUND

"A wireless ad hoc [or sensor] network (WANET) is a decentralised type of wireless network. The network is ad hoc because it does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data from other nodes, so the determination of which nodes forward data is made dynamically on the basis of network connectivity. In addition to the classic routing, ad hoc networks can use flooding for forwarding data"(Gupta and Kumar, 2000).

This project is focused on a particular type of network called an ad-hoc network. Ad-hoc and sensor networks are a particular type of wireless networks composed of nodes (or modules) which require no infrastructure and can connect without any pre-established hierarchy. Such networks are very important for environmental, civil, and military applications. For example, the sensors can measure the temperature or humidity in the soil (environmental), can measure tiny vibrations in the debris of a collapsed building, thus signalling the existence of survivors (civil), or can measure vibrations in the ground signalling a moving person in a forbidden area (military). These types of networks have huge advantages over classical wireless networks because they require no infrastructure or pre-setup, but with this comes inherent problems which have to be solved. This project focuses on a specific problem present in this type of a network.

GENERAL AIMS

There are times in which the modules cannot be placed in a specific area (e.g. because there is a lake or enemy territory). In such cases, communication can be disrupted by the lack of modules in that region creating so-called holes in the network due to the modules' physical limitation of wireless transmission. An ad-hoc network with holes is called a sparse network, thus an ad-hoc network without holes and with a uniform distribution of modules are called dense networks. This problem is currently solved by relaying information over multi-hop paths around the region with no modules (Fig. 1). But this method is problematic. In fact, relaying the information along the same multi-hop path will in time deplete the power source of the relaying modules disconnecting them in the end and therefore disrupting the network. Although these modules do not actually generate any new information, they only carry information generated by modules which could not directly communicate due to the huge distance between them. An alternative method of transmitting through these holes could be theoretically used. This method used distributed phase-shift beam-forming (further DPSBF) which enables clusters (groups) of modules to tune their transmission signal in such a way that their signals perfectly overlaps at the receiver (resonate). This allows the groups of wireless transmitters to behave as a single, virtual, more powerful radio transmitter and thus cover a larger radio communication distance (Fig. 2). The goal of this project is to construct an application that will be able to simulate these two different methods of information transmission in a specifically defined ad-hoc network to compare the difference of these two methods for an identical setup.

Figure 1: Relaying information over a multi-hop path around the zone with no modules (a lake for example)

Figure 2: Modules transmit in the same time the same information using DPSBF and achieve together higher communication distance

1. GENERAL DESIGN

This section presents the general structure of the application with the inputs, the general processing structure and the outputs of the programme.

1.1. DATA INPUT AND DEFINITION

This section presents and explains the design of the simulator and the motivations behind the design. The main research in this area was about the use of graphs in representation of the network and the use of the coordinate system. The use of graph system to study a network is very natural in fact the network theory is a subset of the graph theory and is used in many scientific articles such as Newman's "The Structure and Function of Complex Networks."

The simulator application takes many input parameters, but they can all be divided in two main categories:

- Network Definition
- Module Definition

The first category – *Network Definition* – contains everything that is concerned in the topology of the network, while the second category – *Module Definition* – contains all the physical parameters for the simulation.

To define the topology of the network and therefore of the simulation, we need the location of all the nodes that will participate at the transmissions. For this purpose a simple two dimensional Cartesian coordinate system. This was done because it is a very simple and efficient way to locate objects in this system and work with them.

In this system a module/node is represented by a point which the location is expressed using two coordinates: the abscissa representing the value on the x axis and the ordinate representing the value on the y axis. The hole in the network is a square and it can therefore be sufficiently represented by only two points which define the diagonal. The user can thereby input only the top left point and the bottom right point coordinates to define the hole, and this simulation calculates the square from this simple input, using simple geometry. The simulator uses the graph naming convention; in graph theory, vertex and node are synonymous and can be considered as equivalent in this project. The last topological parameter is the starting and the ending node. These are the nodes where the information transmission respectively starts and ends. This means that the first module in the transmission path is the starting node and the last receiving node is the ending node.

