Pacemaker Vulnerability:

How Medical Technology and Biological Augmentation

Create New Vulnerabilities

James Waugh, Megan Clark, Elizabeth Hubbard, Daniel Elice

COSC, Towson University

jwaugh4@students.towson.edu, mclark51@students.towson.edu,
ehubba2@students.towson.edu, delice1@students.towson.edu

***Abstract*—This document gives an overview of the Network Security project that focused on the vulnerabilities relating to pacemakers.**

***Keywords*— Pacemaker, Man in the Middle, Denial of Service, Vulnerabilities, NS3, Simulation**

I. Problem Statement

The main problem that is being addressed in this project is, how the most recent pacemakers have the ability to communicate to a patient's doctor via a computer network. Vulnerabilities exist in this situation which may lead to a variety of exploits/attacks. One example is the man in the middle attack. A man in the middle attack is when communication between two devices is intercepted by a malicious actor. In our case, a malicious actor would intercept communication between the pacemaker and the doctor while information is being transferred over the network. The attacker can maliciously modify, alter, and/or delete important data. For example, a malicious actor can "deliver shocks patients don't need or withhold ones they do" [7]. This threat could lead to the patient getting incorrect treatment and could result in fatal consequences. In addition, the doctor may be fed incorrect information from the patient's device. If an attacker adds incorrect statistics regarding the patient's health, the doctor will be unable to give proper care to the patient. These vulnerabilities exist and are extremely dangerous. The next steps would be for the companies that produce these medical devices to take the necessary actions to secure the products for which they provide for their customers. Unfortunately, as we have seen with similar IoT devices, companies are not interested in securing their products but rather, making them easy to use for their customers, as well as ensuring the company is making a strong profit. Although ease of use is important for patients using this newer pacemaker technology, it is not worth the risk of losing one's life with a vulnerable device.

II. Motivation

The main motivation for doing more work on this topic is because the grandfather of one of our group members, currently has a pacemaker. It is said that the older pacemakers are made to last about 10 years. Her grandfather has had his pacemaker for about 9 years now.

This being said it is time for the device to be replaced. Since the newer devices communicate over a network they are vulnerable to potential attacks such as stealing information or taking control of the pacemaker. We wanted to look at the details surrounding these new pacemakers communication techniques in order to find potential vulnerabilities and to research the risk associated with the newer technology. This research can help our group members grandfather determine whether he would like one of the newer pacemakers or the older ones. This information should also be known to the public because if these pacemaker devices are vulnerable people have the right to know.

In addition to family members being affected by this issue, our group also became very interested in the medical field related to network security when the case study about the insulin pump was discussed in class. We learned from the case study discussed in class, that an insulin pump could be remotely hacked when connected through a network such as an app. A person was able to change the amount of insulin, stop the insulin from being given or even give more insulin than needed. Since the pump was unencrypted, the hacker could have also gained personal information about the user such as the treatment, the amount of insulin given, and the data of the device. In regard to the pump, the company who created it had no interest in trying to resolve the security vulnerability but rather just to keep the product easy to use for their customers. This led us to believe that similar situations could affect other medical devices other than simply just the insulin pump. An insulin pump is extremely critical to life for those with blood glucose levels, yet the pacemaker is even more serious for one's everyday life. Our group wanted to look more into the pacemaker to examine the risks associated with the newer technology and to examine the potential vulnerabilities that exist today in these devices. We think it is also important to note the amount of people that have pacemakers. Most people know of someone in their family or a friend that these potential vulnerabilities could affect. In 2016 there was a global number of 1.14 million pacemakers and by 2023 there is expected to be an estimated 1.43 million pacemakers [10]. This means that this is affecting a handful of the population and companies need to ensure that these people are protected against potential vulnerabilities which could lead to life-threatening consequences.

