Dynamic Emergency Vehicular Evacuation (D.E.V.E)

Conor Powderly, Ashish Shota, Vanshika Sinha, Mudit garg

April 2022

1 Collaboration From Project two

Both of our project two networks focused on vehicle to vehicle communication meaning an easy compromise of ideal was found. Our collaborative knowledge base consisted of TCP, UDP, DHCP, DSRC,V2V and to pedestrian use peer to peer, Client server was used overall, rdt 3.0, Selective repeat, Threading, pipe-lining, Socket programming, ICMP beacon frames, Serialization data packet. These topics were used to develop this project and can be seen in our write up and implementation. We also had a greater knowledge of the link layer and wireless communication from our progression in this module that we were able to implement in this project.

2 Use Case

The automotive industry is now advancing towards building more safer and feature rich vehicles. The emerging trends in the industry such as connected self-driving cars can greatly improve the safety of everyone on the road by eliminating human error and other factors such as latency during peak action. This results in enhanced traffic control safety during normal days as well as during the state of emergencies when everything is in chaos

The concern that our network model intends to assist with evacuations times of crises, like war or serious weather events caused by climate change. In many of these crises the servers that provide connection to virtual navigation tools can be damaged or they may not have an accurate topology of the map in the event of a crises.

The goal of our project is to provide accurate real time data sharing between these vehicles and stationary nodes placed at road junctions to calculate the best evacuation route.

While the focus of this project is vehicular evacuation this could be evolved to include pedestrian nodes, which could organising ride sharing in evacuation situations. It could also be used by an over sight organisation like the UN to monitor the evacuation and provide support where needed.

3 Communication Model

DEVE is a peer-to-peer based network fabric which aims at finding the safest and shortest path which updates real time using the information shared from the peers. The network consists of two types of nodes, stationary nodes and mobile nodes. Stationary nodes are placed at the stationary elements of the streets like the street lamps and mobile nodes are the moving vehicles which travel on the roads.

The stationary nodes together form the topology of the map. Are placed at the the same junction to account from the difference direction in which the cars travel.

Mobile node shares the time taken to reach from one stationary node to another and the weight of the edge from one node to another is updated based on the average time taken to reach from one stationary node to another. Stationary nodes will compute the shortest path to the goal evacuation node and inform the user.

DEVE can be implemented physically using Zigbee network controllers and use to coordinator at the intersection with Zigbee end devices hooked up in the vehicles. It uses IEEE 802.15.4 specification for a suite of high-level communication protocol used to create wireless personal area network using low-power. The stationary nodes will have solar panels to charge the low power network devices so our network is off the grid and emergency ready. Communication though Zigbee is secured by a 127-bit symmetric encryption key so it adds a layer of security to our network.

4 Application Layer

The each stationary network remains dormant and listening on a universal port and IP until it receives a message from a mobile node signifying that an evacuation has begun. Then the stationary nodes boot up with a expected map topology store in their ROM. This the uses Dijkstra's Shortest Path Algorithm to calculate the way fastest to an evacuation point.

Each time a vehicle passes a new travel time to a node the shortest path algorithm runs again to keep the make up to date. When no vehicle has travelled over a particular edge for a certain period of time, it is considered to be blocked and alternative routes are explored.

categories - green, orange and red zone. Green nodes are stationary node with constant flow of traffic passing though that node whereas orange nodes are the nodes where the flow is inconsistent slow and red nodes are stationary peers which are rarely visited.

We implemented this algorithm using a priority queue to greedily choose the nearest node that has not been visited yet and execute this process on all of its edges.

Dijkstra's algorithm provides an advantage over other algorithms because once it is run, we can save the results somewhere and can look up to our algorithm again and again without having to actually run the algorithm itself.

In our network model, we have used an adjacency matrix representation of graph and calculated the shortest path between two nodes using Dijkstra algorithm and then passing the information to the transfer layer. In our path finding algorithm, we are exposing the function interfaces of our graph.py file which are, "printgraph" and "getgraph" in our main file of client.py. For calculating the time in travelling from one node to another node, we have included the concept of blocked roads, which implies that if we have a blocked road and there is no update on the road for 10 minutes, then the weight of the road rises by 100 which leads to blockage. most of the roads are in a certain range of length and few important roads connect different areas together.