To define the module specification means that we need several physical parameters describing this module. These parameters are as follows:

- data-rate
- battery capacity
- transmitter power
- transmitter frequency
- receiver sensitivity
- antenna gain
- packet size
- voltage

The data-rate or also bit rate is the number of bits per second. It is the speed of data transmission. The battery capacity is the number of charges that the battery contains and is expressed in mAh. The transmitter power is the maximal power that the transmitter can output in the antenna and is expressed in mW. The transmitter frequency is the frequency of the waves that are transmitted by the module and is expressed in MHz. The transmitter sensitivity is the minimal magnitude of input signal required for the signal to be recognisable by the receiver. The antenna gain is conveying how much it focuses the signal in comparison to an ideal dipole and is expressed in dBi. The packet is a data unit in networking, packet size is its size in bytes. Voltage is the potential difference of the current in the module circuits expressed in V.

These physical characteristics are expressed in units most commonly found in the networking industry. They are not equivalent, and the application therefore needs to perform several unit conversion procedures when dealing with these inputs. This is so that the user does not have to make the conversions by himself/herself.

I had to decide what programming platform to use for the development. As I knew I would need to be able to develop in a fast way and the application should preferably be available for multiple operating systems (Win and OSX), I chose Java. Java has the big advantage of great free libraries and running on the Java Virtual Machine (JVM) can be easily run on many platforms.

1.2. SIMULATION PROCESSING STRUCTURE

After the input data has been defined by the user through the UI menu, the programme can start the simulation process. This section walks the reader through the logical structure of procedures followed in the application.

I needed a simple but effective way to save the output of the simulation. In fact the output of the simulation was clearly a big amount of data as each of the networking modules needed an evolution of energy consumption over time. After considering multiple options such as xml or even a database system, I chose a simple comma separated value system (CSV), not only for its simplicity, but also for the overwhelming compatibility with every system imaginable. In fact, a CSV file is a simple text file and only by the way that the data is formatted, i.e. separated by commas over multiple lines, can be understood by programmes such as Microsoft Excel, Libre Office, and others. This is very important because it means that the output of my programme is not locked to my application, i.e. it can be understood, opened, and visualised with a standard programme. It also enables the user to evaluate the data in many more ways that would be able to do normally. In fact, this format allows the user to easily visualise the trends and make graphs out of it in a spreadsheet application of his choice.

Figure 3: Simulator internal structure

The application follows loosely the structure of model/controller where the model is mostly responsible for object storage and the control performs most of the actions. The procedure starts with the module entity calculating additional module characteristics from the user input such as maximal transmission distance or the wavelength; these are then stored in this model entity for further use and reference. Then the relay entity steps in by calling on the graph entity which constructs a weighted graph that the relay method needs for its operations. The graph entity uses the information from the module and the node entities to construct this weighted graph that represents the modules in the network with the connections representing the possibility of communication and their weight the distance between the respective modules. The relay entity uses the graph as a basis for the relay method of information transmission in the network.

The beam-forming entity has a very similar function to the relay entity, but instead of the relay method it takes care of the beam-forming method. To do this, it also uses the same weighted graph constructed by the graph entity. In addition, it needs the information from the region entity which tells it the location of the hole/lake in the network. The two entities relay and beam-forming construct and fill the power consumption tables and send them to the data entity which saves them respectively as RelayBrief.csv and BFBrief.csv in the working directory.

1.3. RELAY DESIGN

This section provides more concrete information on the inner workings of the relaying method. I also needed to deepen my knowledge of graph theory which was really important for the path-finding in the network. Then I needed to research Java libraries that are focused on this field. I found multiple of them including Jung, JgraphT, and JgraphX. I chose JgraphT as it is focused on the processing of graphs and contains a lot of useful algorithms such as a collection of algorithms used to find the shortest possible path in a graph. This meant that I would be able to use the library to implement the path-finding in the network.