III. Background Research/Survey/Referenced Approaches

A pacemaker is a medical device that is placed in one's chest to control the hearts timing to beat. It uses electrical pulses to prompt the users heart to beat at a normal rate. There are two main parts of a pacemaker: The pulse generator and leads. The pulse generator is a small metal container that has batteries and electrical circuitry that regulates the rate of electrical pulses sent to the user's heart. The leads or electrodes are one to three flexible insulated wires that are placed in a chamber or chambers of the user's heart in order to deliver the electrical pulses to adjust one's heart rate [9].

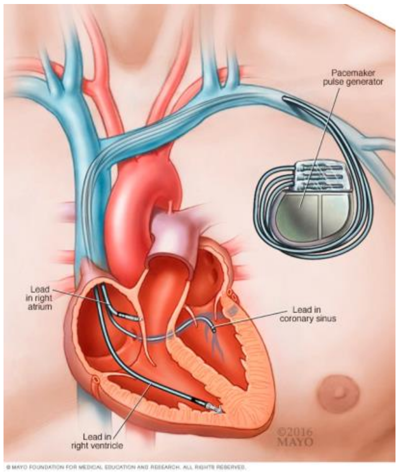
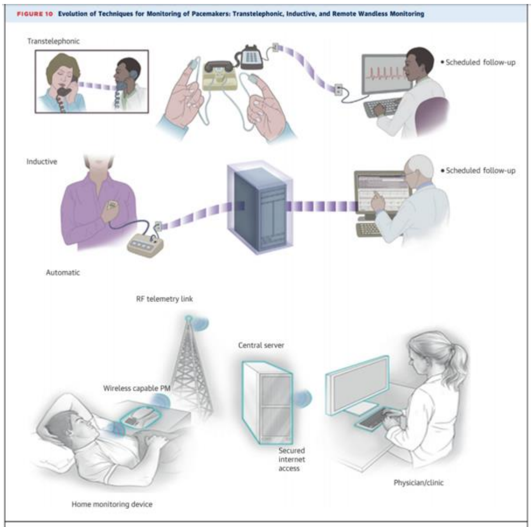


Figure 1: Shows the different parts of a pacemaker. [9]

A variety of pacemakers are able to add a remote monitoring feature which some doctors require their patients to have. The terms used are RM and remote interrogation (RI), which “refer to acquisition of system or patient information from a cardiovascular implantable electronic device and transmitting it to a clinic distant from the patient to enhance care” [5]. This device is able to help doctors detect abnormal heart rhythms and issues related to the heart device faster. There are a few different methods of transmitting data to the doctor [4]. One way the patients are able to communicate their health information to their doctors is to use an application on their smartphones. Patients can transmit secure data from their pacemakers to their physicians through an application. These pacemaker devices that connect directly to a phone through an application allow not only for the patients to communicate directly to the doctor but also to create a personalized profile on the devices website to manage their pacemaker information and data transmissions, confirm the date of their most recent transmission of pacemaker information and receive email or text reminders, confirmations and notifications of their data transmissions [6]. Another method of transmitting data from the patient to doctor is through inductive transmission, one of the most common methods used. Inductive transmission works by having “a patient hold a transmitter’s inductive wand over the pacemaker to perform a full interrogation.” [5]. The way the information is then transmitted back to the doctor is either through a landline or cellular connection to a server. The report that is sent back includes information that the doctor would have taken through a regular pacemaker checkup, which includes “battery status, lead integrity, lead sensing and pacing function, activity sensor statistics, pacing frequency, and stored arrhythmic events” [5]. This method requires the patient’s interaction with the device to send the information back to the doctor. There is another method which does the work without the patient needing to do anything, a method known as radiofrequency RM. Instead of the patient needing to wave the wand over the device, a transmitter which has radiofrequency capabilities is needed. With this radiofrequency it can transmit the data directly from the device without the patient doing anything, as long as the transmitter is in range. The main difference with this device compared to the others is that it “checks patient and device status on a daily basis (as opposed to every 3 months), is fully automatic, and can verify transmissions and generate alerts when they are absent” [5].