5 Transport Layer

On the transport layer, we are using UDP as it fast and best suited for the nature of the network. To add reliability to our model we added RDT (reliable data transfer) 3.0 which waits for the conformation that the packet is received before sending next sequence of packet in between network transport and application layer.

6 Network layer

Each node broadcasts their IP port and time to live every 20 seconds. This broadcast is caught by other nodes who add this information to a list of peers. If a node receives duplicate information, it restarts this node's time to live in its table. When a particular time to live expires, it is assumed that the node to which this information belongs to, has left the network. Each node initializes one socket to communicate with every member of this list that is still alive and waits its turn to transmit its data. This approach to node discovery and loss is for the dynamic element of our network.

Each node has a buffer which stores the address of the peers and their time to live. To join the network, a new peer must broadcast its network address and all the existing peers add it to their buffer and the stationary node reply the new peer with the address of all the peers in the network.

7 Link Layer

Due to our network using a shared medium of Wi-Fi, collisions are a concern to our network. We have two ways that we've decided to approach this issue, the first being that we may listen to the channel and if there's nothing, we send. If we hear something coming in while we're sending, we stop sending and we'll back off and the other peer does the same. This is an inefficient way because it leaves a lot of time idle on the channel. However, if a collision occurs, the information is useless anyway to the receiver. We considered this a necessary evil. Another solution by placing our stationary nodes on the roads themselves so that when a car passes by them, stationary node is swamped with that signal due to the proximity between nodes.

Finally, before we send our data into the channel, we encode using a CRC (cyclic redundancy check), that is, add the sending node and amending the remainder bits to the end has redundancy error bits. We then, at the receiver, perform this check again to make sure that no errors have occurred over the transmission and if errors are detected, it is our protocol that we will discard this frame. We have chosen to discard the frame rather than to attempt to fix the error because our network has built in redundancy with the fact that many cars will be carrying very similar data, many cars we moving from node A to B will be traveling roughly at the same speed so no one nodes' information is vital to our network

8 Testing and validation

The UDP Broadcast for node recognition we proposed to use was not an option in the implementation on the raspberry pi's so we instead manually entered the ports and IP. For the same reason the collision detection method proposed was not implement. Aside from these two aspects of our design the error messaged were encoded for our CRC check, our RDT ACKs and our timeout. The simulation consisted of one stationary node and 3 mobile nodes retrieving the shortest path to their evacuation point.

9 Peer Learning: Conor Powderly

Its been a great learning experience with regards to team work, understanding data fabric and their implementation. I was able to learn how use python thanks to Ashish's guidance and practical problems to solve. I gained a deeper insight into the difficulties of implementing reliable data transfer. I liked freedom in which we developed our idea we constantly had to tear our network down and start again.

10 Peer Learning: Ashish Shota

It's been a great learning experience through project 3. I was able to learn a lot from my group members. Along the way we advanced our knowledge about how to design and implement a network model from scratch, right from the idea phase up till programming the network. We had several team meetings which also included pair coding sessions where we met in person to implement the network. Overall through this project I was able to develop leadership quality and team management skills along with the learning the technical knowledge required to fabricate a network.

11 Peer Learning: Vanshika Sinha

My learning experience in context with the Project-3 of Computer Networks was fantastic. First of all, I learned that how my simple idea of Project-1 concerning the vehicular communication using P2P network could be upscaled to such a higher level taking into account the global concern of evacuation during natural calamities. Talking about the important concepts used in this project, this project provided me a chance to dive deeper into the concept of designing a network model from scratch which could actually prove to be of great help for the real-world and is feasible to design. Most importantly, apart from my personal development academically, this project provided me a grand opportunity to work in a team and develop skills specific to collaborative efforts. All in all, I was able to refine my understanding through discussions and feedback from my peers, and develop stronger communication skills.

12 Peer Learning: Mudit Garg

The learning journey of project 3 taught me and solidified for me, a lot of concepts of computer networks. I don't think that just attending lectures and consuming resources present online would have impacted in this much. Overall, through this project I was able to learn about the practical aspects of networking and socket programming and get an idea about how connection and communication between devices in a network is possible in today's vastly interconnected world