I had to research the possible algorithms used to find the path as each excels in a different aspect. I started my search on Wikipedia as it gives a nice and concise overview continuing by searching the information about the specific algorithms and their implementation in Java on various programming forums. I also looked at the site and knowledge repository of Cisco, a major manufacturer of networking technologies, to find what algorithms are they using in their hardware. Finally, I chose the Dijkstra's algorithm as it is used in real networking protocols such as OSPF (Cisco, 2003).

The relay method has a starting node and an ending node. There is a set amount of packets that need to be transported from the starting node to the end node. In the relay method this is done by relaying the packets from node to node in a specific manner. The path for the transport is found by a particular form of the breadth-first searching algorithm called Dijkstra's algorithm. This algorithm searches the graph and finds the shortest path between two nodes. In the search for the shortest path, only the number of transmissions is taken into account. This is an algorithm that is really used in networking technologies such as the RIP networking protocol. The data is then sent packet per packet through the path each time calculating the power consumption for pairs of modules participating at the transmission in the proper path step. This means that the end node is only receiving.

The modules in this project use a single transmission range this means that a node cannot transmit and receive at the same time. This is why the transmission time is basically halved and there is always a transmission period for a module and then a receiving period.

This means that a module first receives a packet and then it transmits the packet further. Because of this, the simulator needs to calculate the proper transmission time from the absolute time.

1.4. BEAM-FORMING DESIGN

This section provides more concrete information on the inner workings of the beam-forming method. The beam-forming method has the same starting and ending node as the relay method and the same number of packets needs to be transported from the start node to the end node. But the process is quite different.

In fact the beam-forming method can be divided into two main phases:

- setup/preparation phase
- transportation phase

The setup phase consists in finding the paths that the information will take in the network including the finding of the beam-forming pairs and the transportation phase assists in the simulation calculations and data construction.

SETUP/PREPARATION PHASE

1. Calculate the shore distance D
2. Find beam-forming pairs using combinatorial from the red zone (Fig.4)
3. Calculate the range for all pairs
4. Choose the pair that can reach farthest
5. Choose beam-forming target from the green zone (Fig.4) and closest to the end node

Figure 4: Beam-forming setup

The red zone is defined partly by the lake constraints and a maximal distance from shore D . The green zone is also partly defined by the constraints of the lake and by a circle with the centre as the midpoint of the chosen beam-forming pair and radius equal to the maximal transmission range of the beam-forming pair denoted $dMax$. The maximal transmission is discussed further in 2.2. *Derived Equations Equation (4)*.

The distance D is calculated using the following formula:

$$D = \frac{\lambda}{4\pi} \times \sqrt{\frac{P_T}{\Delta P}} - d$$

Where D is the maximal distance from shore, the variable λ is the wavelength, d is the length of the lake, and π is a mathematical constant with the approximate value of 3.14159. The variable P_T is the transmitter maximal power and ΔP is how much we want the second module in the beam-forming group to add to the target location in terms of power. This equation makes sure that we look for beam-forming pairs only in the zone where it makes sense. In fact, the farther from the zone we look, the less the module would matter in terms of additional power in the beam-forming transmission.

TRANSPORTATION PHASE:

The transportation phase uses the data from the setup phase. In the second phase there are in fact three path sections. The first is a traditional relay between the start node and the beam-forming group, the second stage is the actual beam-forming transmission, and finally there is another traditional relay between the beam-forming target node and the end node.

This is also reflected in the simulation, the data is first calculated for the first path then for the beam-forming and finally for the second relay path.