FIGURE 10 Evolution of Techniques for Monitoring of Pacemakers: Transtelephonic, Inductive, and Remote Wandless Monitoring



Although transtelephonic and inductive monitoring provide intermittent monitoring, radiofrequency wandless monitoring can provide device- and patient-related information more continuously. Modified from Slotwiner et al. (75). PM = pacemaker; RF = radiofrequency.

Figure 2: Shows the different methods, discussed above, of how the pacemaker can transmit information from the patient to the doctor. [5]

There has been a lot of research on the different exploits and vulnerabilities involved with pacemakers. A recently discovered exploit emerged that allows for an attacker to directly install malware on a patient's pacemaker [7]. Two researchers Billy Rios and Jonathan Butts have performed tests and found multiple different vulnerabilities that exist on Medtronic's pacemakers, some that could lead to life-threatening consequences. Both Billy and

Jonathan have given their information back to Medtronic, and “while Medtronic has remediated some of the issues the researchers discovered, Rios and Butts say that too much remains unresolved, and that the risk remains very real for pacemaker patients” [7]. The first problem they found was not with the direct transmission of the information from the pacemaker to the doctor but instead from the software delivery network. This is how devices involved directly with the pacemaker receive updates, “which healthcare professionals use to tune implanted pacemakers” [7]. The researchers had given the information they had found through their research and their simulated environment to the company but instead of solving the issues, the company “took 10 months to vet the submission, at which point it opted not to take action to secure the network” [7]. This is, again, similar to the insulin pump scenario, with how companies are aware that vulnerabilities exist within their devices but decide not to address the problem. Similar to the insulin pump, the one device these researchers looked more into, the device lacked “of digital code signing” [7]. This can allow for non-validated updates on the device which an attacker could use to their benefit. Overall, there is an obvious major problem with the medical devices that are may result in life-threatening consequences to the people using these devices.

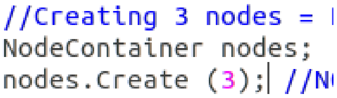
IV. Challenges

The concept that the pacemaker, a device meant to keep somebody alive by monitoring their heart beats, has multiple vulnerabilities that have previously been exploited is alarming. For obvious reasons our group could not actually attempt to exploit vulnerabilities on an actual pacemaker, so we needed to find a software that was able to simulate attacks. Our group used NS3 as our network simulation tool. The first challenge we had was setting up NS3. The software required a large amount of space. We needed to find a computer that had plenty of available memory. NS3 was eventually loaded on a desktop computer. To configure the software, a YouTube tutorial was followed. Using VMWare Workstation 15 Player with Ubuntu 18.04 installed, NS3 successfully downloaded and we configured it properly to allow the simulator to run as expected and create a pacemaker scenario.

An additional challenge of this project was not exactly knowing how these devices communicate between one another. When first starting this project, we thought that we would have more information about how the devices interact with each other which we quickly learned was not true. We are not able to know how exactly they are being encrypted which is most likely not available to prevent attacks against the pacemakers. Since we do not know exactly how they are sending information back and forth and whether or not it is encrypted, we just looked at the vulnerabilities from transferring information over the networks without any kind of encryption. An additional challenge was setting up attacks using the NS3 software. Once the pacemaker scenario was established, the group wanted to simulate a man in the middle attack to show the consequences of such a vulnerability. NS3 made this difficult to do. Instead, the group ended up simulating a Denial of Service attack, where many packets were sent from the client to the server, causing congestion and some packets to be dropped.

V. Solution

During our initial setup of our pacemaker simulation, we decided to create 3 nodes. Node 0 was the actual pacemaker. Node 1 was the patient’s home transmitter. Node 2 was the doctor’s access point, to include the onsite database at the hospital. This setup is very similar to the interaction between the patient’s pacemaker and doctor’s access point.



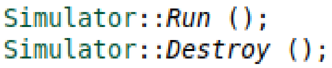
```
//Creating 3 nodes = 
NodeContainer nodes;
nodes.Create (3); //N
```

Figure 3: 3 Nodes have been created using NS3 simulation software

After creating the three nodes, we connect them using the Point-to-Point Protocol. This is a data link layer communications protocol used to establish a direct connection between two nodes. The pacemaker (Node 0) and the home transmitter (Node 1) directly communicate using this protocol and the home transmitter and doctor’s access point (Node 2) directly communicate using this protocol. While connecting the three nodes, attributes and IP addresses were set for the nodes.