2. MATHEMATICAL DESIGN AND BACKGROUND

This section introduces the reader to the mathematical and physical background present in this project. It presents the equations used and explains where they come from. The physical theory used in this project is a genuine scientific endeavor as it is taken from the "Antenna Theory Analysis and Design" by Balanis (2005). The same theory for one antenna is also presented in a book by Haupt (2010). It has also been used in a number of scientific articles on the subject (Moucha, Cerny and Kubr). We can therefore see that the theory is used in practice for similar projects and is a good option to use.

2.1. INITIAL MATHEMATICAL THEORY

The initial theory used is the free-space propagation model explained in "Antenna Theory Analysis and Design" by Balanis (2005). This section presents these general physical formulas that are included in this model.

$$P_R = P_T \times G_T \times G_R \times \left(\frac{\lambda}{4\pi d}\right)^2$$

Equation (2) is a general formula to calculate the received power P_R when transmitter uses power P_T and the transmitter and receiver have an antenna with a respective gain of G_T and G_R . The variable λ is the wavelength, d is the distance between the transmitter and the receiver, and π is a mathematical constant with the approximate value of 3.14159. This equation is in traditional units with P_R and P_T expressed in *milliwatt (mW)*, G_T and G_R expressed as a ratio between the surface of the radiation pattern of the dipole and the surface of the antenna's radiation pattern. The variables λ and d are expressed in meters.

$$P_R = P_T + G_T + G_R + 10 \log_{10} \left(\frac{\lambda}{4\pi d}\right)^2$$

Equation (3) is an equivalent equation to equation (2) but this time it is expressed in *decibels (dB)*, a logarithmic unit of measurement. This means that P_R and P_T are expressed in *dBm*, a decibel equivalent of power measurement. While G_T and G_R are expressed in *dBd*, a decibel measurement unit for the gain of an antenna compared with the isotropic antenna/ideal dipole.

2.2. DERIVED EQUATIONS

The initial theory general formulas were not the specific equations needed for the simulator to make the calculations. Different, more specific formulas needed to be derived from these general equations. This section presents the formulas used in the simulator and the way they were derived from the general formulas.

Equation (4) is used to determine the maximal distance of transmission between two modules of the same type. This is done by the module entity (Fig. 3) and is considered as a complex module characteristic that is derived from the user inputs. A slightly modified equation (4) is also used to determine the maximal transmission distance of the beam-forming group. In this modified equation the P_r is changed to P_r^2 because the group is composed of two modules.

$$P_R = P_T \times G_T \times G_R \times \left(\frac{\lambda}{4\pi d}\right)^2 \Leftrightarrow \sqrt{\frac{P_R}{P_T \times G_T \times G_R}} = \frac{\lambda}{4\pi d} \Leftrightarrow \sqrt{\frac{P_R}{P_T \times G_T \times G_R}} \times 4\pi d = \lambda$$

$$\Leftrightarrow d = \frac{\lambda}{\sqrt{\frac{P_R}{P_T \times G_T \times G_R}} \times 4\pi}$$

Equation (5) is used by the relay entity and the beam-forming entity to calculate the power that is transmitted in a second between two identical modules during transmission.

The relay entity uses this equation for all steps in the communication path as every step in the method are only two modules participating in the communication. While on the other hand, the beam-forming entity uses this only to compute the data transmission from the starting node to the beam-forming coordinator and then from the beam-forming target to the ending node. This is because the transmission done by the beam-forming group is effectuated at maximum power and does not therefore need to be computed.

$$P_R = P_T + G_T + G_R + 10 \log_{10} \left(\frac{\lambda}{4\pi d}\right)^2 \Leftrightarrow P_R = P_T + G_T + G_R + 20 \log_{10} \left(\frac{\lambda}{4\pi d}\right)$$

$$\Leftrightarrow P_T = P_R - G_T - G_R - 20 \log_{10} \left(\frac{\lambda}{4\pi d}\right)$$

Equation (6) is used to compute the power consumption of a transmission in both relay and beam-forming entities. PC is the power consumption expressed in *Ampere-hour (Ah)*. PT is the transmitted power expressed in *watts (W)*. t is the duration of the transmission expressed in *seconds (s)* and V is the potential difference of the current in the module circuits expressed in *volt (V)*.

$$PC = \frac{PT \times t}{V \times 3600}$$

3. TESTING

This section shows the principles used during the testing of the application and its inner workings as a simulator while showing examples along the way.