Later, using the UdpEchoClientHelper class, we create a “server application which waits for input UDP packets and sends them back to the original sender” [11]. We install the doctor’s access point (Node 2) to this and create a UdpEchoClientHelper class next for the pacemaker (Node 0). Our nodes are now able to communicate and send packets to each other.

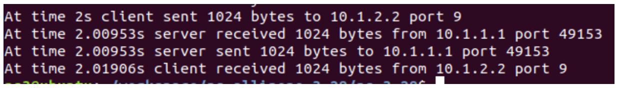
Running our simulator allows us to examine packets and interaction between nodes:



```
Simulator::Run ();
Simulator::Destroy ();
```

Figure 4: NS3’s built in command to run our simulated environment.

We run our environment and view the following output:



```
At time 2s client sent 1024 bytes to 10.1.2.2 port 9
At time 2.00953s server received 1024 bytes from 10.1.1.1 port 49153
At time 2.00953s server sent 1024 bytes to 10.1.1.1 port 49153
At time 2.01906s client received 1024 bytes from 10.1.2.2 port 9
```

Figure 5: Viewing the output after running the simulation in the terminal

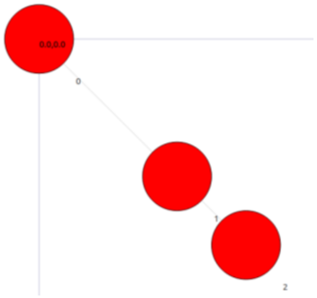


Figure 6: A more user-friendly output after running the simulation using NetAnim

At 2 seconds, the client (or pacemaker) sends packets to IP address 10.1.2.2 (the doctor’s access point). Afterwards, the server (doctor’s access point) sends packets back to 10.1.1.1 (the pacemaker). The home transmitter then sends the packets to the pacemaker. Although a simple simulation, it demonstrates how the pacemaker communicates over the network.

Imagine a Denial of Service attack occurs resulting in loss of packets. Information will be lost in transmission. The doctor will receive incomplete pacemaker data resulting in incorrect diagnoses and treatment. Imagine a man-in-the-middle attack occurs. An attacker may add extra packets resulting in false information being transmitted to/from the doctor and patient. These vulnerabilities exist and need to be addressed by the companies producing these products.

To simulate a Denial of Service attack, we sent 10,000 packets from the client to server in a matter of 8 seconds. As a result, some packets were dropped and the communication between the client and server was interrupted. If this was a real scenario, packets and important information would not have been received by the doctor, potentially resulting in false treatment and harmful consequences.

Denial of Service attacks are very difficult to prevent, yet the effect of the attack can be mitigated. A load balancer would help keep the nodes from being unable to function. In addition, limits should be assigned to networks. If a Denial of Service attack occurs, the load balancer will hopefully drop any connections that are exceeding the maximum values permitted. To prevent man-in-the-middle attacks, all the information sent from the pacemaker to the home transmitter to the doctor's access point and back should be encrypted. In addition, authentication certificates should be implemented. These certificates can be automated so the patient would not need to worry about securing their information. Companies should be responsible for implementing these prevention techniques so that the patient's lives are not endangered if a malicious attack was to occur.

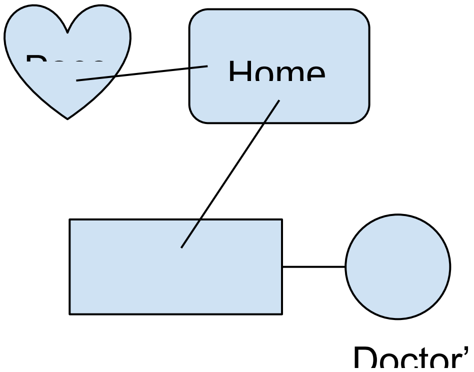


Figure 7: Explains how our simulation was set up to communicate between nodes

VI. Evaluation

The simulation exhibits how the doctor-to-pacemaker communication process transmits wirelessly, and as a result of the wireless communications, vulnerabilities in the system arise. In a real-world scenario, it is likely an attack would be carried out on the hospital system, as hospital system security is often not up to current security standards due to budget or operational constraints. Performing a Denial of Service attack on the system will satisfy the outcome of a real attack on the system, in which the patient's pacemaker is no longer communicating with the hospital database and the doctor's inputs. It can be inferred from a successful Denial of Service attack on the hospital system that the attacker could then attack the vulnerable home transmitter and send information to the pacemaker.

By simulating a scenario where a patient's pacemaker is communicating with a home transmitter, which in turn is communicating with the hospital database and the doctor's access point, we collected information on the system. The system had to have wireless communication between the home transmitter and the hospital database. In a recent research study conducted at the Black Hat information security conference, two security professionals, Jonathon Butts of QED Secure Solutions and Billy Kim Rios of Whitescope, performed an attack on a pacemaker system. "To take control of the pacemaker, Rios and Butts went up the chain, hacking the system that a doctor would use to program a patient's pacemaker. Their hack rewrote the system to replace the background with an ominous skull, but a real hack could modify the system invisibly, while ensuring that any pacemaker connected to it would be programmed with harmful instructions" [3]. The article written by Alex Hern, a technology reporter employed by The Guardian writing from Las Vegas where the conference took place, continues: "'You can obviously issue a shock,' Butts said, 'but you can also deny a shock.' Because the devices are implanted for a reason, he added, withholding treatment can be as damaging as active attempts to harm" [3].

In regard to data authenticity, confidentiality, and integrity, the vulnerability of the system is, beyond reasonable doubt, failing to secure all three facets of security. The authenticity of the doctor’s inputs once an attacker has conducted a man-in-the-middle attack is in question, once the hospital’s system is compromised. The confidentiality of patient information is not assured if the hospital’s security measures are not prepared to defend against modern-day cyberattacks, and the integrity of the system is not up to satisfaction since the attack can be carried out to fruition. The confidentiality of the patient’s data is easily breached with this vulnerability because now an unauthorized user can easily access private or confidential patient information, such as medical records, which includes their name, address, SSN, and other personal information. If an attacker gains access to the pacemaker device, then the whole integrity of the system is breached. The pacemaker can then be changed by the unauthorized user and set to no longer perform its intended functions free from deliberate manipulation. Availability can also be breached because an attacker may choose to completely block out the doctor from accessing the information from the patient’s device, or vice versa. Either way the attacker is denying services to authorized users, hence breaching availability. Overall, all elements of the CIA Triad are easily breached with the different attacks and vulnerabilities in the present-day pacemakers.

The additional concepts of the CIA Triad also need to be taken into consideration for the pacemaker. There should be functions to authenticate both the user as well as the doctor. This can assure that it is actually the person who is designated to use the device. Unfortunately, from our research, there is nothing in place right now that authenticates the users. We also believe there should be features integrated into the device to address accountability. This would be ideal so that if something does change, it can be traced back directly to the entity that made those changes. From our research we have not found anything that directly states that this function is put in place.

VII. Analysis and Inferences

It appears that due to the nature of the pacemaker’s paired home transmitter device, and the necessity for wireless communication in order for doctors to access information from the pacemaker remotely, the system is vulnerable to attacks that exploit the communication among devices. This would be a man-in-the-middle attack, where an attacker adopts the “persona” of the doctor in order to transmit malicious inputs to the pacemaker. As the system communicates wirelessly, it may be possible for an attacker to spoof the address of the home transmitter after gaining access to the patient’s home network and obtain information about the doctor’s system as the doctor attempts to communicate with the pacemaker. Alternatively, the attacker may gain access to the hospital network to intercept packets sent to and from the home transmitter. As a number of recent reports reveal, hospitals are notoriously unprepared for malicious attacks on their network, as they utilize outdated security measures and improper network security measures are implemented.