To test the application, a collection of data was used. A specific topology was constructed which was fed to the simulator. Meanwhile, a series of test calculations were made by hand. The testing topology contains a series of fourteen nodes arranged so that they allow for the use of both transportation methods.

Figure 5: Testing network topology

The graph was then verified using human made calculations and visualised using Geogebra. As the equations are the same for each node couples, it is sufficient to do human made calculations for sample data and not for all of the nodes, as if we find the same results for a couple it will be consistent across the whole graph.

The input data are as follows:

*Data rate = 115 kbps, battery capacity = 200 mAh, transmitter power = 63 mW,
transmitter frequency = 2450 Mhz, receiver sensitivity = -64 dBm, antenna gain = 2.15 dBi,
packet size = 128 B, voltage = 5 V*

$$dMax = \frac{\lambda}{\sqrt{\frac{P_R}{P_T \times G_T \times G_R}} \times 4\pi}$$

As we know from equation (4) the 2.2. Derived equations. P_R is the sensitivity of the transmitter equal to -64 dBm .

$$\lambda = \frac{c}{f} = \frac{2.998 \times 10^8}{2450 \times 10^6} = 0.1223 (m)$$

Where c is the speed of light in vacuum and f is the frequency of the light-wave.

We need additional conversions as the units must correspond one with each other.

$$P_R = -64 \text{ dBm} = 3.981 \times 10^{-7} \text{ mW}$$

$$G_T = G_R = 2.15 \text{ dBi} = 10^{2.15/10} = 1.6405 (\text{ratio})$$

This conversion needs to take place as we cannot use decibels and we must convert the gain in decibels in a ratio that does not have a unit. It becomes a constant multiplier. This was further explained in the Section 2.1. Initial Mathematical Theory.

This gives us

$$dMax = \frac{0.1223}{\sqrt{\frac{3.981 \times 10^{-7}}{63 \times 1.6405^2}} \times 4\pi} = 200.9 (m)$$

while the simulator gives us a value of $200.9675...$ without cutting the value short.

And for the

$$dBFM_{ax} = \frac{0.1223}{\sqrt{\frac{3.981 \times 10^{-7}}{63^2 \times 1.6405^2}} \times 4\pi} = 284.2 (m)$$

while the simulator gives us a value of 284.2110.. without cutting the value short.

We can therefore see that the testing was conclusively correct.

When the graph construction was deemed correct, the testing proceeded to the next phase, the testing of the power consumption. This was done in the same spirit as the graph testing i.e. the power consumption was calculated for a pair of nodes and compared to the simulator data. In this section, the testing found a lot of inconsistency and problems. In fact, the values differed wildly from the human made calculations. After recalculating the results to know there was not a fault in the human factor, it was sure that there must be a problem in the simulation. A step-by-step debugging process began where the simulation calculation needed to be divided in the smallest steps possible to see where the inconsistency was. In fact, there was not only one inconsistency found, but several, including a missing bracket and an incorrect unit conversion.

Figure 7: Simulator test beam-forming method path

In fact, it was very challenging to write the quite complex equations used in the programming language. This was because the equations needed to be written inline using only brackets. In this way even quite simple to understand equations become very unreadable by the human eye. After the debugging process was done, the power consumption testing results were compared to the newly simulated data and deemed correct.

$$P_T = P_R - G_T - G_R - 20 \log_{10} \left(\frac{\lambda}{4\pi d} \right), \text{ see equation (5) in 2.2.Derived Equations.}$$

We calculate using this, the power constant for a specific couple of nodes.

$$P_T = -64 - 2.15 - 2.15 - 20 \log_{10} \left(\frac{0.12236}{4\pi 114.127} \right) = -63.8 - 20 \log_{10} \left(\frac{0.12236}{1434.1621} \right) = 13.074 \text{ (dBm)}$$

$$P_T = 0.0203 \text{ (W)}$$

Now the final derived equation is to be used to calculate the battery capacity used during a transmission at a time that the simulator does the same calculation. The value of t was chosen *14262 seconds*, the first value in the table with real transmigration time of *6957 seconds*.

$$PC[Ah] = \frac{PT \times t}{V \times 3600}, \text{ see equation (6) in 2.2.Derived Equations.}$$

$$PC[mAh] = \frac{0.0203 \times 6957}{5 \times 3.600} = 7.845 \text{ (mAh)}$$

Now the only thing left to do is to subtract this value from the battery capacity of the module.

$BC - PC = 200 - 7.845 = 192.155 \text{ (mAh)}$ which sufficiently compares to the simulator value of *192.147(mAh)* as we have a precision at the tenth of a *mAh*.

4. RESULTS AND CONCLUSIONS

There are many ways that the simulator result could be analysed in order to answer the initial question that the simulator should answer. The question remains, which method is more efficient?

One way to look at it is to look at the evolution of the energy of the whole system, not only the individual modules. Another way would be to examine the modules, see which are depleted first, and what would that means for the network; will it still be able to relay information or not?

For the purpose of our project, the most interesting analysis method is the first one that explains the network system in its entirety. This method itself can be done in multiple ways, either we can consider a positive value that would represent all the energy of all the modules in the network and see how this energy is depleted over time, the second would be to consider a negative value that would represent only the depletion of the energy over time. The second method is a better approach in the case of this work because the first method would include in the energy levels all the nodes in the network. This would mean that the energy level would include even nodes that do not participate at the transmissions. That is why the second method is much more appropriate and representative of the actual situation in the network.

To apply this method, each value level of energy of each module must be added together in the same time frame while for each we subtract the maximal battery capacity of the module. We get the following formula: $E_{loss} = (E_{N0} - BC) + (E_{N1} - BC) + \dots + (E_{Nn} - BC)$ where E_{loss} is the overall energy loss of the whole system, E_{Nn} is the current energy level of a module with number n , and BC is the maximal battery capacity of the module. If we do this for both briefs that the simulator gives us, we get the following results.

Time [s]	v0	v2	v4	v8	v9	v10	Overall energy use
0	200	200	200	200	200	200	
14262	192.14727	176.7107	176.557003	187.9421013	185.45516	192.1473035	-89.0404167169
28524	184.29455	153.4215	153.113977	175.8841718	170.91028	184.2945769	-178.0809914962
42786	176.44182	130.1322	129.67095	163.8262423	156.3654	176.4418503	-267.1215662755
57048	168.58909	106.8429	106.227924	151.7683128	141.82053	168.5891237	-356.1621410548
71310	160.73637	83.55359	82.7848971	139.7103832	127.27565	160.7363971	-445.2027158342
85572	152.88364	60.2643	59.3418705	127.6524537	112.73077	152.8836806	-534.2432805628
99834	145.03092	36.97501	35.8988739	115.5945242	98.185913	145.0309439	-623.2838067215
114096	137.1782	13.68573	12.4558474	103.5365947	83.641036	137.1782173	-712.3243815008
128358	129.32547	-9.603562	-10.9871792	91.47866515	69.096158	129.3254907	-801.3649562801
142620	121.47274	-32.89285	-34.4302058	79.42073562	54.551281	121.4727641	-890.4055310594

Figure 8: Relay Brief Overall Energy Level Analysis

These tests were made with the same parameters as the testing in the section 4. Testing of this paper. As we can see in both Figures 8 and 9, the overall energy use linearly increases.