It can be inferred that medically implanted devices similar to pacemakers will inherit vulnerabilities of this nature as long as they are transmitting information wirelessly. However, it might be within manufacturers’ interests to increase the quality of their system security so that attacks on patients that may result in harm or death do not lead to aggressive lawsuits. Failure to comply with security standards for medically implanted devices associated with vital bodily processes would surely lead to public outcry and politically motivated legal action.

VIII. Conclusion and Future Direction

Common medical devices are vulnerable and not properly secured against cyber-attacks, and though an instance of an attack on a medical device such as a pacemaker has not yet been carried out, a vulnerability in a person's physical safety brought about by exploits in medical devices that are expected to be secure should be a top priority for security professionals. Albeit, the goal of common cyber criminals is to steal sensitive data or obtain money, a politically-motivated, personally-motivated, or state-sanctioned attack on an important individual is not an unreasonable consideration. Having a vulnerable medical device that allows the person to be targeted by remote attacks, potentially untraceable attacks, or affected by malware on a different device that spreads to the medical device is a violation of the device manufacturer's due diligence.

Inherent flaws in medical devices such as pacemakers may persist as the devices become integrated with more vital components of the human body, such as the brain. Assuming that a vulnerable device could be connected to the brain in such a way that would allow the device to communicate with either other devices or an access point wirelessly, then it can be deduced that the device is therefore vulnerable to attacks carried out over the wireless connection. An attacker issuing malevolent commands to a compromised device, possibly even a modern-day pacemaker, could end a life remotely from anywhere in the world. Should implanted devices become a widespread phenomenon, a common augmentation of the human body, a particularly aggressive attack may be able to end a multitude of lives in a relatively short amount of time.

Another consideration that could resolve the issue of companies not doing something about the risk is having legal actions put in place to ensure companies adhere to a set of security standards. Since there are no current laws directly charging companies for not taking action on these vulnerabilities, there could be specific laws put into place to do so. By creating laws that would require companies to make sure their products are of the proper level of security, it could lower the risk of patients being put in danger of losing their life. Now these laws would not just be for pacemakers but could be expanded to any medical devices requiring a connection over a network. The laws could be that if a company is aware of the vulnerabilities within their device and choose not to take the proper action to mediate them, they can be charged a fee. It does cost money to resolve the security vulnerabilities in their devices, as of right now they do not need to spend the extra money to solve the problems, but if this law were to be put in place then the companies would be forced to spend the money to fix the problem or pay an even greater amount of money to pay the fee. As a result, although there would be benefits in creating legal action to hold companies accountable for their devices, it could ultimately lower the amount of companies willing to put medical devices on the market.

IX. References

[1] Choudhary, H. (2014, November 24). Retrieved November 12, 2018, from https://www.youtube.com/watch?v=T8NwCPROYYA

[2] Hamlyn-Harris, J. H. (2018, September 19). Three reasons why pacemakers are vulnerable to hacking. Retrieved from http://theconversation.com/three-reasons-why-pacemakers-are-vulnerable-to-hacking-83362

[3] Hern, A. (2018, August 9). Retrieved November 16, 2018, from https://www.theguardian.com/technolog

y/2018/aug/09/implanted-medical-devices-hacking-risks-medtronic

[4] Ilov, N., Abdulkadyrov, A., & Nechepurenko, A. (2016). Early detection of defibrillator lead failure by Medtronic Carelink remote monitoring. *Annaly Aritmologii,13*(1), 55-58. doi:10.15275/annaritmol.2016.1.7 https://www.medtronic.com/content/dam/medtronic-com/us-en/corporate/documents/Medtronic_2090_Security_Bulletin_02272018.pdf

[5] Madhavan, M., Mulpuru, S. K., Cha, C. J., & Friedman, P. A. (2017, January 09). Advances and Future Directions in Cardiac Pacemakers. Retrieved from http://www.onlinejacc.org/content/69/2/211