This is predictable as the path is invariable and the conditions stay the same during the whole simulation. We can see that if we compare the energy use in the relay method in Figure 8 and the energy use of the beam-forming method in Figure 9 the overall energy use of the beam-forming system is smaller than the energy use of the relay system. This conclusively shows that the beam-forming method is more efficient in this network than the relay method.

Time [s]	v0	v1	v7/v6	v11	Overall Energy Use
0	200	200	200	200	
14262	192.1472733973	179.0740929901	175.6502416696	197.452762388	-80.0253878855
28524	184.2945467946	158.1481591971	151.3004833391	194.905524776	-160.050802554
42786	176.4418201919	137.2222254042	126.9507250087	192.358287164	-240.0762172226
57048	168.5890935892	116.2962916112	102.6009666783	189.811049552	-320.1016318912
71310	160.7363669865	95.3703578182	78.2512083478	187.2638119399	-400.1270465597
85572	152.8836403838	74.4444240253	53.9014500174	184.7165743279	-480.1524612283
99834	145.0309238317	53.5184902323	29.5517228522	182.1693399761	-560.1778002555
114096	137.178197229	32.5925564393	5.2019645217	179.6221023641	-640.203214924
128358	129.3254706263	11.6666226464	-19.1477938087	177.0748647521	-720.2286295926
142620	121.4727440236	-9.2593111466	-43.4975521391	174.5276271401	-800.2540442611

Figure 9: BF Brief Overall Energy Level Analysis

It is worth mentioning though that the difference is significant but not enormous. We can also use look at the evolution of energy of the individual modules if we use a chart to visualise the data from the relay brief and the beam-forming brief.

Figure 10: Relay node comparison

Figure 11: Beam-forming node comparison

As we know from the overall energy loss analysis, the beam-forming method is more efficient than the relay method, but if we look at Figures 10 and 11 we can see that the beam-forming group constituted from node v7 and v6 do most of the "heavy lifting" which means that even though they save quite a lot of energy in the system, these nodes in particular are depleted faster than any nodes in the relay method. This fact results in a sooner severing of the connection through this path in the beam-forming method use. This means that the use of beam-forming is definitely a useful and more efficient tool but should be used in particular situations and not as a general rule of thumb.

In further inquiries there is the potential to examine the situation of beam-forming vs. relaying for bigger beam-forming groups. In fact, using these bigger beam-forming groups would probably not change the overall efficiency but could push the severing of the network beyond the relay method as there would be more nodes participating at the same beam-forming transmission. But with this comes more inherent problems such as interference of the beam-forming group with the rest of the network, etc. which would be very difficult to predict without real experimental testing. ■

REFERENCES

- Cisco. (2003) *OSPF Incremental SPF*. [Online]. Available at: http://www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/ospfispf.html (Accessed 17 November 2015).
- Constantine A. Balanis, John Wiley and Sons. (2005) *Antenna Theory Analysis and Design*. 3rd edn., pp. 962-977.
- Gupta, P. and Kumar, P. (2000) 'The capacity of wireless networks', *IEEE Transactions on Information Theory*, 46(2), pp. 388-404.
- Haupt, Randy L. (2010) *Antenna Arrays*. Hoboken, N.J.: Wiley-IEEE Press, pp. 17-19.
- Moucha, A., Cerny, V. and J Kubr, J. (2012) 'Distributed Phase-Shift Beamformed Power Balancing In Ad-Hoc And Sensor Networks', *The Telecommunications Systems Journal*.
- Newman, M.E.J. (no date) 'The Structure and Function Of Complex Networks', *Department of Physics*, University of Michigan.
- Weisstein, Eric W. (2016) "Graph." From *MathWorld*--A Wolfram Web Resource. Available at: [Mathworld.wolfram.com](http://mathworld.wolfram.com). (Accessed 9 January 2016).