[6] Medtronic enables pacemaker monitoring by smartphone. (2015, November 20). Retrieved from https://www.healthcareitnews.com/news/medtronic-enables-pacemaker-monitoring-smartphone

[7] Newman, L. H. (2018, August 09). A New Pacemaker Hack Puts Malware Directly on the Device. Retrieved from https://www.wired.com/story/pacemaker-hack-malware-black-hat/

[8] ICS- CERT Medtronic Search https://search.usa.gov/search?utf8=✓&affiliate=us-cert-ics&query=Medtronic&commit=Search

[9] Pacemaker. (2018, March 21). Retrieved from https://www.mayoclinic.org/tests-procedures/pacemaker/about/pac-20384689

[10] Pacemakers market volume worldwide in units 2023 forecast | Statistic. (n.d.). Retrieved from https://www.statista.com/statistics/800794/pacemakers-market-volume-in-units-worldwide/

[11] *ns-3: ns3::MinstrelHtWifiManager Class Reference*. [Online]. Available: https://www.nsnam.org/doxygen/classns3_1_1_udp_echo_server_helper.html#details. [Accessed: 30-Nov-2018].

X. Appendix

The following is the full code used to create our simulation:

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/netanim-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE
("FirstScriptExample");

int
main (int argc, char *argv[])
{
        CommandLine cmd;
        cmd.Parse (argc, argv);

        Time::SetResolution (Time::NS);
        LogComponentEnable
("UdpEchoClientApplication",
LOG_LEVEL_INFO);
        LogComponentEnable
("UdpEchoServerApplication",
LOG_LEVEL_INFO);

        //Creating 3 nodes = Pacemaker, Home
Transmitter, & Doctor's Access Point
        NodeContainer nodes;
```

```
    nodes.Create (3); //N0 =Client (Pacemaker), N1= Router (Home Transmitter), N2= Server (Doctor's Access Point)

    //P2P 1
    PointToPointHelper pointToPoint1;
    pointToPoint1.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
    pointToPoint1.SetChannelAttribute ("Delay", StringValue ("2ms"));

    //P2P 2
    PointToPointHelper pointToPoint2;
    pointToPoint2.SetDeviceAttribute ("DataRate", StringValue ("10Mbps"));
    pointToPoint2.SetChannelAttribute ("Delay", StringValue ("5ms"));

    InternetStackHelper stack;
    stack.Install (nodes);

    Ipv4AddressHelper address;
    address.SetBase ("10.1.1.0", "255.255.255.0");

    NetDeviceContainer devices;
    devices = pointToPoint1.Install (nodes.Get(0), nodes.Get(1)); //node 0 = 10.1.1.1 & node 1 = 10.1.1.2
    Ipv4InterfaceContainer interfaces = address.Assign (devices); //left half of network

    //right half of network
    devices = pointToPoint2.Install (nodes.Get(1), nodes.Get (2));
    address.SetBase("10.1.2.0", "255.255.255.0");
    interfaces = address.Assign(devices);


    Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

    UdpEchoServerHelper echoServer (9);

    ApplicationContainer serverApps = echoServer.Install(nodes.Get (2));
    serverApps.Start (Seconds (1.0));
    serverApps.Stop (Seconds (10.0));

    UdpEchoClientHelper echoClient (interfaces.GetAddress(1), 9); //IP address of second node, port number 9
    echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
    echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
    echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

    ApplicationContainer clientApps = echoClient.Install(nodes.Get (0)); //installing client on node 0
    clientApps.Start (Seconds (2.0));
    clientApps.Stop (Seconds (10.0));

    AnimationInterface anim ("pacemaker.xml");
    anim.SetConstantPosition (nodes.Get(0), 0.0, 0.0);
    anim.SetConstantPosition (nodes.Get(1), 2.0, 2.0);
    anim.SetConstantPosition (nodes.Get(2), 3.0, 3.0);

    Simulator::Run ();
    Simulator::Destroy ();

    return 0;

}
```